

Random-Restart Best-Response Dynamics for Large-Scale Integer Programming Games and Their Applications

Hyunwoo Lee* Robert Hildebrand† Wenbo Cai‡ İ. Esra Büyüктаhtakin§

February 24, 2026

Abstract

This paper presents scalable algorithms for computing pure Nash equilibria (PNEs) in large-scale integer programming games (IPGs), where existing exact methods typically handle only small numbers of players. Motivated by a county-level aquatic invasive species (AIS) prevention problem with 84 decision makers, we develop and analyze *random-restart best-response dynamics* (RR-BRD), a randomized search framework for PNEs. For IPGs with finite action sets, we model RR-BRD as a Markov chain on the best-response state graph and show that, whenever a PNE exists and the restart law has positive probability of reaching a PNE within the round cap, RR-BRD finds a PNE almost surely. We also propose a Monte Carlo sampling-and-simulation procedure to estimate success behavior under a fixed round cap, which informs our instance-dependent performance characterization. We then embed RR-BRD as a randomized local-search subroutine within the zero-regret (ZR) framework, yielding *BRD-incorporated zero-regret* (BZR). Using solver callbacks, RR-BRD searches for and supplies PNEs, while ZR separates and adds equilibrium inequalities to tighten the formulation. We introduce *edge-weighted budgeted maximum coverage* (EBMC) games to model AIS prevention and establish PNE existence results for both selfish and locally altruistic utilities. Computational experiments on synthetic EBMC and knapsack problem game instances show that RR-BRD and BZR scale equilibrium computation up to $n \leq 30$ players. We further solve a real-world EBMC game derived from the Minnesota AIS dataset with $n = 84$ county players.

Keywords: Integer Programming Games, Pure Nash Equilibrium, Socially Optimal PNE, Best-Response Dynamics, Sink Equilibria, Aquatic Invasive Species

Funding: This work was supported in part by the Air Force Office of Scientific Research (AFOSR) and the Minnesota Aquatic Invasive Species Research Center (MAISRC).

1 Introduction

Integer programming games (IPGs) stand at the crossroads of integer programming and non-cooperative game theory. They study settings in which each player solves an optimization model with discrete deci-

*Grado Department of Industrial and Systems Engineering, Virginia Tech. Email: hyunwoo.lee@vt.edu

†Grado Department of Industrial and Systems Engineering, Virginia Tech. Email: rhil@vt.edu

‡Mechanical and Industrial Engineering, New Jersey Institute of Technology. Email: cai@njit.edu

§Grado Department of Industrial and Systems Engineering, Virginia Tech. Email: esratoy@vt.edu

sions, and these decisions influence one another’s utilities. IPGs are non-cooperative, complete-information games: players are self-interested and have complete information about each other’s utilities and strategy sets [Carvalho et al., 2022]. A pure Nash equilibrium (PNE), as defined by Nash [1950], is a strategy profile in which no player can improve their utility by unilaterally changing their strategy. Social welfare is the sum of all players’ utilities. In this paper, we develop efficient algorithms for large-scale IPGs to (i) compute a PNE, (ii) compute multiple PNEs, and (iii) identify the PNE with the highest social welfare.

Computing multiple PNEs is an important task in non-cooperative games, as some equilibria may be more desirable than others. In particular, a socially optimal PNE—a PNE that maximizes total social welfare—is a natural target in settings with mediators or central planners who care about system-wide performance. This notion is also central in algorithmic game theory [Roughgarden, 2010], where it underpins standard measures of equilibrium inefficiency relative to socially optimal outcomes. In the IPG literature, two recent lines of work address (i) equilibrium computation and enumeration and (ii) selecting a welfare-maximizing equilibrium: Dragotto and Scatamacchia [2023] for PNEs, and Crönert and Minner [2024] for mixed Nash equilibria (MNEs). However, both approaches remain limited to small-scale instances.

Although recent advances in IPG algorithms have improved our ability to compute PNEs, most methods remain limited to small-scale games with only two or three players ($n \leq 3$) [Dragotto and Scatamacchia, 2023] or with a smaller number of variables ($m \leq 5$) [Sagratella, 2016, Schwarze and Stein, 2023]. Given the maturity of the field, it is timely to bridge this gap by addressing the large-scale, multi-stakeholder instances required for real-world decision-making. Compelling examples arise in resource allocation decisions across decentralized administrative structures, such as the many states or counties in the United States, naturally forming large-scale IPG settings. Motivated by Minnesota’s AIS prevention efforts, which are independently managed by 84 counties, we introduce a new class of non-cooperative IPGs: *edge-weighted budgeted maximum coverage* (EBMC) games. These games model county-level decision-making under budget constraints and inter-county externalities. Solving such problems requires scalable and efficient algorithmic approaches.

To address this challenge, we focus on *random-restart best-response dynamics* (RR-BRD), a randomized variant of best-response dynamics (BRD) [Matsui, 1992]. We adopt a Markov chain framework and model RR-BRD as a random walk on the best-response (BR) state graph of pure strategy profiles, as introduced by Goemans et al. [2005]. A basic observation is that the induced random walk eventually enters a recurrent class of the BR state graph; these *sink equilibria* include singleton PNEs as well as non-PNE sink components. RR-BRD mitigates such traps via a restart strategy with a fixed cap on the number of steps per attempt. We show that, whenever a PNE exists and the restart distribution assigns positive probability to its basin of attraction, RR-BRD finds a PNE almost surely, with an instance-dependent bound on the expected number of best-response solves. In practice, we incorporate RR-BRD as a randomized local-search subroutine within the zero-regret (ZR) framework of Dragotto and Scatamacchia [2023]. This hybrid approach dramatically improves the scalability of PNE computation, increasing the number of solvable players from small-scale cases ($n = 2, 3$) to up to 30 players in random datasets and 84 players in real-world datasets.

As a motivating application for large-scale IPGs, we build on optimization-based approaches to invasive species control [Büyüktaktakın and Haight, 2018, Chen et al., 2023, Kıbış et al., 2021] by introducing EBMC games, which arise naturally in decentralized, county-level AIS prevention across U.S. lakes. Previous

studies have applied the EBMC formulation [Caskurlu et al., 2014] to model AIS inspection at the county [Haight et al., 2021] and state levels [Haight et al., 2023], while others have examined AIS spread across network graphs [Escobar et al., 2018]. In EBMC games, each player controls a subset of vertices in a network graph, and each player maximizes their edge coverage under budget constraints. We define two variants of the utility functions—selfish and locally altruistic—and establish theoretical conditions for the existence of a PNE in these models. Additionally, we validate our IPG framework through experimental results on EBMC games using random and real-world datasets, as well as on knapsack problem games (KPGs) with random datasets. Our findings reveal that IPGs, coupled with the enhanced solvability of the BZR algorithm presented in this paper, offer a powerful and scalable framework for modeling and solving large-scale multi-agent problems.

1.1 Literature Review

Integer programming games (IPGs) are gaining attention for their ability to model strategic interactions beyond traditional normal-form games. Various applications of IPGs have been studied, including facility location and design games [Crönert and Minner, 2024], fixed-charge transportation problems [Sagrattella et al., 2020], kidney exchange games [Carvalho et al., 2017], quadratic IPGs (qIPGs) [Sagrattella, 2016, Schwarze and Stein, 2023], and knapsack problem games (KPGs) [Carvalho et al., 2025, Dragotto and Scatamacchia, 2023]. In these models, each player solves a player-specific integer program (IP), and strategic interactions are typically captured through reciprocally bilinear terms that couple players’ decision variables, with either linear objectives (e.g., KPG) or quadratic objectives (e.g., qIPG).

Carvalho et al. [2023] provide a comprehensive overview of the IPG solution methods, categorizing them into six approaches. Among these, sample generation method (SGM) [Carvalho et al., 2022], enumerative SGM (eSGM) [Crönert and Minner, 2024], and Cut-and-Play (CnP) [Carvalho et al., 2025] compute MNEs, while branching method (BM) [Sagrattella, 2016], zero-regret (ZR) [Dragotto and Scatamacchia, 2023], and branch-and-prune (BnP) [Schwarze and Stein, 2023] focus on PNEs. Notably, Carvalho et al. [2022] and Crönert and Minner [2024] use normal-form game methods to compute MNEs in sampled games before verifying them in the original setting. Among exact methods to compute PNEs, the zero-regret (ZR) algorithm [Dragotto and Scatamacchia, 2023] is state-of-the-art for computing socially optimal PNEs via a cutting-plane approach. For approximate NEs, Sankaranarayanan [2024] presents a best-response algorithm to compute approximate PNE and Duguet et al. [2025] show a piecewise linear approximation to compute approximate MNE in the mixed-integer nonlinear programming cases based on SGM.

In most approaches, equilibrium computation is ultimately reduced to one or more joint optimization problems that couple all (or a subset of) players’ strategy sets, either explicitly in normal form or implicitly via extended MIP or complementarity formulations. Consequently, reported computational studies focus on instances with small numbers of players and integer variables per player (typically with $n \cdot m \leq 300$, where n is the number of players and m is the number of integer variables per player; see also Section 4.3). This provides a useful baseline but leaves open the question of how far one can push the tractability of equilibrium computation in IPGs when the number of players is much larger—for example, when $n \cdot m$ is in the thousands.

Best-response dynamics (BRD) [Matsui, 1992] is a foundational and commonly used method for computing PNEs in normal-form games, where players take turns solving their best-response problems while keeping the strategies of the others fixed. In well-structured classes such as potential games [Monderer and Shapley, 1996] and, more generally, best-response–weakly acyclic games [Apt and Simon, 2015], the convergence of BRD to a PNE is well understood. However, for arbitrary normal-form games the best-response state graph may contain directed cycles and non-singleton sink strongly connected components. Goemans et al. [2005] interpret the long-run behavior of such dynamics through *sink equilibria*, which capture recurrent behavior of best-response walks; see also the broader dynamics-based perspective of Papadimitriou and Piliouras [2019]. Relatedly, Hakim et al. [2024] study *limit* (infinite-horizon) absorption over sink SCCs of a (better-or-equal) response graph under a different noisy dynamics; our analysis uses a Markov chain induced by (improving) best-response updates and focuses on finite-horizon PNE-hitting probabilities.

Recent work in algorithmic game theory (AGT) has studied randomized variants of BRD that still follow a single trajectory but randomize either the initial profile or the updating player at each step. For example, Amiet et al. [2021] and Heinrich et al. [2023] consider games with random payoffs: Amiet et al. [2021] study BRD in games with binary strategy profiles, and Heinrich et al. [2023] analyze BRD with randomized playing sequences and randomly drawn initial strategy profiles. These works introduce randomness through the play sequence and initial draws from a known or assumed distribution, but they do not employ restart mechanisms. In the IPG context, Sankaranarayanan [2024] analyzes a best-response algorithm with fixed playing sequence for lattice convex-quadratic simultaneous games and shows that, when the sequence of strategy profiles enters a finite trap (detected by a repeated profile), one can extract an approximate MNE from the strategies in that trap. To our knowledge, prior to this work, BRD-type procedures had not been developed as a scalable, restart-based algorithm for computing PNEs in IPGs with finite strategy sets.

1.2 Contribution

Our detailed contributions are summarized as follows:

- We introduce *random-restart BRD* (RR-BRD) into the IPG literature and analyze it as a randomized search framework for computing PNEs in large-scale games. For IPGs with finite strategy sets, we adopt the best-response (BR) state-graph viewpoint of Goemans et al. [2005] and model RR-BRD as a Markov chain on this graph. We show that, whenever a PNE exists and the restart distribution assigns positive probability to states from which a PNE is reachable within the round cap, RR-BRD finds a PNE with probability one and admits an instance-dependent bound on the expected number of BR solves. We further contribute to the normal-form game literature by providing a convergence guarantee for BRD under restarts stated directly in terms of BR-state-graph reachability, without assuming random payoffs or restrictive structural conditions (e.g., potential games or BR-weakly acyclic games).
- We propose a Monte Carlo sampling-and-simulation procedure, in the spirit of Hakim et al. [2024], to estimate per-attempt success under a fixed round cap and to inform an instance-dependent performance characterization of RR-BRD. We also discuss best- and worst-case behavior on general BR

state graphs and give a knapsack-style IPG instance illustrating worst-case behavior.

- Building on this framework, we propose *BRD-incorporated zero-regret* (BZR), which embeds RR-BRD as a randomized local-search subroutine within the ZR [Dragotto and Scatamacchia, 2023]. Using solver callbacks, BZR invokes RR-BRD to search for and supply PNEs, while ZR adds equilibrium inequalities to tighten the formulation. In BZR, integer-feasible incumbents produced by the mixed-integer program (MIP) solver provide randomized initial strategy profiles for RR-BRD, and equilibrium inequalities are generated from the best-response profiles encountered along RR-BRD trajectories. We also discuss how RR-BRD can serve as a modular local-search subroutine within other IPG algorithms.
- We introduce edge-weighted budgeted maximum coverage (EBMC) games, a novel class of non-cooperative integer programming games where players maximize edge coverage under budget constraints on a network. We examine two variants of the players’ utility functions—locally altruistic and selfish—and prove the existence of a PNE in locally altruistic EBMC games via potential game arguments, as well as provide sufficient conditions for the existence of a PNE in selfish EBMC games.
- We conduct extensive experiments to assess the scalability and effectiveness of RR-BRD and BZR on synthetic EBMC and KPG instances with up to 30 players. Across 135 test cases (27 per variant for the two EBMC and three KPG types), RR-BRD finds a PNE in 128 instances. When embedded in ZR, BZR substantially improves performance: it finds at least one PNE in 128 instances (vs. 33 for ZR) and identifies 179 PNEs for EBMC (vs. 35) and 3160 for KPGs (vs. 38). Overall, RR-BRD is effective both as a standalone algorithm and as a local-search subroutine that greatly improves ZR’s scalability.
- We implement our algorithms on a real-world EBMC instance with 84 county planners in Minnesota’s AIS inspection programs. For the selfish EBMC game, RR-BRD finds a socially optimal PNE; without this algorithm, it would be unclear whether a PNE exists, making PNE-based benchmarks difficult to use. For the locally altruistic EBMC game, we identify a socially optimal solution that is also a PNE. Comparing these equilibria, we show how the behavioral specification (selfish vs. locally altruistic) changes both system-wide performance and the distribution of individual utilities, with locally altruistic equilibria often yielding higher individual utilities.

This paper is organized as follows. Section 2 provides preliminaries on IPGs, normal-form games, and the best-response state graph. Section 3 presents our RR-BRD framework and its probabilistic convergence guarantees. Section 4 shows how RR-BRD can be incorporated into the ZR framework and other algorithms. Section 5 formally defines EBMC games as an illustrative IPG, highlighting variations in utility functions across players. Section 6 establishes conditions for the existence of PNEs in EBMC games. Section 7 reports computational experiments that demonstrate the effectiveness and generalizability of RR-BRD and BZR on both random and real-world datasets, including random knapsack game datasets representing broader classes of resource-constrained IPGs. Finally, Section 8 offers concluding remarks. For quick reference on acronyms and notation used throughout the paper, see Appendix 1.

2 Preliminaries

This section provides the modeling and state-graph foundations used throughout the paper. Section 2.1 defines IPGs and PNE, and Section 2.2 introduces the normal-form best-response state graph (and its Markov-chain reduction) used to describe and analyze our RR-BRD in Section 3.

2.1 Integer Programming Games and Pure Nash Equilibria

We introduce the fundamental definitions and concepts related to IPGs and PNEs. Following Köppe et al. [2011], an IPG is defined as a tuple $\mathcal{G} = (N, (\mathcal{X}_i)_{i \in N}, (u_i)_{i \in N})$, where $N := \{1, \dots, n\}$ denotes the set of players, $\mathcal{X}_i := \{\mathbf{x}^i \in \mathbb{Z}^{m_i} : A^i \mathbf{x}^i \leq \mathbf{b}^i\}$ represents the pure-integer strategy set for player i with rational matrix A^i , vector \mathbf{b}^i , and dimension m_i , and $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) : \mathcal{X} \rightarrow \mathbb{R}$ is the utility function for player i , parametrized by the strategies \mathbf{x}^{-i} of all other players. Here, $\mathcal{X} := \mathcal{X}_1 \times \dots \times \mathcal{X}_n$ denotes the joint strategy space.

Throughout, we assume that each \mathcal{X}_i is nonempty and *polyhedrally bounded*, so that each player has a finite number of pure strategies and the joint strategy space \mathcal{X} is finite. This assumption is used when relating IPGs to finite normal-form games in the sequel.

The utility function of player i is often expressed as

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{x}^i)^\top Q^i \mathbf{x}^i + (\mathbf{d}^i)^\top \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i, \quad (1)$$

where Q^i is a symmetric matrix (not necessarily positive semidefinite), \mathbf{d}^i is a linear coefficient vector, and the terms $(\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i$ capture inter-player interactions through *reciprocally bilinear* terms.

In an IPG, players act non-cooperatively, each solving an individual integer program (IP) to maximize their utility. The best-response problem for player i is

$$\max_{\mathbf{x}^i} \{u_i(\mathbf{x}^i, \mathbf{x}^{-i}) : \mathbf{x}^i \in \mathcal{X}_i\}. \quad (2)$$

We assume that the best-response problems (2) are tractable and solved to optimality.

A *strategy profile* $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$ is a collection of all players' strategies. A profile $\hat{\mathbf{x}}$ is a *pure Nash equilibrium (PNE)* if, for each player $i \in N$, the strategy $\hat{\mathbf{x}}^i$ is a best response (BR) to the others' strategies $\hat{\mathbf{x}}^{-i}$, i.e., $u_i(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^{-i}) \geq u_i(\mathbf{x}^i, \hat{\mathbf{x}}^{-i})$, $\forall \mathbf{x}^i \in \mathcal{X}_i$. Carvalho et al. [2018] show that deciding whether a general IPG admits a PNE is a Σ_2^P -complete problem and that PNEs may fail to exist even in simple two-player IPGs.

We define the social welfare function as $\phi(\mathbf{x}) := \sum_{i \in N} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$, and refer to a *socially optimal* strategy profile (or *optimal social-welfare (OSW) solution*) as $\hat{\mathbf{x}}_{\text{sw}}^* \in \arg \max_{\mathbf{x} \in \mathcal{X}} \phi(\mathbf{x})$. Let $\mathcal{S}_{\text{pne}}(\mathcal{G})$ denote the set of all PNEs of the IPG instance \mathcal{G} . The *best PNE* is then defined as $\hat{\mathbf{x}}_{\text{pne}}^* \in \arg \max_{\mathbf{x} \in \mathcal{X} \cap \mathcal{S}_{\text{pne}}(\mathcal{G})} \phi(\mathbf{x})$. The *price of stability (POS)* [Roughgarden, 2010], a common measure of inefficiency in game theory, is defined as the ratio $\text{POS} = \frac{\phi(\hat{\mathbf{x}}_{\text{sw}}^*)}{\phi(\hat{\mathbf{x}}_{\text{pne}}^*)}$. A larger POS indicates a larger gap in social welfare between the best PNE and the socially optimal solution.

IPGs as finite normal-form games. When each player’s pure-integer feasible set \mathcal{X}_i is finite and polyhedrally bounded, an IPG can be viewed as a finite normal-form game by setting $S_i := \mathcal{X}_i$ as the pure strategy set of player i and defining utilities u_i at each joint profile to equal the IP objective values. This representation preserves payoffs and best responses and induces the same best-response state graph and Markov chain as those defined below. We emphasize that this is a conceptual viewpoint—explicit enumeration of $\prod_{i \in N} \mathcal{X}_i$ is generally intractable—and our methods avoid it by computing BRs via player-specific IPs.

2.2 Normal-Form Games, Best-Response State Graph, and Markov-Chain Reduction

Let $N = \{1, \dots, n\}$ denote the players. For each $i \in N$, let S_i be a *finite* set of pure strategies and set $S := \prod_{i \in N} S_i$. Utilities $u_i : S \rightarrow \mathbb{R}$ are arbitrary but deterministic. For each $i \in N$ and profile $s = (s_i)_{i \in N} \in S$, define the set of strictly improving best responses (BRs) as $\mathcal{BR}_i^+(s) := \{t_i \in \arg \max_{r_i \in S_i} u_i(r_i, s_{-i}) : u_i(t_i, s_{-i}) > u_i(s_i, s_{-i})\}$. We say that player i has an *improving BR move* at s if $\mathcal{BR}_i^+(s) \neq \emptyset$; otherwise, player i is already best-responding at s (possibly with ties), and we regard this as a *no-move* situation for player i .

Best-response state graph and a time-homogeneous Markov chain. The *best-response (BR) state graph* is $\mathcal{U} = (S, E)$ with a directed edge $s \rightarrow (t_i, s_{-i})$ for each $s \in S$, $i \in N$, and $t_i \in \mathcal{BR}_i^+(s)$; that is, edges correspond to strictly improving BR moves. We also interpret no-move updates as self-loops $s \rightarrow s$. A profile s^* is a PNE iff $\mathcal{BR}_i^+(s^*) = \emptyset$ for all $i \in N$ —equivalently, s^* has no outgoing strictly improving BR edges, so all BR updates reduce to self-loops.

Fix a scheduler π with $\pi_i > 0$ for all $i \in N$ (e.g., uniform $\pi_i = 1/n$). For each player $i \in N$ and state $s \in S$, we specify a *tie-breaking kernel* $\lambda_i(\cdot | s)$, a probability distribution on S , such that:

- if $\mathcal{BR}_i^+(s) \neq \emptyset$, then $\lambda_i(\cdot | s)$ is supported on the improving BR successors $\{(t_i, s_{-i}) : t_i \in \mathcal{BR}_i^+(s)\}$, and each such successor has positive probability (for instance, uniform choice over $\mathcal{BR}_i^+(s)$);
- if $\mathcal{BR}_i^+(s) = \emptyset$, then $\lambda_i(s | s) = 1$, i.e., player i does not move when already best-responding (even if there are other tied best responses).

We then define transition probabilities

$$P(s, s') := \sum_{i \in N} \pi_i \lambda_i(s' | s), \quad s, s' \in S. \quad (3)$$

Thus $(X_t)_{t \geq 0}$ with $X_{t+1} \sim P(X_t, \cdot)$ is a finite, time-homogeneous Markov chain whose positive-probability transitions are exactly the edges of \mathcal{U} (including self-loops). By the Markov property, the next state depends on the current state and a fresh draw $I_t \sim \pi$ together with the tie-breaking kernels. At a PNE s^* , $\mathcal{BR}_i^+(s^*) = \emptyset$ for all i , hence $\lambda_i(s^* | s^*) = 1$ and $P(s^*, s^*) = 1$, so $\{s^*\}$ is an absorbing class. Standard notions such as communicating classes, recurrent and transient states, hitting probabilities, and expected hitting times apply.

3 Convergence of Best-Response Dynamics in Finite Normal-Form Games

We establish convergence guarantees for random-restart best-response dynamics (RR-BRD) in IPGs via the following steps: (i) cast an IPG with finite pure-integer feasible sets as a finite normal-form game; (ii) construct the best-response (BR) state graph; (iii) compute its strongly connected components (SCCs) and the associated condensation directed acyclic graph (DAG); and (iv) identify sink SCCs, distinguishing singleton *PNE sinks* from multi-node *non-PNE sinks*. Steps (i)–(ii) are described in Section 2. The baseline fact is that a random walk on the BR state graph eventually enters a sink SCC and remains there; thus it converges either to a singleton PNE sink or to a non-PNE sink (trap) [Goemans et al., 2005]. Our addition is to introduce *random restarts* aimed at reaching singleton PNE sinks, yielding a Las Vegas–style guarantee: if a PNE exists and the restart distribution places positive mass on any state from which some PNE sink is reachable, then RR-BRD finds (and certifies) a PNE almost surely.

3.1 SCCs, the Condensation DAG, and Recurrent Classes

Let $\text{SCC}(\mathcal{U})$ denote the set of strongly connected components of \mathcal{U} , and let \mathcal{D} be the *condensation DAG* of \mathcal{U} , obtained by contracting each SCC (so \mathcal{D} is finite and acyclic). An SCC S is a *sink SCC* if and only if no BR edge leaves S (equivalently, the node corresponding to S is a sink in \mathcal{D}). We distinguish: (i) a *PNE sink*, i.e., a singleton sink SCC $\{s^*\}$ (necessarily a PNE), and (ii) a *non-PNE sink* (or *trap*), i.e., a sink SCC S with $|S| > 1$. This structure is illustrated in Figure 1.

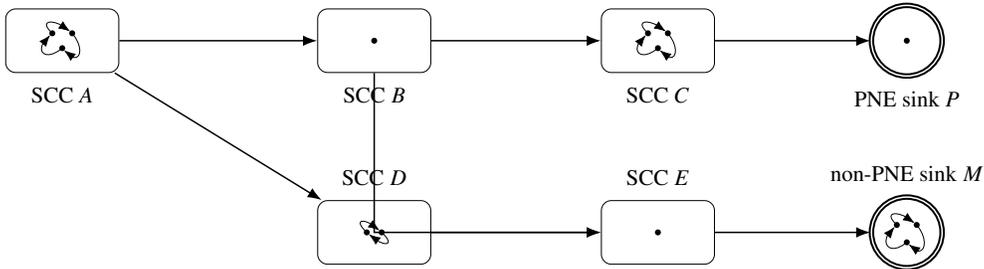


Figure 1: Illustration of a Connected Condensation DAG with Two Terminal Sinks: a singleton *PNE sink* P and a multi-node *non-PNE sink* M . Boxes and circles are SCCs; internal dots depict the number of profiles each SCC contains (singletons vs. multi-node SCCs). Arrows are edges of the acyclic condensation graph. SCC labels (A, B, C, D, E, P, M) are arbitrary and used solely for illustration. In potential or BR-weakly acyclic games, non-PNE sinks do not occur; all sink SCCs are singleton PNEs, so the branch labeled M is absent.

Figure 1 illustrates that best-response dynamics may converge either to a singleton PNE sink or to a non-PNE sink, from which escape is impossible—motivating the use of random restarts developed in Section 3.3. Because (3) assigns positive probability exactly to BR edges, the Markov chain’s communicating classes coincide with the SCCs of \mathcal{U} ; in particular, its *recurrent classes* are precisely the *sink SCCs*. Thus, every PNE is a singleton recurrent (absorbing) class, while any non-PNE sink is a closed, non-singleton trap. BRD as a random walk (i.e., equivalently, with randomized playing sequences) samples $I_t \in N$ i.i.d. with $\Pr(I_t = i) = \pi_i > 0$ and, conditional on $I_t = i$, updates s_{t+1} by applying one improving best-response move for player i according to $\lambda_i(\cdot | s_t)$. This produces exactly the Markov chain with transition kernel P in (3).

Randomized playing sequences [Heinrich et al., 2023] can help escape *non-sink* cycles (for example,

cycles in an SCC with at least one outgoing edge, such as the SCCs labeled *A*, *C* and *D* in Figure 1), whereas a fixed update order can get stuck there (i.e., the fixed update order can prevent the escaping BR move). However, using a randomized playing sequence is not enough; once the chain enters a non-PNE trap, it cannot leave. In particular, when non-PNE traps exist, a random playing sequence alone does *not* ensure reaching a PNE, as emphasized in the sink-equilibrium framework of Goemans et al. [2005].

3.2 Deterministic Convergence in Structured Subclasses

Potential games and best-response weak acyclicity. (compare Fig. 1: non-PNE sink is absent in these subclasses) A finite game is a (weighted) *potential game* if some $\psi : S \rightarrow \mathbb{R}$ tracks unilateral improvements: for all i and $t_i \in S_i$, $u_i(t_i, s_{-i}) - u_i(s_i, s_{-i}) = w_i(\psi(t_i, s_{-i}) - \psi(s_i, s_{-i}))$ with $w_i > 0$. By the finite improvement property, every BR path strictly increases ψ and therefore reaches a PNE in finitely many steps [Monderer and Shapley, 1996]; classic subclasses include congestion games [Rosenthal, 1973] and player-specific congestion games [Milchtaich, 1996]. A game is *BR-weakly acyclic* if every sink SCC in the BR state graph is a singleton PNE (equivalently, from every start, there exists a finite BR path to a PNE). In such games, *every* BR trajectory (under any scheduler) reaches a PNE in finite time because infinite trajectories would have to reside in a non-PNE sink, which is excluded by definition; see, e.g., Apt and Simon [2015] for systematic treatments and related characterizations.

Bridge to the general case. Outside these structured subclasses, the BR state graph may contain non-PNE trap components, which can trap BRD even under randomized playing sequences (see the sink-equilibrium viewpoint of Goemans et al. 2005). The next subsection shows that introducing *random restarts* yields Las Vegas-style guarantees and instance-dependent bounds in arbitrary finite games.

3.3 Round Random-Restart Best-Response Dynamics (RRR-BRD)

RRR-BRD is a round-based variant of BRD that incorporates *random restarts*. A *restart* means that if the dynamics fails to reach a PNE within a prescribed cap, we abandon the current trajectory, reinitialize the strategy profile (according to a restart distribution), and repeat. Instead of updating one player at a time, RRR-BRD *batches* updates into *rounds*: in each round, every player plays one improving BR move, in a uniformly random order. This makes local PNE certification simple: if no player changes in a round (i.e., no improving BR exists when each player is selected), then the current profile is a PNE. Hereafter, we use RRR-BRD to denote the implemented algorithm; we retain RR-BRD as a conceptual umbrella term.

The guarantees we develop are *Las Vegas* rather than deterministic: randomness affects the trajectory and runtime, but the algorithm never returns an incorrect solution—whenever it returns, it outputs a *certified* PNE. If a PNE exists and a capped attempt succeeds with positive probability, then independent unbounded restarts imply almost-sure convergence to a PNE. We now present RRR-BRD (Algorithm 1).

Algorithm 1 Round Random-Restart BRD (RRR–BRD)

Require: restart law μ on S ; round cap $R \in \mathbb{N}$

```
1: while true do
2:   sample  $s \sim \mu$  ▷ fixed-cap restart: if no PNE within  $R$  rounds, restart
3:   for  $r = 1, 2, \dots, R$  do
4:     draw a random permutation  $\Pi$  of  $\{1, \dots, n\}$  ▷ uniform
5:      $s^{\text{start}} \leftarrow s$ 
6:     for  $i$  in order  $\Pi$  do apply one improving best-response update for player  $i$ 
7:     if  $s = s^{\text{start}}$  then return  $s$  ▷ PNE certified by end-of-round no-change
```

Non-PNE sinks and the avoidance of trap detection. The BR state graph may contain non-singleton SCCs and, in particular, non-PNE traps. A natural question is whether Algorithm 1 could implement a “stop-on-trap” rule—e.g., restart immediately (Line 2) once the trajectory enters a non-PNE trap. While appealing in principle, this requires detecting sink membership, which is computationally intractable in general for succinctly represented games. To the best of our knowledge, there is no general complexity classification of the IN-A-SINK problem for arbitrary succinct normal-form representations; existing hardness results are established for specific classes. For example, IN-A-SINK is PSPACE-complete for graphical games/BGP [Fabrikant and Papadimitriou, 2008], and across several natural game classes (e.g., anonymous, player-specific or weighted congestion, valid-utility, and two-sided markets) sink membership and the existence of non-singleton sinks are PSPACE-complete, while even PNE existence can be NP-hard [Mirrokni and Skopalik, 2009]. Accordingly, our algorithm avoids trap detection entirely: Algorithm 1 uses only inexpensive *local* checks for PNE certification and fixed-cap restarts, and never requires constructing or reasoning over the full BR state graph.

Round-based (macro) Markov chain and hitting times. For a permutation π of $\{1, \dots, n\}$, let F_π denote the (random) operator that applies exactly one improving BR update for each player in the order π , using the tie-breaking kernels $(\lambda_i)_{i \in N}$ from Section 2.2. If $(\Pi_r)_{r \geq 0}$ are i.i.d. uniform permutations, then the round-boundary process $S_{r+1} = F_{\Pi_r}(S_r)$, $r = 0, 1, 2, \dots$, is a time-homogeneous Markov chain on the finite state space S . PNEs are still absorbing states. Moreover, any sink SCC of the stepwise BR state graph remains closed under round transitions, so the recurrent classes are unchanged. For any $C \subseteq S$, define the (round) hitting time $\tau_C := \inf\{r \geq 0 : S_r \in C\}$ and write τ_{PNE} for the hitting time of the set of PNE.

Restarts and per-attempt success probability. A restart law μ is any probability distribution on S . For a fixed round cap $R \geq 1$, define the per-attempt success probability

$$q_\mu[R] := \mathbb{E}_{s \sim \mu} [\Pr(\tau_{\text{PNE}} \leq R)], \quad (4)$$

i.e., the probability that a single RRR–BRD attempt started from $s \sim \mu$ reaches (and certifies) a PNE within at most R rounds. Note that $q_\mu[R] > 0$ whenever μ assigns positive probability to at least one state from which a PNE is reachable via improving BR moves within R rounds. (In contrast, Hakim et al. [2024] study

limit (infinite-horizon) absorption probabilities into *sink* SCCs under a different noisy dynamics.)

We now establish the following result; proof is given in Appendix 2.

Theorem 1 (RRR–BRD: almost surely success and expected complexity). *Assume a PNE exists and the restart law μ assigns positive mass to states from which a PNE is reachable via improving best-response moves. If $q_\mu[R] > 0$, then: (i) RRR–BRD finds a PNE with probability 1 and (ii) The expected total number of best-response solves is bounded by $\mathbb{E}[\#BR \text{ solves}] \leq \frac{nR}{q_\mu[R]}$.*

Remark 1. *If a PNE exists and μ has full support on S , then $q_\mu[R] > 0$ for all $R \geq 1$ because μ assigns positive probability to every state, including each PNE state (and thus a PNE can be reached within one capped attempt). Therefore, the condition $q_\mu[R] > 0$ in Theorem 1 holds under any full-support restart law and more generally whenever μ assigns positive mass to the attraction basin of some PNE.*

Theorem 1 concerns the idealized unbounded-restart variant; in computation we cap the number of restarts at L and report failure if no PNE is reached. If each BR solve takes time at most T_{BR} , then the expected time until discovery $\mathbb{E}[\text{time}] \leq \mathbb{E}[\#BR \text{ solves}] T_{BR} \leq \frac{nR}{q_\mu[R]} T_{BR}$. Here T_{BR} depends on the BR implementation and may be difficult to characterize.

RRR-BRD in IPGs. We now specialize RRR–BRD to IPGs. The scheme is analogous to Algorithm 1, but states are represented by integer-feasible strategy profiles $\mathbf{x} = (\mathbf{x}^1, \dots, \mathbf{x}^n)$, and BR updates are implemented by solving player-specific IPs. The round-based random permutation is encoded in Line 4, and random restarts are encoded in Line 8.

Algorithm 2 RRR-BRD Algorithm for Integer Programming Games

Require: IPG instance, initial strategy profile $\bar{\mathbf{x}}_{\text{init}}$, max rounds R , max restarts L

- 1: Initialize current profile $\bar{\mathbf{x}} \leftarrow \bar{\mathbf{x}}_{\text{init}}$ (optional) **or** $\bar{\mathbf{x}} \leftarrow \text{RANDOM_GENERATION}(\mathcal{X})$
 - 2: **for** $l = 1$ to L **do**
 - 3: **for** $r = 1$ to R **do**
 - 4: draw a random permutation of players $N \leftarrow \text{RANDOMPERMUTATION}(N)$
 - 5: **for** each player i in N **do**
 - 6: If an improving best response exists, compute $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i} u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ and update $\bar{\mathbf{x}} \leftarrow (\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$.
 - 7: **if** consecutive profiles are identical **then return** $\bar{\mathbf{x}}$ \triangleright PNE certified by end-of-round no-change
 - 8: $\bar{\mathbf{x}} \leftarrow \text{RANDOM_GENERATION}(\mathcal{X})$ \triangleright restart: generate a new initial strategy profile
 - 9: **return** None \triangleright no PNE found within caps R, L
-

We now present the following corollary; proof is given in Appendix 2.

Corollary 1 (RRR–BRD for IPGs). *Consider an IPG $\mathcal{G} = (N, (\mathcal{X}_i)_{i \in N}, (u_i)_{i \in N})$ with polyhedrally bounded feasible sets $\mathcal{X}_i = \{\mathbf{x}^i \in \mathbb{Z}^{m_i} : A^i \mathbf{x}^i \leq b^i\}$, so that $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ is finite and can be viewed as the state space S of a finite normal-form game. Let μ be the restart law induced by $\text{RANDOM_GENERATION}(\mathcal{X})$ in Algorithm 2. If \mathcal{G} admits a PNE and $q_\mu[R] > 0$, then Algorithm 2 with unbounded restarts ($L \rightarrow \infty$) finds a PNE with probability 1 and satisfies $\mathbb{E}[\#BR \text{ solves}] \leq \frac{nR}{q_\mu[R]}$. In particular, if RANDOM_GENERATION has full support on \mathcal{X} and a PNE exists, then $q_\mu[R] > 0$ for every $R \geq 1$ (Remark 1).*

Any feasible-solution generator with broad coverage over \mathcal{X} can be used to implement μ . We provide explicit random feasible-solution generators—both maximal and full-support variants—for KPG and EBMC

in Appendix 3. We alternate these variants across restarts; the resulting mixture retains full support on \mathcal{X} . We also cap R and L , since we do not know *a priori* whether a PNE exists: if no PNE exists, the conceptual RRR–BRD would never terminate. Moreover, when integrating with a MIP solver, the solver supplies natural initial profiles, so imposing finite caps is practical.

In terms of runtime, if each BR solve takes at most $T_{\mathcal{BR}}$, then the expected runtime bound follows immediately by multiplying the BR-solve bound by $T_{\mathcal{BR}}$. In IPGs, each BR solve is implemented by solving a player-specific IP to optimality, so $T_{\mathcal{BR}}$ can be difficult to characterize theoretically. In practice, one may treat $T_{\mathcal{BR}}$ as an empirical per-solve runtime estimate under the chosen solver settings.

3.4 Complexity Analysis for RRR–BRD

The practical question is how to select the round cap R in RRR–BRD. As R increases, $q_\mu[R]$ (4) is nondecreasing, creating a trade-off in expected time. Taking R to be very large is impractical in a failed attempt, and the idealized “stop-on-trap” behavior is unavailable. Throughout our study, we therefore fix a modest cap $R \in [10, 20]$, which empirically balances per-attempt cost and $q_\mu[R]$.

Because the BR state graph can contain SCCs of size > 1 and non-PNE traps, performance can be poor even when a PNE exists: if $q_\mu[R]$ is extremely small, then after r restarts the total work nRr can exceed crude enumeration scales with non-negligible probability. In particular, as a *worst-case event*, more than r restarts are required (equivalently, the algorithm fails for r consecutive attempts), which occurs with probability $\Pr(R^* > r) = (1 - q_\mu[R])^r$. See Appendix 4 for a schematic and a knapsack-style example where most restarts are absorbed by a non-PNE trap and $q_\mu[R]$ is exponentially small. As a *best-case event*, the first attempt can succeed, which occurs with probability $q_\mu[R]$. Very roughly, $q_\mu[R]$ captures instance difficulty for RRR–BRD under μ and R . Our guarantees are Las Vegas–style: success occurs almost surely when $q_\mu[R] > 0$, while efficiency is governed by these instance-dependent success probabilities.

Since $q_\mu[R]$ is not directly computable in large IPGs, we estimate it empirically via Monte Carlo simulation over restart states. This aligns with the sampling-and-simulation approach used to estimate absorption behavior on response graphs [Hakim et al., 2024]. While their work defines transition via weakly improving deviations and studies infinite-horizon absorption into sink SCCs under noisy dynamics, we estimate a finite-horizon probability of hitting the PNE set within $\leq R$ rounds under improving best-response updates.

Given a restart law μ , we estimate $\tilde{q}_\mu[R] := \mathbb{E}_{s \sim \mu}[\mathbf{1}\{\text{RRR–BRD from } s \text{ reaches a PNE within } R \text{ rounds}\}]$ using Monte Carlo by averaging the indicator over a finite set of starts. We report `avg_rounds`, i.e., the mean number of rounds over all attempts (using R for failures and the realized round count for successes). Furthermore, based on $\tilde{q}_\mu[R]$, we report the expected step bound (ESB) for each instance.

We examine RRR–BRD in two settings:

- (i) **Normal-form reductions of small IPGs.** We consider instances of the form $\max_{\mathbf{x}^i \in \{0,1\}^m} \{\mathbf{d}_i^\top \mathbf{x}^i + \sum_{p \in N: p \neq i} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i\}$, where $\mathbf{d}_i \in \mathbb{Z}^p$, and $Q_p^i \in \mathbb{Z}^{m \times m}$, with entries in $[-100, 100]$. For $n \in \{2, 3\}$ and $m \in \{3, 4\}$, we sample 10 random games per (n, m) , retaining only instances that admit at least one PNE. For each instance, μ is taken from a uniform distribution over all pure profiles S . Since $|S|$ is modest, we set $R = 10$ and estimate $\tilde{q}_\mu[R]$ by testing every starting state.

(ii) **Knapsack problem games (KPG)**. (see Section 7.4 for the full definition). We implement RRR–BRD for IPGs (Algorithm 2) with a round cap $R = 20$ and a restart cap $L = 100$. μ is taken from the empirical distribution over 100 randomly generated feasible starts per instance using a feasible-solution generator (Appendix 3); $\tilde{q}_\mu[R]$ is then estimated from these starts. We use the KPG instances from Dragotto and Scatamacchia [2023] for which at least one PNE has been verified in their experiments.

The summaries for (i) and (ii) appear in Tables 1 and 2. Full results are in Appendix 5. In all simulated cases $\tilde{q}_\mu[R] > 0$, and the average $\tilde{q}_\mu[R]$ value often exceeds 0.5, implying high success probability with a modest number of restarts. While $|S|$ can be counted exactly for (i), the number of integer-feasible profiles for (ii) cannot be counted exactly; nevertheless, it is expected to be substantially larger than in (i) due to the many binary decisions, even under knapsack constraints. Note that the average ESB is a conservative upper bound; the realized number of best-response solves is often much smaller.

Table 1: Normal-Form Reductions: Empirical Success Within $R = 10$ Rounds Under Uniform Restarts.

n	k	$ S $	# inst. with $\tilde{q}_\mu[R] > 0$	avg. $\tilde{q}_\mu[R]$	avg_rounds	avg_ESB ($\frac{nR}{q_\mu[R]}$)
2	3	64	10/10	1.00	2.6	20
2	4	256	10/10	0.925	3.2	26
3	3	512	10/10	0.58	6.4	94.4
3	4	4096	10/10	0.72	5.7	62.4

Table 2: KPGs: Empirical Success Within $R = 20$ Rounds Under Random Restarts.

type	# inst. with $\tilde{q}_\mu[R] > 0$	avg. $\tilde{q}_\mu[R]$	avg_rounds	avg_ESB ($\frac{nR}{q_\mu[R]}$)
A	23/23	1	3.6	49.6
B	20/20	1	3.4	48
C	7/7	0.60	10.1	2135.2

Types (A) and (B) draw the interaction coefficients equally and uniformly from the set in $[1, 100]$ and independently and uniformly from the set in $[1, 100]$, respectively. Type (C) draws them from a uniform distribution over $[-100, 100]$.

This Monte Carlo exercise is used only to contextualize complexity, not as a prerequisite for implementation. Note that the estimate $\tilde{q}_\mu[R]$ is inherently *algorithm- and parameter-dependent* outside structured classes (e.g., potential games or BR-weakly acyclic games). Accordingly, we do not propose estimating $\tilde{q}_\mu[R]$ in practice; rather, we run RRR–BRD directly and treat $\tilde{q}_\mu[R]$ as an *ex post* diagnostic. In particular, even a *single* successful run that returns a PNE immediately certifies $q_\mu[R] > 0$ for that (μ, R) (and yields a PNE to report), whereas failure to hit a PNE within a finite budget does not rule out $q_\mu[R] > 0$.

4 Best-response Dynamics Incorporated Zero-regret Algorithm

This section develops our algorithmic framework for computing PNEs in large-scale IPGs. We first review the lifted formulation and equilibrium inequalities (EIs) used in ZR (Section 4.1), and then present our best-response-incorporated variant and implementation details (Section 4.2). We also show how RRR–BRD can be used as a randomized local-search subroutine (Section 4.3) and, if it terminates without identifying any PNE, how it can be adapted to provide an approximate-equilibrium certificate (Section 4.4).

4.1 Lifted Space and Perfect Equilibrium Formulation

Following Dragotto and Scatamacchia [2023], we work in a lifted space in which all players' utilities are linearized. Let $\mathbf{x} := (\mathbf{x}^1, \dots, \mathbf{x}^n)$ collect the players' decision vectors and \mathbf{z} denote auxiliary variables used to linearize nonlinear terms (e.g., bilinear products). Let \mathcal{K} be the lifted feasible set, $\mathcal{K} := \{(\mathbf{x}, \mathbf{z}) : \mathbf{x}^i \in \mathcal{X}_i \forall i \in N, (\mathbf{x}, \mathbf{z}) \text{ satisfies all linearization constraints and bounds}\}$, and let $\text{conv}(\mathcal{K})$ be its convex hull. Any integer point $(\mathbf{x}, \mathbf{z}) \in \mathcal{K}$ corresponds to a unique strategy profile \mathbf{x} ; conversely, every pure-strategy profile has at least one lifted representative in \mathcal{K} .

Let $\mathcal{S}_{\text{pne}}(\mathcal{G})$ denote the set of all PNE profiles of the IPG instance \mathcal{G} . Define the *perfect equilibrium formulation* as $\mathcal{E} := \{(\mathbf{x}, \mathbf{z}) \in \text{conv}(\mathcal{K}) : \mathbf{x} \in \text{conv}(\mathcal{S}_{\text{pne}}(\mathcal{G}))\}$. Let $\mathcal{BR}_i(\mathbf{x}^{-i})$ denote the set of BRs of player i to a profile $\mathbf{x}^{-i} \in \prod_{k \in N, k \neq i} \mathcal{X}_k$. ZR is a cutting-plane algorithm using EIs, that is, linear inequalities valid for \mathcal{E} . Such inequalities are obtained from BRs: for any player i and any BR $\hat{\mathbf{x}}^i \in \mathcal{BR}_i(\mathbf{x}^{-i})$, the inequality

$$u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\mathbf{x}^i, \mathbf{x}^{-i}), \quad (5)$$

is valid for all $(\mathbf{x}, \mathbf{z}) \in \mathcal{E}$ (with utilities evaluated in the lifted space). Dragotto and Scatamacchia [2023] show that collecting all such inequalities over all players and all BRs yields an *equilibrium closure* that coincides with \mathcal{E} (Theorem 1 in Dragotto and Scatamacchia [2023]).

Given a candidate point $(\bar{\mathbf{x}}, \bar{\mathbf{z}}) \in \text{conv}(\mathcal{K})$, the *equilibrium separation oracle* solves, for each player i , $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} \{u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})\}$. If $u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) > u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$, the associated EI (5) is violated at $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ and is added as a cut; if no such violation occurs for any i , then $\bar{\mathbf{x}}$ is a PNE.

The original ZR algorithm iteratively solves a master problem of the form

$$\max\{\phi(\mathbf{x}, \mathbf{z}) : (\mathbf{x}, \mathbf{z}) \in \mathcal{K}, (\mathbf{x}, \mathbf{z}) \text{ satisfies all cuts in } \Omega\}, \quad (6)$$

where Ω is a growing family of EIs. At each iteration, ZR solves (6) to optimality in the lifted space, queries the equilibrium separation oracle at the optimal solution $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, and either (i) certifies $\bar{\mathbf{x}}$ as a welfare-maximizing PNE or (ii) adds violated EIs to Ω and repeats. This yields a cutting-plane method that is provably correct and finitely terminating whenever a PNE exists. We note that regardless of the choice of the objective function in (6), the capability of computing a PNE via the separation procedure remains valid.

Motivation for the BZR algorithm. The BZR algorithm is motivated by the complementary strengths and weaknesses of ZR and RRR-BRD.

(i) *Limitations of ZR and how RRR-BRD complements it.* For large-scale IPGs, the ZR approach faces several practical challenges. Solving the joint optimization problem (6) to optimality can be very expensive when \mathcal{X} encodes many players and variables. Even when the POS is small, many cuts may be needed to move from the socially optimal solution $\bar{\mathbf{x}}_{\text{sw}}^*$ to $\hat{\mathbf{x}}_{\text{pne}}^*$. Moreover, the original implementation only adds EIs for BRs that are currently violated, even though additional BRs could, in principle, be used to cut off larger portions of the non-equilibrium region. RRR-BRD (Algorithm 2) addresses these issues by rapidly finding PNEs at scale and by generating many BRs along its trajectories. Embedding RRR-BRD as a randomized local-search subroutine inside ZR supplies PNEs in instances where plain ZR struggles to find any PNE

within the time limit, and provides a rich pool of best responses that can be recycled as additional EIs.

(ii) *Limitations of RRR-BRD and how ZR complements it.* RRR-BRD itself has only probabilistic guarantees: it can efficiently find PNEs when they exist but cannot certify that no PNE exists. When integrated with the lifted ZR formulation, RRR-BRD benefits from diverse, welfare-oriented starting points produced by the MIP solver through callbacks, which inject both randomness and structure into the initial profiles. At the same time, the combined BZR algorithm inherits ZR’s structural capabilities: given sufficient time, ZR can certify “no PNE” by proving infeasibility of the equilibrium formulation and can be configured to enumerate PNEs (e.g., via Hamming-distance cuts). Thus, ZR supplies the certification and enumeration machinery that RRR-BRD lacks, while RRR-BRD supplies the scalable search capability that ZR lacks.

4.2 Algorithm Details

We now present the BZR algorithm (Algorithm 3) and illustrate its behavior in Figure 2, which depicts the joint strategy space \mathcal{X} for two players. The BZR algorithm can be viewed as a combination of a randomized local-search subroutine and a cutting-plane method with delayed (lazy) separation of EIs.

Callback function and separation procedure: We use the MIP solver’s callback mechanism that triggers on integer-feasible solutions (e.g., MIPSOL in Gurobi [Gurobi Optimization, 2024]). While many types of intervention in the branch-and-bound tree are possible (e.g., custom branching rules), we restrict our use of callbacks to (i) collect integer-feasible points and (ii) inject EIs. Each incumbent integer solution $\bar{\mathbf{x}}$ returned by the callback is passed to the separation procedure (Lines 4–7). The oracle checks whether $\bar{\mathbf{x}}$ is a PNE; if not, it adds violated EIs. In Figure 2, these cuts are shown in green.

RRR-BRD as a randomized local-search subroutine: If $\bar{\mathbf{x}}$ is not a PNE, it is used as an initial strategy profile for RRR-BRD (Line 9). Along each RRR-BRD trajectory, we store the BRs encountered; for any BR $\hat{\mathbf{x}}^i$ whose associated EI has not yet been added, we may inject it into Ω as an additional cut (Lines 10–12). In Figure 2, the RRR-BRD trajectory is illustrated by the blue arrow, and these additional EIs are drawn in red. Note that this step is valid regardless of whether RRR-BRD finds a PNE in that run: every trajectory consists of valid BRs, hence all associated EIs are valid. If a newly found PNE improves the current best value, we update the lower bound and set $\hat{\mathbf{x}}_{\text{pne}}^* \leftarrow \bar{\mathbf{x}}$ (Line 13). If RRR-BRD fails to find a PNE within the caps (R, L) , no primal update occurs and the algorithm simply proceeds to the next callback solution.

Termination: The dual (upper) bound UB is updated by solving $UB = \max \{ \phi(\mathbf{x}, \mathbf{z}) \mid \mathbf{x} \in \mathcal{K}_{\text{relax}} \cap \Omega \}$, where $\mathcal{K}_{\text{relax}}$ denotes the current relaxation of \mathcal{K} and Ω is the set of accumulated EIs. The BZR algorithm terminates once $UB - LB \leq \text{tol}$ (Line 14–15), where tol is the solver’s default optimality tolerance. This ensures that either the best PNE has been found or that the PNE formulation is infeasible (no PNE exists).

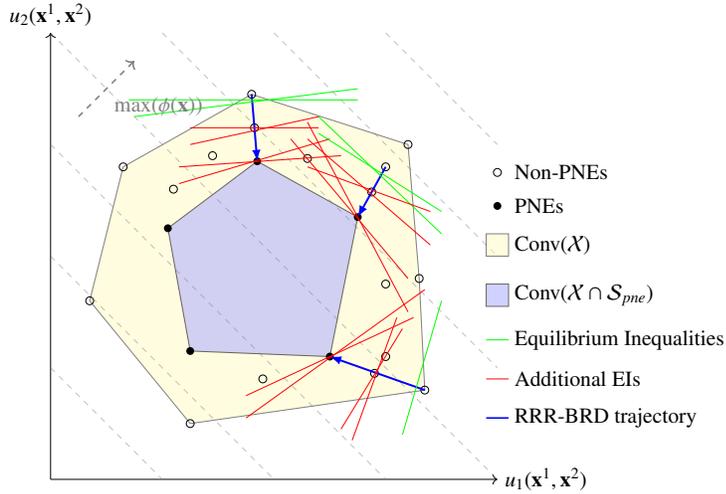
Remark 2. *The set $\mathcal{K}_{\text{relax}}$ consists of the LP relaxation of \mathcal{K} , augmented with general-purpose cutting planes generated by the solver. In our implementation, this dual bound UB may remain above $\bar{\mathbf{x}}_{\text{sw}}^*$ because BZR does not require solving $\max \{ \phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K} \cap \Omega \}$ to optimality at each iteration. Since UB is shaped jointly by general cuts and a subset of all possible EIs, we do not attempt to add every available EI. Instead, we add only a limited number per callback (e.g., up to n), and only for players whose BRs have not yet been used to generate cuts.*

Algorithm 3 Best-Response Dynamics Incorporated Zero-regret (BZR) Algorithm

Require: IPG instance, parameters max rounds R , max restarts L , tolerance tol

Ensure: Best pure Nash Equilibrium $\hat{\mathbf{x}}_{\text{pne}}^*$ or None

- 1: Initialize: $LB \leftarrow -\infty$, $UB \leftarrow \max\{\phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K}_{\text{relax}}\}$, $\Omega \leftarrow \emptyset$, $\mathcal{H} \leftarrow \emptyset$ (\mathcal{H} stores best responses)
 - 2: **while** $UB - LB > \text{tol}$ and time limit not exceeded **do**
 - 3: $\bar{\mathbf{x}} \leftarrow$ incumbent integer solution from MIPSOL of $\max\{\phi(\mathbf{x}, \mathbf{z}) \mid \mathbf{x} \in \mathcal{K} \cap \Omega\}$ ▷ Callback Function
 - 4: **for** $i \in N$ **do**
 - 5: Compute a best response: $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} \{u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})\}$
 - 6: **if** $u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) > u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$ **then**
 - 7: Add $u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\bar{\mathbf{x}}^i, \mathbf{x}^{-i})$ to Ω and add $\hat{\mathbf{x}}^i$ to \mathcal{H} ▷ Add violated EIs
 - 8: **if** any violation is found ($\bar{\mathbf{x}}$ is not a PNE) **then**
 - 9: $\bar{\mathbf{x}} \leftarrow$ RRR-BRD($\bar{\mathbf{x}}, R, L$) (Algorithm 2) and store the best responses found in \mathcal{H} ▷ Apply RRR-BRD
 - 10: **for** (a limited number of) best responses $\hat{\mathbf{x}}^i \in \mathcal{H}$, $i \in N$ **do**
 - 11: **if** the EI for $\hat{\mathbf{x}}^i$ has not yet been added **then**
 - 12: Add $u_i(\hat{\mathbf{x}}^i, \mathbf{x}^{-i}) \leq u_i(\bar{\mathbf{x}}^i, \mathbf{x}^{-i})$ to Ω ▷ Add additional EIs
 - 13: **if** $\bar{\mathbf{x}}$ is a PNE and $\phi(\bar{\mathbf{x}}) \geq LB$ **then** $\hat{\mathbf{x}}_{\text{pne}}^* \leftarrow \bar{\mathbf{x}}$ and $LB \leftarrow \phi(\bar{\mathbf{x}})$
 - 14: $UB \leftarrow \max\{\phi(\mathbf{x}, \mathbf{z}) \mid (\mathbf{x}, \mathbf{z}) \in \mathcal{K}_{\text{relax}} \cap \Omega\}$
 - 15: **if** $\hat{\mathbf{x}}_{\text{pne}}^* \neq \text{None}$ **then return** $\hat{\mathbf{x}}_{\text{pne}}^*$ **else return** None ▷ Return best PNE found or No PNE found
-



Remark 3. Figure 2 depicts many EIs, with some appearing “deeper” (e.g., those that are binding at PNEs). However, we do not establish any general dominance relation among EIs. For general IPGs, we are not aware of a principled way to identify a strictly stronger subclass of EIs. A baseline observation is that including EIs associated with all BRs yields the perfect-equilibrium formulation \mathcal{E} . Accordingly, BZR focuses on generating a practically useful subset of EIs, rather than attempting a full characterization.

Figure 2: Illustration of the BZR Algorithm in a Two-Player Game.

An illustrative two-player example that works explicitly through the BZR computations is provided in the Appendix 6. To complement this small-scale walkthrough, we also include an exponential-separation example between ZR and BZR in Appendix 7.

4.3 RRR-BRD as a Randomized Local-Search Subroutine for Other Algorithms

Carvalho et al. [2023] organize IPG algorithms around three building blocks: an *approximate* phase, which constructs an inner or outer approximation of each player’s strategy set, yielding sets $\tilde{\mathcal{X}}_i$; a *play* phase, which computes a tentative equilibrium *with respect to* this approximated game; and an *improve* phase, which verifies the tentative solution against the original game and refines the approximation as needed. In existing methods, the play phase is typically implemented via one or more *joint* optimization problems that couple all (or a subset of) players’ decisions, either explicitly in normal form or implicitly through extended MIP and

complementarity-based formulations. RRR-BRD provides a complementary play-phase mechanism: rather than solving a global joint problem over a potentially very large feasible region, it performs a randomized walk on the BR state graph, repeatedly solving BR problems without ever constructing the full normal form.

Table 3 summarizes the main exact algorithms for computing Nash equilibria in IPGs, adapted from Carvalho et al. [2023] and extended with RRR-BRD. The first six methods are based on *deterministic* existence and correctness proofs for their targeted equilibrium notions (MNE or PNE) in the classes considered. By contrast, RRR-BRD provides a *probabilistic* (Las Vegas) guarantee. This trades deterministic worst-case guarantees for a random-walk-based mechanism that scales well in practice.

Table 3: Summary of Algorithms for Computing Nash equilibria in IPGs.

Method	NE Type	Approx	Play	Alg Feas	Largest IPG Instances Tested	Reference
SGM	Mixed	Inner	FP	Deter.	KPG: $n \leq 3, m \leq 100$	[Carvalho et al., 2022]
eSGM	Mixed	Inner	MIP	Deter.	KPG: $n \leq 3, m \leq 60$	[Crönert and Minner, 2024]
CnP	Mixed	Outer	CP	Deter.	KPG: $n \leq 3, m \leq 100$	[Carvalho et al., 2025]
BM	Pure	Outer	CP	Deter.	qIPG: $n \leq 6, m \leq 3$	[Sagratella, 2016]
ZR	Pure	Outer	MIP	Deter.	KPG: $n \leq 3, m \leq 100$	[Dragotto and Scatamacchia, 2023]
BnP	Pure	Outer	MIP	Deter.	qIPG: $n \leq 3, m \leq 5$	[Schwarze and Stein, 2023]
RR-BRD	Pure	Inner	BRD	Prob.	KPG: $n \leq 30, m \leq 100$, EBMC: $n \leq 84$	This paper

Note. This table is adapted from Table 1 in Carvalho et al. [2023] and extended to include RR-BRD.

‘Approx’ = approximate phase, ‘Play’ = play phase. ‘Alg Feas’ indicates whether the algorithm has a *deterministic* (Deter.) or *probabilistic* (Prob.) theoretical guarantee to find its targeted equilibrium whenever one exists in the considered class of IPGs. FP = feasibility problem, MIP = mixed-integer problem, CP = complementarity problem, KPG = knapsack problem games, qIPG = quadratic IPG. n : number of players; $m = m_i, i \in N$: number of integer variables per player.

Conceptually, RRR-BRD can be plugged into any IPG algorithm that occasionally produces an integer-feasible joint strategy profile $\mathbf{x} \in \mathcal{X}$. Once such a profile is available, RRR-BRD can be launched from \mathbf{x} (on the induced finite game) to search locally for a PNE. If a PNE is found, it can be recorded as a feasible solution without altering the logical structure or exactness guarantees of the base algorithm; if not, the base algorithm simply continues unaffected. In this sense, RRR-BRD acts as a modular, PNE-oriented local-search layer that can be invoked whenever integer-feasible joint strategies become available.

Branching-type methods (BM and BnP). In the *branching method* (BM) of Sagratella [2016] and the *branch-and-prune* (BnP) algorithm of Schwarze and Stein [2023], the search for equilibria is organized as a tree whose nodes correspond to restricted subgames defined by additional bounds or fixings on players’ variables. At each node, a continuous auxiliary problem (e.g., a relaxed NEP or complementarity system) is solved, pruning tests may discard parts of the search region, and branching refines players’ feasible sets. Integer-feasible joint profiles arise, for example, from rounding, local search, or exhaustive enumeration within a node’s box. Whenever such a profile is available, RRR-BRD can be run on the corresponding restricted subgame, using that profile as a starting point to search for a PNE within the node.

Sampling-based and outer-approximation methods (SGM, eSGM, CnP). In the sample generation method (SGM) [Carvalho et al., 2022], its enumeration variant eSGM [Crönert and Minner, 2024], and the Cut-and-Play (CnP) algorithm [Carvalho et al., 2025], an IPG is approximated by a finite normal-form game built from sampled or externally generated pure strategies. At intermediate iterations, these methods main-

tain (i) finite strategy subsets $\tilde{S}_i \subseteq S_i$ for each player and (ii) one or more integer-feasible joint profiles in $\tilde{S} := \prod_i \tilde{S}_i$ (e.g., candidate equilibria or best-response combinations in the sampled game). RRR–BRD can then be applied to the current finite game restricted to \tilde{S} , using any such profile as a starting point. If RRR–BRD reaches a PNE of the sampled game, then (by construction) this profile corresponds to a degenerate mixed Nash equilibrium of the original IPG and can be stored as a candidate solution or benchmark.

4.4 Approximate PNE Certificates upon Early Termination

RRR–BRD can also return an *approximate-equilibrium certificate* when it does not reach an exact PNE within the allotted budget. For any profile $\bar{\mathbf{x}}$, let $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ be a best response for player i , and assume $u_i(\mathbf{x}^i, \bar{\mathbf{x}}^{-i}) > 0$ for all feasible profiles. Define

$$\alpha_i(\bar{\mathbf{x}}) := \frac{u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})}{u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})}, \quad \alpha(\bar{\mathbf{x}}) := \max_{i \in N} \alpha_i(\bar{\mathbf{x}}). \quad (7)$$

Then $\bar{\mathbf{x}}$ is an $\alpha(\bar{\mathbf{x}})$ -approximate PNE in the multiplicative sense, and $\alpha(\bar{\mathbf{x}}) = 1$ if and only if $\bar{\mathbf{x}}$ is an exact PNE. Implementation and reporting details are provided in Appendix 13.

5 Edge-weighted Budgeted Maximum Coverage (EBMC) Games

This section presents the edge-weighted budgeted maximum coverage (EBMC) game as a motivating example of large-scale IPGs. Section 5.1 introduces the EBMC problem, and Section 5.2 develops two EBMC games with different utility functions. Section 5.3 characterizes the role of types and network structure, and Section 5.4 connects EBMC to AIS-prevention resource allocation.

5.1 The Edge-weighted Budgeted Maximum Coverage (EBMC) Problem

The budgeted maximum coverage (BMC) problem [Chekuri and Kumar, 2004] aims to maximize element coverage by selecting subsets within a limited budget. The EBMC problem is a specialized case of BMC, where edges represent elements, and vertices represent subsets. Both problems are NP-hard [Chekuri and Kumar, 2004], and Caskurlu et al. [2014] further demonstrate that EBMC remains NP-hard even when restricted to bipartite graphs.

We model the underlying network as a directed graph $(\mathcal{I}, \mathcal{A})$, where \mathcal{I} is the set of vertices and \mathcal{A} is the set of arcs. The players are indexed by $N := \{1, \dots, n\}$. Each player $i \in N$ controls a subset of vertices $\mathcal{I}_i \subseteq \mathcal{I}$, and these subsets form a partition of the vertex set, i.e., $\bigsqcup_{i \in N} \mathcal{I}_i = \mathcal{I}$. We write $\mathcal{A}_i \subseteq \mathcal{A}$ for the set of arcs whose endpoints both lie in \mathcal{I}_i , and use $\delta_+(\mathcal{I}_i)$ (resp. $\delta_-(\mathcal{I}_i)$) for the set of arcs with tail (resp. head) in \mathcal{I}_i and head (resp. tail) in $\mathcal{I} \setminus \mathcal{I}_i$. Each arc $(j, k) \in \mathcal{A}$ has a nonnegative weight w_{jk} , and each player $i \in N$ has a budget \mathcal{B}_i . The decision variables are $x_j \in \{0, 1\}$, indicating whether vertex $j \in \mathcal{I}$ is selected, and $y_{jk} \in \{0, 1\}$, indicating whether arc $(j, k) \in \mathcal{A}$ is *covered* (i.e., at least one of its endpoints is selected).

The Social Welfare (SW) model ((8a)–(8c)) represents the centralized planning perspective. The objective function (8a) maximizes total edge coverage over the arc set \mathcal{A} . Constraint (8b) enforces that an edge

(j, k) can be covered only if at least one of its adjacent vertices, j or k , is selected. Constraint (8c) imposes the individual budget limit for each player $i \in N$.

The Social Welfare (SW) Model

$$\max_{\substack{x_j \in \{0,1\} \quad \forall j \in \mathcal{I} \\ y_{jk} \in \{0,1\} \quad \forall (j,k) \in \mathcal{A}}} \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} \quad (8a)$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j, k) \in \mathcal{A} \quad (8b)$$

$$\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i \quad \forall i \in N. \quad (8c)$$

The optimal solution to the SW model is the OSW solution $(\bar{\mathbf{x}}_{\text{sw}}^*)$, as introduced in Section 2.

5.2 A Non-cooperative EBMC Game Among N Players

Although the SW model in (8) allocates budgets to each player, it primarily represents the objectives of the central planner, not the players. Consequently, the optimal solutions of this model may not align with the individual goals of the players. We start by defining a player-specific optimization problem that is contingent on the decisions of other players. Subsequently, we introduce two types of utility functions to model player interactions: a locally altruistic function and a selfish function, based on the different sets of arcs.

For each player $i \in N$, let \mathcal{I}_{-i} denote the set of vertices not controlled by player i , and let $\bar{\mathbf{x}}^{-i} \in \{0, 1\}^{\mathcal{I}_{-i}}$ be the corresponding decisions of all other players. The optimization model for a single player is given by:

The Single-Player (SP) Model

$$\max \quad \tilde{u}_i(\mathbf{y}) \quad (9a)$$

$$\text{s.t.} \quad y_{jk} \leq x_j + x_k \quad \forall (j, k) \in \mathcal{A} \quad (9b)$$

$$\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i \quad (9c)$$

$$x_j = \bar{x}_j^{-i} \quad \forall j \in \mathcal{I}_{-i} \quad (9d)$$

$$x_j \in \{0, 1\} \quad \forall j \in \mathcal{I}_i \quad (9e)$$

$$y_{jk} \in \{0, 1\} \quad \forall (j, k) \in \mathcal{A}. \quad (9f)$$

The SP model (9), defined by (9a)–(9f), describes player i 's problem parameterized by $\bar{\mathbf{x}}^{-i}$. When an edge connects a vertex $j \in \mathcal{I}_i$ of player i to a vertex $k \in \mathcal{I}_{-i}$ of some other player, a positive decision $x_k = 1$ allows edge (j, k) to be covered even if $x_j = 0$. Thus, the actions of other players can have a positive external effect on player i 's objective. The objective function $\tilde{u}_i(\mathbf{y})$ in (9a) is left generic so that it can be tailored to the specific interests of player i . Since each y_{jk} can be written as $y_{jk} = \max(x_j, x_k) = x_j + x_k - x_j x_k$, the objective can equivalently be expressed as a function of \mathbf{x} , denoted by $u_i(\mathbf{x})$. This equivalence enables interchangeable use of $\tilde{u}_i(\mathbf{y})$ and $u_i(\mathbf{x})$ under the condition that $y_{jk} = \max(x_j, x_k)$. While using \mathbf{y} -variables emphasizes edge-level outcomes influenced by other players, expressing the utility directly in terms of \mathbf{x} -variables reveals player-level interactions via bilinear terms, which is standard in many IPG formulations.

Including all edges in Constraints (9b) and (9f) is computationally inefficient, especially for large graphs. Instead, arc sets can be restricted to relevant edges using the refined definition introduced below.

Definition 1 (Induced Arc Sets). *Let $\mathcal{D} = (\mathcal{I}, \mathcal{A})$ be a directed graph and let the subset of vertices $\mathcal{I}_i, i \in N$ be a partition of \mathcal{I} , i.e., $\mathcal{I} := \bigsqcup_{i \in N} \mathcal{I}_i$. The induced neighborhood arc set $\mathcal{A}[\mathcal{I}_i]$ consists of all arcs that have at least one endpoint in \mathcal{I}_i . The induced inbound arc set $\mathcal{A}^-[\mathcal{I}_i]$ consists of all arcs that have their terminal points in \mathcal{I}_i . Formally, $\mathcal{A}[\mathcal{I}_i] := \{(j, k) \in \mathcal{A}_i \cup \delta_-(\mathcal{I}_i) \cup \delta_+(\mathcal{I}_i)\}$ and $\mathcal{A}^-[\mathcal{I}_i] := \{(j, k) \in \mathcal{A}_i \cup \delta_-(\mathcal{I}_i)\}$.*

Regardless of the choice of induced arc sets in Definition 1, the joint constraints of the SP model (9)–interpreted by treating the parameterized variables \bar{x}^{-i} as decision variables in Constraint (9d)–correspond to those of the SW model (8b)–(8c). Based on these arc-set definitions, we consider two variants of the EBMC game, using different utility functions : (i) *locally altruistic* utilities, where player i maximizes edge coverage over $\mathcal{A}[\mathcal{I}_i]$, i.e., all edges incident to \mathcal{I}_i (including outgoing arcs); and (ii) *selfish* utilities, where player i maximizes edge coverage over $\mathcal{A}^-[\mathcal{I}_i]$, i.e., only edges whose head lies in \mathcal{I}_i (incoming arcs). Interpreting an edge (j, k) as a risky or critical edge (e.g., for environmental risk, security, or regulatory responsibility), Case (i) models a player who shares responsibility for any risk connected to their nodes, whereas Case (ii) models a player who is concerned only with risks incident to its own vertices \mathcal{I}_i .

Locally Altruistic Game In this game, the utility function of the player i is the coverage of the edges using the induced neighborhood arcs. Each player i solves the following optimization problem:

Locally Altruistic Single-Player (LASP) Model

$$\max \quad \tilde{u}_i^{\text{Alt}}(\mathbf{y}) := \sum_{(j,k) \in \mathcal{A}[\mathcal{I}_i]} w_{jk} y_{jk} \quad (10a)$$

$$\text{s.t. } y_{jk} \leq x_j + x_k \quad \forall (j, k) \in \mathcal{A}[\mathcal{I}_i] \quad (10b)$$

$$(9c) - (9e) \quad (10c)$$

$$y_{jk} \in \{0, 1\} \quad \forall (j, k) \in \mathcal{A}[\mathcal{I}_i]. \quad (10d)$$

Note that the arc sets \mathcal{A} in Constraints (9b) and (9f) are replaced by $\mathcal{A}[\mathcal{I}_i]$. The locally altruistic game, denoted by \mathcal{G}^{Alt} , is an EBMC game with N players, where each player solves a LASP model (10), defined by (10a)–(10d). The locally altruistic game is not fully altruistic: player i 's utility depends only on arcs incident to \mathcal{I}_i not on the entire network.

Selfish Game In this game, the utility function of player i is the edge coverage of the induced inbound arc set, which contains incoming arcs controlled by player i . Then, each player i solves the following problem:

Selfish Single-Player (SSP) Model

$$\max \quad \tilde{u}_i^{\text{Self}}(\mathbf{y}) := \sum_{(j,k) \in \mathcal{A}^-[\mathcal{I}_i]} w_{jk} y_{jk} \quad (11a)$$

$$\text{s.t. } y_{jk} \leq x_j + x_k \quad \forall (j,k) \in \mathcal{A}^-[\mathcal{I}_i] \quad (11b)$$

$$(9c) - (9e) \quad (11c)$$

$$y_{jk} \in \{0, 1\} \quad \forall (j,k) \in \mathcal{A}^-[\mathcal{I}_i]. \quad (11d)$$

Note that the arc sets \mathcal{A} in Constraints (9b) and (9f) are replaced by $\mathcal{A}^-[\mathcal{I}_i]$. The selfish game, denoted by $\mathcal{G}^{\text{Self}}$, is an EBMC game with N players, where each player solves their SSP model (11), defined by (11a)–(11d). We provide an illustrative two-player example of selfish EBMC interactions and the resulting BRD trajectory in the Appendix 8.

Selfish Non-Game In this scenario, each player i solves the SSP model (11) independently, without engaging in any strategic interaction—that is, each player assumes that no other players will select any vertices. Consequently, player i determines its strategy by solving $\bar{\mathbf{x}}_{\text{ng}}^i \in \arg \max_{\mathbf{x}^i \in \{0,1\}^{\mathcal{I}_i}} u_i(\mathbf{x}^i, \mathbf{0}^{-i})$, where $\mathbf{0}^{-i}$ denotes the zero vector for all players other than i . The combination of these individually optimal strategies forms the *selfish non-game* strategy profile, denoted by $\bar{\mathbf{x}}_{\text{ng}}$.

Equilibrium Inequality for Selfish EBMC Games We will show in Section 6.1 that the OSW solution $\bar{\mathbf{x}}_{\text{sw}}^*$ is a PNE for locally altruistic games. Thus, $\bar{\mathbf{x}}_{\text{sw}}^*$ will be treated as the best PNE in this setting, and no further algorithmic comparisons are needed. In contrast, for selfish games, we derive the following equilibrium inequality (EI). For any player $i \in N$ and any best response $\hat{\mathbf{x}}^i$ to a profile $\bar{\mathbf{x}}^{-i}$, we have

$$\sum_{(j,k) \in \mathcal{A}_i} w_{jk} (\hat{x}_j + \hat{x}_k - \hat{x}_j \hat{x}_k) + \sum_{(j,k) \in \delta_-(\mathcal{I}_i)} w_{jk} (x_j + \hat{x}_k - x_j \hat{x}_k) \leq \sum_{(j,k) \in \mathcal{A}_i \cup \delta_-(\mathcal{I}_i)} w_{jk} y_{jk},$$

for all $x_j, x_k, y_{jk} \in \{0, 1\}$ satisfying Constraints (11b)–(11d). Note that only the best-response variables \hat{x}_j with $j \in \mathcal{I}_i$ can be fixed in the EI. Moreover, in EBMC the \mathbf{y} -variables serve as auxiliary variables in the lifted formulation, so when constructing EIs we can substitute $y_{jk} = x_j + x_k - x_j x_k$ and fix the appropriate \mathbf{x} -variables as implied by the relevant arc-set definitions.

Explicit Reformulation to Standard IPGs The objective functions of the locally altruistic and selfish games can be reformulated into the standard IPG form by substituting \mathbf{y} -variables with \mathbf{x} -variables. These reformulations are detailed in Appendix 9.

5.3 Types and Network Structure in EBMC

We enrich the EBMC formulation with a finite set of *types* \mathcal{L} , which may represent invasive species, commodity classes, or risk categories. For each vertex $j \in \mathcal{I}$ and type $l \in \mathcal{L}$, let $f_{jl} \in \{0, 1\}$ indicate whether j has type l , and let $n_{jk} \geq 0$ denote the interaction intensity from j to k (e.g., traffic or flow). An edge (j, k) is

type- l risky if $f_{jl} = 1$ and $f_{kl} = 0$. We aggregate risk into the edge weight $w_{jk} := \sum_{l \in \mathcal{L}} f_{jl} (1 - f_{kl}) n_{jk}$, so only interactions that may transmit at least one type from j to k contribute to w_{jk} .

This structure induces a useful graph-theoretic dichotomy. When $|\mathcal{L}| = 1$, vertices split into those with and without the unique type, and risky edges run only from the former to the latter, yielding a bipartite network. When $|\mathcal{L}| \geq 2$, vertices can differ across types, and both (j, k) and (k, j) may be risky (for different types), so the network is generally non-bipartite.

5.4 A Resource Allocation Game among Minnesota Counties

The EBMC games in Section 5.2 are motivated by aquatic invasive species (AIS) prevention in Minnesota (MN). Currently, 84 county planners manage AIS inspection programs independently, each with its own AIS prevention budget supported by direct state aid [MDR, 2025]. The Minnesota Department of Natural Resources (MN DNR) facilitates information sharing (e.g., workshops and tools such as AIS Explorer [MAISRC, 2025]) but does not enforce a formal inter-county coordination mechanism for placing inspection stations. Counties report inspection activity (e.g., hours) to MN DNR, which can be used to infer deployed stations, and the increasing use of AIS Explorer facilitates transparency about allocations.

In this paper, we therefore model the *current* practice as a non-cooperative game with complete information, where each county chooses how to allocate its own inspection stations and positive externalities across counties arise endogenously from these choices. The statewide goal of the MN DNR is captured by the social-welfare objective (total inspected risky boat movements). Designing regulatory interventions or incentive schemes that better align county-level decisions with statewide welfare is an important direction for future work but lies beyond the scope of this study. Instead, we focus on two utility specifications: a *selfish* utility, where counties care only about risky boats entering lakes in their own \mathcal{I}_i , and a *locally altruistic* utility, where they also value inspection of boats leaving their lakes (Section 5.2). Our central questions are: *given these utility specifications and network types, when do PNE solutions exist in the resulting EBMC games, how do the resulting PNE solutions compare—both theoretically and computationally—for both the selfish and locally altruistic formulations, and what social welfare levels do these PNE solutions achieve?*

The vertices $j \in \mathcal{I}$ represent lakes, partitioned by county so that \mathcal{I}_i denotes the lakes controlled by county $i \in N$. For each ordered pair of lakes (j, k) , we let n_{jk} denote the estimated number of boat trips from j to k over a given period, obtained from survey-based lake-to-lake movement data [Kao et al., 2021]. Since Minnesota lakes host different AIS types, the edge weight w_{jk} aggregates type-specific risk following the setup in Section 5.3. Following the empirical AIS literature [Haight et al., 2021, 2023], we focus on lake-to-lake risk driven by the *most recent* trip on a given route; in particular, we do not model spread along multi-edge boat routes, both because current data and practice are organized at the origin–destination level and because infestation risk is driven primarily by short-term carryover from the immediately visited infested lake. Each county i has a budget \mathcal{B}_i representing the maximum number of inspection stations it can deploy, under the assumption that one station covers one lake.

6 Pure Nash Equilibrium for EBMC Games

As discussed in Section 2, the existence of a PNE is not guaranteed for general IPGs. In particular, when the interaction matrices Q_p^i in (1) are dense, a player's utility is heavily influenced by externalities (i.e., the decisions of other players), making it unlikely that there are simple, general sufficient conditions for PNE existence. However, if an IPG is a potential game [Monderer and Shapley, 1996], then a PNE is guaranteed to exist. Building on this idea, we first show that locally altruistic EBMC games admit a PNE. We then establish sufficient conditions for the existence of a PNE in selfish EBMC games, with a particular focus on large-scale IPG settings where externalities tend to dominate internal utility contributions.

We define the objective function of the SW model (8a) as the global function $\tilde{\phi}(\mathbf{y}) = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} = \sum_{(j,k) \in \mathcal{A}} w_{jk} \max(x_j, x_k) = \sum_{(j,k) \in \mathcal{A}} w_{jk} (x_j + x_k - x_j x_k) = \phi(\mathbf{x})$. The next proposition shows that $\phi(\mathbf{x})$ coincides with the sum of selfish utilities $u_i^{\text{Self}}(\mathbf{x})$ as defined in (11a) where $\tilde{u}_i^{\text{Self}}(\mathbf{y})$ is rewritten as $u_i^{\text{Self}}(\mathbf{x})$.

Proposition 1 (Social welfare as sum of selfish utilities). *Let $\mathcal{D} = (\mathcal{I}, \mathcal{A})$ be a directed graph and let the subsets of vertices \mathcal{I}_i , $i \in N$, form a partition of \mathcal{I} , i.e., $\mathcal{I} := \bigsqcup_{i \in N} \mathcal{I}_i$. Then $\mathcal{A} = \bigsqcup_{i \in N} \mathcal{A}^-[\mathcal{I}_i]$, and, for any feasible (\mathbf{x}, \mathbf{y}) with $y_{jk} = \max(x_j, x_k)$, the social welfare objective satisfies $\phi(\mathbf{x}) = \sum_{i \in N} u_i^{\text{Self}}(\mathbf{x})$.*

The proof is given in Appendix 2. Note that $\phi(\mathbf{x})$ is not equal to the sum of players' utilities in the locally altruistic game. Thus, under the standard definition of social welfare (SW) as the sum of utilities, $\phi(\mathbf{x})$ coincides with SW only for the selfish game. Nevertheless, since $\phi(\mathbf{x})$ maximizes edge coverage over the entire network, we refer to it as the SW function throughout, including in the locally altruistic setting.

6.1 Existence of a PNE in the Locally Altruistic Games

We now show that locally altruistic EBMC games are exact potential games. A game is an *exact potential game* if there exists a function $\psi : \mathcal{X} \rightarrow \mathbb{R}$ (an *exact potential*) such that for all $i \in N$, any strategies $\bar{\mathbf{x}}^{-i} \in \mathcal{X}_{-i}$ and $\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^i \in \mathcal{X}_i$ satisfy $\psi(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - \psi(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) = u_i(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - u_i(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i})$. Monderer and Shapley [1996] show that every potential game possesses at least one PNE: any strategy profile that maximizes the potential function. This existence result assumes that each player's strategy set \mathcal{X}_i is nonempty and compact. In our setting, each \mathcal{X}_i is a finite set of binary-feasible solutions (hence compact), and it is nonempty because the all-zero strategy is feasible. We now establish the following results; the proofs are given in Appendix 2.

Theorem 2. *The social welfare function ϕ is an exact potential function for locally altruistic games.*

Corollary 2. *The strategy profile $\bar{\mathbf{x}}_{\text{sw}}^*$ is a PNE for locally altruistic EBMC games.*

This corollary allows central planners to recommend an optimal strategy profile $\bar{\mathbf{x}}_{\text{sw}}^*$ if all players are locally altruistic. Additionally, assuming that players have not decided on a utility function, central planners can encourage players to adopt a more altruistic approach by highlighting mutual benefits. Decision makers at any level, central planner or player level, can directly solve the SW model (8) and implement the solution.

6.2 Existence of a PNE in the Selfish Games

Next, we establish sufficient conditions for the existence of a PNE in selfish EBMC games, particularly in regimes where externalities from other players' decisions $\bar{\mathbf{x}}^{-i}$ dominate the utility generated by a player's own choices \mathbf{x}^i . We denote the selfish game with a single type by $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and with multiple types by $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Let $\mathcal{I} = \mathcal{I}^+ \cup \mathcal{I}^0$, where $\mathcal{I}^+ := \{j \in \mathcal{I} : \exists l \in \mathcal{L} \text{ s.t. } f_{jl} = 1\}$ and $\mathcal{I}^0 := \{j \in \mathcal{I} : f_{jl} = 0 \text{ for all } l \in \mathcal{L}\}$. Thus, \mathcal{I}^+ is the set of vertices that have at least one type, while \mathcal{I}^0 is the set of vertices with no types. The following result formalizes these conditions; its proof is provided in Appendix 2.

Theorem 3. *Let \mathcal{I}^{ng} be the set of vertices selected in the non-game strategy profile $\bar{\mathbf{x}}_{\text{ng}}$. If $\mathcal{I}^{\text{ng}} \subseteq \mathcal{I}^0$, then $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE for both $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. Additionally, if $\mathcal{I}^{\text{ng}} \subseteq \mathcal{I}^+$, $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$. \square*

We now examine conditions aligned with Theorem 3. For a vertex $j \in \mathcal{I}_i$, let $w(j)$ denote the total weight of inbound arcs incident to j within the induced inbound arc set $\mathcal{A}^-[\mathcal{I}_i]$. In the random graphs we generated, the condition $\mathcal{I}_i^{\text{ng}} \subseteq \mathcal{I}_i^0$ for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ frequently holds. This phenomenon arises from the structure of $\mathcal{A}^-[\mathcal{I}_i]$, where $w(k)$ for vertices $k \in \mathcal{I}_i^0$ often exceeds $w(j)$ for vertices $j \in \mathcal{I}_i^+$. As the number of players increases, the aggregate weight from incoming arcs typically exceeds that from internal arcs, leading to a non-game strategy profile $\bar{\mathbf{x}}_{\text{ng}}^i$ that favors vertices in \mathcal{I}_i^0 .

Even when $\mathcal{I}_i^{\text{ng}} \subseteq \mathcal{I}_i^0$ is not strictly satisfied, we observe computationally (in randomly generated instances) that $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ almost always admits a PNE. Intuitively, large (though not strictly dominant) externalities can guide BRD to a PNE in only a few iterations. Typically, $\mathcal{I}_i^{\text{ng}}$ consists primarily of vertices in \mathcal{I}_i^0 due to the high volume of risky incoming arcs. If BRD starts from a conservative profile such as $\bar{\mathbf{x}}_{\text{ng}}$, players benefit iteratively: selecting vertices in \mathcal{I}_i^+ induces other players to select previously uncovered vertices in \mathcal{I}_i^0 . Importantly, selecting vertices in \mathcal{I}_i^0 benefits only player i , whereas selecting vertices in \mathcal{I}_i^+ benefits others. Since \mathcal{I}_i^+ is finite, this improvement process eventually terminates, yielding a PNE. In Section 7.2, we confirm that all instances of the selfish game $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ admit a PNE.

However, this dynamic pattern typically holds under symmetric or balanced edge benefits. We obtain the counterexamples in Appendix 10 either via carefully calibrated edge weights and graph structures (for $|\mathcal{L}| = 1$) or via multiple AIS types with conflicting priorities (for $|\mathcal{L}| \geq 2$).

Summary and pointers. Section 6 establishes PNE existence for locally altruistic EBMC games via an exact potential and provides sufficient conditions (and counterexamples) for the selfish games. For a brief discussion of the complexity of deciding PNE existence, see Appendix 9.

7 Computational Results

This section evaluates the effectiveness of RRR–BRD (Algorithm 2), BZR (Algorithm 3), and the ZR algorithm of Dragotto and Scatamacchia [2023] for computing PNEs and identifying high-welfare (best-found) PNEs in EBMC games and KPGs. To implement ZR, we use Algorithm 3 with Lines 8–12 disabled. The main manuscript reports (i) summary results for RRR–BRD under multiple initial profiles and (ii) summary

results and selected comparisons of BZR versus ZR for EBMC and KPG instances; complete results are provided in Appendix 11–12. For instances where no PNE is found across all runs, we report approximate-PNE certificates in Appendix 13. Section 7.1 describes the computational settings and performance metrics. Section 7.2 reports results for EBMC games on randomly generated instances with varying numbers of players ($n = 2, 3, 5, 8, 10, 15, 20, 25, 30$). Section 7.3 presents results for the Minnesota dataset with 84 county-level players and discusses managerial insights. Finally, Section 7.4 reports results for KPGs using the instance-generation scheme of Dragotto and Scatamacchia [2023], scaling from small ($n = 2, 3$) to larger games ($n = 5, 8, 10, 15, 20, 25, 30$).

7.1 Computational Setup and Metrics

All experiments were run on a machine with an Intel(R) Core(TM) i9-13900F CPU @ 3.10GHz (24 cores) and 64GB RAM. All algorithms (RRR–BRD, BZR, ZR) were implemented in Python and solved using Gurobi Optimizer (v11.0.3) with 16 threads. Equilibrium inequalities were enforced via Gurobi callbacks and lazy constraints. Code, instances, and results are available on GitHub.¹

Model construction time for best-response problems is non-negligible, especially in large-scale IPGs where best responses must be solved repeatedly across players. To reduce overhead, we construct each player’s best-response model once and then update only the components that change across solves (e.g., fixing \mathbf{x}^{-i}), consistently across all algorithms. When integrating RRR–BRD trajectories into callbacks (e.g., within BZR), we also control the number of additional best-response inequalities generated from a trajectory; in most instances, we add at most n such inequalities. For the Minnesota instance, we disable these additional cuts, since adding many inequalities can substantially slow the overall solving process.

Table 4 summarizes the acronyms and shorthands used in the paper and in this section. While OSW and POS are defined in Section 2, we report time-limited approximations—denoted $\widetilde{\text{OSW}}$ and $\widetilde{\text{POS}}$ —due to the scale of the instances. For both BZR and ZR, the time limit was 1800 seconds. The SW model (8) was also solved with a 1800-second limit to compute $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{1800})$. For RRR–BRD, we use two initial strategy profiles: $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$. For standalone RRR–BRD, we set the maximum numbers of restarts and rounds to $L = 10$ and $R = 20$, respectively. When used as a subroutine within BZR, we use $L = 3$ and $R = 20$. Finally, all objective values (ϕ) are reported in thousands and rounded to the nearest tenth in all tables.

We report the number of players (n), the budget ratio (BG), and additional performance metrics.

- **RRR–BRD metrics:** For each instance and each initial profile ($\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$), we report the best-found PNE (in terms of ϕ), runtime, number of rounds, $\widetilde{\text{OSW}}$, and $\widetilde{\text{POS}}$ (computed using the higher-welfare PNE among the two runs). We also include reference values from BZR, such as the best PNE found and the best dual bound. The time to obtain $\tilde{\mathbf{x}}_{\text{sw}}^{300}$ is included in the reported runtime of RRR–BRD.
- **ZR and BZR metrics:** For each instance, we report the number of distinct PNEs found ($\#_{\text{pne(ZR)}}$, $\#_{\text{pne(BZR)}}$), the best-found PNE, the dual bound, time to the first PNE (e.g., $T(\text{BZR}^{1\text{st}})$), total runtime (e.g., $T(\text{BZR})$), $\widetilde{\text{OSW}}$, $\widetilde{\text{POS}}$ (based on the best-found PNE), and the number of equilibrium inequalities added (e.g., $\#_{\text{cuts(BZR)}}$). As noted in Remark 2, the gap between the best-found PNE and the dual bound

¹<https://github.com/HyunwooLee0429/Best-response-dynamics-IPG>

Table 4: Acronyms and Shorthands

Symbol / Abbreviation	Full Term / Meaning
$\mathbf{0}^N$	Initial strategy profile: all-zero vector
$\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}$	Initial strategy profile obtained by solving the SW model with time limit (secs)
$\tilde{\mathbf{x}}_{\text{ng}}$	Non-game strategy profile
$\hat{\mathbf{x}}_{\text{pne}}^1 / \hat{\mathbf{x}}_{\text{pne}}^*$	First identified PNE / Best identified PNE (maximizing ϕ) within time limit
$\phi(\cdot)$	Social welfare function; values are reported in thousands and rounded to the nearest tenth.
$\widetilde{\text{OSW}}$	Approximate optimal social welfare ($\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})$)
$\widetilde{\text{POS}}$	Approximated price of stability given by $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}) / \phi(\hat{\mathbf{x}}_{\text{pne}}^*)$
$\#\text{cuts}(\text{ZR}) / \#\text{cuts}(\text{BZR})$	Number of equilibrium inequalities added by ZR / BZR
$\#\text{pne}(\text{ZR}) / \#\text{pne}(\text{BZR})$	Number of distinct PNEs found by ZR / BZR
$\text{T}(\text{ZR}^{1\text{st}}) / \text{T}(\text{BZR}^{1\text{st}})$	Time to find the first PNE using ZR / BZR
$\text{T}(\text{ZR}) / \text{T}(\text{BZR})$	Total runtime of the ZR / BZR algorithm

can remain large at termination, reflecting the large-scale nature of IPG instances. The best-found PNE is certified as welfare-optimal when $\phi(\hat{\mathbf{x}}_{\text{pne}}^*)$ coincides with the dual bound (within a tolerance).

7.2 EBMC Games with Random Dataset

We generate random EBMC instances by varying the number of counties, budget levels, infestation patterns, and network density. The number of counties is chosen from $\{2, 3, 5, 8, 10, 15, 20, 25, 30\}$, and each county controls 50 lakes. For the single-AIS case ($|\mathcal{L}| = 1$), each lake is independently infested with a probability drawn from $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. For the multiple-AIS case ($|\mathcal{L}| \geq 2$), infestation is generated independently by AIS type, with type-specific probabilities drawn from nested candidate sets: $\{0.2, 0.4, 0.6, 0.8, 1.0\}$ (type 1), $\{0.2, 0.4, 0.6, 0.8\}$ (type 2), $\{0.2, 0.4, 0.6\}$ (type 3), and $\{0.2, 0.4\}$ (type 4); this construction biases instances so that most lakes remain uninfested for at least one AIS type. We consider three budget ratios, $\{0.3, 0.5, 0.8\}$, and set each county’s budget as the number of infested lakes in that county multiplied by the chosen ratio. Arc weights are drawn i.i.d. uniformly from $[10, 20]$. To avoid unrealistically dense (nearly complete) networks, we start from a complete directed graph and randomly delete arcs: 20% when $|\mathcal{L}| = 1$ and 50% when $|\mathcal{L}| \geq 2$, with the higher deletion rate also helping control instance size. Overall, this procedure yields 27 instances per case (single AIS and multiple AIS), corresponding to the 9 county counts combined with the 3 budget ratios.

Table 5 reports standalone RRR–BRD performance over these 27 instances using two initial profiles, $\mathbf{0}^N$ and $\tilde{\mathbf{x}}_{\text{sw}}^{300}$; complete results are provided in Appendix 11. Relative to the single-AIS game $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, the multiple-AIS game $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$ requires substantially more rounds on average, suggesting that strategic dynamics become markedly more complex when multiple AIS types are present.

Table 5: Summarized RRR-BRD Results with Different Initial Strategy Profiles for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$.

type \mathcal{L}	Avg [†] PNE (ϕ)		Avg Price of Stability		Avg References		Avg Time (secs)		Avg Round	
	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{300}$
$ \mathcal{L} = 1$	786.8	1010.5	1055.5	1.051	1023.6	1089.5	7.5	212.1	2.0	1.6
$ \mathcal{L} \geq 2$	3658.0	3658.0	4010.0	1.097	3658.0	5409.3	25.6	289.5	4.0	4.7

Note. †: Averages of ϕ , OSW, and POS are calculated from instances where a PNE is found by both initial strategy profiles.

Tables 6 summarizes the performance of BZR and ZR on $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. We report (out of 27 instances) how often a PNE was found, multiple PNEs were identified, the best PNE was obtained, and a tighter dual bound was achieved. “Ties” indicate the number of instances in which both algorithms either identify the same best PNE or attain the same dual bound. The full results for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ appear in Table 7, while the full results for $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ are reported in Appendix 12. In both settings, BZR consistently outperforms ZR in (i) time to the first PNE, (ii) the number of distinct PNEs found, and (iii) the best PNE value identified.

Table 6: Summarized Comparison of BZR and ZR for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

type \mathcal{L}	# PNEs Found		Finding a PNE		Best PNE			Dual Bounds			Avg POS	# Avg EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	Ties	ZR	BZR	Ties	ZR	BZR	$\overline{\text{POS}}$	# _{cuts(ZR)}	# _{cuts(BZR)}
$ \mathcal{L} = 1$	32	150	11	27	10	0	17	9	4	14	1.033	59.3	94.4
$ \mathcal{L} \geq 2$	3	29	3	23	3	0	20	5	18	4	1.097	66.0	116.3

For the single-AIS case ($|\mathcal{L}| = 1$), BZR typically finds a PNE within a few seconds when $n \leq 10$, and within roughly 100 seconds for $15 \leq n \leq 30$, indicating strong scalability. In contrast, ZR returns a PNE in only 11 of the 27 instances within the time limit; when both algorithms find a PNE, BZR more frequently identifies the best PNE. BZR also tends to provide tighter dual bounds in this setting.

For the multiple-AIS case ($|\mathcal{L}| \geq 2$), the performance gap widens: BZR finds a PNE in 23 of the 27 instances, whereas ZR succeeds in only 3 instances within the time limit. Although BZR continues to identify high-quality PNEs, its dual bounds are often not substantially tightened, even after adding additional equilibrium inequalities. This behavior is expected for three reasons: (i) the time spent in RRR–BRD can be significant in the hardest instances, (ii) dual progress is driven largely by the LP relaxation and general-purpose cutting planes, and (iii) the added equilibrium inequalities primarily enforce equilibrium feasibility rather than strongly strengthening the continuous relaxation. Since the primary goal of ZR/BZR is to compute one or more PNEs (and, when possible, a best PNE), we treat dual bounds as auxiliary metrics.

7.3 EBMC Games with Minnesota Dataset

Tables 8 and 9 report results for the selfish EBMC game $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ on the Minnesota dataset: Table 8 summarizes standalone RRR–BRD performance, and Table 9 compares BZR and ZR. The dataset, provided by the Minnesota Aquatic Invasive Species Research Center (MAISRC), contains lake-to-lake boat movement estimates among 9,182 lakes and uses 2018 infestation records, following the construction in Kao et al. [2021]. We exclude counties with no lakes or with no incoming or outgoing risky movements.

To reflect the scale of county-level inspection programs, we model statewide support corresponding to 436 inspection stations. Minnesota allocates roughly \$10 million per year to AIS inspection programs and employs approximately 800–1000 inspectors [MDR, 2017]; assuming two inspectors per lake, this staffing level corresponds to operating about 400–500 inspection stations statewide. We allocate county budgets proportionally to county-level funding levels [MDR, 2017]. We model four AIS types—Zebra Mussels, Starry Stonewort, Eurasian Watermilfoil, and Spiny Water Fleas—yielding $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Because this instance is extremely large (84 players, roughly 100 lakes per county on average, and millions of arcs), we use a 3600-second time limit for ZR, BZR, and the SW model, and a 600-second limit

Table 7: Comparison of BZR and ZR for Selfish EBMC Games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
2	0.3	1	1	19.2	19.2	19.2	19.2	19.2	1.000	0.2	0.2	0.2	0.4	8	13
	0.5	6	7	22.3	22.3	22.3	22.3	22.3	1.000	0.9	1.7	0.2	2.2	11	25
	0.8	1	6	23.3	23.3	23.3	23.3	24.9	1.066	1.7	1.7	0.3	2.0	12	26
3	0.3	1	6	27.6	27.6	27.6	27.6	27.6	1.000	2.4	2.4	0.6	3.4	9	19
	0.5	5	4	42.1	42.1	42.1	42.1	42.1	1.000	8.2	11.1	0.5	12.1	15	27
	0.8	1	1	53.1	53.1	53.1	53.1	53.1	1.000	0.6	0.6	0.5	1.0	12	19
5	0.3	1	8	90.4	90.4	90.4	90.4	90.4	1.000	12.9	13.1	1.3	15.0	17	38
	0.5	1	4	100.2	100.6	101.3	101.2	100.6	1.000	1474.5	1800.5	1.9	1800.3	94	45
	0.8	10	25	156.7	156.7	157.9	157.4	156.7	1.000	1134.4	1800.9	1.5	1800.3	182	58
8	0.3	3	5	134.1	134.1	134.2	134.2	149.9	1.118	40.4	74.5	6.0	113.1	41	68
	0.5	0	3	-	252.2	260.1	260.2	252.2	1.000	-	1800.8	4.9	1800.2	28	51
	0.8	0	10	-	349.2	379.6	353.4	381.3	1.092	-	1800.3	5.0	1800.6	28	83
10	0.3	2	5	317.2	317.2	317.3	317.3	317.2	1.000	535.8	1359.9	7.9	971.2	54	82
	0.5	0	24	-	361.7	372.5	372.4	361.7	1.000	-	1801.5	5.2	1801.0	28	105
	0.8	0	3	-	538.1	558.8	549.3	539.8	1.003	-	1800.2	7.6	1800.2	31	59
15	0.3	0	4	-	725.4	743.5	745.9	725.4	1.000	-	1800.1	16.3	1800.0	51	94
	0.5	0	6	-	708.4	729.6	730.1	857.5	1.210	-	1800.1	22.3	1800.1	75	143
	0.8	0	2	-	1216.5	1328.9	1249.2	1219.7	1.003	-	1800.1	18.0	1800.1	49	91
20	0.3	0	3	-	904.5	937.7	922.5	904.5	1.000	-	1800.3	42.6	1800.1	79	138
	0.5	0	3	-	1282.1	1469.8	1472.8	1529.1	1.193	-	1800.3	38.2	1800.1	88	147
	0.8	0	2	-	2024.4	2441.1	2207.2	2415.5	1.193	-	1800.3	38.5	1800.2	78	136
25	0.3	0	4	-	1765.0	1823.2	1819.5	1765.0	1.000	-	1800.3	61.6	1800.1	93	166
	0.5	0	4	-	2597.4	2782.7	2767.9	2597.7	1.000	-	1800.3	61.1	1800.1	100	173
	0.8	0	2	-	3307.4	3883.9	3541.1	3312.8	1.002	-	1800.3	51.1	1800.3	86	156
30	0.3	0	3	-	2448.1	2572.0	2556.0	2448.3	1.000	-	1800.4	84.2	1800.2	98	180
	0.5	0	2	-	3260.8	3509.6	3482.4	3263.8	1.001	-	1800.4	90.9	1808.0	130	218
	0.8	0	3	-	4910.4	5574.9	5399.1	4919.7	1.002	-	1800.4	82.1	1800.3	103	189

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

to compute the warm-start profile $\tilde{\mathbf{x}}_{\text{sw}}^{600}$. Standalone RRR-BRD requires more than 400 seconds from either $\mathbf{0}^N$ or $\tilde{\mathbf{x}}_{\text{sw}}^{600}$ (Table 8), indicating a substantially more difficult best-response landscape than in our synthetic instances. Table 9 shows that BZR identifies multiple PNEs within the time limit, whereas ZR finds none.

Table 8: RRR-BRD Results for Selfish EBMC Games $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ in Minnesota Dataset.

n	CR-BRD (ϕ)		Price of Stability		References		Time (secs)		Round	
	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$	init: $\mathbf{0}^N$	init: $\tilde{\mathbf{x}}_{\text{sw}}^{600}$
84	758.8	761.4	887.7	1.166	761.4	861.7	425.4	586.5	5	7

Table 9: Comparison of BZR and ZR for Selfish EBMC Games $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ in Minnesota Dataset.

n	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
84	0	5	-	761.4	861.7	861.7	887.7	1.166	-	3600.8	790.7	3600.8	363	363

We emphasize that PNE existence in this instance is not guaranteed a priori by our sufficient condition (Theorem 3). The Minnesota results therefore provide a computational witness of existence for the selfish game, complementing the locally altruistic case where a PNE is immediate via Corollary 2. Table 10 reports county-level selfish utilities on the Minnesota instance under several strategy profiles: the non-game profile $\tilde{\mathbf{x}}_{\text{ng}}$, the first PNE $\hat{\mathbf{x}}_{\text{pne}}^1$, the best PNE $\hat{\mathbf{x}}_{\text{pne}}^*$ found by BZR, and the welfare benchmark $\tilde{\mathbf{x}}_{\text{sw}}^{3600}$. For each county i , we compare utilities column-wise via $u_i^{\text{Self}}(\tilde{\mathbf{x}}_{\text{ng}}) \leq u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1)$, $u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^1) \leq u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*)$, $u_i^{\text{Self}}(\hat{\mathbf{x}}_{\text{pne}}^*) \leq u_i^{\text{Self}}(\tilde{\mathbf{x}}_{\text{sw}}^{3600})$, and highlight any violations in bold. The complete 84-county table is provided in Appendix 14.

Overall, the results suggest that (i) many counties benefit from strategic interaction relative to the non-game baseline, due to unintended positive spillovers; (ii) among equilibria, the best PNE improves utilities for most counties; and (iii) the welfare-oriented benchmark $\tilde{\mathbf{x}}_{\text{sw}}^{3600}$ (and, in the locally altruistic game, the OSW solution) often yields the highest utilities, although exceptions exist (e.g., Sibley and Winona).

These comparisons illustrate why computed equilibria are policy-relevant benchmarks. A PNE is stable in the sense that no county has an incentive to deviate unilaterally, and the socially best PNE for the selfish game offers a tractable compromise between stability and statewide performance. Moreover, the locally altruistic specification highlights a concrete policy target: if counties can adopt (or are incentivized to adopt) utilities that internalize spillovers, then the welfare-optimal solution is self-enforcing—indeed, Corollary 2 shows that the OSW solution is a PNE under locally altruistic utilities. Nevertheless, the best selfish-game PNE can still deviate substantially from the welfare benchmark, as reflected by a POS of 1.166. This motivates future work on (i) incentive design that induces counties to internalize externalities and (ii) equilibrium-informed coordination that leverages equilibrium information from selfish games to improve social welfare.

Table 10: County-level Utility Values Using Different Strategy Profiles.

County	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{pne}}^1)$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{pne}}^*)$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{sw}}^{3600})$	County	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{ng}})$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{pne}}^1)$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{pne}}^*)$	$u_c^{\text{Self}}(\tilde{\mathbf{x}}_{\text{sw}}^{3600})$
Anoka	11292	12466	12495	15723	Mcleod	7194	7479	7503	8933
Becker	24219	26489	26497	31712	Meeker	8866	9423	9428	9964
Hennepin	21136	22442	22474	27024	Sherburne	5077	5557	5564	6698
Hubbard	22780	24426	24444	29543	Sibley	2390	2388	2388	2429
Isanti	3635	3876	3884	4227	Stearns	20200	21544	21554	26076
Lyon	2752	2925	2939	3314	Winona	1561	1579	1579	1444
Mahnomen	4177	4424	4428	6012	Wright	22047	23782	23805	29464

Note. Counties, where the inequalities regarding utility value comparisons do not hold, are highlighted in bold.

7.4 Large-scale Knapsack Problem Games Datasets and Results

To further assess the effectiveness of RRR–BRD and BZR, we report extensive computational results for large-scale knapsack problem games (KPGs), a standard benchmark in the IPG literature [Carvalho et al., 2025, 2022, Dragotto and Scatamacchia, 2023, Crönert and Minner, 2024]. In a KPG, each player i solves a binary knapsack problem of the form $\max_{\mathbf{x}^i} \left\{ \sum_{j=1}^m p_j^i x_j^i + \sum_{k=1, k \neq i}^n \sum_{j=1}^m f_{k,j}^i x_j^i x_j^k : \sum_{j=1}^m w_j^i x_j^i \leq \mathcal{B}^i, \mathbf{x}^i \in \{0, 1\}^m \right\}$, where $x_j^i = 1$ if player i selects item j (and 0 otherwise). The parameters p_j^i , w_j^i , and \mathcal{B}^i denote item profit, item weight, and player budget, respectively. Strategic interactions are captured by the bilinear coefficients $f_{k,j}^i$, which contribute to player i 's payoff when both players i and k select item j . To linearize $x_j^i x_j^k$, we introduce auxiliary variables $z_j^{i,k}$ imposing the following constraints: $z_j^{i,k} \geq x_j^i + x_j^k - 1$, $z_j^{i,k} \leq x_j^i$, $z_j^{i,k} \leq x_j^k$, $\forall j \in \{1, \dots, m\}$, $\forall i \in N$, $\forall k \in N \setminus \{i\}$.

Following Dragotto and Scatamacchia [2023], we also incorporate two classes of problem-specific equilibrium inequalities—strategic dominance and strategic payoff inequalities—along with the standard equilibrium inequalities. For larger instances ($n \geq 15$), identifying dominance relations and minimal interaction sets for strategic payoff inequalities becomes computationally expensive, so we omit these inequalities.

We adopt the data generation scheme of Dragotto and Scatamacchia [2023] for interaction types (A)

and (B), and use a modified version of type (C) to better match our experimental goal. In type (A), each $f_{k,j}^i$ is set to a constant drawn uniformly from $[1, 100]$; in type (B), each $f_{k,j}^i$ is drawn independently from $[1, 100]$. In the original type (C), $f_{k,j}^i \sim \text{Unif}[-100, 100]$, which can induce very strong negative (adversarial) interactions; as also observed in Dragotto and Scatamacchia [2023], such instances often fail to admit a PNE even for small sizes ($n = 2, 3$), and this issue becomes more pronounced as n grows. Because our convergence guarantees for RRR–BRD (and hence BZR’s use of it as a subroutine) are conditional—*on PNE existence*—we prioritize stress-testing instances that remain adversarial yet are not overwhelmingly likely to be PNE-free. Concretely, we modify type (C) by sampling $f_{k,j}^i$ from $[-20, 0]$, which preserves negative interactions but reduces their magnitude to an “admissible” level that more frequently yields equilibria in practice. Budgets are set as a fraction of total item weight: $\mathcal{B}^i \in \{0.2 \sum_{j=1}^m w_j^i, 0.5 \sum_{j=1}^m w_j^i, 0.8 \sum_{j=1}^m w_j^i\}$. We start from the largest KPG setting in Dragotto and Scatamacchia [2023] ($n \in \{2, 3\}$, $m = 100$) and scale to $n \in \{2, 3, 5, 8, 10, 15, 20, 25, 30\}$ with $m = 100$, yielding 27 generated instances per interaction type.

Table 11 summarizes RRR–BRD performance under two initial profiles, $\mathbf{0}^N$ and $\bar{\mathbf{x}}_{\text{sw}}^{300}$; complete results appear in Appendix 11. Across interaction types, we observe that type (C) instances exhibit larger $\overline{\text{POS}}$, and require substantially more rounds before reaching a PNE. These trends indicate that negative interactions produce markedly more complex best-response dynamics than types (A) and (B).

Table 11: Summarized RRR-BRD Results with Different Initial Strategy Profiles for Type A, B, and C KPG.

type	Avg [†] PNE (ϕ)		Avg Price of Stability		Avg References		Avg Time (secs)		Avg Round	
	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$
Type A	700.7	711.0	714.1	1.012	711.2	715.6	2.6	192.9	7.7	3.9
Type B	706.1	714.7	717.1	1.010	715.5	719.2	2.9	198.2	8.6	4.6
Type C	14.0	14.1	16.9	1.171	14.2	16.3	1.8	73.2	29.5	16.5

Note. †: The averages of ϕ , OSW, and POS are computed only from instances where a PNE is found under both initial solutions.

Table 12 provides a summarized comparison of BZR and ZR across the three KPG datasets, reporting how often a PNE was found, multiple PNEs were identified, the best PNE was obtained, and a tighter dual bound was achieved. Full results for type (B) are given in Table 13, while those for types (A) and (C) appear in Appendix 12. Overall, BZR consistently outperforms ZR across all interaction types in (i) time to the first PNE, (ii) the number of distinct PNEs found, and (iii) the best PNE value identified. For dual bounds, the advantage is clearer in types (A) and (B), whereas in type (C) the two methods exhibit complementary strengths: as shown in Table A17 (Appendix 12), BZR generally produces tighter dual bounds for $n \leq 15$, while ZR tends to perform better for $n \geq 20$.

Table 12: Summarized Comparison of BZR and ZR for Type A, B, and C KPG.

type	# PNEs Found		Finding a PNE		Best PNE			Dual Bounds			Avg POS	Avg # EI Cuts	
	# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	Ties	ZR	BZR	Ties	ZR	BZR	$\overline{\text{POS}}$	# _{cuts(ZR)}	# _{cuts(BZR)}
Type A	14	719	7	27	7	0	20	10	0	17	1.010	1206.7	1012.0
Type B	12	1766	5	27	5	0	22	7	1	19	1.006	945.9	966.6
Type C	12	675	7	24	7	0	17	8	7	12	1.175	2180.9	2516.4

Table 13: Comparison of BZR and ZR for Type B KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
2	0.2	3	9	8.2	8.2	8.2	8.2	8.3	1.018	9.7	13.4	0.7	10.0	1944	1913
	0.5	4	125	14.2	14.2	14.2	14.2	14.4	1.010	72.9	130.9	0.3	72.9	2281	2154
	0.8	1	9	19.1	19.1	19.1	19.1	19.1	1.002	1.2	1.4	0.3	2.2	1224	1239
3	0.2	0	195	-	15.6	15.7	15.6	15.9	1.023	-	1800.3	1.1	1800.3	2679	2161
	0.5	0	222	-	28.9	29.2	29.1	29.4	1.016	-	1800.3	0.6	1800.2	2744	2100
	0.8	3	291	40.0	40.0	40.1	40.1	40.3	1.007	590.5	1800.3	1.2	1801.3	2287	2361
5	0.2	0	84	-	40.5	40.8	40.8	40.9	1.012	-	1800.2	2.0	1800.2	663	832
	0.5	0	129	-	78.0	78.8	78.8	78.8	1.011	-	1800.4	2.2	1802.9	1377	1325
	0.8	0	107	-	108.3	108.7	108.6	108.7	1.003	-	1800.5	4.0	1800.4	1790	983
8	0.2	0	75	-	91.6	92.7	92.7	92.6	1.011	-	1800.3	4.4	1800.5	755	1055
	0.5	0	89	-	188.6	190.0	190.0	189.9	1.007	-	1801.1	10.0	1801.3	874	1282
	0.8	0	12	-	272.9	273.4	273.2	273.1	1.001	-	1801.2	6.9	1800.8	998	176
10	0.2	0	43	-	135.9	137.1	137.0	136.9	1.007	-	1801.1	2.3	1800.4	523	721
	0.5	0	28	-	287.8	289.5	289.4	289.0	1.004	-	1800.8	6.1	1801.2	306	508
	0.8	0	27	-	426.9	428.4	428.3	427.8	1.002	-	1800.5	7.4	1801.4	667	435
15	0.2	0	41	-	287.3	290.6	290.6	289.5	1.008	-	1801.9	20.4	1801.0	358	960
	0.5	0	19	-	642.1	645.7	645.5	644.0	1.003	-	1800.9	20.4	1801.5	252	466
	0.8	0	12	-	963.3	967.1	966.6	965.2	1.002	-	1800.9	23.0	1801.4	258	312
20	0.2	0	16	-	488.4	493.2	493.1	491.2	1.006	-	1800.7	13.4	1800.6	334	496
	0.5	0	11	-	1123.1	1129.2	1129.1	1125.5	1.002	-	1801.2	27.6	1801.3	309	402
	0.8	0	10	-	1671.7	1677.5	1677.3	1672.9	1.001	-	1800.8	24.5	1800.9	264	367
25	0.2	0	12	-	760.9	767.6	767.5	763.5	1.003	-	1800.8	21.6	1801.0	315	444
	0.5	0	11	-	1716.7	1728.7	1728.5	1723.7	1.004	-	1800.9	19.2	1800.9	338	448
	0.8	0	9	-	2610.3	2623.1	2622.8	2614.3	1.002	-	1800.9	28.9	1800.7	371	420
30	0.2	0	10	-	1083.3	1094.7	1094.6	1088.8	1.005	-	1801.0	34.1	1800.7	399	540
	0.5	0	11	-	2459.8	2473.3	2473.4	2466.1	1.003	-	1800.7	98.9	1801.0	237	584
	0.8	0	8	-	3748.6	3764.1	3764.0	3750.8	1.001	-	1800.8	27.0	1800.6	425	430

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

How Interaction Coefficients and the Number of Players Affect $q_\mu[R]$. When interaction coefficients are predominantly positive, improving BR moves more often enter equilibrium basins, leading to larger $\tilde{q}_\mu[R]$ and faster PNE discovery; as interactions become more complex—e.g., when coefficients are mixed or negative—the dynamics can become more complicated and $q_\mu[R]$ may decrease. This viewpoint aligns with our EBMC variants: in $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, other players’ selections are typically helpful for player i (Section 6.2), whereas in $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$, other players’ updates can affect player i either positively or negatively, yielding more complicated BR trajectories. Similarly, KPG types (A) and (B) behave like predominantly positive-interaction instances, while our type (C) is closer to a mixed regime. Finally, increasing the number of players can amplify externalities; for example, the original type (C) construction in Dragotto and Scatamacchia [2023] can exhibit a more adversarial landscape as externalities increase (Section 7.4), which may drive $q_\mu[R]$ to be very small. This is consistent with our results: $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and KPG types (A) and (B) find PNEs in all instances, whereas $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ fails in four instances and our KPG type (C) fails in three instances.

8 Conclusion

Computing PNEs in IPGs remains challenging, primarily because most existing exact approaches rely on joint formulations that couple players’ strategy sets, and computational evidence is therefore largely restricted to instances with only a few players and modest per-player model sizes. This scalability gap is particularly salient in large-scale resource-allocation settings with many stakeholders, such as the EBMC games studied in this paper. To address this challenge, we introduce round random-restart best-response

dynamics (RRR–BRD) for IPGs and provide a probabilistic convergence guarantee whenever a PNE exists and the restart law assigns positive probability to basins that reach a PNE.

We further embed RRR–BRD as a randomized local-search subroutine within the ZR framework, yielding substantial practical gains in finding PNEs and high-social-welfare PNEs, and often improving dual bounds. Overall, these improvements translate into an order-of-magnitude enhancement in the tractable scale of equilibrium computation for our tested IPG classes. We also discuss how RRR–BRD can serve as a local-search subroutine within other IPG algorithms whenever integer-feasible profiles are available. On the modeling side, we study two EBMC game variants—locally altruistic and selfish—by specifying how players value edge coverage, and we establish sufficient conditions for PNE existence in each setting. Using the MN dataset, we illustrate how equilibrium outcomes depend on these behavioral specifications and provide managerial insights for allocating limited inspection resources at both the statewide and county levels.

Finally, our results point to two important directions for future research. First, our probabilistic guarantees are expressed through the instance-dependent success probability under restarts, $q_\mu[R]$, which is difficult to compute in general; deriving tractable estimators/bounds for $q_\mu[R]$ under additional structural assumptions, and designing restart laws that improve it, are promising extensions. Second, while we focus on equilibrium computation and comparative analysis under different utility specifications, we do not design incentive mechanisms; developing implementable policy levers that steer equilibria toward socially desirable outcomes, and analyzing welfare–stability trade-offs and sensitivity, are natural next steps.

Acknowledgments. H. Lee and R. Hildebrand were partially funded by AFOSR grant FA9550-21-1-0107. H. Lee and İ. E. Büyükahtakın have also been funded by the Grado Department of ISE at VT. S. Cai was partially funded by MAISRC Subaward H009064601, and all authors were partially funded by the MAISRC Subaward H0113450M1. The authors are grateful to MAISRC for making the Minnesota dataset publicly accessible, and Amy Kinsley (MAISRC), Nick Phelps (MAISRC), Alex Bajcz (MAISRC), Bob Haight (USFS), Tina Fitzgerald (DNR) and Adam Doll (DNR) for pointing us to the Minnesota dataset and for helpful discussion on the AIS inspection problem in MN. Any opinions, findings, conclusions, or recommendations expressed in this study are those of the authors and do not necessarily reflect the views of the Air Force Office of Scientific Research and MAISRC.

References

- Ben Amiet, Andrea Collevocchio, Marco Scarsini, and Ziwen Zhong. Pure Nash Equilibria and Best-response Dynamics in Random Games. *Mathematics of Operations Research*, 46(4):1552–1572, 2021.
- Krzysztof R Apt and Sunil Simon. A classification of weakly acyclic games. *Theory and Decision*, 78(4): 501–524, 2015.
- I Esra Büyükahtakın and Robert G Haight. A Review of Operations Research Models in Invasive Species Management: State of the Art, Challenges, and Future Directions. *Annals of Operations Research*, 271(2):357–403, 2018.

- Margarida Carvalho, Andrea Lodi, Joao Pedro Pedroso, and Ana Viana. Nash Equilibria in the Two-player Kidney Exchange Game. *Mathematical Programming*, 161:389–417, 2017.
- Margarida Carvalho, Andrea Lodi, and Joao Pedro Pedroso. Existence of Nash Equilibria on Integer Programming Games. In A. Ismael F. Vaz, João Paulo Almeida, José Fernando Oliveira, and Alberto Adrego Pinto, editors, *Operational Research: IO2017, Valença, Portugal, June 28-30 XVIII*, pages 11–23. Springer, 2018.
- Margarida Carvalho, Andrea Lodi, and João P Pedroso. Computing Equilibria for Integer Programming Games. *European Journal of Operational Research*, 303(3):1057–1070, 2022.
- Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. Integer Programming Games: A Gentle Computational Overview. In *Tutorials in Operations Research: Advancing the Frontiers of OR/MS: From Methodologies to Applications*, pages 31–51. INFORMS, 2023.
- Margarida Carvalho, Gabriele Dragotto, Andrea Lodi, and Sriram Sankaranarayanan. The Cut-and-Play algorithm: Computing Nash Equilibria via Outer Approximations. *Operations Research*, 2025.
- Bugra Caskurlu, Vahan Mkrtchyan, Ojas Parekh, and K Subramani. On Partial Vertex Cover and Budgeted Maximum Coverage Problems in Bipartite Graphs. In Josep Diaz, Ivan Lanese, and Davide Sangiorgi, editors, *IFIP International Conference on Theoretical Computer Science*, pages 13–26. Springer, 2014.
- Chandra Chekuri and Amit Kumar. Maximum Coverage Problem with Group Budget Constraints and Applications. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 72–83. Springer, 2004.
- Chen Chen, Wenbo Cai, İ Esra Büyüktaktakın, and Robert G Haight. A Game-theoretic Approach to Incentivize Landowners to Mitigate an Emerald Ash Borer Outbreak. *IISE Transactions*, pages 1–15, 2023.
- Tobias Crönert and Stefan Minner. Equilibrium Identification and Selection in Finite Games. *Operations Research*, 72(2):816–831, 2024.
- Gabriele Dragotto and Rosario Scatamacchia. The Zero Regrets Algorithm: Optimizing over Pure Nash Equilibria via Integer Programming. *INFORMS Journal on Computing*, 2023.
- Aloïs Duguet, Margarida Carvalho, Gabriele Dragotto, and Sandra Ulrich Ngueveu. Computing Approximate Nash Equilibria for Integer Programming Games. *Optimization Letters*, pages 1–25, 2025.
- Luis E Escobar, Sophie Mallez, Michael McCartney, Christine Lee, Daniel P Zielinski, Ratna Ghosal, Przemyslaw G Bajer, Carli Wagner, Becca Nash, Megan Tomamichel, et al. Aquatic Invasive Species in the Great Lakes Region: an Overview. *Reviews in Fisheries Science & Aquaculture*, 26(1):121–138, 2018.
- Alex Fabrikant and Christos H Papadimitriou. The Complexity of Game Dynamics: BGP Oscillations, Sink Equilibria, and Beyond. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 844–853, 2008.

- Michel Goemans, Vahab Mirrokni, and Adrian Vetta. Sink Equilibria and Convergence. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*, pages 142–151. IEEE, 2005.
- Gurobi Optimization. Gurobi Optimizer Reference Manual, 2024. URL <https://docs.gurobi.com/projects/optimizer/en/current/>.
- Robert G Haight, Amy C Kinsley, Szu-Yu Kao, Denys Yemshanov, and Nicholas BD Phelps. Optimizing the Location of Watercraft Inspection Stations to Slow the Spread of Aquatic Invasive Species. *Biological Invasions*, 23(12):3907–3919, 2021.
- Robert G Haight, Denys Yemshanov, Szu-Yu Kao, Nicholas BD Phelps, and Amy C Kinsley. A Bi-Level Model for State and County Aquatic Invasive Species Prevention Decisions. *Journal of Environmental Management*, 327:116855, 2023.
- Rashida Hakim, Jason Milionis, Christos Papadimitriou, and Georgios Piliouras. Swim Till You Sink: Computing the Limit of a Game. In *International Symposium on Algorithmic Game Theory*, pages 205–222. Springer, 2024.
- Torsten Heinrich, Yoojin Jang, Luca Mungo, Marco Pangallo, Alex Scott, Bassel Tarbush, and Samuel Wiese. Best-Response Dynamics, Playing Sequences, and Convergence to Equilibrium in Random Games. *International Journal of Game Theory*, 52(3):703–735, 2023.
- Szu-Yu Zoe Kao, Eva A Enns, Megan Tomamichel, Adam Doll, Luis E Escobar, Huijie Qiao, Meggan E Craft, and Nicholas BD Phelps. Network Connectivity of Minnesota Waterbodies and Implications for Aquatic Invasive Species Prevention. *Biological Invasions*, 23(10):3231–3242, 2021.
- Eyyüb Y Kılış, İ Esra Büyüктаhtakın, Robert G Haight, Najmaddin Akhundov, Kathleen Knight, and Charles E Flower. A Multistage Stochastic Programming Approach to the Optimal Surveillance and Control of the Emerald Ash Borer in Cities. *INFORMS Journal on Computing*, 33(2):808–834, 2021.
- Matthias Köppe, Christopher Thomas Ryan, and Maurice Queyranne. Rational Generating Functions and Integer Programming Games. *Operations Research*, 59(6):1445–1460, 2011.
- MAISRC. AIS Explorer: Data-Driven Approach to Inform Decision Making, 2025. <https://www.aisexplorer.umn.edu/>.
- Akihiko Matsui. Best Response Dynamics and Socially Stable Strategies. *Journal of Economic Theory*, 57(2):343–362, 1992.
- MDR. Minnesota Department of Revenue 2017, Aug 2017. URL https://www.revenue.state.mn.us/sites/default/files/2017-08/aispa_2018certification.pdf. Last accessed August 26, 2024.
- MDR. Local Aquatic Invasive Species Prevention Aid, 2025. <https://www.dnr.state.mn.us/invasives/ais/prevention/index.html>.

- Igal Milchtaich. Congestion Games with Player-specific Payoff Functions. *Games and Economic Behavior*, 13(1):111–124, 1996.
- Vahab S Mirrokni and Alexander Skopalik. On the Complexity of Nash Dynamics and Sink Equilibria. In *Proceedings of the 10th ACM conference on Electronic Commerce*, pages 1–10, 2009.
- Dov Monderer and Lloyd S Shapley. Potential Games. *Games and Economic Behavior*, 14(1):124–143, 1996.
- John F Nash. Equilibrium Points in N-person Games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
- Christos Papadimitriou and Georgios Piliouras. Game Dynamics as the Meaning of a Game. *ACM SIGecom Exchanges*, 16(2):53–63, 2019.
- Robert W Rosenthal. A Class of Games Possessing Pure-strategy Nash Equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- Tim Roughgarden. Algorithmic Game Theory. *Communications of the ACM*, 53(7):78–86, 2010.
- Simone Sagratella. Computing All Solutions of Nash Equilibrium Problems with Discrete Strategy Sets. *SIAM Journal on Optimization*, 26(4):2190–2218, 2016.
- Simone Sagratella, Marcel Schmidt, and Nathan Sudermann-Merx. The Noncooperative Fixed Charge Transportation Problem. *European Journal of Operational Research*, 284(1):373–382, 2020.
- Sriram Sankaranarayanan. Best-Response Algorithms for Lattice Convex Quadratic Simultaneous Games. *arXiv preprint arXiv:2405.07119*, 2024.
- Stefan Schwarze and Oliver Stein. A Branch-and-Prune Algorithm for Discrete Nash Equilibrium Problems. *Computational Optimization and Applications*, 86(2):491–519, 2023.

Appendix A: Acronyms and Notations

Table A1 lists acronyms used throughout the paper. Tables A2–A3 summarize the notation.

Table A1: List of Acronyms Used in the Manuscript and Appendix

Acronym	Meaning
IPG / IP	Integer programming game / integer program.
PNE	Pure Nash equilibrium.
BR	Best response (noun); best-response (adjective).
SCC	Strongly connected component in a directed graph.
DAG	Directed acyclic graph.
RR-BRD	Random-Restart best-response dynamics.
RRR-BRD	Round RR-BRD, i.e., a round-based implementation of RR-BRD.
EI	Equilibrium inequality.
ZR	Zero-regret (algorithm).
BZR	BRD-incorporated zero-regret (algorithm).
KPG	Knapsack problem game.
EBMC	Edge-weighted budgeted maximum coverage (game/model).
AIS	Aquatic invasive species.
OSW	Optimal social welfare (social optimum).
POS	Price of stability.

Table A2: Summary of Notation Used in the Manuscript and Appendix (Core Notation)

Symbol	Meaning
IPGs.	
\mathcal{G}	An integer programming game (IPG); the IPG tuple as defined in Section 2.1.
$N = \{1, \dots, n\}$	Set of players; $n := N $.
$m_i; m$	Number of integer variables of player i ; when m_i is the same for all $i \in N$, we write m .
X_i	Polyhedrally bounded pure-integer set of player i in the IPG (i.e., a finite strategy set).
X	Joint strategy set, $X := X_1 \times \dots \times X_n$.
$\mathbf{x}^i \in X_i; \mathbf{x} = (\mathbf{x}^i)_{i \in N}$	Player i 's strategy; joint strategy profile.
$u_i(\mathbf{x}^i, \mathbf{x}^{-i})$	Utility (objective value) of player i under profile \mathbf{x} .
$\phi(\mathbf{x})$	Social welfare function, $\phi(\mathbf{x}) = \sum_{i \in N} u_i(\mathbf{x})$.
$\bar{\mathbf{x}}_{sw}^*$	OSW solution.
$\hat{\mathbf{x}}_{pne}^*$	Best PNE (socially optimal PNE).
Normal-Form Games, BR State Graph, and RRR-BRD.	
S_i	Set of pure strategies of player i in the finite normal-form game.
$S = \prod_{i \in N} S_i$	Set of pure strategy profiles (normal-form viewpoint).
$s = (s_i, s_{-i})$	Strategy profile; s_{-i} denotes the strategies of all players except i .
$\mathcal{BR}_i^+(s_{-i})$	Set of (improving) best responses at s_{-i} .
SCC; sink SCC	Strongly connected component; a sink SCC has no outgoing edges in the condensation DAG.
Condensation DAG	Directed acyclic graph obtained by contracting SCCs of the BR state graph.
μ	Restart distribution over S used by RRR-BRD.
R	Round cap per attempt in RRR-BRD.
$q_\mu[R]$	Probability that an attempt from $s \sim \mu$ reaches a PNE within R rounds.
Equilibrium Inequalities and Lifted Formulations.	
$S_{PNE}(\mathcal{G})$	Set of pure Nash equilibria of game \mathcal{G} .
\mathcal{K}	Lifted/linearized formulation of the joint strategy profile.
\mathcal{E}	Equilibrium closure.
Ω	Set over which equilibrium inequalities are generated/enforced.

Table A3: Summary of Notation Used in the Manuscript and Appendix (Applications and Experiments)

Symbol	Meaning
EBMC Games.	
$i \in N$	Player index in the EBMC model/games.
\mathcal{I}_i	Set of vertices controlled by player i (player-level decision set).
x_j	Binary variable indicating whether vertex j is selected.
y_{jk}	Binary variable indicating whether arc (j, k) is covered.
\mathcal{B}_i	Budget of player i (inspection budget).
w_{jk}	Edge weight for edge (j, k) .
$\mathcal{G}_{ \mathcal{L} =1}^{\text{Self}}$	Selfish EBMC game with a single AIS type.
$\mathcal{G}_{ \mathcal{L} \geq 2}^{\text{Self}}$	Selfish EBMC game with multiple AIS types.
\mathcal{G}^{Alt}	Locally altruistic EBMC game.
KPGs.	
J	Set of items in the knapsack problem game.
$x_j^i \in \{0, 1\}$	Binary selection of item j by player i .
\mathcal{B}_i	Knapsack capacity of player i (in KPG).
p_j^i	Profit of item j for player i .
$f_{k,j}^i$	Bilinear coefficient associated with item j for players i and k .
Computational Results.	
$\tilde{q}_\mu[R]$	$q_\mu[R]$ estimated via Monte Carlo simulation.
$\mathbf{0}^N$	Initial strategy profile (all-zero vector).
$\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}$	Initial strategy profile obtained by solving the SW model with a time limit (seconds).
$\tilde{\mathbf{x}}_{\text{ng}}$	Non-game strategy profile.
$\hat{\mathbf{x}}_{\text{pne}}^1 / \hat{\mathbf{x}}_{\text{pne}}^*$	First identified PNE / best identified PNE (maximizing ϕ) within the time limit.
$\phi(\cdot)$	Social welfare function; values are reported in thousands and rounded to the nearest tenth.
$\widetilde{\text{OSW}}$	Approximate optimal social welfare, $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}})$.
$\widetilde{\text{POS}}$	Approximate price of stability, $\phi(\tilde{\mathbf{x}}_{\text{sw}}^{\text{time}}) / \phi(\hat{\mathbf{x}}_{\text{pne}}^*)$.
$\#\text{cuts}(\text{ZR}) / \#\text{cuts}(\text{BZR})$	Number of equilibrium inequalities added by ZR / BZR.
$\#\text{pne}(\text{ZR}) / \#\text{pne}(\text{BZR})$	Number of distinct PNEs found by ZR / BZR.
$\text{T}(\text{ZR}^{\text{1st}}) / \text{T}(\text{BZR}^{\text{1st}})$	Time to find the first PNE using ZR / BZR.
$\text{T}(\text{ZR}) / \text{T}(\text{BZR})$	Total runtime of the ZR / BZR algorithm.

Appendix B: Proofs

Proof of Theorem 1

Proof. (i) For attempt r , let $I_r := \mathbf{1}\{\tau_{\text{PNE}} \leq R \text{ in attempt } r\}$, i.e., $I_r = 1$ iff the r -th attempt reaches (and certifies) a PNE within at most R rounds. Across attempts we independently resample $s \sim \mu$ and re-draw fresh round permutations; hence $(I_r)_{r \geq 1}$ are i.i.d. Bernoulli with parameter $q_\mu[R]$. Let

$$R^* := \inf\{r \geq 1 : I_r = 1\}$$

be the number of attempts until success. Then

$$\Pr(R^* > r) = (1 - q_\mu[R])^r \xrightarrow{r \rightarrow \infty} 0,$$

so $\Pr(R^* < \infty) = 1$ and the algorithm finds a PNE almost surely.

(ii) Let L_r be the number of BR solves in attempt r . For each failed attempt $r < R^*$, exactly R rounds are executed, so $L_r = nR$. On the successful attempt, a PNE is first hit during one of the R rounds, and the algorithm certifies it by the end-of-round no-change test; thus $L_{R^*} \leq nR$. Taking expectations and using $\mathbb{E}[R^*] = 1/q_\mu[R]$ (geometric),

$$\mathbb{E}\left[\sum_{r=1}^{R^*} L_r\right] \leq nR \mathbb{E}[R^*] = \frac{nR}{q_\mu[R]}.$$

This establishes the bound in part (ii). □

Proof of Corollary 1

Proof. Because each \mathcal{X}_i is a bounded integer polyhedron, it contains finitely many integer points, so $\mathcal{X} = \prod_{i \in N} \mathcal{X}_i$ is finite and can be viewed as the pure strategy space S of a finite normal-form game. The RRR–BRD–IPG algorithm (Algorithm 2) is exactly RRR–BRD (Algorithm 1) applied to this finite game, with restart law μ induced by `RANDOM_GENERATION`. If \mathcal{G} admits a PNE and $q_\mu[R] > 0$, then the hypotheses of Theorem 1 hold, so the almost-sure convergence and the bound on the expected number of BR solves follow directly. □

Proof of Proposition 1

Proof. By Definition 1, $\mathcal{A}^-[I_i] := \delta_-(I_i) \cup \mathcal{A}_i$. Fix any edge $(j, k) \in \mathcal{A}$. Since $\mathcal{I} = \bigsqcup_{i \in N} I_i$ is a partition, there is a *unique* index $i \in N$ such that $k \in I_i$. There are two cases: (i) if $j \in I_i$ as well, then $(j, k) \in \mathcal{A}_i \subseteq \mathcal{A}^-[I_i]$; (ii) if $j \notin I_i$, then $(j, k) \in \delta_-(I_i) \subseteq \mathcal{A}^-[I_i]$. Hence every edge $(j, k) \in \mathcal{A}$ belongs to $\bigcup_{i \in N} \mathcal{A}^-[I_i]$. Furthermore, since each edge (j, k) is unique, it must exclusively belong to one set $\mathcal{A}^-[I_i]$. Therefore, $\mathcal{A} = \bigsqcup_{i \in N} \mathcal{A}^-[I_i]$.

Using this partition and the definition of u_i^{Self} , we obtain

$$\phi(\mathbf{x}) = \sum_{(j,k) \in \mathcal{A}} w_{jk} y_{jk} = \sum_{i \in N} \sum_{(j,k) \in \mathcal{A}^-[I_i]} w_{jk} y_{jk} = \sum_{i \in N} \tilde{u}_i^{\text{Self}}(\mathbf{y}) = \sum_{i \in N} u_i^{\text{Self}}(\mathbf{x}),$$

which proves the claim. \square

Proof of Theorem 2

Proof. Fix a player $i \in N$. The arcs of the entire graph can be partitioned into those that are incident to I_i and those that are not, i.e., $\mathcal{A} = \mathcal{A}[I_i] \sqcup \mathcal{A}_{-i}$, where $\mathcal{A}[I_i] = \mathcal{A}_i \cup \delta_+(I_i) \cup \delta_-(I_i)$ and \mathcal{A}_{-i} collects all remaining arcs. For a given profile $(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$, the potential can be written as

$$\begin{aligned} \phi(\mathbf{x}) &= \sum_{j,k \in I_i} w_{jk} \cdot \max(x_j, x_k) + \sum_{j \in I_i, k \in I_{-i}} w_{jk} \cdot \max(x_j, \bar{x}_k) + \sum_{j \in I_{-i}, k \in I_i} w_{jk} \cdot \max(\bar{x}_j, x_k) + \sum_{j,k \in I_{-i}} w_{jk} \cdot \max(\bar{x}_j, \bar{x}_k) \\ &= \sum_{(j,k) \in \mathcal{A}[I_i]} u_i^{\text{Alt}}(\mathbf{x}^i, \bar{\mathbf{x}}^{-i}) + \sum_{j,k \in I_{-i}} w_{jk} \cdot \max(\bar{x}_j, \bar{x}_k). \end{aligned}$$

The second term depends only on $\bar{\mathbf{x}}^{-i}$ and is therefore constant with respect to changes in \mathbf{x}^i . The first term is precisely $u_i^{\text{Alt}}(\mathbf{x}^i, \bar{\mathbf{x}}^{-i})$ (where $\tilde{u}_i^{\text{Alt}}(\mathbf{y})$ is rewritten as $u_i^{\text{Alt}}(\mathbf{x})$). Hence, for any $\bar{\mathbf{x}}^i, \hat{\mathbf{x}}^i \in \mathcal{X}_i$,

$$\phi(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - \phi(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) = u_i^{\text{Alt}}(\hat{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}) - u_i^{\text{Alt}}(\bar{\mathbf{x}}^i, \bar{\mathbf{x}}^{-i}),$$

which shows that ϕ is an exact potential function for the locally altruistic game. \square

Proof of Corollary 2

Proof. By Theorem 2, the locally altruistic EBMC games are exact potential games with potential function $\phi(\mathbf{x})$, and $\bar{\mathbf{x}}_{\text{sw}}^*$ is an optimal solution maximizing $\phi(\mathbf{x})$ over $\mathbf{x} \in \mathcal{X}$. By Lemma 2.1 of Monderer and Shapley [1996], which states that any maximizer of a potential function is a PNE, $\bar{\mathbf{x}}_{\text{sw}}^*$ is a PNE for locally altruistic EBMC games. \square

Proof of Theorem 3.

Proof. Case 1: $\mathcal{I}^{\text{ng}} \subseteq \mathcal{I}^0$. Fix $i \in N$. Since vertices in \mathcal{I}^0 have no types, other players' selections in \mathcal{I}^0 create no relevant coverage benefit for player i in the selfish utility. Hence, for all $\mathbf{x}^i \in \mathcal{X}_i$,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) = u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}).$$

By definition of $\bar{\mathbf{x}}_{\text{ng}}^i$,

$$\bar{\mathbf{x}}_{\text{ng}}^i \in \arg \max_{\mathbf{x}^i \in \mathcal{X}_i} u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}),$$

so $\bar{\mathbf{x}}_{\text{ng}}^i$ is also a best response to $\bar{\mathbf{x}}_{\text{ng}}^{-i}$. Thus $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE (for both $|\mathcal{L}| = 1$ and $|\mathcal{L}| \geq 2$).

Case 2: $\mathcal{I}^{\text{ng}} \subseteq \mathcal{I}^+$ and $|\mathcal{L}| = 1$. Fix $i \in N$. For any $\mathbf{x}^i \in \mathcal{X}_i$, overlap of covered edges implies

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\mathbf{x}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Since $\bar{\mathbf{x}}_{\text{ng}}^i$ maximizes $u_i^{\text{self}}(\cdot, \mathbf{0}^{-i})$,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Under $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, the instance is bipartite as in Section 5.3, with each coverable edge having one endpoint in \mathcal{I}^+ and one in \mathcal{I}^0 . With $\mathcal{I}^{\text{ng}} \subseteq \mathcal{I}^+$, there is no double counting of edges, so

$$u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \mathbf{0}^{-i}) + u_i^{\text{self}}(\mathbf{0}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) = u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}).$$

Therefore,

$$u_i^{\text{self}}(\mathbf{x}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \leq u_i^{\text{self}}(\bar{\mathbf{x}}_{\text{ng}}^i, \bar{\mathbf{x}}_{\text{ng}}^{-i}) \quad \forall \mathbf{x}^i \in \mathcal{X}_i,$$

so $\bar{\mathbf{x}}_{\text{ng}}^i$ is a best response to $\bar{\mathbf{x}}_{\text{ng}}^{-i}$. Since i is arbitrary, $\bar{\mathbf{x}}_{\text{ng}}$ is a PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$. □

Appendix C: Random Feasible-Start Generators for KPG and EBMC

We provide two randomized feasible-start generators for each IPG class studied in this paper: (i) a *maximal* generator, which tends to produce strategies that are close to capacity-filling and thus resemble typical best responses; and (ii) a *full-support* generator, which assigns strictly positive probability to every feasible pure strategy. In our RRR–BRD implementation, we alternate between these two variants (e.g., using the maximal generator on odd-numbered restarts and the full-support generator on even-numbered restarts), although any mixture is admissible.

The design goals are: (a) every feasible profile in \mathcal{X} has nonzero probability under the full-support generator (for the theoretical guarantees in Section 3.3); and (b) the generators remain lightweight (expected linear time) so that restarts are inexpensive.

Notation. Let $N = \{1, \dots, n\}$ be the set of players/counties. In KPG, let J be the set of items, $w_j^i \geq 0$ the weight of item $j \in J$ for player $i \in N$, and $\mathcal{B}_i \geq 0$ the capacity. A pure strategy for player i is $\mathbf{x}^i \in \{0, 1\}^{|J|}$ with $\sum_{j \in J} w_j^i x_j^i \leq \mathcal{B}_i$. In EBMC, let \mathcal{I}_i denote the lakes in county i and $\mathcal{B}_i \in \mathbb{Z}_{\geq 0}$ its inspection budget; a pure strategy is $\mathbf{x}^i \in \{0, 1\}^{|\mathcal{I}_i|}$ with $\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i$.

Generators for Knapsack Problem Games (KPG)

Maximal generator (random-order greedy). This generator matches the implementation used in our code: for each player, we take a randomized ordering of items and perform a single greedy pass, adding an item whenever it fits in the remaining capacity.

Algorithm 4 MAXRANDOMGENERATION-KPG (random-order greedy)

Require: Items J , capacities $(\mathcal{B}_i)_{i \in N}$, weights $(w_j^i)_{i \in N, j \in J}$
Ensure: Feasible profile $\bar{\mathbf{x}} \in \prod_{i \in N} \{\mathbf{x}^i \in \{0, 1\}^{|J|} : \sum_j w_j^i x_j^i \leq \mathcal{B}_i\}$

- 1: **for each** $i \in N$ **do**
- 2: $S \leftarrow \emptyset$, $R \leftarrow \mathcal{B}_i$
- 3: $\pi \leftarrow \text{RANDOMPERMUTATION}(J)$ ▷ encodes shuffle
- 4: **for each** $j \in \pi$ **do**
- 5: $w \leftarrow w_j^i$
- 6: **if** $w \leq R$ **then**
- 7: $S \leftarrow S \cup \{j\}$; $R \leftarrow R - w$
- 8: Set $x_j^i \leftarrow \mathbf{1}\{j \in S\}$ for all $j \in J$
- 9: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$

Algorithm 4 runs in $O(|N||J|)$ time and always returns a feasible strategy for each player, since items are only accepted when the remaining capacity R allows. Because all randomness sits in the permutation π , it yields a diverse collection of near-maximal solutions that typically resemble best responses.

Full-support generator (Bernoulli + rejection). The full-support generator uses a Bernoulli–rejection scheme. For each player i , we choose a Bernoulli parameter p_i that reflects the “density” of the knapsack, typically on the order of the mean fill ratio \mathcal{B}_i/W_i , where $W_i := \sum_{j \in J} w_j^i$:

Every feasible knapsack pattern for player i can be generated with positive probability in a single Bernoulli

Algorithm 5 RANDOMGENERATION-KPG (full-support, expected linear time)

Require: Items J , capacities $(\mathcal{B}_i)_{i \in N}$, weights $(w_j^i)_{i \in N, j \in J}$

Ensure: Feasible profile $\bar{\mathbf{x}} \in \prod_{i \in N} \{\mathbf{x}^i \in \{0, 1\}^{|J|} : \sum_j w_j^i x_j^i \leq \mathcal{B}_i\}$

- 1: **for each** $i \in N$ **do**
 - 2: $W_i \leftarrow \sum_{j \in J} w_j^i$
 - 3: Choose p_i in $(0, 1)$, e.g. $p_i \leftarrow \mathcal{B}_i/W_i$
 - 4: **repeat**
 - 5: **for each** $j \in J$ **do**
 - 6: Draw $x_j^i \sim \text{Ber}(p_i)$ independently
 - 7: **until** $\sum_{j \in J} w_j^i x_j^i \leq \mathcal{B}_i$ ▷ reject if capacity violated
 - 8: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$
-

draw (because each coordinate is independent and $p_i \in (0, 1)$), and is accepted whenever it satisfies the capacity constraint. Thus, after accounting for the rejection step, RANDOMGENERATION-KPG induces a full-support distribution over the feasible set. For moderate p_i and capacities, the expected number of rejection loops is small, so the expected runtime remains linear in $|J|$ per player.

In RRR–BRD, we alternate these two generators across restarts (e.g., MAXRANDOMGENERATION-KPG for odd restarts and RANDOMGENERATION-KPG for even restarts), combining capacity-filling starts with fully supported random perturbations.

Generators for EBMC

For EBMC, budgets are cardinality bounds, so we can work directly with subsets of lakes.

Maximal generator (uniform cardinality sampling). Here we sample a subset of lakes of size $b_i := \min\{\mathcal{B}_i, |\mathcal{I}_i|\}$ uniformly without replacement.

Algorithm 6 MAXRANDOMGENERATION-EBMC (uniform cardinality)

Require: County lakes $(\mathcal{I}_i)_{i \in N}$, budgets $(\mathcal{B}_i)_{i \in N}$

Ensure: Feasible profile with $\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i$ for all $i \in N$

- 1: **for each** $i \in N$ **do**
 - 2: $b_i \leftarrow \min\{\mathcal{B}_i, |\mathcal{I}_i|\}$
 - 3: $S \leftarrow \text{SAMPLEWITHOUTREPLACEMENT}(\mathcal{I}_i, b_i)$ ▷ uniform over $\binom{|\mathcal{I}_i|}{b_i}$
 - 4: Set $x_j \leftarrow \mathbf{1}\{j \in S\}$ for all $j \in \mathcal{I}_i$
 - 5: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$
-

This generator runs in $O(|\mathcal{I}_i|)$ per county (via a shuffle-and-take routine) and samples uniformly among all maximal (cardinality- b_i) feasible strategies.

Full-support generator (Bernoulli + rejection). To obtain full support in EBMC, we again use a Bernoulli–rejection scheme. Here the natural density proxy is the ratio $\mathcal{B}_i/|\mathcal{I}_i|$:

As in the KPG case, every feasible subset of \mathcal{I}_i has positive probability under this procedure, and the expected number of rejection loops remains small for moderate p_i and budgets.

Remark. In summary, for both KPG and EBMC we alternate between maximal and full-support generators across restarts. Formally, let μ^{\max} and μ^{full} denote the restart laws on \mathcal{X} induced when all restarts use

Algorithm 7 RANDOMGENERATION-EBMC (full-support)

Require: County lakes $(\mathcal{I}_i)_{i \in N}$, budgets $(\mathcal{B}_i)_{i \in N}$

Ensure: Feasible profile with $\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i$ for all $i \in N$

- 1: **for each** $i \in N$ **do**
 - 2: Choose p_i in $(0, 1)$, e.g. $p_i \leftarrow \mathcal{B}_i / |\mathcal{I}_i|$
 - 3: **repeat**
 - 4: **for each** $j \in \mathcal{I}_i$ **do**
 - 5: Draw $x_j \sim \text{Ber}(p_i)$ independently
 - 6: **until** $\sum_{j \in \mathcal{I}_i} x_j \leq \mathcal{B}_i$
 - 7: **return** $\bar{\mathbf{x}} = (\mathbf{x}^i)_{i \in N}$
-

the maximal generator and the full-support generator, respectively. If a PNE exists and the full-support generator is used (so the restart law is μ^{full}), then $q_{\mu^{\text{full}}}[R] > 0$ for all $R \geq 1$. In practice, however, the maximal generator often produces starting profiles that are closer to best responses and can empirically yield larger success probabilities ($q_{\mu^{\text{max}}}[R] > q_{\mu^{\text{full}}}[R]$), even though these quantities are not computable in closed form. Alternating the two variants balances theoretical robustness (via the full-support generator) with the potential efficiency of more structured, capacity-filling starts.

Appendix D: Extremely Small $q_\mu[R]$

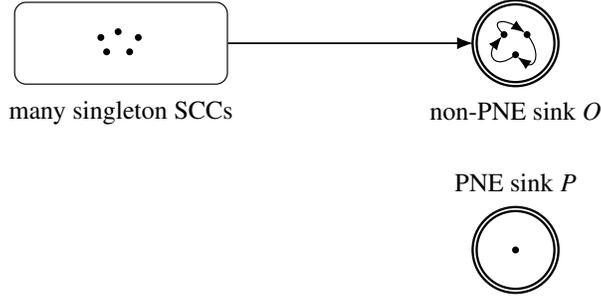


Figure A1: A Condensation DAG Showing That $q_\mu[R]$ Can Be Extremely Small.

Figure A1 highlights that the instance-dependent success probability $q_\mu[R]$ (4) in RRR–BRD can be extremely small even when a PNE exists. Assuming that the restart law μ has full support on the restart set and that R is sufficiently large, the diagram can be interpreted as the condensation DAG of the best-response state graph. In the schematic, O is an absorbing non-PNE sink SCC, P is an absorbing PNE sink SCC, and the remaining node(s) represent (possibly many) transient SCCs that flow into O . In particular, P may be a singleton and may be *unreachable* from any non-PNE start under best-response moves, so that a run succeeds only if the restart lands directly in P . Under full-support restart laws (e.g., uniform sampling over feasible profiles), the success probability $q_\mu[R]$ can therefore be made arbitrarily small by scaling the number of transient states (and hence their μ -mass) relative to P .

A knapsack-style IPG inducing vanishing $q_\mu[R]$. We now give a concrete IPG with a single knapsack constraint per player for which $q_\mu[R]$ decreases exponentially in M , matching the structure in Figure A1. Consider a two-player IPG ($N = \{1, 2\}$) with $m := M$ binary decisions indexed by $j \in \{1, \dots, M\}$. Each player i chooses a binary vector $\mathbf{x}^i \in \{0, 1\}^m$ subject to the single knapsack constraint

$$\sum_{j=1}^M x_j^i \leq M,$$

so every subset is feasible.

We use the standard IPG utility form

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{d}^i)^\top \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top Q_p^i \mathbf{x}^i,$$

with $Q^i \equiv 0$ (no within-player quadratic term). Define the cyclic successor map on indices $g : \{1, \dots, M\} \rightarrow \{1, \dots, M\}$ by $g(j) = j + 1$ for $j = 1, \dots, M - 1$ and $g(M) = 1$.

Set linear coefficients $\mathbf{d}^1 = \mathbf{d}^2 = -c \mathbf{1}$ for some $c > 0$, and define the interaction matrices by the following nonzero entries (all others are 0):

$$Q_2^1[j, j] = T \quad (j = 1, \dots, M), \quad Q_1^2[g(j), j] = T \quad (j = 1, \dots, M),$$

for some $T > c$. Under these utilities, player 1's unique best response to \mathbf{x}^2 is to choose exactly the same subset $\mathbf{x}^1 = \mathbf{x}^2$, and player 2's unique best response to \mathbf{x}^1 is to choose the shifted subset $x_{g(j)}^2 = x_j^1$ for all j .

Identification of the sinks O and P and the transient region. The profile $(\mathbf{x}^1, \mathbf{x}^2) = (\mathbf{0}, \mathbf{0})$ is a PNE: each player obtains payoff 0, while any deviation to a nonzero subset yields negative payoff due to the linear cost. Thus the singleton $\{(\mathbf{0}, \mathbf{0})\}$ is the PNE sink P in Figure A1.

In contrast, no profile with $\mathbf{x}^1 \neq \mathbf{0}$ or $\mathbf{x}^2 \neq \mathbf{0}$ can be a PNE. If $\mathbf{x}^2 \neq \mathbf{0}$, then player 1's unique best response is $\mathbf{x}^1 = \mathbf{x}^2 \neq \mathbf{0}$. If $\mathbf{x}^1 \neq \mathbf{0}$, then player 2's unique best response is the shifted subset $g(\mathbf{x}^1) \neq \mathbf{x}^1$, and hence \mathbf{x}^2 must be updated unless it already equals that shift. Restricting to the “active” profiles with $\mathbf{x}^1 \neq \mathbf{0}$ and $\mathbf{x}^2 \neq \mathbf{0}$, best-response updates cycle on the orbit induced by g , forming an absorbing strongly connected component O containing no PNE. All other non-PNE profiles are transient singleton SCCs that flow into O under improving best responses. This matches the schematic: a (possibly very large) transient region feeding into the non-PNE sink O , with a separate singleton PNE sink P .

Moreover, $(\mathbf{0}, \mathbf{0})$ is not reachable from any non-PNE start under improving best-response moves: when the opponent plays a nonzero subset, deviating to $\mathbf{0}$ yields payoff 0, whereas the prescribed best response yields strictly positive payoff because $T > c$. Thus, all non-PNE starts eventually enter O and remain there.

Implication for $q_\mu[R]$ under full-support restarts. Let the restart law μ have full support over the restart set, e.g., uniform over $\{0, 1\}^m \times \{0, 1\}^m$. In this instance, $P = \{(\mathbf{0}, \mathbf{0})\}$ is not reachable from any non-PNE start under improving best responses, whereas all other starts enter the non-PNE sink O and fail for any finite round cap R . Consequently,

$$q_\mu[R] = \mu(P) = \mu((\mathbf{0}, \mathbf{0})) = \frac{1}{2^{2M}},$$

which can be made arbitrarily small by increasing M . Equivalently, the expected number of restarts required to hit a PNE is $1/q_\mu[R] = 2^{2M}$, so RRR-BRD (Algorithm 2) can be exponentially weak in expectation on this instance.

Moreover, since the number of restarts until success is geometric, tail probabilities are non-negligible:

$$\Pr(R^* > r) = (1 - q_\mu[R])^r.$$

With $q_\mu[R] = 2^{-2M}$, we have

$$\Pr(R^* > 2^{2M}) = (1 - 2^{-2M})^{2^{2M}} \approx e^{-1},$$

so even at the mean scale 2^{2M} there remains a constant probability of requiring more restarts (and hence more work).

Appendix E: Full Simulation Results

Tables A4–A5 report full simulation results for the normal-form reductions and KPG instances. Most settings exhibit large $\tilde{q}_\mu[R]$ and small average rounds, while a minority of instances have much smaller $\tilde{q}_\mu[R]$, leading to larger ESB and longer runs; this instance dependence is most visible in the harder KPG configurations (notably type C).

Table A4: Full Simulation Results for Normal-form reductions

n	k	inst	trials	successes	$\tilde{q}_\mu[R]$	avg_rounds_all	ESB ($\frac{nR}{q_\mu[R]}$)	Time (secs)
2	3	1	64	64	1.00	2.84	20.0	0.01
2	3	2	64	64	1.00	2.83	20.0	0.00
2	3	3	64	64	1.00	2.52	20.0	0.00
2	3	4	64	64	1.00	2.58	20.0	0.00
2	3	5	64	64	1.00	2.66	20.0	0.00
2	3	6	64	64	1.00	2.44	20.0	0.00
2	3	7	64	64	1.00	2.73	20.0	0.00
2	3	8	64	64	1.00	2.63	20.0	0.00
2	3	9	64	64	1.00	2.39	20.0	0.00
2	3	10	64	64	1.00	2.34	20.0	0.00
2	4	1	256	256	1.00	2.34	20.0	0.02
2	4	2	256	64	0.25	8.04	80.0	0.11
2	4	3	256	256	1.00	3.18	20.0	0.05
2	4	4	256	256	1.00	2.62	20.0	0.03
2	4	5	256	256	1.00	2.84	20.0	0.05
2	4	6	256	256	1.00	2.88	20.0	0.04
2	4	7	256	256	1.00	2.20	20.0	0.02
2	4	8	256	256	1.00	3.63	20.0	0.05
2	4	9	256	256	1.00	2.13	20.0	0.02
2	4	10	256	256	1.00	2.41	20.0	0.04
3	3	1	512	410	0.80	6.32	37.5	0.13
3	3	2	512	287	0.56	6.58	53.5	0.12
3	3	3	512	176	0.34	7.56	87.3	0.14
3	3	4	512	332	0.65	5.86	46.3	0.13
3	3	5	512	45	0.09	9.37	341.3	0.16
3	3	6	512	512	1.00	2.89	30.0	0.06
3	3	7	512	69	0.13	8.98	222.6	0.20
3	3	8	512	334	0.65	5.79	46.0	0.12
3	3	9	512	310	0.61	6.66	49.5	0.14
3	3	10	512	512	1.00	3.80	30.0	0.09
3	4	1	4096	4096	1.00	3.42	30.0	1.09
3	4	2	4096	2542	0.62	6.86	48.3	2.20
3	4	3	4096	612	0.15	9.31	200.8	2.95
3	4	4	4096	3173	0.77	6.04	38.7	1.92
3	4	5	4096	4096	1.00	3.41	30.0	1.16
3	4	6	4096	2542	0.62	6.47	48.3	2.10
3	4	7	4096	3566	0.87	4.65	34.5	1.59
3	4	8	4096	943	0.23	8.36	130.3	2.62
3	4	9	4096	4096	1.00	3.93	30.0	1.30
3	4	10	4096	3702	0.90	4.58	33.2	1.61

Table A5: Full Simulation Results for KPGs

type	n	items	budget	trials	successes	$\tilde{q}_\mu[R]$	avg_rounds_all	ESB ($\frac{nR}{q_\mu[R]}$)	Time (secs)
A	2	25	2	200	200	1.00	2.50	40.0	0.30
A	2	25	5	200	200	1.00	2.72	40.0	0.34
A	2	25	8	200	200	1.00	2.94	40.0	0.44
A	2	50	2	200	200	1.00	3.18	40.0	0.58
A	2	50	5	200	200	1.00	2.58	40.0	0.56
A	2	50	8	200	200	1.00	2.71	40.0	0.66
A	2	75	2	200	200	1.00	3.25	40.0	0.75
A	2	75	5	200	200	1.00	3.57	40.0	1.14
A	2	75	8	200	200	1.00	3.46	40.0	1.62
A	2	100	2	200	200	1.00	3.26	40.0	1.03
A	2	100	5	200	200	1.00	3.66	40.0	1.56
A	2	100	8	200	200	1.00	3.17	40.0	4.50
A	3	25	2	200	200	1.00	3.21	60.0	0.60
A	3	25	5	200	200	1.00	3.40	60.0	0.79
A	3	25	8	200	200	1.00	3.15	60.0	0.75
A	3	50	2	200	200	1.00	3.59	60.0	0.99
A	3	50	5	200	200	1.00	3.17	60.0	1.25
A	3	50	8	200	200	1.00	3.78	60.0	1.56
A	3	75	2	200	200	1.00	4.34	60.0	1.71
A	3	75	5	200	200	1.00	3.42	60.0	1.81
A	3	75	8	200	200	1.00	4.13	60.0	2.35
A	3	100	5	200	200	1.00	4.66	60.0	7.34
A	3	100	8	200	200	1.00	4.02	60.0	9.56
B	2	25	2	200	200	1.00	3.08	40.0	0.34
B	2	25	5	200	200	1.00	3.22	40.0	0.45
B	2	25	8	200	200	1.00	2.85	40.0	0.42
B	2	50	2	200	200	1.00	3.52	40.0	0.63
B	2	50	5	200	200	1.00	3.71	40.0	0.87
B	2	50	8	200	200	1.00	3.04	40.0	0.70
B	2	75	2	200	200	1.00	3.28	40.0	0.88
B	2	75	5	200	200	1.00	3.97	40.0	1.21
B	2	75	8	200	200	1.00	3.69	40.0	2.95
B	2	100	2	200	200	1.00	3.26	40.0	1.08
B	2	100	5	200	200	1.00	3.58	40.0	5.61
B	2	100	8	200	200	1.00	3.31	40.0	5.99
B	3	25	2	200	200	1.00	3.45	60.0	0.82
B	3	25	5	200	200	1.00	3.64	60.0	0.83
B	3	25	8	200	200	1.00	3.46	60.0	0.74
B	3	50	2	200	200	1.00	4.05	60.0	1.26
B	3	50	5	200	200	1.00	4.43	60.0	1.57
B	3	50	8	200	200	1.00	3.63	60.0	1.34
B	3	75	5	200	200	1.00	4.55	60.0	2.19
B	3	75	8	200	200	1.00	4.11	60.0	2.37
C	2	25	8	200	200	1.00	2.62	40.0	0.27
C	2	50	2	200	200	1.00	3.66	40.0	0.63
C	2	50	8	200	200	1.00	3.04	40.0	0.38
C	2	75	2	200	3	0.02	19.75	2666.7	5.36
C	2	75	8	200	74	0.37	13.85	108.1	2.92
C	2	100	2	200	156	0.78	7.53	51.3	2.59
C	3	25	2	200	1	0.01	19.92	12000.0	3.42

Appendix F: Example for Best-Response Dynamics incorporated Zero-Regret Algorithm

We exemplify how RRR–BRD, BZR, and ZR operate on a small two-player instance. In this example the performance gap is not pronounced, but the mechanics are transparent.

$$\textbf{Player 1: } \max_{x^1 \in \{0,1\}^3} 3x_1^1 + 2x_2^1 + 2x_3^1 - 4x_1^1x_1^2 - 5x_2^1x_2^2 - 3x_3^1x_3^2 \quad \text{s.t.} \quad x_1^1 + x_2^1 + x_3^1 \leq 2.$$

$$\textbf{Player 2: } \max_{x^2 \in \{0,1\}^3} 2x_1^2 + x_2^2 + 0x_3^2 + 5x_1^2x_1^1 + 2x_2^2x_2^1 + x_3^2x_3^1 \quad \text{s.t.} \quad x_1^2 + x_2^2 + x_3^2 \leq 2.$$

The game has a unique PNE $\hat{\mathbf{x}}_{\text{pne}}^* = \{(0, 0, 1), (1, 1, 0)\}$ with social welfare $\phi(\hat{\mathbf{x}}_{\text{pne}}^*) = 5$. We use $\mathbf{0}^N = \{(0, 0, 0), (0, 0, 0)\}$ as a starting profile.

RRR–BRD: Assume the (round-wise) random playing sequences are $[1, 2] \rightarrow [2, 1] \rightarrow [1, 2]$. Starting from $\mathbf{0}^N$, RRR–BRD follows the BR path

$$\{(0, 0, 0), (0, 0, 0)\} \xrightarrow{1} \{(1, 1, 0), (0, 0, 0)\} \xrightarrow{2} \{(1, 1, 0), (1, 1, 0)\} \xrightarrow{1} \{(0, 0, 1), (1, 1, 0)\},$$

and the final round $[1, 2]$ certifies this profile via the end-of-round no-change test.

BZR: In our implementation, BZR first calls the separation oracle at an incumbent integer solution. Suppose the MIP solver produces $\{(0, 0, 0), (0, 0, 0)\}$. This profile is not a PNE: player 1 has BR $(1, 1, 0)$ and player 2 has BR $(1, 1, 0)$, yielding two violated equilibrium inequalities (EIs), which are added:

$$5 - 4x_1^2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3, \quad 3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3.$$

Since $\{(0, 0, 0), (0, 0, 0)\}$ is not a PNE, BZR then invokes RRR–BRD and finds the PNE $\{(0, 0, 1), (1, 1, 0)\}$ under $[1, 2] \rightarrow [2, 1] \rightarrow [1, 2]$. Along this BRD trajectory, player 1's BRs are $(1, 1, 0)$ and $(0, 0, 1)$, and player 2's BR is $(1, 1, 0)$; thus player 1's BR $(0, 0, 1)$ provides an additional inequality:

$$2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3.$$

After adding these inequalities, BZR resolves and terminates once the separation oracle confirms that the incumbent $\{(0, 0, 1), (1, 1, 0)\}$ (maximizing social welfare) is indeed the best PNE.

ZR: ZR first checks $\{(1, 0, 1), (1, 1, 0)\}$ (social welfare = 9), generating the violated EI from player 1's BR $(0, 0, 1)$:

$$2 - 3x_3^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3.$$

Next it evaluates $\{(1, 1, 0), (1, 0, 1)\}$ (social welfare = 8), generating two violated EIs from player 1's BR $(0, 1, 0)$ and player 2's BR $(1, 1, 0)$:

$$2 - 5x_2^2 \leq 3x_1^1 + 2x_2^1 + 2x_3^1 - 4z_1 - 5z_2 - 3z_3,$$

$$3 + 5x_1^1 + 2x_2^1 \leq 2x_1^2 + x_2^2 + 0x_3^2 + 5z_1 + 2z_2 + z_3.$$

Finally, $\{(0, 0, 1), (1, 1, 0)\}$ is tested and confirmed as the best PNE.

Comparison: BZR requires one separation call at the initial incumbent, one RRR–BRD run, and three EIs in total (two from the initial separation and one from the BRD trajectory), resulting in two separation calls overall (including the final call that certifies the best PNE). ZR requires three separation calls and generates three EIs. This gap can widen substantially in large-scale IPGs where many separation rounds may be needed. We include this example for *illustrative* purposes; in the large-scale IPG instances studied in this paper, the performance gap between BZR and ZR is often substantial.

Appendix G: Exponential Separation Between ZR and BZR

This section gives a simple worst-case construction showing an exponential separation between ZR and BZR when *maximizing social welfare over PNEs* (i.e., the welfare-maximizing equilibrium problem). We use the n -fold product of the classical Prisoner’s Dilemma and the standard welfare $\phi = u_1 + u_2$ to illustrate that there can be exponentially many high-welfare non-equilibria, forcing welfare-guided equilibrium search to reject many candidates before certifying the optimal PNE. (We use $O(\cdot)$ for upper bounds and $\Theta(\cdot)$ when the bound is tight up to constant factors.)

Construction. Two players choose $\mathbf{x}, \mathbf{y} \in \{0, 1\}^n$. Each coordinate $j \in \{1, \dots, n\}$ is an independent Prisoner’s Dilemma, where 0 = Cooperate and 1 = Defect, with per-coordinate payoffs shown in Table A6.

Table A6: Per-Coordinate Payoffs for the Prisoner’s Dilemma Component Game in \mathcal{P}_n .

	$y_j = 0$ (C)	$y_j = 1$ (D)
$x_j = 0$ (C)	(3, 3)	(0, 5)
$x_j = 1$ (D)	(5, 0)	(1, 1)

Note. (0 = cooperate, 1 = defect)

Structure. Defection strictly dominates cooperation in each coordinate for both players. Hence the unique PNE is $(\mathbf{1}, \mathbf{1})$ with $\phi(\mathbf{1}, \mathbf{1}) = 2n$, while the social optimum is $(\mathbf{0}, \mathbf{0})$ with $\phi(\mathbf{0}, \mathbf{0}) = 6n$. Moreover, every profile other than $(\mathbf{1}, \mathbf{1})$ has strictly larger welfare, so exactly $4^n - 1$ profiles satisfy $\phi > 2n$.

Complexity gap. ZR must reject all $4^n - 1$ non-PNE profiles with $\phi > 2n$ before it can certify that $(\mathbf{1}, \mathbf{1})$ is the optimal PNE, yielding $\Theta(4^n)$ separation-oracle calls in the worst case. By contrast, BZR finds the PNE in $O(1)$ best-response steps (player 1 best-responds to $\mathbf{1}$, then player 2 best-responds to $\mathbf{1}$). Verification is still exponential and in fact $\Theta(2^n)$ in this construction: because defection is dominant, equilibrium inequalities are essentially non-restrictive and do not cut off high-welfare non-PNE profiles, so BZR must rule out $(\mathbf{1}, \mathbf{y})$ for each $\mathbf{y} \neq \mathbf{1}$. There are $2^n - 1$ such \mathbf{y} , and for each one player 2’s best response to $\mathbf{1}$ is $\mathbf{1} \neq \mathbf{y}$, hence $(\mathbf{1}, \mathbf{y})$ is not a PNE. Thus verification requires $\Theta(2^n)$ checks and the overall separation is $\Theta(2^n)$.

Table A7: Worst-Case Complexity Comparison on \mathcal{P}_n : ZR versus BZR.

Algorithm	Finding	Verification	Total
ZR	–	(included)	$\Theta(4^n)$
BZR	$O(1)$	$\Theta(2^n)$	$\Theta(2^n)$
Gap:			$\Theta(2^n)$

Takeaway. This dominant-strategy example shows that certifying the welfare-optimal PNE can be exponentially harder than finding a PNE.

Appendix H: Illustration: A Two-Player Selfish EBMC Game

Figure A2 illustrates the interplay between two selfish players, each controlling three vertices and selecting one under a budget constraint of one. The bold circles and the dashed lines represent the selected vertices and the edges covered by the choices of players, respectively. For example: In Fig. (A2a), edges from B_1 to A_1 are covered at vertex A_1 , while those from B_1 to B_2 are not covered. Assume that the vertices differ in status between two task types. Following standard BRD, Player A first chooses vertex A_1 . Player B benefits from this choice because the edge (A_1, B_1) is now covered. Subsequently, Player B chooses vertex B_1 , covering additional arcs. This in turn activates the edge (B_1, A_1) , allowing Player A to consider selecting vertex A_3 as a new best response. Player B maintains the same choice. At this point, the game reaches a PNE, as neither player has an incentive to deviate from their current strategy. This example illustrates how one player's best response may incidentally benefit the other, even in a selfish setting. However, these benefits are not intentional but rather emergent consequences of self-interested decisions. Importantly, mutual gains observed during the BRD do not guarantee monotonic increases in utility for all players. For example, Player B benefits from Player A's initial choice of A_1 , but the subsequent choice of A_3 does not influence Player B's utility, so that it continues to prefer vertex B_1 . From Player B's perspective, Player A's earlier strategy (choosing A_1 alone) yields a higher selfish objective value than the final strategy (choosing A_3).

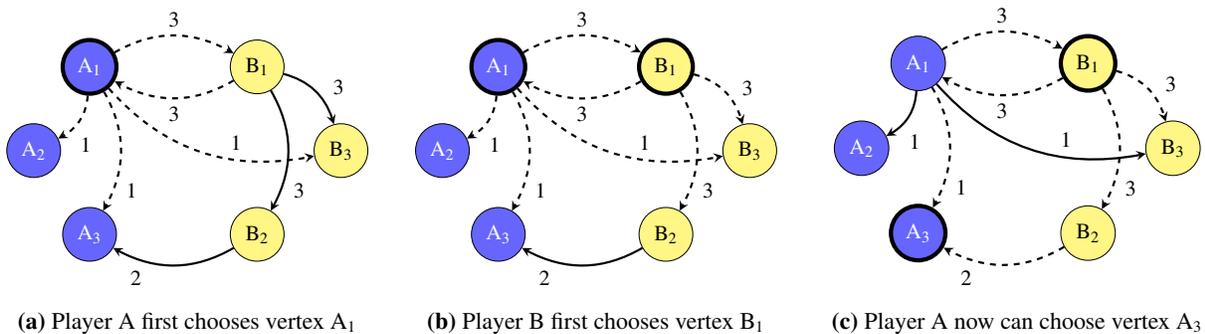


Figure A2: The Interactions in the Selfish Game with Two Players.

Appendix I: EBMC as an IPG: Reformulation and Complexity Implications

The objective functions of the locally altruistic and selfish EBMC games can be reformulated in the standard (quadratic) IPG form by eliminating the edge variables \mathbf{y} and expressing utilities purely in terms of vertex variables \mathbf{x} . Throughout, vertex decisions are binary ($x_j \in \{0, 1\}$), and edge coverage is defined by $y_{jk} = \max\{x_j, x_k\}$. Hence, for every arc (j, k) we have the equation $y_{jk} = x_j + x_k - x_j x_k$. Using this equation, for each player $i \in N$ we can write the EBMC utility as

$$u_i(\mathbf{x}^i, \mathbf{x}^{-i}) = (\mathbf{x}^i)^\top \mathbf{Q}^i \mathbf{x}^i + \sum_{p \in N \setminus \{i\}} (\mathbf{x}^p)^\top \mathbf{Q}_p^i \mathbf{x}^i + (\mathbf{d}^i)^\top \mathbf{x}^i + c_i(\mathbf{x}^{-i}),$$

where \mathbf{Q}^i captures within-player quadratic terms, \mathbf{Q}_p^i captures cross-player interactions (rows indexed by variables in \mathbf{x}^p and columns by variables in \mathbf{x}^i), and \mathbf{d}^i captures linear terms in \mathbf{x}^i . The last summation collects all terms that depend only on opponents' decisions \mathbf{x}^{-i} . Since $c_i(\mathbf{x}^{-i})$ does not depend on \mathbf{x}^i , it is an additive constant from player i 's perspective and therefore does not affect best responses or equilibria.

Coefficient construction. The reformulation follows by expressing the three core utility components $\sum_{(j,k) \in \mathcal{A}_i} w_{jk} y_{jk}$, $\sum_{(j,k) \in \delta_+(\mathcal{I}_i)} w_{jk} y_{jk}$, $\sum_{(j,k) \in \delta_-(\mathcal{I}_i)} w_{jk} y_{jk}$ in terms of \mathbf{x} via $y_{jk} = x_j + x_k - x_j x_k$. Below we describe how each arc set contributes to the coefficients in $u_i(\mathbf{x}^i, \mathbf{x}^{-i})$. (As usual, if one stores \mathbf{Q}^i as a symmetric matrix, then a term $-w_{jk} x_j x_k$ may be split across (j, k) and (k, j) to avoid double counting.)

- **Internal arcs.** For an internal arc $(j, k) \in \mathcal{A}_i$ (so $j, k \in \mathcal{I}_i$), the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the j -th and k -th entries of \mathbf{d}^i , and adds $-w_{jk}$ to the (j, k) entry of \mathbf{Q}^i (representing the bilinear term $-w_{jk} x_j x_k$).
- **Outgoing arcs.** For an outgoing arc $(j, k) \in \delta_+(\mathcal{I}_i)$ with $j \in \mathcal{I}_i$ and $k \in \mathcal{I}_p$ for some $p \neq i$, the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the j -th entry of \mathbf{d}^i (the x_j term), adds the bilinear coefficient $-w_{jk}$ to the (k, j) entry of \mathbf{Q}_p^i (row indexed by x_k , column indexed by x_j), and adds the opponent-only linear term $w_{jk} x_k$ into $c_i(\mathbf{x}^{-i})$.
- **Incoming arcs.** For an incoming arc $(j, k) \in \delta_-(\mathcal{I}_i)$ with $j \in \mathcal{I}_p$ for some $p \neq i$ and $k \in \mathcal{I}_i$, the contribution $w_{jk}(x_j + x_k - x_j x_k)$ adds w_{jk} to the k -th entry of \mathbf{d}^i (the x_k term), adds the bilinear coefficient $-w_{jk}$ to the (j, k) entry of \mathbf{Q}_p^i (row indexed by x_j , column indexed by x_k), and adds the opponent-only linear term $w_{jk} x_j$ into $c_i(\mathbf{x}^{-i})$.

Applying these transformations to the appropriate arc sets yields an explicit representation of EBMC games in the standard (quadratic) IPG format. In particular, the interaction structure of EBMC, captured by the family $\{\mathbf{Q}_p^i\}_{p \neq i}$, mirrors that of quadratic IPGs (qIPGs).

Finally, although $u_i(\mathbf{x}^i, \mathbf{x}^{-i})$ includes the term $c_i(\mathbf{x}^{-i})$, this term depends only on \mathbf{x}^{-i} and therefore does not affect player i 's best response; it can be safely ignored when computing best responses or separating equilibrium inequalities.

Complexity considerations. Both the selfish and locally altruistic EBMC games are instances of quadratic integer programming games (qIPGs): each player solves a binary optimization problem over an individual feasible set \mathcal{X}_i , and utilities admit the qIPG representation. Moreover, any term depending only on opponents' decisions can be absorbed into $c_i(\mathbf{x}^{-i})$ and therefore does not affect best responses or equilibria. Carvalho et al. [2018] show that deciding whether a general IPG admits a pure Nash equilibrium (PNE) is Σ_2^P -complete; consequently, the PNE-existence problem restricted to EBMC games lies in Σ_2^P . In the locally altruistic EBMC game, we establish that the game is an exact potential game and hence a PNE always exists (each \mathcal{X}_i is finite and nonempty). For the selfish EBMC game, PNE existence is not guaranteed; while the decision problem lies in Σ_2^P , establishing Σ_2^P -hardness (and thus completeness) for the EBMC subclass would require a dedicated reduction, which we leave for future work.

Appendix J: Counterexamples for the Existence of PNE

This section presents counterexamples for the existence of a PNE in the EBMC games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$ and $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$. For $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, we identify a counterexample using modest integer weights between 1 and 10 with a total of 9 vertices and two players with budgets 2 and 1. Table A8 shows that there is a unilateral selfish deviation for any strategy profile. We demonstrate the existence of a selfish deviation for any strategy profile rather than showing cycles, as the latter case does not conclusively indicate the absence of a PNE.

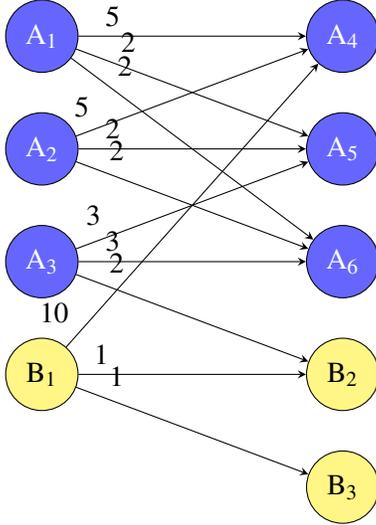


Figure A3: No PNE for $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

Table A8: Strategy Profiles and Selfish Deviations.

Strategy profiles	Selfish unilateral deviation
$x_{A_3} = 1, \mathbf{x}^B \neq (1, 0, 0)$	$\hat{\mathbf{x}}^B = (1, 0, 0)$
$x_{A_3} \neq 1, \mathbf{x}^B \neq (0, 1, 0)$	$\hat{\mathbf{x}}^B = (0, 1, 0)$
$x_{B_1} = 1, \mathbf{x}^A \neq (1, 1, 0, 0, 0, 0)$	$\hat{\mathbf{x}}^A = (1, 1, 0, 0, 0, 0)$
$x_{B_1} \neq 1, \mathbf{x}^A \neq (0, 0, 1, 1, 0, 0)$	$\hat{\mathbf{x}}^A = (0, 0, 1, 1, 0, 0)$

This table displays all potential strategy profiles for players A and B. A has a budget of 2 while B has a budget of 1. Each row describes a set of strategy profiles. The first row contains the strategy profile $\mathbf{x}^A = (0, 0, 1, 1, 0, 0)$ and $\mathbf{x}^B = (0, 1, 0)$. In this case, player B would deviate to $\hat{\mathbf{x}}^B = (1, 0, 0)$. In another scenario, if $\mathbf{x}^A = (1, 0, 1, 0, 0, 0)$ and $\mathbf{x}^B = (0, 1, 0)$, possible deviations are $\hat{\mathbf{x}}^B = (1, 0, 0)$ for player B and $\hat{\mathbf{x}}^A = (0, 0, 1, 1, 0, 0)$ for player A.

Compared to $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$, counterexamples in $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$ are more readily identifiable. An example is presented in Figure A4, where the vertices A_1 and A_2 are of type-1, and the vertices B_1 and B_2 are of type-2. Assuming that each player has only one budget, no PNE exists; for every possible strategy profile, there is a selfish unilateral deviation, as detailed in Table A9.

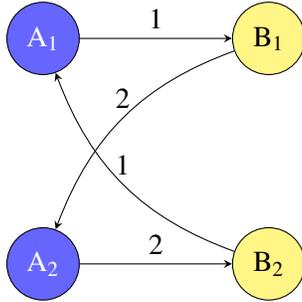


Figure A4: No PNE for $\mathcal{G}_{|\mathcal{L}|\geq 2}^{\text{Self}}$.

Table A9: Strategy Profiles and Selfish Deviations.

Strategy profiles	Selfish unilateral deviation
$x_{A_1} = 1, \mathbf{x}^B \neq (0, 1)$	$\hat{\mathbf{x}}^B = (0, 1)$
$x_{A_1} \neq 1, \mathbf{x}^B \neq (1, 0)$	$\hat{\mathbf{x}}^B = (1, 0)$
$x_{B_1} = 1, \mathbf{x}^A \neq (1, 0)$	$\hat{\mathbf{x}}^A = (1, 0)$
$x_{B_1} \neq 1, \mathbf{x}^A \neq (0, 1)$	$\hat{\mathbf{x}}^A = (0, 1)$

This table displays all potential strategy profiles for players A and B. For instance, if $\mathbf{x}^A = (1, 0)$ and $\mathbf{x}^B = (1, 0)$, player B would deviate to $\hat{\mathbf{x}}^B = (0, 1)$.

Appendix K: RRR-BRD Results

We present the full results of the RRR-BRD algorithm using different initial strategy profiles in various problem settings. Specifically, we report results for selfish EBMC games with $|\mathcal{L}| = 1$ and $|\mathcal{L}| \geq 2$, as well as KPG instances with interaction types (A), (B), and (C). The metrics and computational settings used in these experiments are described in Section 7.1.

Table A10: RRR-BRD Results with Different Initial Strategy Profiles for Selfish EBMC Games $\mathcal{G}_{|\mathcal{L}|=1}^{\text{Self}}$.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$
2	0.3	19.2	19.2	19.2	1.000	19.2	19.2	0.0	0.1	2	1
	0.5	21.6	22.3	22.3	1.000	22.3	22.3	0.1	2.3	2	1
	0.8	22.6	22.6	24.9	1.102	23.3	23.3	0.1	0.4	2	2
3	0.3	22.0	27.6	27.6	1.000	27.6	27.6	0.2	2.4	2	1
	0.5	38.8	42.1	42.1	1.000	42.1	42.1	0.2	6.9	2	1
	0.8	53.1	53.1	53.1	1.000	53.1	53.1	0.1	0.2	2	1
5	0.3	83.5	90.4	90.4	1.000	90.4	90.4	0.4	13.8	2	1
	0.5	85.0	100.6	100.6	1.000	100.6	101.2	0.5	301.0	2	1
	0.8	146.9	156.7	156.7	1.000	156.7	157.4	0.5	301.1	2	1
8	0.3	130.1	130.1	149.9	1.152	134.1	134.2	1.6	5.3	2	3
	0.5	169.1	252.2	252.2	1.000	252.2	260.2	1.6	302.8	2	1
	0.8	269.9	341.4	381.3	1.117	349.2	353.4	1.4	232.6	2	3
10	0.3	308.1	317.2	317.2	1.000	317.2	317.3	2.6	304.5	2	1
	0.5	289.8	361.7	361.7	1.000	361.7	372.4	1.6	302.8	2	1
	0.8	412.2	539.8	539.8	1.000	538.1	549.3	2.0	303.8	2	1
15	0.3	630.4	725.4	725.4	1.000	725.4	745.9	4.7	307.8	2	1
	0.5	576.9	576.9	857.5	1.487	708.4	730.1	6.1	23.7	2	3
	0.8	886.3	1212.2	1219.7	1.006	1216.5	1249.2	5.7	311.3	2	3
20	0.3	656.1	904.5	904.5	1.000	904.5	922.5	10.4	319.2	2	1
	0.5	1089.0	1398.8	1529.1	1.093	1282.1	1472.8	11.1	319.8	2	1
	0.8	1317.6	1701.4	2415.5	1.420	2024.4	2207.2	13.1	323.9	2	3
25	0.3	1419.5	1765.0	1765.0	1.000	1765.0	1819.5	18.0	328.9	2	1
	0.5	2140.7	2597.5	2597.7	1.000	2597.4	2767.9	19.0	334.7	2	2
	0.8	2246.8	3294.0	3312.8	1.006	3307.4	3541.1	18.9	336.4	2	3
30	0.3	1780.2	2448.2	2448.3	1.000	2448.1	2556.0	26.1	345.6	2	2
	0.5	2704.9	3263.8	3263.8	1.000	3260.8	3482.4	29.8	349.1	2	1
	0.8	3724.0	4919.7	4919.7	1.000	4910.4	5399.1	28.1	346.4	2	1

Table A11: RRR-BRD Results with Different Initial Strategy Profiles for Selfish EBMC Games $\mathcal{G}_{|L|\geq 2}^{\text{Self}}$.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$
2	0.3	51.4	51.4	56.3	1.096	51.4	51.4	0.4	1.6	3	3
	0.5	83.8	83.8	86.6	1.033	83.8	86.0	0.8	58.8	3	2
	0.8	92.1	91.4	95.2	1.034	92.8	94.3	1.0	5.6	3	2
3	0.3	115.0	115.0	139.7	1.215	115.0	115.0	0.7	12.1	4	4
	0.5	178.6	178.3	200.3	1.122	178.6	196.7	0.7	301.5	4	4
	0.8	252.2	252.2	253.7	1.006	252.2	257.8	0.8	301.8	4	6
5	0.3	380.0	380.0	399.8	1.052	380.0	380.0	1.7	303.5	3	3
	0.5	557.6	557.6	577.3	1.035	557.6	618.5	1.7	303.1	3	3
	0.8	650.1	650.1	661.9	1.018	650.1	677.5	2.1	303.9	4	4
8	0.3	862.1	862.1	995.8	1.155	862.1	980.4	5.2	309.4	4	4
	0.5	1214.5	1214.5	1387.9	1.143	1214.5	1535.1	5.6	309.2	4	4
	0.8	1661.5	1661.5	1721.0	1.036	1661.5	1771.3	6.1	309.3	4	6
10	0.3	1265.2	1265.2	1521.6	1.203	1265.2	1526.5	8.0	319.6	3	13
	0.5	2084.2	2084.2	2253.9	1.081	2083.2	2635.7	8.7	315.5	4	5
	0.8	2612.3	2612.3	2685.2	1.028	2612.3	2757.3	9.4	318.6	5	9
15	0.3	2916.4	2916.4	3440.2	1.180	2916.4	3523.7	19.5	329.0	4	4
	0.5	4476.8	4476.8	5030.1	1.124	4476.8	5750.7	29.3	337.9	6	5
	0.8	-	-	6206.3	-	-	6428.7	402.9	687.6	200	200
20	0.3	5373.6	5373.6	6284.2	1.169	5373.6	6483.1	44.9	366.4	5	4
	0.5	-	-	9019.4	-	-	11452.5	874.9	1195.1	200	200
	0.8	10149.3	10149.7	10510.8	1.036	10149.7	10858.2	45.7	377.6	4	6
25	0.3	8023.9	8023.9	9502.9	1.184	8023.9	17369.6	51.8	394.3	3	4
	0.5	12640.7	12640.7	13936.0	1.102	12640.7	17424.1	77.7	417.1	4	4
	0.8	-	-	16806.1	-	-	17521.5	1789.4	1949.1	200	200
30	0.3	11151.4	11151.4	13520.2	1.212	11151.4	24217.3	119.6	460.2	5	4
	0.5	17342.4	17342.4	16969.0	0.978	17342.4	25103.4	146.5	503.8	6	6
	0.8	-	-	24180.9	-	-	25028.6	2910.7	3240.5	200	200

Table A12: RRR-BRD Results with Different Initial Strategy Profiles for Type A KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	$\widetilde{\text{OSW}}$	$\widetilde{\text{POS}}$	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$	init: $\mathbf{0}^N$	init: $\bar{\mathbf{x}}_{\text{sw}}^{300}$
2	0.2	18.4	18.5	18.7	1.011	18.5	18.5	0.1	0.6	3	2
	0.5	17.6	18.3	18.3	1.002	18.3	18.3	0.0	0.4	3	2
	0.8	8.1	8.2	8.4	1.022	8.3	8.3	0.0	0.4	3	2
3	0.2	37.2	37.5	37.8	1.010	37.6	37.6	0.3	3.1	4	3
	0.5	31.8	32.4	32.8	1.011	32.7	32.7	0.1	2.0	3	3
	0.8	14.5	14.8	15.2	1.027	14.8	14.8	0.0	0.4	4	2
5	0.2	129.2	131.3	131.3	1.000	131.3	131.3	0.9	47.0	4	1
	0.5	59.6	60.0	61.4	1.024	60.6	61.0	0.8	33.5	7	3
	0.8	27.0	27.2	29.3	1.079	27.3	27.3	0.1	2.0	4	3
8	0.2	289.8	294.8	295.1	1.001	295.1	295.4	4.0	301.1	8	2
	0.5	110.5	112.1	116.7	1.041	112.4	114.0	0.9	235.1	6	5
	0.8	85.6	86.6	87.8	1.013	86.9	87.6	0.3	25.8	7	4
10	0.2	500.5	504.6	508.7	1.008	505.5	506.1	6.1	302.5	8	3
	0.5	255.0	260.4	262.8	1.009	260.9	263.0	1.0	302.3	7	4
	0.8	139.5	141.6	143.9	1.016	141.2	142.9	0.5	301.3	7	5
15	0.2	1050.2	1057.4	1058.8	1.001	1058.1	1060.8	6.5	302.8	8	3
	0.5	583.1	599.7	604.2	1.007	600.4	605.2	1.9	304.1	9	6
	0.8	246.4	254.2	257.7	1.014	255.3	258.3	0.8	302.4	8	6
20	0.2	1663.5	1688.0	1689.9	1.001	1689.5	1693.9	8.6	303.2	10	4
	0.5	1103.4	1120.6	1126.2	1.005	1120.4	1129.0	2.2	302.8	13	7
	0.8	598.6	610.2	612.8	1.004	610.8	615.2	1.2	302.6	15	4
25	0.2	2597.6	2626.4	2628.1	1.001	2627.4	2637.1	8.3	306.1	7	3
	0.5	1789.9	1825.8	1833.3	1.004	1827.5	1838.4	3.7	307.2	17	8
	0.8	690.1	706.2	711.5	1.008	706.6	714.5	1.4	303.8	11	5
30	0.2	3311.2	3334.8	3351.2	1.005	3331.6	3356.8	15.1	307.4	10	4
	0.5	2490.0	2528.8	2535.3	1.003	2528.7	2544.7	3.3	305.2	14	6
	0.8	1070.2	1095.9	1104.6	1.008	1095.7	1109.7	1.4	304.4	8	6

Table A13: RRR-BRD Results with Different Initial Strategy Profiles for Type B KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}	\widetilde{OSW}	\widetilde{POS}	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}	init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}
2	0.2	8.0	8.2	8.3	1.018	8.2	8.2	0.0	0.1	3	2
	0.5	14.1	14.2	14.4	1.016	14.2	14.2	0.2	0.3	3	3
	0.8	18.8	19.0	19.1	1.005	19.1	19.1	0.2	0.3	3	3
3	0.2	14.8	15.4	15.9	1.035	15.6	15.6	0.0	0.3	5	3
	0.5	28.3	28.7	29.4	1.025	28.9	29.1	0.1	2.8	5	4
	0.8	39.0	39.8	40.3	1.012	40.0	40.1	0.9	2.6	4	2
5	0.2	39.6	39.6	40.9	1.033	40.4	40.8	0.2	5.8	6	5
	0.5	76.7	77.9	78.8	1.012	78.1	78.8	0.6	19.5	9	5
	0.8	106.9	107.7	108.7	1.009	108.3	108.6	4.0	24.9	10	3
8	0.2	87.7	91.1	92.6	1.017	91.6	92.7	0.4	253.4	5	4
	0.5	185.2	187.5	189.9	1.013	188.6	189.9	1.8	184.4	13	7
	0.8	267.6	272.3	273.1	1.003	272.8	273.2	4.5	302.5	6	4
10	0.2	132.5	135.4	136.9	1.011	136.0	137.0	0.1	300.6	10	3
	0.5	281.0	286.8	289.0	1.008	287.6	289.4	0.7	302.0	7	4
	0.8	423.2	426.1	427.8	1.004	426.9	428.3	5.6	302.9	9	4
15	0.2	279.2	285.8	289.5	1.013	287.0	290.5	0.6	302.5	7	7
	0.5	627.1	641.3	644.0	1.004	642.7	645.4	1.1	302.1	9	6
	0.8	954.7	963.8	965.2	1.001	963.8	966.5	6.6	303.1	9	3
20	0.2	479.9	486.6	491.2	1.009	488.7	493.1	0.7	301.9	10	4
	0.5	1108.2	1121.6	1125.5	1.004	1123.8	1129.0	1.0	302.0	9	4
	0.8	1661.3	1671.9	1672.9	1.001	1670.7	1677.3	8.4	302.7	11	2
25	0.2	753.5	759.8	763.5	1.005	761.4	767.5	0.9	303.4	9	4
	0.5	1696.0	1717.7	1723.7	1.004	1719.9	1728.6	4.0	304.2	20	8
	0.8	2586.8	2608.5	2614.3	1.002	2610.0	2622.9	12.1	306.5	13	5
30	0.2	1055.2	1085.3	1088.8	1.003	1085.2	1094.7	2.3	305.8	12	14
	0.5	2409.3	2460.4	2465.8	1.002	2460.3	2473.3	3.8	305.1	13	5
	0.8	3729.8	3745.0	3750.8	1.002	3749.0	3764.0	17.0	310.2	11	6

Table A14: RRR-BRD Results with Different Initial Strategy Profiles for Type C KPG.

n	BG	RRR-BRD (ϕ)		Price of Stability		References from BZR		Time (secs)		Round	
		init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}	\widetilde{OSW}	\widetilde{POS}	BZR(ϕ)	BZR(Bound)	init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}	init: $\mathbf{0}^N$	init: \bar{x}_{sw}^{300}
2	0.2	4.7	4.7	4.7	1.007	4.7	4.7	0.0	0.1	3	2
	0.5	7.8	7.8	7.8	1.007	7.8	7.8	0.0	0.1	3	2
	0.8	8.4	8.4	8.4	1.008	8.4	8.4	0.1	0.1	3	2
3	0.2	7.1	7.1	7.2	1.012	7.1	7.1	0.0	0.3	4	3
	0.5	10.3	10.3	10.4	1.011	10.3	10.3	0.1	0.3	4	2
	0.8	11.5	11.6	11.8	1.024	11.6	11.6	0.1	0.1	5	3
5	0.2	10.1	10.1	10.4	1.025	10.3	10.3	0.1	0.7	3	3
	0.5	14.2	14.1	14.6	1.028	14.3	14.4	0.5	1.6	4	3
	0.8	13.6	13.7	15.0	1.095	13.7	13.7	0.3	0.4	4	3
8	0.2	15.6	15.6	16.2	1.034	15.8	16.0	0.1	11.4	4	3
	0.5	15.6	15.6	17.6	1.122	15.8	16.5	0.5	1.9	6	4
	0.8	15.6	15.5	18.5	1.192	15.7	16.6	0.2	1.3	5	4
10	0.2	16.3	16.4	17.5	1.070	16.7	17.3	0.1	13.1	6	10
	0.5	16.7	16.7	20.3	1.212	16.9	18.3	6.2	16.4	85	126
	0.8	16.1	16.1	20.3	1.262	16.3	18.0	0.3	4.5	6	5
15	0.2	19.6	19.6	21.8	1.114	19.8	21.5	0.5	224.2	8	17
	0.5	15.9	16.1	23.0	1.435	16.5	19.8	0.4	72.6	10	7
	0.8	16.3	16.4	23.0	1.398	16.8	19.8	0.3	42.8	9	8
20	0.2	20.5	20.7	24.5	1.185	20.9	24.0	4.3	304.7	77	54
	0.5	-	-	24.7	-	-	22.0	6.4	308.5	200	200
	0.8	16.8	16.8	25.5	1.519	17.1	23.1	5.0	303.7	131	50
25	0.2	20.6	20.4	26.2	1.272	20.7	26.4	15.1	303.9	196	20
	0.5	-	-	25.7	-	-	24.9	10.2	313.0	200	200
	0.8	-	-	26.4	-	-	25.4	11.5	315.3	200	200
30	0.2	20.6	-	27.7	1.350	20.5	40.8	17.7	320.2	196	200
	0.5	15.9	15.7	27.3	1.720	16.3	33.0	4.6	305.5	74	33
	0.8	15.6	-	27.6	1.772	15.8	30.4	2.9	323.8	78	200

Appendix L: BZR Full Results

We present the full BZR results of the EBMC $|\mathcal{L}| \geq 2$ games, KPG types (A) and (C). The metrics and computational settings used in these experiments are described in Section 7.1.

Table A15: Comparison of BZR and ZR for Selfish EBMC Games $\mathcal{G}_{|\mathcal{L}| \geq 2}^{\text{Self}}$.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pne(ZR)}	# _{pne(BZR)}	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts(ZR)}	# _{cuts(BZR)}
2	0.3	1	1	51.4	51.4	51.4	51.4	56.3	1.096	9.7	9.8	0.8	12.8	12	19
	0.5	0	1	-	83.8	85.7	86.0	86.6	1.033	-	1800.3	1.4	1800.1	11	20
	0.8	0	6	-	92.8	94.1	94.3	95.2	1.027	-	1800.1	2.1	1800.1	14	26
3	0.3	1	1	115.0	115.0	115.0	115.0	139.7	1.215	1423.4	1423.7	1.9	1040.1	18	29
	0.5	0	2	-	178.6	197.9	196.7	200.3	1.122	-	1800.4	2.3	1800.5	15	29
	0.8	0	1	-	252.2	257.7	257.8	253.7	1.006	-	1800.3	1.9	1800.4	15	32
5	0.3	1	1	380.0	380.0	380.0	380.0	399.8	1.052	489.2	665.9	5.7	642.9	28	49
	0.5	0	1	-	557.6	626.7	618.5	577.3	1.035	-	1800.3	5.6	1800.0	25	43
	0.8	0	1	-	650.1	671.1	677.5	661.9	1.018	-	1800.7	5.4	1800.5	25	57
8	0.3	0	1	-	862.1	975.9	980.4	995.8	1.155	-	1800.1	14.8	1800.1	40	70
	0.5	0	1	-	1214.5	1527.4	1535.1	1387.9	1.143	-	1800.1	14.4	1800.1	40	75
	0.8	0	1	-	1661.5	1770.3	1771.3	1721.0	1.036	-	1800.1	14.8	1800.2	40	78
10	0.3	0	1	-	1265.2	1517.8	1526.5	1521.6	1.203	-	1800.1	24.9	1800.1	50	95
	0.5	0	1	-	2083.2	2635.2	2635.7	2253.9	1.082	-	1800.1	23.2	1800.1	50	93
	0.8	0	1	-	2612.3	2755.8	2757.3	2685.2	1.028	-	1800.3	25.7	1800.3	50	100
15	0.3	0	1	-	2916.4	3501.5	3523.7	3440.2	1.180	-	1800.3	56.7	1800.1	75	131
	0.5	0	1	-	4476.8	5728.7	5750.7	5030.1	1.124	-	1800.3	58.6	1800.2	75	137
	0.8	0	0	-	-	6423.5	6428.7	6206.3	-	-	1800.3	-	1800.3	75	150
20	0.3	0	1	-	5373.6	6483.6	6483.1	6284.2	1.169	-	1800.4	118.5	1800.3	100	178
	0.5	0	0	-	-	10639.6	11452.5	9019.4	-	-	1800.4	-	1800.4	100	200
	0.8	0	1	-	10149.7	10856.7	10858.2	10510.8	1.036	-	1800.4	117.4	1800.2	99	190
25	0.3	0	1	-	8023.9	9787.5	17369.6	9502.9	1.184	-	1800.4	166.9	1800.8	125	210
	0.5	0	1	-	12640.7	17424.1	17424.1	13936.0	1.102	-	1800.6	176.5	1800.5	125	236
	0.8	0	0	-	-	17512.8	17521.5	16806.1	-	-	1800.4	-	1800.5	125	175
30	0.3	0	1	-	11151.4	13751.2	24217.3	13520.2	1.212	-	1800.6	284.1	1801.1	150	267
	0.5	0	1	-	17342.4	25103.4	25103.4	16969.0	0.978	-	1800.8	288.5	1800.9	150	272
	0.8	0	0	-	-	25019.9	25028.6	24180.9	-	-	1800.6	-	1800.7	150	180

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Table A16: Comparison of BZR and ZR for Type A KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pnc} (ZR)	# _{pnc} (BZR)	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts} (ZR)	# _{cuts} (BZR)
2	0.2	1	3	18.5	18.5	18.5	18.5	18.7	1.011	1.1	1.7	0.1	2.5	2031	2040
	0.5	1	9	18.3	18.3	18.3	18.3	18.3	1.002	1.1	1.5	0.1	1.2	155	116
	0.8	1	10	8.3	8.3	8.3	8.3	8.4	1.007	2.7	3.1	0.1	2.3	671	657
3	0.2	5	9	37.6	37.6	37.6	37.6	37.8	1.006	12.0	14.8	1.1	9.6	1058	1009
	0.5	2	131	32.7	32.7	32.7	32.7	32.8	1.003	958.7	1646.6	0.2	174.2	3394	805
	0.8	0	61	-	14.8	14.9	14.8	15.2	1.025	-	1800.2	0.2	339.1	2739	1167
5	0.2	3	8	131.3	131.3	131.3	131.3	131.3	1.000	7.7	81.5	1.8	47.9	58	78
	0.5	0	93	-	60.6	61.0	61.0	61.4	1.014	-	1800.3	0.9	1800.1	2892	1730
	0.8	1	18	27.3	27.3	27.4	27.3	29.3	1.072	1784.9	1800.1	0.5	183.8	3982	3857
8	0.2	0	17	-	295.1	295.6	295.4	295.1	1.000	-	1800.7	5.0	1800.3	1657	200
	0.5	0	35	-	112.4	114.2	114.0	116.7	1.038	-	1800.4	1.5	1800.3	4833	4680
	0.8	0	49	-	86.9	87.7	87.6	87.8	1.010	-	1800.5	0.6	1800.9	571	649
10	0.2	0	8	-	505.5	506.5	506.1	508.7	1.006	-	1800.3	6.2	1800.6	2604	2157
	0.5	0	36	-	260.9	263.1	263.0	262.8	1.007	-	1800.9	2.6	1801.0	583	608
	0.8	0	54	-	141.2	143.0	142.9	143.9	1.019	-	1800.8	1.1	1800.6	1584	1653
15	0.2	0	13	-	1058.1	1061.3	1060.8	1058.8	1.001	-	1800.5	12.9	1801.5	344	254
	0.5	0	31	-	600.4	605.3	605.2	604.2	1.006	-	1800.8	11.3	1801.5	461	755
	0.8	0	35	-	255.3	258.3	258.3	257.7	1.009	-	1800.7	9.1	1801.6	325	822
20	0.2	0	13	-	1689.5	1694.4	1693.9	1689.9	1.000	-	1801.6	12.9	1800.6	279	372
	0.5	0	15	-	1120.4	1129.0	1129.0	1126.2	1.005	-	1800.7	3.8	1800.8	215	514
	0.8	0	14	-	610.8	615.5	615.2	612.8	1.003	-	1800.6	3.0	1800.8	348	395
25	0.2	0	8	-	2627.4	2637.5	2637.1	2628.1	1.000	-	1800.9	17.3	1801.1	309	355
	0.5	0	10	-	1827.5	1838.6	1838.4	1833.3	1.003	-	1800.7	5.6	1801.0	290	447
	0.8	0	12	-	706.6	714.5	714.5	711.5	1.007	-	1801.4	3.9	1800.9	332	547
30	0.2	0	7	-	3331.6	3356.9	3356.8	3351.2	1.006	-	1800.7	26.3	1802.9	197	397
	0.5	0	9	-	2528.7	2545.1	2544.7	2535.3	1.003	-	1800.8	8.4	1801.3	357	475
	0.8	0	11	-	1095.7	1109.9	1109.7	1104.6	1.008	-	1800.9	4.7	1801.0	312	586

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Table A17: Comparison of BZR and ZR for Type C KPG.

n	BG	# PNEs Found		Best PNE (ϕ)		Dual Bounds		Price of Stability		Time (secs)				# EI Cuts	
		# _{pnc} (ZR)	# _{pnc} (BZR)	ZR	BZR	ZR	BZR	$\overline{\text{OSW}}$	$\overline{\text{POS}}$	T(ZR ^{1st})	T(ZR)	T(BZR ^{1st})	T(BZR)	# _{cuts} (ZR)	# _{cuts} (BZR)
2	0.2	1	1	4.7	4.7	4.7	4.7	4.7	1.007	0.4	0.6	0.1	0.5	4075	4085
	0.5	3	14	7.8	7.8	7.8	7.8	7.8	1.007	2.3	3.3	0.1	3.7	4159	4196
	0.8	1	2	8.4	8.4	8.4	8.4	8.4	1.008	0.6	0.8	0.1	0.9	3567	3583
3	0.2	1	3	7.1	7.1	7.1	7.1	7.2	1.012	3.6	3.9	0.1	1.8	4385	4407
	0.5	2	5	10.3	10.3	10.3	10.3	10.4	1.011	39.1	51.2	0.3	12.3	4442	4451
	0.8	2	4	11.6	11.6	11.6	11.6	11.8	1.024	1.3	1.4	0.1	1.6	5629	5647
5	0.2	0	14	-	10.3	10.3	10.3	10.4	1.012	-	1800.4	0.3	243.4	4198	4274
	0.5	0	20	-	14.3	14.5	14.4	14.6	1.023	-	1800.6	0.5	1800.5	4986	5144
	0.8	2	33	13.7	13.7	13.7	13.7	15.0	1.092	235.0	446.0	0.8	177.6	4119	4227
8	0.2	0	19	-	15.8	16.1	16.0	16.2	1.024	-	1800.8	0.4	1801.0	2713	2856
	0.5	0	34	-	15.8	16.8	16.5	17.6	1.108	-	1800.9	1.0	1800.5	2267	2612
	0.8	0	146	-	15.7	16.8	16.6	18.5	1.183	-	1800.5	0.7	1800.5	3033	4072
10	0.2	0	19	-	16.7	17.4	17.3	17.5	1.051	-	1800.5	0.7	1800.4	1559	1749
	0.5	0	49	-	16.9	18.7	18.3	20.3	1.196	-	1800.7	6.0	1800.6	1564	2377
	0.8	0	156	-	16.3	18.6	18.0	20.3	1.244	-	1800.5	1.1	1800.8	1775	3841
15	0.2	0	18	-	19.8	21.7	21.5	21.8	1.104	-	1800.5	22.3	1800.4	1608	1831
	0.5	0	39	-	16.5	20.3	19.8	23.0	1.397	-	1800.3	21.7	1800.3	1651	2302
	0.8	0	37	-	16.8	20.2	19.8	23.0	1.366	-	1800.3	21.4	1800.4	1633	2362
20	0.2	0	11	-	20.9	24.1	24.0	24.5	1.170	-	1800.6	4.4	1800.4	120	340
	0.5	0	0	-	-	21.8	22.0	24.7	-	-	1800.2	-	1800.2	160	240
	0.8	0	13	-	17.1	22.7	23.1	25.5	1.492	-	1800.2	3.7	1800.3	200	352
25	0.2	0	7	-	20.7	25.9	26.4	26.2	1.267	-	1800.3	10.1	1800.4	150	360
	0.5	0	0	-	-	24.7	24.9	25.7	-	-	1800.3	-	1800.6	175	250
	0.8	0	0	-	-	25.5	25.4	26.4	-	-	1800.3	-	1800.5	175	250
30	0.2	0	2	-	20.5	29.3	40.8	27.7	1.355	-	1800.5	14.5	1800.8	150	526
	0.5	0	11	-	16.3	27.5	33.0	27.3	1.675	-	1800.5	8.7	1800.9	180	719
	0.8	0	18	-	15.8	27.9	30.4	27.6	1.747	-	1800.4	5.6	1801.5	210	891

Note. Boldfaced values indicate better solutions found within the time limit, tighter dual bounds, or more PNEs.

Appendix M: Approximate-PNE

This appendix reports *approximate* equilibrium certificates for the instances in which our algorithms did not report an *exact* pure Nash equilibrium (PNE) within the prescribed computational limits. The goal is not to claim that a PNE exists in these instances, but rather to quantify the stability of the best-observed strategy profiles against unilateral deviations.

For each “no-PNE-reported” instance (both EBMC games and KPGs), we run RRR–BRD with a single restart and a fixed round cap: $L = 1$ and $R = 20$. Let $\mathbf{x}^{(r)}$ denote the strategy profile at the *end* of round $r \in \{1, \dots, 20\}$ (i.e., after all players have played once in that round under a randomized play order). We store these end-of-round profiles, yielding at most 20 candidate profiles per instance.

For any candidate profile $\mathbf{x}^{(r)}$, we compute a multiplicative approximation factor as follows. For each player $i \in N$, we solve the best-response problem against \mathbf{x}^{-i} and obtain a best response $\hat{\mathbf{x}}^i \in \arg \max_{\mathbf{x}^i \in \chi_i} u_i(\mathbf{x}^i, \mathbf{x}^{-i})$. Assuming $u_i(\mathbf{x}^i, \mathbf{x}^{-i}) > 0$ for all feasible profiles, we define

$$\alpha_i(\mathbf{x}^{(r)}) := \frac{u_i(\hat{\mathbf{x}}^i, (\mathbf{x}^{(r)})^{-i})}{u_i(\mathbf{x}^{(r),i}, (\mathbf{x}^{(r)})^{-i})}, \quad \alpha(\mathbf{x}^{(r)}) := \max_{i \in N} \alpha_i(\mathbf{x}^{(r)}).$$

Then $\mathbf{x}^{(r)}$ is an $\alpha(\mathbf{x}^{(r)})$ -approximate PNE in the multiplicative sense, and $\alpha(\mathbf{x}^{(r)}) = 1$ if and only if $\mathbf{x}^{(r)}$ is an exact PNE. Among the 20 end-of-round candidates, we identify the profile with the smallest approximation factor,

$$\alpha^* := \min_{r=1, \dots, 20} \alpha(\mathbf{x}^{(r)}),$$

and report: (i) the best round r^* attaining α^* , (ii) the time taken for this RRR–BRD run, and (iii) the social welfare (global objective value) evaluated at $\mathbf{x}^{(r^*)}$.

Tables A18 and A19 summarize these results for EBMC games and KPGs, respectively.

Table A18: Approximate PNE for the instances without a PNE in EBMC Games

Type	n	BG	Approx BRD (ϕ)	Best α	Time (secs)	Best Round
$ \mathcal{L} \geq 2$	15	0.8	6043.8	1.000003	106.3	11
$ \mathcal{L} \geq 2$	20	0.5	8292.9	1.000003	210.4	3
$ \mathcal{L} \geq 2$	25	0.8	16255.9	1.000002	393.8	19
$ \mathcal{L} \geq 2$	30	0.8	23509.4	1.000004	639.2	8

Table A19: Approximate PNE for the instances without a PNE in KPGs

Type	n	BG	Approx BRD (ϕ)	Best α	Time (secs)	Best Round
C	20	0.5	15.2	1.003140	0.9	9
C	25	0.5	14.5	1.003591	1.1	19
C	25	0.8	15.9	1.005714	1.2	20

Appendix N: County-level Utility Values

Table A20 presents the complete set of county-level utility values under different strategy profiles, as introduced in Table 10 of the main manuscript.

Table A20: County-level Utility Values Using Different Strategy Profiles.

County	$u_i^{\text{Self}}(\bar{x}_{\text{ng}})$	$u_i^{\text{Self}}(\hat{x}_{\text{pne}}^1)$	$u_i^{\text{Self}}(\hat{x}_{\text{pne}}^*)$	$u_i^{\text{Self}}(\bar{x}_{\text{sw}}^{3600})$	County	$u_i^{\text{Self}}(\bar{x}_{\text{ng}})$	$u_i^{\text{Self}}(\hat{x}_{\text{pne}}^1)$	$u_i^{\text{Self}}(\hat{x}_{\text{pne}}^*)$	$u_i^{\text{Self}}(\bar{x}_{\text{sw}}^{3600})$
Aitkin	23746	24956	24982	29013	Martin	24	24	24	24
Anoka	11292	12466	12495	15723	McLeod	7194	7479	7503	8933
Becker	24219	26489	26497	31712	Meeker	8866	9423	9428	9964
Beltrami	18720	20049	20061	23756	Mille lacs	1778	1818	1818	1886
Benton	1571	1688	1688	1779	Morrison	5243	5460	5466	6184
Big Stone	4555	4827	4827	5664	Mower	43	43	43	48
Blue Earth	4712	4924	4932	5439	Murray	3481	3576	3590	3906
Brown	2338	2436	2436	2805	Nicollet	32	32	32	32
Carlton	5393	5808	5813	6910	Nobles	2328	2396	2396	2520
Carver	6664	7306	7331	8211	Norman	8	8	8	8
Cass	40151	42450	42500	48926	Olmsted	587	596	598	852
Chippewa	60	60	60	60	Otter tail	43479	47839	47863	58200
Chisago	5463	5923	5930	6827	Pennington	22	22	22	22
Clay	2075	2481	2485	2893	Pine	6061	6313	6321	7088
Clearwater	8484	9210	9235	10484	Pipestone	38	73	73	78
Cook	28157	29949	30031	34442	Polk	5584	6031	6037	6906
Cottonwood	2713	2824	2824	3212	Pope	10157	11020	11032	13728
Crow Wing	38044	39727	39789	46254	Ramsey	6995	7726	7761	9372
Dakota	6958	7400	7406	8664	Redwood	177	177	177	177
Dodge	7	7	7	7	Renville	837	866	871	1056
Douglas	16827	18127	18136	22158	Rice	4496	4839	4847	5076
Faribault	251	251	251	251	Rock	0	0	0	0
Freeborn	1172	1172	1172	1172	Roseau	234	234	234	234
Goodhue	770	845	845	879	Saint Louis	78691	81719	81770	91172
Grant	6635	7389	7391	8817	Scott	4983	5249	5253	6381
Hennepin	21136	22442	22474	27024	Sherburne	5077	5557	5564	6698
Hubbard	22780	24426	24444	29543	Sibley	2390	2388	2388	2429
Isanti	3635	3876	3884	4227	Stearns	20200	21544	21554	26076
Itasca	51214	53588	53640	61367	Steele	372	372	372	372
Jackson	3061	3167	3171	3474	Stevens	2938	3340	3382	4142
Kanabec	3697	3917	3927	4634	Swift	1287	1461	1461	1778
Kandiyohi	15551	16438	16450	18723	Todd	9108	9832	9842	11661
Kittson	254	254	254	254	Traverse	1427	1466	1466	1523
Koochiching	2217	2278	2284	2314	Wabasha	331	331	331	331
Lac Qui Parle	1331	1376	1376	1405	Wadena	1834	1950	1950	2289
Lake	30194	31584	31640	36702	Waseca	2940	3074	3080	3089
Lake of the woods	1009	1009	1009	1009	Washington	11252	11946	11968	15845
Le Sueur	6804	7135	7157	7929	Watsonwan	946	955	955	1114
Lincoln	4221	4292	4292	4501	Wilkin	94	102	102	128
Lyon	2752	2925	2939	3314	Winona	1561	1579	1579	1444
Mahnomen	4177	4424	4428	6012	Wright	22047	23782	23805	29464
Marshall	4585	5076	5078	6081	Yellow Medicine	829	858	858	938

Note. Counties, where the inequalities regarding utility value comparisons do not hold, are highlighted in bold.