

A New Adaptive Randomized Equilibrium Search for Generalized Nash Equilibrium Problems

Benjamin Benteke Longaou

bbenteke@uoguelph.ca

University of Guelph

Tangi Migot

Monica-Gabriela Cojocaru

University of Guelph

Research Article

Keywords: GNEPs, local search, trust region, best response dynamics

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11175847/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

A New Adaptive Randomized Equilibrium Search for Generalized Nash Equilibrium Problems

Benjamin Benteke Longaou^{1*}, Tangi Migot^{2†} and
Monica-Gabriela Cojocaru^{1†}

^{1*}Department of mathematics and Statistics, University of Guelph, Stone
Rd E 50, Guelph, N1G 2W1, Ontario, Canada.

*Corresponding author(s). E-mail(s): bbenteke@uoguelph.ca;
Contributing authors: tangi.migot@gmail.com; mcojocar@uoguelph.ca;

[†]These authors contributed equally to this work.

Abstract

Generalized Nash equilibrium problems (GNEPs) provide a framework for modeling strategic interactions among multiple decision-makers with coupled constraints. Solving general GNEPs is challenging because of coupled feasibility, possible nonconvexity, equilibrium-existence issues, and potentially empty player feasible sets. This paper proposes a simple and practical local search heuristic for solving GNEPs, termed adaptive randomized equilibrium search (ARES). The proposed method combines best-response information, local action sampling, a Gauss-Southwell-type player-selection rule [Nesterov \(2012\)](#); [Nutini et al. \(2015\)](#), and an adaptive trust-region mechanism. We evaluate ARES on academic jointly-convex and non-shared-constraint GNEPs, as well as on several application-inspired jointly-convex GNEPs. Illustrative examples further show that ARES can avoid cycling or stagnation encountered by best-response dynamics and Gauss-Seidel updates on selected GNEPs. We compare ARES with a deterministic variant based on full best-response directions, using success rate and solution diversity as the main performance measures.

For application-inspired jointly-convex GNEPs, both methods generally identify GNEs, while ARES can discover a more diverse set of equilibria, as illustrated by the autonomous-racing example. For GNEPs with non-shared constraints, randomized partial moves substantially improve robustness on several instances and generally maintain high solution diversity. These results suggest that randomized partial best-response moves are particularly useful when full best-response updates interact poorly with coupled constraints.

Keywords: GNEPs, local search, trust region, best response dynamics

1 Introduction

We consider a noncooperative game with N players, each with an objective function and a feasible strategy set. Both the objective function and the strategy set of each player may depend on the player’s own strategy as well as the strategies of the other players. Such a game is called a generalized Nash game, and finding its equilibria is called the generalized Nash equilibrium problem (GNEP). [Nash Jr \(1950\)](#) originally introduced the notion of Nash equilibrium for games in which only the objective functions depend on other players’ strategies. Later, [Debreu \(1952\)](#) and [Arrow and Debreu \(1954\)](#) extended this concept to the generalized Nash equilibrium (GNE), taking into account strategy sets that also depend on the strategies of other players. The GNEP was initially inspired by economics as an abstract economy or social equilibrium problem. Since then, it has been used to model a wide range of problems, such as autonomous racing [Dreves and Gerdtz \(2018\)](#); [Le Cleac’h et al. \(2022\)](#); [Pustilnik et al. \(2025\)](#); [Pustilnik and Borrelli \(2025\)](#), electricity markets [Jing-Yuan and Smeers \(1999\)](#); [Le Cadre et al. \(2020\)](#); [Contreras et al. \(2004\)](#); [Hobbs and Pang \(2007\)](#), environmental control [Krawczyk \(2005\)](#); [Breton et al. \(2006\)](#), forecast of prices and sales [Wang \(2023\)](#), mobile cloud computing [Cardellini et al. \(2016\)](#), traffic routing [Benenati and Grammatico \(2024\)](#), network optimization [Kelly et al. \(1998\)](#), supply chains [Hall et al. \(2024\)](#), electric vehicle charging [Ma et al. \(2016, 2011\)](#), transportation [Stein and Sudermann-Merx \(2018\)](#), and railroad pricing [Bassanini et al. \(2002\)](#). This broad range of applications motivates the development of reliable numerical methods for computing GNEs.

Computing a GNE is challenging because coupled constraints make each player’s feasible strategy set depend on the decisions of the other players. When constraints are not shared, unilateral updates may also make the strategies of other players infeasible. Feasible sets may even become empty, and standard equilibrium-existence assumptions need not hold.

Jointly-convex GNEPs are tractable because they admit a convex structure in the joint strategy space, which enables stronger existence guarantees and more stable algorithms. Common approaches include variational inequality reformulations [Facchinei et al. \(2007\)](#); [Kulkarni and Shanbhag \(2012\)](#), global optimization via the Nikaidô–Isoda function [URYAs’ Ev and Rubinstein \(2002\)](#); [Von Heusinger and Kanzow \(2009\)](#); [Krawczyk and Uryasev \(2000\)](#), and Karush–Kuhn–Tucker (KKT)-based methods [Dreves \(2011\)](#). In this setting, the most well-known solution concept is the variational equilibrium is a GNE where all players have the same Lagrange multiplier for any shared active constraints. Many methods for jointly-convex GNEPs target variational equilibria, however, in this work we are interested more broadly in computing GNEs.

Finding GNEs for general GNEPs is substantially more challenging because convexity and regularity properties available in the jointly-convex setting may be lost, player feasible sets may become empty, and equilibrium existence is more difficult to guarantee. In many cases, researchers reformulate general GNEPs as quasi-variational inequalities [Pang and Fukushima \(2005\)](#). The augmented Lagrangian method (ALM)-based techniques [Kanzow and Steck \(2016, 2018\)](#), as well as inexact and exact penalty

methods [Fukushima \(2011\)](#); [Ba and Pang \(2022\)](#); [Facchinei and Kanzow \(2010b\)](#); [Facchinei and Lampariello \(2011\)](#), are commonly used to handle these difficulties.

Best response dynamics (BRD) form a widely used class of algorithms for computing GNEs. In this approach, players iteratively update their strategies according to their best response. The updating scheme may be synchronous, sequential, or totally asynchronous. BRD enjoys convergence guarantees only in special classes of games, including potential games [Monderer and Shapley \(1996\)](#), aggregative games [Dindoš and Mezzetti \(2006\)](#), acyclic games [Fabrikant et al. \(2010\)](#), and quasi-acyclic games [Friedman and Mezzetti \(2001\)](#); [Takahashi and Yamamori \(2002\)](#). It has also been extended to GNEPs. In particular, [Thiran et al. \(2024\)](#) apply BRD in distributed GNEP settings, while [Lisboa et al. \(2025\)](#) demonstrate its performance on both benchmark and real-world problems, and the internet switching model. Although simple and intuitive, BRD may cycle in games with multiple equilibria and fail to converge [Thiran et al. \(2024\)](#). To mitigate this issue, better-response dynamics [Amiet et al. \(2021\)](#) allow players to switch to any improving strategy, not necessarily the best one. Another modification is the Krasnosel’skiĭ–Mann update [Neto et al. \(2025\)](#), which combines the previous iterate with the best response using a relaxation parameter in $(0, 1)$. For GNEPs with coupled non-shared constraints, full best-response updates may additionally be problematic because an update that is feasible for one player can render the joint strategy infeasible for another player.

A related class of best-response-based algorithms consists of decomposition methods. These approaches decompose the game into a sequence of smaller optimization problems, typically one per player, which are solved iteratively while keeping other players’ strategies fixed. Classical schemes include Jacobi and Gauss–Seidel methods. The Jacobi method enables parallel updates, whereas Gauss–Seidel updates players sequentially, see [Facchinei and Kanzow \(2010a\)](#); [Facchinei and Lampariello \(2011\)](#); [Facchinei et al. \(2011\)](#); [Fischer et al. \(2014\)](#); [Nie et al. \(2021\)](#); [Kim \(2021\)](#); [Migot and Cojocaru \(2021\)](#). [Migot and Cojocaru \(2020a\)](#) show that GNEPs can be interpreted through projected dynamical systems and sweeping processes whose stable points correspond to equilibria.

Recent work has also explored learning-based approaches. [Graf et al. \(2024\)](#) uses deep reinforcement learning (DRL), specifically deep deterministic policy gradient, to compute Nash equilibria in Bertrand competition. [Benteke Longaou et al. \(2025\)](#) extend multi-agent DRL methods, including multi-agent deep deterministic policy gradient and multi-agent proximal policy optimization, to both shared and non-shared constraints GNEPs.

GNEPs may admit multiple or infinitely many equilibria. Although most numerical methods target a single GNE, identifying different equilibria may be valuable when the solution set is nonunique. Multi-start heuristic methods provide a natural mechanism for exploring such solution diversity. Parametrized variational inequality formulations were introduced for jointly-convex GNEPs [Nabetani et al. \(2011\)](#); [Cojocaru et al. \(2018\)](#); [Dreves and Kanzow \(2011\)](#) and later extended to general GNEPs [Migot and Cojocaru \(2020b\)](#). [Dreves \(2014, 2017\)](#) study affine GNEPs.

Researchers have also developed local search methods to compute Nash equilibria via support exploration. [Porter et al. \(2008\)](#) propose algorithms that systematically

explore support profiles, pruning infeasible ones based on dominance relations and favoring small supports. Extending this idea, [Ceppi et al. \(2010\)](#) introduce an anytime algorithm that combines support enumeration with local search, framing equilibrium computation as a combinatorial optimization problem over supports.

Heuristic and metaheuristic methods have also been proposed to improve effectiveness and solution diversity. These include evolutionary-inspired algorithms, stochastic gradient descent (SGD) methods [Fields \(2021\)](#); [Benteke et al. \(2024\)](#), and the generalized Nash equilibrium iterative Sierpinski solver (GNEISS) [Wild \(2017\)](#). Population-based approaches such as the whale slime mould Algorithm (CWSMA) [Wang and Jia \(2025\)](#) and coevolutionary immune quantum particle swarm optimization [Liu and Jia \(2020\)](#) typically produce a single equilibrium per run, but can be used repeatedly to obtain multiple solutions. While these methods perform well on benchmark jointly-convex GNEPs, their behavior on general GNEPs is still not fully understood. In particular, there remains limited evidence on simple local-search mechanisms specifically designed to cope with the feasibility difficulties created by non-shared constraints.

In this paper, we propose Adaptive Randomized Equilibrium Search (ARES), a local-search heuristic for computing GNEs. ARES uses best-response information to define promising search directions but, rather than moving directly to a best response, samples randomized partial moves along these directions. Candidate actions are filtered according to the feasibility of the resulting joint strategy, and a Gauss–Southwell-type rule selects the player to update. An adaptive search radius controls the magnitude of the sampled actions. ARES computes a single equilibrium per run, while repeated runs can be used to explore multiple GNEs, either from different initial points or, because of randomization, from the same initial point. ARES does not enforce monotonic decrease of the equilibrium residual: because the best response is recomputed after each update, an accepted partial move may lead to an increase in the resulting residual. This nonmonotonicity gives the search additional flexibility compared with methods that require improvement at every iteration. The main contributions of this work are the development of a randomized local-search framework based on partial best-response moves and an adaptive search radius, together with a feasibility mechanism that accounts for the effect of unilateral updates on all players’ constraints and thereby enables the treatment of GNEPs with non-shared constraints. We also provide an empirical evaluation on academic jointly-convex and non-shared-constraint GNEPs, as well as several application-inspired jointly-convex models, including comparisons with a deterministic full best-response variant. Finally, we investigate how randomized partial moves affect robustness and the exploration of multiple equilibria.

The remainder of this paper is organized as follows. Section 2 provides a brief review of GNEPs. Section 3 introduces the proposed algorithm. Section 4 reports the performance metrics used for evaluation and the numerical experiments on nineteen benchmark GNEPs, including both academic and real-world. Finally, Section 5 concludes the study and outlines potential directions for future research.

2 Background

This section introduces the main concepts used throughout the paper. A GNEP is a non-cooperative game with N players. For each player $\nu \in \mathcal{V} := \{1, \dots, N\}$, the objective function $\theta_\nu(x_\nu, x_{-\nu})$ and feasible set may depend on both the player's own strategy x_ν and the strategies of the other players $x_{-\nu}$. Here, $x_\nu \in \mathbb{R}^{n_\nu}$ is the strategy of player ν , while $x_{-\nu} := (x_\mu)_{\mu \neq \nu} \in \mathbb{R}^{n-n_\nu}$ collects the strategies of all other players. The total dimension is $n = \sum_{\nu=1}^N n_\nu$, with $n_{-\nu} = n - n_\nu$, and the joint strategy vector is $x = (x_\nu)_{\nu=1}^N \in \mathbb{R}^n$. We occasionally write $(x_\nu, x_{-\nu})$ instead of x , without implying any reordering of components. For each player ν , the feasible set is defined through a vector-valued constraint function $g_\nu : \mathbb{R}^n \rightarrow \mathbb{R}^{m_\nu}$. Given the strategies of the other players $x_{-\nu}$, the feasible set of player ν is defined as

$$X_\nu(x_{-\nu}) := \{x_\nu \in \mathbb{R}^{n_\nu} \mid g_\nu^j(x_\nu, x_{-\nu}) \leq 0, j = 1, \dots, m_\nu\}.$$

The goal is to find a strategy vector $x^* = (x_1^*, \dots, x_N^*)$ such that no player can improve its objective by unilateral deviation, i.e.,

$$\theta_\nu(x_\nu^*, x_{-\nu}^*) \leq \theta_\nu(x_\nu, x_{-\nu}^*), \quad \forall x_\nu \in X_\nu(x_{-\nu}^*), \forall \nu \in \mathcal{V},$$

and such that feasibility holds: $x_\nu^* \in X_\nu(x_{-\nu}^*)$, $\forall \nu \in \mathcal{V}$. The vector x^* is called a generalized Nash equilibrium (GNE). Equivalently, an N -player GNEP is a system of N coupled optimization problems, where each player ν optimizes over n_ν variables under constraints depending on the other players' strategies:

$$P_\nu(x_{-\nu}): \min_{x_\nu \in \mathbb{R}^{n_\nu}} \theta_\nu(x_\nu, x_{-\nu}) \text{ s.t. } x_\nu \in X_\nu(x_{-\nu}),$$

We denote by

$$\Omega := \{x \mid x_\nu \in X_\nu(x_{-\nu}), \forall \nu \in \mathcal{V}\}$$

the joint feasible strategy set. Because our proposed approach allows a controlled relaxation of feasibility during the search, we also define the following Δ -feasible set, with $\Delta \geq 0$. For each player ν and a fixed strategy profile $x_{-\nu}$ of the remaining players, the Δ -feasible set is given by:

$$X_\nu^\Delta(x_{-\nu}) := \left\{ x'_\nu \in \mathbb{R}^{n_\nu} : \max_{j=1, \dots, m_\nu} g_\nu^j(x'_\nu, x_{-\nu}) \leq \Delta \right\} \quad (1)$$

We then define classical assumptions used to state standard existence and best-response results.

Assumption 1 There exist N nonempty, convex, and compact sets $L_\nu \subset \mathbb{R}^{n_\nu}$ such that, for every $x \in \mathbb{R}^n$ with $x_\nu \in L_\nu$ for every ν , $X_\nu(x_{-\nu})$ is nonempty, closed, and convex, $X_\nu(x_{-\nu}) \subseteq L_\nu$, and X_ν , as a point-to-set map, is both upper and lower semicontinuous.

Assumption 2 For every player ν , the function $\theta_\nu(\cdot, x_{-\nu})$ is quasi-convex on $X_\nu(x_{-\nu})$.

These conditions provide sufficient conditions for the existence of a GNE [Facchinei and Kanzow \(2010a\)](#), but they are not required in our proposed approach as global assumptions. For the algorithm to be well defined, however, we assume that the best-response problem of each player admits a solution at every strategy profile encountered by the method. If the GNEP admits no equilibrium, then the exact best-response residual cannot attain zero. In this case, our proposed approach may still search for a strategy profile with small residual, but reaching the maximum number of iterations without satisfying the prescribed tolerance should be interpreted only as algorithmic failure, not as a certificate of nonexistence of a GNE.

Definition 1 (Jointly-convex GNEP [Facchinei and Kanzow \(2010a\)](#)) A GNEP is jointly convex if, for every player ν , $\theta_\nu(\cdot, x_{-\nu})$ is convex, $X_\nu(x_{-\nu})$ is closed and convex, and there exists a nonempty, closed, and convex set $X \subseteq \mathbb{R}^n$ such that $X_\nu(x_{-\nu}) := \{x_\nu \in \mathbb{R}^{n_\nu} \mid (x_\nu, x_{-\nu}) \in X\}$.

The best response mapping of player ν given the strategies of the other players $x_{-\nu}$ is defined as

$$\mathcal{BR}_\nu(x_{-\nu}) := \underset{x'_\nu \in X_\nu(x_{-\nu})}{\operatorname{argmin}} \theta_\nu(x'_\nu, x_{-\nu}). \quad (2)$$

A feasible x^* is a GNE if and only if $x_\nu^* \in \mathcal{BR}_\nu(x_{-\nu}^*)$, $\forall \nu \in \mathcal{V}$. Equivalently, $x^* \in \mathcal{BR}(x^*) := \mathcal{BR}_1(x_{-1}^*) \times \cdots \times \mathcal{BR}_N(x_{-N}^*)$.

If Assumption 1 holds and, for each player ν , the function θ_ν is continuous, then the existence of a best response is guaranteed.

In general, the best-response correspondence $\mathcal{BR}(x)$ may be set-valued. A shadow coordinate of x_ν is the selected best response of player ν to $x_{-\nu}$, and it is denoted by $\hat{x}_\nu$. Then, $\hat{x} = (\hat{x}_1, \dots, \hat{x}_N)$ is called the selected best response (or shadow point) to $x = (x_1, \dots, x_N)$, with $\hat{x}_\nu \in \mathcal{BR}_\nu(x_{-\nu})$, $\forall \nu = 1, \dots, N$.

The exact best-response residual is

$$\hat{f}(x) := \operatorname{dist}(x, \mathcal{BR}(x)) := \inf_{\hat{x} \in \mathcal{BR}(x)} \|\hat{x} - x\|_2. \quad (3)$$

Under some of our assumptions, the best-response set is compact, so the infimum above can equivalently be written as a minimum.

In practice, when a best response is not available analytically, we approximate a selected best response numerically using an optimization solver. We continue to denote the resulting point by $\hat{x}$, with the understanding that the computed residual depends on the accuracy of the underlying best-response solve. In this context, the merit or fitness function becomes

$$f_{\hat{x}}(x) := \|\hat{x} - x\|_2, \quad (4)$$

where the dependence of f to $\hat{x}$ is in general clear from the context and omitted.

Proposition 1 *The function $f(x)$ is an upper bound on the distance from x to the best-response set, i.e.,*

$$f_{\hat{x}}(x) \geq \operatorname{dist}(x, \mathcal{BR}(x)).$$

Proof Since $\hat{x}$ is a best-response candidate, while $\text{dist}(x, \mathcal{BR}(x)) = \min_{y \in \mathcal{BR}(x)} \|x - y\|_2$ is the minimum distance over all best responses, it follows immediately that

$$\|\hat{x} - x\|_2 \geq \text{dist}(x, \mathcal{BR}(x)).$$

Hence, $f(x)$ is an upper bound on the distance from x to the best-response set. $\square$

This residual or fitness function (4) plays the role of an equilibrium merit function. Unlike the Nikaidô–Isoda function, which measures objective improvement gaps, it measures the distance to a selected best-response Fields (2021); Benteke et al. (2024).

A feasible point x^* is a GNE if and only if $\hat{f}(x^*) = 0$. In practice, our proposed approach uses the selected best-response residual f as a computable fitness measure.

3 Adaptive Randomized Equilibrium Search algorithm

We now present Adaptive Randomized Equilibrium Search (ARES) for approximately minimizing the residual f (4). At each iteration, ARES uses the current selected best response to generate randomized partial moves for each player. Candidate moves are filtered according to joint feasibility, and a Gauss–Southwell-type rule selects one player to update. The magnitude of the sampled moves is controlled by an adaptive search radius.

ARES starts from a jointly feasible strategy profile x^0 and generates iterates x^k , for $k = 1, \dots, K$. At each iteration k , the selected best response $\hat{x}_\nu^{k-1}$ defines the best-response direction

$$\hat{\beta}_\nu^k := \hat{x}_\nu^{k-1} - x_\nu^{k-1}.$$

Unlike classical best-response dynamics, ARES does not necessarily move directly to the best response. Instead, it considers randomized partial moves along best-response directions, which can provide additional flexibility when full best-response updates lead to cycling or violate joint feasibility.

ARES does not require the residual to decrease monotonically, since the selected best response is recomputed after each accepted update. Finally, ARES updates its strategy for only one player at each iteration, and it has an adaptive mechanism inspired by trust-region to find their current strategies.

3.1 Randomized partial best-response search

In this subsection, we present the main components of the proposed ARES algorithm, as outlined in Algorithm 1.

3.1.1 Action generation

To construct the randomized partial best-response search, normalize each player ν 's best-response direction and then scale it by the trust-region radius δ , yielding

$$\hat{\sigma}_\nu := \frac{\delta}{\|\hat{\beta}_\nu\|_2} \hat{\beta}_\nu, \quad (5)$$

where the iteration indices are omitted here.

Within the ARES framework, we study randomized moves aligned component-wise with the best-response direction. Specifically, each player ν samples M candidate actions $\mathcal{A}_\nu = \{a_{\nu,i}\}_{i=1}^M$ by independently drawing each coordinate between 0 and the corresponding component of $\hat{\sigma}_\nu \in \mathbb{R}^{n_\nu}$. For each $i = 1, \dots, M$, the candidate action $a_{\nu,i} \in \mathbb{R}^{n_\nu}$ is generated component-wise according to

$$(a_{\nu,i})_j \sim \text{Unif}(\min(0, (\hat{\sigma}_\nu)_j), \max(0, (\hat{\sigma}_\nu)_j)), \quad j = 1, \dots, n_\nu,$$

where Unif denotes the uniform distribution over the specified interval.

The sampling neighborhood is bounded by δ , so that the resulting moves satisfy $\|a_{\nu,i}\|_2 \leq \delta$. Thus, ARES explores a bounded region in the direction of the best response while allowing different components to move by different amounts.

If $\|\hat{\beta}_\nu\|_2 = 0$, then the current strategy of player ν already coincides with its selected best response and $\|\hat{\sigma}_\nu\| = 0$. In this case, no normalization is required, and the natural candidate action is simply $a_\nu = 0$.

Algorithm 1 Adaptive Randomized Equilibrium Search (ARES)

Require: Initial feasible strategy x^0 , maximum iterations K , number of sampled actions M , initial $\delta_0 > 0$, $\delta_{\max}$, tolerance $\varepsilon > 0$, and constants $0 < \rho_0, \rho_1 < 1$.

- 1: Initialize: $k \leftarrow 0$, $\delta \leftarrow \delta_0$
- 2: Compute an approximate best response $\hat{x}^0$ to x^0 and $f^0 \leftarrow \|\hat{x}^0 - x^0\|_2$.
- 3: **while** $k \leq K$ or $(f^k > \varepsilon$ or $c^k > \varepsilon)$ **do**
- 4: **for** $\nu = 1$ **to** N **do**
- 5: $(a_\nu, \gamma_\nu, f_\nu) \leftarrow \text{BESTACTIONANDFITNESS}(\nu, \hat{x}^{k-1}, x_\nu^{k-1}, x_{-\nu}^{k-1}, \varepsilon, \delta)$
- 6: **end for**
- 7: Select player with lowest fitness: $\nu^* \leftarrow \arg \min_\nu f_\nu$
- 8: Update strategy for selected player: $x_{\nu^*}^k \leftarrow x_{\nu^*}^{k-1} + a_{\nu^*}$, $x_{-\nu^*}^k \leftarrow x_{-\nu^*}^{k-1}$
- 9: $c^k \leftarrow \max_{\nu=1, \dots, N} \max_{j=1, \dots, m_\nu} g_\nu^j(x^k)$
- 10: Compute an approximate best response $\hat{x}^k$ to x^k
- 11: Compute the fitness: $f^k \leftarrow \|\hat{x}^k - x^k\|_2$
- 12: $\delta \leftarrow \text{RADIUSUPDATE}(\delta, \delta_{\max}, f^k, f^{k-1}, (a_\nu)_{\nu=1}^N, (\gamma_\nu)_{\nu=1}^N, \varepsilon, \rho_0, \rho_1, \nu^*)$
- 13: $k \leftarrow k + 1$
- 14: **end while**
- 15: **return** x^k or report failure if maximum iterations reached

3.1.2 Joint feasibility control

Feasibility is a key challenge in GNEPs because players' feasible sets are interdependent. In particular, an update that is feasible for the updating player may violate the constraints of other players, so player-wise feasibility does not necessarily imply feasibility of the resulting joint strategy.

For GNEPs with shared constraints, this issue does not arise when players are updated one at a time: if the updated player's action satisfies the shared constraints while the other strategies remain fixed, then the resulting joint strategy remains feasible. However, this property does not necessarily hold for GNEPs with non-shared constraints. In such cases, the updating of a single player's strategy may preserve the feasibility of that player but violate the constraints of the other players as illustrated by Example 1.

Example 1 Consider the following three-player game:

$$\begin{aligned} P_1 : \quad & \min_{x_1 \in \mathbb{R}} \frac{1}{2}(x_1 - 1)^2 - x_1 x_2 \quad \text{s.t. } x_1 + x_2 + x_3 \leq 1, \\ P_2 : \quad & \min_{x_2 \in \mathbb{R}} \frac{1}{2}(x_2 - 1)^2 + x_1 x_2 \quad \text{s.t. } x_1 + x_2 + x_3 \leq 1, \\ P_3 : \quad & \min_{x_3 \in \mathbb{R}} \frac{1}{2}(x_3 - 1)^2 \quad \text{s.t. } 0 \leq x_3 \leq x_1 + x_2. \end{aligned}$$

Consider the feasible strategy $x^{k-1} = (x_1^{k-1}, x_2^{k-1}, x_3^{k-1}) = (1.0, -0.5, 0.2)$. At iteration k , Player 1's best-response problem is

$$\min_{x_1 \in \mathbb{R}} \frac{1}{2}(x_1 - 1)^2 - x_1 x_2^{k-1} \quad \text{subject to } x_1 + x_2^{k-1} + x_3^{k-1} \leq 1,$$

and the best response is $\hat{x}_1^{k-1} = 1 + x_2^{k-1} = 0.5$.

Suppose Player 1 updates its strategy using the best response to x_1^{k-1} :

$$x_1^k = x_1^{k-1} + \hat{\beta}_1^k = \hat{x}_1^{k-1} = 0.5.$$

The updated profile is therefore $x^k = (0.5, -0.5, 0.2)$. The update is feasible for Player 1; however, the updated profile violates Player 3's constraint $x_3^k \leq x_1^k + x_2^k$.

To address this issue, ARES controls the feasibility of the overall strategy profile when updating a player's strategy. Specifically, a new iterate x^k is chosen such that the resulting constraint violation satisfies

$$c^k = \max_{\nu=1,\dots,N} \left(\max_{j=1,\dots,m_\nu} g_\nu^j(x^k) \right) \leq \Delta_k. \quad (6)$$

Therefore, all iterates produced by ARES remain Δ_k -feasible. Moreover, since $\Delta_k = \max(\varepsilon, \delta_{k-1})$, then $\Delta_k \rightarrow \varepsilon$ whenever $\delta_{k-1} \rightarrow 0$ as $k \rightarrow \infty$.

Accordingly, the Δ -feasible set for player ν is redefined as

$$X_\nu^\Delta(x_{-\nu}) := \left\{ x'_\nu \in \mathbb{R}^{n_\nu} : \max_{\mu=1,\dots,N} \left(\max_{j=1,\dots,m_\mu} g_\mu^j(x'_\nu, x_{-\nu}) \right) \leq \Delta \right\}. \quad (7)$$

3.1.3 Action selection

Each player evaluates the fitness of candidate strategies obtained by adding each action to the previous strategy, $x_\nu^{k-1} + a_{\nu,i}$, for $i = 1, \dots, M$. To avoid recomputing

a best response for every sampled candidate, ARES ranks candidate strategies using their distance to the current selected best response.

An action $a_{\nu,i}$ is Δ -feasible if it satisfies $x_{\nu}^{k-1} + a_{\nu,i} \in X_{\nu}^{\Delta}(x_{-\nu}^{k-1})$ for all ν . Among the Δ -feasible candidate actions, the player selects the one that yields the lowest fitness, while the strategies of the other players remain fixed. Specifically,

$$a_{\nu}^{\star} := \operatorname{argmin}_{a_{\nu,i} \in \mathcal{A}_{\nu}^k} \left\{ \left\| (\hat{x}_{\nu}^{k-1}, \hat{x}_{-\nu}^{k-1}) - (x_{\nu}^{k-1} + a_{\nu,i}, x_{-\nu}^{k-1}) \right\|_2 : x_{\nu}^{k-1} + a_{\nu,i} \in X_{\nu}^{\Delta}(x_{-\nu}^{k-1}) \right\}$$

The joint Δ -feasibility criterion and its role in controlling constraint violations are discussed in Section 3.1.2.

```

1: procedure RADIUSUPDATE( $\delta, \delta_{\max}, f^k, f^{k-1}, (a_{\nu})_{\nu=1}^N, (\gamma_{\nu})_{\nu=1}^N, \varepsilon, \rho_0, \rho_1, \nu^{\star}$ )
2:   Compute the progress indicator:  $b^{\star} = f_{\nu^{\star}}^{k-1} - f_{\nu^{\star}}^k$ 
3:   if  $b^{\star} = 0 \wedge \forall \nu (\gamma_{\nu} = 0)$  then                                      $\triangleright$  Stagnation
4:      $\delta \leftarrow \rho_0 \times \delta$                                                   $\triangleright$  Reduction
5:   else if  $b^{\star} < \rho_1$  then                                              $\triangleright$  No progress or deterioration
6:      $\delta \leftarrow \rho_0 \times \delta$                                               $\triangleright$  Reduction
7:   else if  $b^{\star} > \rho_1$  then                                            $\triangleright$  Sufficient progress
8:     if  $\|a_{\nu^{\star}}\| - \delta \leq \varepsilon$  then                                    $\triangleright$  Active trust region
9:        $\delta \leftarrow \min(2.0 \times \delta, \delta_{\max})$                               $\triangleright$  Expansion
10:    else if  $\|a_{\nu^{\star}}\| - \delta > \varepsilon \wedge \forall \nu (\gamma_{\nu} \neq 0)$  then
11:       $\delta \leftarrow \delta$                                                       $\triangleright$  Preservation
12:    else if  $\|a_{\nu^{\star}}\| - \delta > \varepsilon \wedge \exists \nu (\gamma_{\nu} = 0)$  then
13:       $\delta \leftarrow \rho_0 \times \delta$                                             $\triangleright$  Reduction
14:    end if
15:  end if
16:  return  $\delta$ 
17: end procedure

```

3.1.4 Player selection and update

ARES uses a GS type of player selection in which the player whose candidate action yields the lowest fitness is selected for update. Specifically,

$$\nu^{\star} = \operatorname{argmin}_{\nu \in \mathcal{V}} \left\| (\hat{x}_{\nu}^{k-1}, \hat{x}_{-\nu}^{k-1}) - (x_{\nu}^{k-1} + a_{\nu}^{\star}, x_{-\nu}^{k-1}) \right\|_2,$$

where the candidate action satisfies

$$x_{\nu}^{k-1} + a_{\nu}^{\star} \in X_{\nu}^{\Delta}(x_{-\nu}^{k-1}).$$

Only the selected player updates its strategy:

$$x_{\nu^{\star}}^k = x_{\nu^{\star}}^{k-1} + a_{\nu^{\star}}^{\star}, \quad x_{-\nu^{\star}}^k = x_{-\nu^{\star}}^{k-1}.$$

Thus, ARES employs a greedy block-selection rule inspired by GS, while alternative selection mechanisms, such as random or cyclic selection, could also be considered.

```

1: procedure BESTACTIONANDFITNESS( $\nu, \hat{x}, x_\nu, x_{-\nu}, \varepsilon, \delta$ )
2:   Initialize:  $f_\nu^* \leftarrow \infty, a_\nu^* \leftarrow 0$ 
3:    $\hat{\beta}_\nu \leftarrow \hat{x}_\nu - x_\nu$ ; Set  $\Delta \leftarrow \max(\delta, \varepsilon)$ 
4:   if  $\|\hat{\beta}_\nu\| \leq 10^{-6}$  then
5:     Set  $a_\nu^* \leftarrow 0$  and  $f_\nu^* \leftarrow \|\hat{x} - x\|_2$ 
6:     Create a new strategy:  $y = (x_\nu, x_{-\nu})$ 
7:     Compute constraint violation:  $c \leftarrow \max_{\nu=1, \dots, N} (\max_{j=1, \dots, m_\nu} g_\nu^j(y))$ 
8:      $\gamma_\nu \leftarrow \begin{cases} 1, & c \leq \Delta, \\ 0, & c > \Delta. \end{cases}$ 
9:   else
10:    Scale to trust region:  $\hat{\sigma}_\nu \leftarrow \frac{\delta}{\|\hat{\beta}_\nu\|_2} \hat{\beta}_\nu$ 
11:    Sample actions:  $\mathcal{A}_\nu = \{a_{\nu,i}\}_{i=1}^M$ , where  $a_{\nu,i} \in \mathbb{R}^{n_\nu}$  and  $(a_{\nu,i})_j \sim \text{Unif}(\min(0, \hat{\sigma}_{\nu,j}), \max(0, \hat{\sigma}_{\nu,j}))$ ,  $j = 1, \dots, n_\nu$ .
12:    for all  $a_\nu \in \mathcal{A}_\nu$  do
13:      Create a new strategy:  $y = (x_\nu + a_\nu, x_{-\nu})$ 
14:      Compute constraint violation:  $c \leftarrow \max_{\nu=1, \dots, N} (\max_{j=1, \dots, m_\nu} g_\nu^j(y))$ 
15:      if  $c \leq \Delta$  then
16:        Compute fitness:  $f_\nu \leftarrow \|\hat{x} - y\|_2$ 
17:        if  $f_\nu < f_\nu^*$  then
18:           $a_\nu^* \leftarrow a_\nu, \gamma_\nu \leftarrow 1, f_\nu^* \leftarrow f_\nu$ 
19:        end if
20:      end if
21:    end for
22:    if  $\gamma_\nu = 0$  then  $\triangleright$  No feasible action exists for player  $\nu$ .
23:       $a_\nu^* \leftarrow 0, f_\nu^* \leftarrow \|\hat{x} - x\|_2$ 
24:    end if
25:  end if
26:  return  $(a_\nu^*, \gamma_\nu, f_\nu^*)$ 
27: end procedure

```

3.1.5 Adaptive search-radius mechanism

The radius δ is adaptively adjusted according to the improvement observed between successive iterations. This adjustment is inspired by the trust-region principle of adapting the size of the search region based on the quality of the progress achieved ?. All steps of the trust-region update are detailed in Procedure RadiusUpdate. Specifically, we measure the absolute improvement for the selected player ν^* by

$$b^* := f_{\nu^*}^{k-1} - f_{\nu^*}^k,$$

where $f_{\nu^*}^k$ denotes the fitness value of the selected player at iteration k . A positive value of b^* indicates an improvement in fitness relative to the previous iteration, $b^* = 0$ indicates no improvement (progress), and a negative value of b^* indicates a deterioration in fitness.

In addition to b^* , we consider the Δ -feasibility of the action a_ν selected by player ν . This feasibility is encoded by the indicator.

$$\gamma_\nu = \begin{cases} 1, & \text{if } c \leq \Delta, \\ 0, & \text{otherwise,} \end{cases}$$

where $c = \max_{\nu=1,\dots,N} (\max_{j=1,\dots,m_\nu} g_\nu^j(x_\nu + a_\nu, x_{-\nu}))$. Thus, $\gamma_\nu = 1$ indicates that the resulting action profile is Δ -feasible, while $\gamma_\nu = 0$ indicates that it is not. This indicator is used in the adjustment of δ .

The value of δ is adjusted according to three modes: contraction, expansion, and preservation, based on b^* , the players' actions, and the feasibility of all players' actions via γ_ν for $\nu \in \mathcal{V}$, even though only one player is updated at each iteration.

In this study, we set ρ_0 and ρ_1 are 0.75 each.

Contraction occurs when the search fails to make sufficient progress ($b^* < \rho_1$), and the search radius is then reduced according to

$$\delta = \rho_0 \times \delta.$$

This contraction applies both when the search makes limited progress, ($0 < b^* < \rho_1$), and when fitness deteriorates, ($b^* < 0$).

Contraction also occurs when no progress is made and no player finds feasible action, i.e., $b^* = 0$ and $\gamma_\nu = 0$ for all ν , then the search neighborhood is contracted, with the radius updated according to

$$\delta = \rho_0 \times \delta.$$

Additionally, contraction is also applied when progress is achieved, but the action a_{ν^*} for the selected player is away from the boundary of the trust-region, and at least one player generates an infeasible action. This is if $b^* \geq \rho_1$, $||a_{\nu^*}|| - \delta| > \varepsilon$, and $\exists \nu (\gamma_\nu = 0)$, then

$$\delta = \rho_0 \times \delta.$$

Reducing δ narrows the search region and encourages more localized exploration around the current solution.

Under the expansion strategy, the trust-region radius increases when adequate progress has been made, and the action a_{ν^*} for the selected player is close to the boundary of the current region. Particularly, whenever $b^* \geq \rho_1$, then a_{ν^*} is checked for its proximity to the boundary of the current trust region radius, that is, satisfying

$$||a_{\nu^*}|| - \delta| \leq \varepsilon,$$

the radius is expanded as $\delta \leftarrow \min(2 \times \delta, \delta_{\max})$, allowing the players to explore a larger neighborhood.

In preservation mode, the radius remains unchanged when sufficient progress has been made, and the action a_{ν^*} for the selected player is away from the boundary of the current region. Specifically, if

$$b^* \geq \rho_1, \quad |||a_{\nu^*}| - \delta| > \varepsilon,$$

and all players generate nonzero actions, the radius is maintained: $\delta = \delta$.

In ARES, a player may choose the best-response direction as the optimal action. If the best-response direction satisfies the trust-region condition and leads to a globally Δ -feasible joint strategy while keeping the other players' strategies fixed, then it is selected, as it provides the greatest improvement.

3.2 ARES with best-response directions

To assess the effect of randomized partial updates, we consider a deterministic variant in which candidate partial moves are replaced by full best-response steps. We refer to this method as ARES with best-response directions (ARES-BR).

In this setting, the selected player ν^* always takes the full best response step. Specifically, the selected player updates along the best-response direction:

$$x_{\nu^*}^k = x_{\nu^*}^{k-1} + a_{\nu^*}^* = x_{\nu^*}^{k-1} + \hat{\beta}_{\nu^*}^k = x_{\nu^*}^{k-1} + (\hat{x} * \nu^{*k-1} - x * \nu^{*k-1}) = \hat{x}_{\nu^*}^{k-1}.$$

The important difference from the Algorithm 1 is that $a_{\nu^*}^*$ is obtained directly from the approximate best response.

The ARES-BR algorithm is a best-response dynamics (BRD) method with a GS player-selection mechanism. Unlike sequential BRD, where players update according to a predetermined order, or synchronous BRD, ARES-BR evaluates the best-response update of each player and selects the player with the lowest fitness for the actual strategy update. Thus, only one player updates at each iteration, while the strategies of the remaining players are kept fixed.

Since ARES-BR directly uses the best-response action, we do not employ a trust-region mechanism. In particular, we set $\delta = 0$ and consequently $\Delta = \max(\delta, \varepsilon) = \varepsilon$, where $\varepsilon > 0$ is the prescribed feasibility tolerance. Therefore, a proposed update is accepted only if its constraint violation satisfies $c^k \leq \Delta = \varepsilon$. If $c^k > \varepsilon$, the resulting iterate is infeasible beyond the prescribed tolerance. In this case, the proposed update is rejected, the algorithm terminates, and ARES-BR is reported as a failure. In other words, ARES-BR does not attempt to reduce the step size or recover from an infeasible update through a trust-region adjustment. Once an infeasible iterate is encountered, the procedure stops.

The ARES-BR procedure is given in Algorithm 2.

Algorithm 2 ARES-BR

Require: Initial feasible strategy x^0 , maximum iterations K , initial approximate best response $\hat{x}^0$, and tolerance $\varepsilon > 0$

- 1: Initialize: $k \leftarrow 0$
- 2: Compute the initial fitness: $f^0 \leftarrow \|\hat{x}^0 - x^0\|_2$
- 3: **while** $k < K$ and $(f^k > \varepsilon$ or $c^k > \varepsilon)$ **do**
- 4: **for** $\nu = 1$ **to** N **do**
- 5: Fix the strategies of all other players: $x_{-\nu}^k \leftarrow x_{-\nu}^{k-1}$
- 6: Compute the best-response action for player ν : $a_\nu^* \leftarrow \hat{x}_\nu^{k-1} - x_\nu^{k-1}$.
- 7: Construct the candidate strategy profile: $x_\nu^k \leftarrow x_\nu^{k-1} + a_\nu^*$, $x_{-\nu}^k \leftarrow x_{-\nu}^{k-1}$
- 8: Compute the fitness of the candidate: $f_\nu^k \leftarrow \|\hat{x}^{k-1} - x^k\|_2$
- 9: **end for**
- 10: Select the player with the lowest fitness: $\nu^* \leftarrow \arg \min_{\nu=1,\dots,N} f_\nu^k$
- 11: Update the strategy of the selected player: $x_{\nu^*}^k \leftarrow x_{\nu^*}^{k-1} + a_{\nu^*}^*$, $x_{-\nu^*}^k \leftarrow x_{-\nu^*}^{k-1}$
- 12: Compute the constraint violation: $c^k \leftarrow \max_{\nu=1,\dots,N} \max_{j=1,\dots,m_\nu} g_\nu^j(x^k)$
- 13: **if** $c^k > \varepsilon$ **then**
- 14: Reject the proposed move: $x^k \leftarrow x^{k-1}$
- 15: **break**
- 16: **end if**
- 17: Compute an approximate best response $\hat{x}^k$ to x^k and $f^k \leftarrow \|\hat{x}^k - x^k\|_2$
- 18: $k \leftarrow k + 1$
- 19: **end while**
- 20: **return** x^k or report failure if the maximum number of iterations is reached.

3.3 Illustrative examples

In this subsection, we present two examples illustrating the advantages of ARES over ARES-BR. Furthermore, the second example demonstrates that ARES converges even where both BRD and the Gauss-Seidel method (GSM) fail to converge. For the two examples, we set $K = 500$, $M = 200$, $\varepsilon = 10^{-4}$, $\delta_0 = 25.0$, and $\delta_{\max} = 50$.

Example 2 Consider the following two-player non-shared constraints GNEP.

$$\begin{array}{ll} P_1(x_2) & \min_{x_1 \in \mathbb{R}} x_1 \\ & \text{s.t.} \quad -1 \leq x_1 \leq 1 \\ & \quad \quad x_1 + x_2 \leq 0 \end{array} \qquad \begin{array}{ll} P_2(x_1) & \min_{x_2} x_2^2 \\ & \text{s.t.} \quad -1 \leq x_2 \leq 1 \\ & \quad \quad x_1 + x_2 \geq 0 \end{array}$$

The corresponding GNE is given by $(x_1^*, x_2^*) = (-1, 1)$.

ARES-BR: One iteration

Suppose that we start at the joint feasible starting point $x^0 = (0, 0)$, with $\hat{x}^0 = (-1, 0)$. At iteration $k = 1$:

1. **Player 1 optimization:** Fixing player 2's strategy ($x_2^0 = 0$), player 1 computes the action $a_1^* = \hat{x}_1^0 = -1$, updating its strategy via:

$$x_1^1 = x_1^0 + a_1^* = 0 - 1 = -1$$

This yields the candidate strategy profile $x^1 = (x_1^1, x_2^0) = (-1, 0)$. The corresponding fitness value for player 1 is evaluated as:

$$f_1^1 = \|\hat{x}^0 - x^1\|_2 = \|(-1, 0) - (-1, 0)\|_2 = 0$$

2. **Player 2 optimization:** Fixing player 1's strategy ($x_1^0 = 0$), player 2 computes the action step $a_2^* = \hat{x}_2^0 = 0$, updating its strategy via:

$$x_2^1 = x_2^0 + a_2^* = 0 - 0 = 0$$

This yields the candidate strategy profile $x^1 = (x_1^0, x_2^1) = (0, 0)$. The corresponding fitness value for player 2 is evaluated as:

$$f_2^1 = \|\hat{x}^1 - x^1\|_2 = \|(-1, 0) - (0, 0)\|_2 = 1$$

3. **Player selection and joint feasibility check:** Since player 1 achieves the lower fitness value ($f_1^1 = 0 < f_2^1 = 1$), player 1 is selected to update its strategy while player 2 remains fixed, leading to the proposed strategy profile $x^1 = (-1, 0)$.

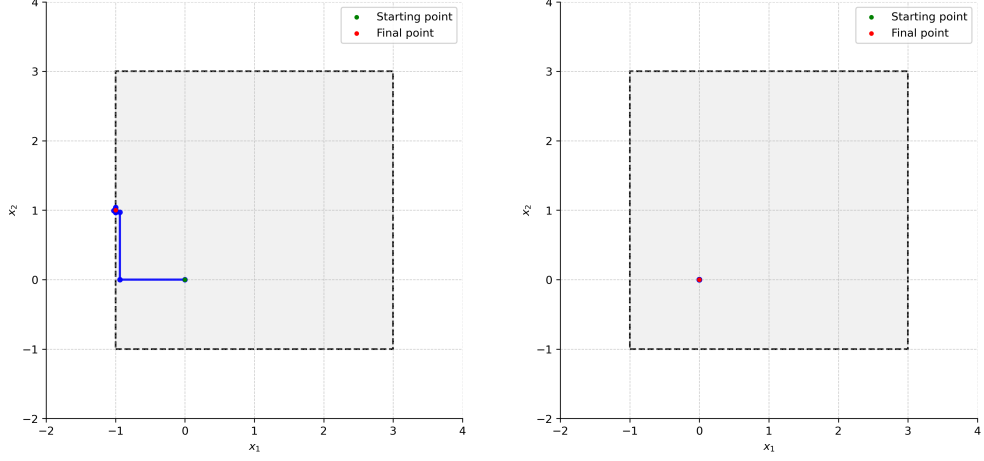
However, $x^1 = (-1, 0)$ is not jointly feasible: while it satisfies player 1's individual feasibility, it violates the feasibility condition for player 2. Consequently, no move is executed, and the algorithm remains stationary at $(0, 0)$ until the maximum number of iterations is reached.

So, ARES-BR fails to converge from $(0, 0)$ due to the violation of joint feasibility constraints. The same scenario is observed for other initial points.

The results of ARES and ARES-BR are also illustrated in Figure 1. As mentioned above, ARES-BR does not reach the GNE. To address this problem, ARES uses a trust-region strategy that lets the algorithm find feasible improvements. For the same initial point, ARES reached the stopping tolerance to the GNE $(-1, 1)$. One reason the trust-region approach works better for ARES is that it shrinks the trust-region radius, making the strategy feasible for subsequent steps.

Example 3 In this example, we compare ARES-BR against three benchmarks in challenging scenarios involving cyclic behavior: (i) classical BRD, using the results from [Thiran et al. \(2024\)](#); (ii) the Gauss-Seidel Method (GSM), which we implement directly; and (iii) ARES-BR.

Let us consider a two-player GNEP $P^{c,c}$ parameterized by $0 < c < 1.5$ is considered [Thiran et al. \(2024\)](#):



(a) Iterative trajectory of the ARES converging to the GNE $(x_1^*, x_2^*) = (-1, 1)$. (b) Iterative trajectory of the ARES-BR not converging to the GNE.

Fig. 1: Comparison of the ARES and ARES-BR, with $\varepsilon = 10^{-4}$.

$$\begin{aligned}
P_1^c(x_2) : \quad & \min_{x_1} \theta_1(x_1, x_2) = (x_1 - cx_2 + 1)^2 & P_2^c(x_1) : \quad & \min_{x_2} \theta_2(x_1, x_2) = (x_2 + cx_1)^2 \\
\text{s.t.} \quad & x_1 \in L_1 = [-2, 2] & \text{s.t.} \quad & x_2 \in L_2 = [1 - 2c, 3 + c], \\
& & & X_2(x_1) = \left[1 - cx_1, 3 - c\frac{x_1}{2} \right].
\end{aligned}$$

Thiran et al. (2024) found that the GNEP has a unique GNE given by $x^E = \left(\frac{c-1}{c^2+1}, \frac{c+1}{c^2+1} \right)$.

We evaluate the algorithmic behavior across different parameter settings $c \in \{0.8, 1.0\}$, with the comparative results detailed in Table 1. In particular, when $c = 0.8$ or $c = 1$, ARES-BR fails to reach a GNE. Conversely, as shown in Table 1, ARES successfully bypasses these limitations and converges to the exact GNE solutions.

Table 1: Comparison of BRD, GSM, and ARES for different values of c .

c	BRD/GSM behavior	ARES
$c = 0.8$	GSM converges to $(-0.1219, 1.0975)$.	Converges to $x^* = (-0.1239, 1.0991)$ with $f(x^*) = 0.0$.
$c = 1$	BRD and GSM become trapped in cycles. For GSM, the iterates follow the cycle $x^0 = (-1.0876, 1.4730) \rightarrow x^1 = (0.4730, 0.5269) \rightarrow x^2 = (-0.4730, 1.4730) \rightarrow x^3 = (0.4730, 0.5269)$.	Converges to $x^* = (0.0000, 1.0000)$ with $f(x^*) = 0.0$.

From the examples 2 and 3, it is clear that ARES can perform better than ARES-BR. This observation motivates the use of ARES-BR as a baseline in the numerical experiments, against which we assess the performance of the proposed randomized approach.

In the next section, where we present extensive numerical experiments on both benchmark and real-world GNEPs.

4 Numerical Results

In this section, we evaluate ARES on academic jointly-convex and non-shared-constraint GNEPs, as well as on several application-inspired jointly-convex models. The analysis is structured as follows. First, we describe ARES's performance on jointly-convex GNEPs to validate our approach under theoretically favorable conditions. Then, we evaluate the practicality of ARES on real-world jointly-convex GNEPs, including autonomous racing, internet switching, and oligopoly model. Finally, we consider GNEPs with non-shared constraints, where unilateral player updates may make the joint strategy globally infeasible.

Throughout this section, we compare ARES with ARES-BR, which serves as the baseline for assessing the effect of randomized partial moves. Unless otherwise stated, both methods are evaluated using the same initialization procedure and stopping criteria.

4.1 Performance Metrics

We assess the performance of ARES using convergence and solution-diversity measures. Convergence is evaluated using the iteration count k^* , final fitness, and success rate (SR), while solution diversity is assessed using scatter plots and the percentage of distinct points (PDP).

Denote $k^*(r)$ the first iteration during run r at which the convergence tolerance ε is satisfied, i.e. $k^*(r) := \min \{k \in \{1, \dots, K\} : f^k \leq \varepsilon\}$, where f^k denotes the fitness value at iteration k during run r . If the above condition is not satisfied, then $k^*(r) = K$. Runs with $k^* = K$ are classified as unsuccessful, although their final fitness values are retained for statistical analysis.

Definition 2 (Success rate) The success rate (SR) is defined as

$$\text{SR} := \frac{\#\{r : k^*(r) < K\}}{R},$$

where $k^*(r)$ is the iteration count to convergence for the r -th run ($r = 1, \dots, R$).

Definition 3 (Final fitness) Let $\mathcal{A}$ be our algorithm that generates a sequence of fitness values $\{f_k\}_{k=1}^{k^*}$. For a run r , the *final fitness* is defined as $f_{\text{final}}^{(r)} := f^{k^*}$. Therefore, the average final fitness is given by

$$\bar{f}_{\text{final}} = \frac{1}{R} \sum_{r=1}^R f_{\text{final}}^{(r)}.$$

The final fitness is the fitness value at termination, either when the stopping tolerance is first reached or when the maximum number of iterations is attained. Unsuccessful runs are included in the final-fitness statistics to characterize the residual attained when the method fails to meet the prescribed tolerance. Additionally, for each GNEP, we report the maximum final fitness value.

Definition 4 (Percentage of distinct points) Let $\mathcal{X} = \{x^1, x^2, \dots, x^{R_s}\}$ be the set of approximate GNEs obtained from R_s successful runs of an algorithm. Two points x^i and x^j are considered distinct if $\|x^i - x^j\| > \gamma$, where $\gamma > 0$ is a predefined tolerance. The set of distinct points identified from $\mathcal{X}$ is denoted by $\mathcal{X}_d$, and its cardinality by $N_d := |\mathcal{X}_d|$. The percentage of distinct points (PDP) is defined as

$$\text{PDP} := \frac{N_d}{R_s} \times 100.$$

A higher PDP indicates that a larger proportion of successful runs produce mutually distinct approximate GNEs and is therefore used as an empirical indicator of solution diversity.

4.2 Experimental Protocol

For all experiments, the algorithm settings are $K_{\max} = 500$, $\rho_0 = \rho_1 = 0.75$, $M = 200$, $\delta_0 = 25.0$, and $\delta_{\max} = 50$. The convergence tolerance is set to $\varepsilon = 10^{-4}$, and the distinct point criterion to $\gamma = 10^{-4}$.

For all academic GNEPs, we uniformly generate starting points from the players' feasible strategy sets until 300 jointly feasible starting points are obtained. We run ARES independently five times from each starting point, resulting in 1,500 runs per problem. Since ARES-BR is deterministic for a fixed initial point, repeating the run from the same initialization yields the same trajectory and is therefore unnecessary. For the real-world GNEPs, we run each algorithm once for each starting point, with the number of starting points specified in the corresponding subsections.

We use the collection of successful runs to construct the set of computed approximate GNEs, from which we evaluate the convergence and diversity metrics.

The experiments were implemented in Python 3.13.9 and conducted on a machine equipped with an Apple M5 processor and 24 GB of RAM.

For each example, we compute the best response sequentially for players using the Sequential Least Squares Programming (SLSQP) algorithm implemented in SciPy's `optimize.minimize` routine.

4.3 Academic Jointly-Convex GNEPs

We first evaluate ARES on the four academic jointly-convex GNEPs listed in Table 2. The results are reported in Table 3, for which reference equilibrium information is available.

These results show that ARES competes with ARES-BR in solving jointly-convex GNEPs. ARES achieves a 100% SR in Ex. 1, Ex. 3, and Ex. 4, matching the ARES-BR in these examples. While ARES gets a lower SR in Ex. 2 (51.9% vs 100%), it finds

Table 2: Academic jointly-convex GNEPs.

Problems	GNEs set
Ex. 1 Nabetani et al. (2011)	
$P_1 : \min_x x^2 - xy - x$ s.t.: $x^2 + y^2 \leq 1, x \geq 0$ $P_2 : \min_y y^2 - \frac{1}{2}xy - 2y$ s.t.: $x^2 + y^2 \leq 1, y \geq 0$	$\left\{ \left(\frac{t}{\sqrt{1-t^2}} \right), 0 \leq t \leq \frac{4}{5} \right\}$
Ex. 2 Harker (1991)	
$P_1(x_2) : \min_{x_1} x_1^2 + \frac{8}{3}x_1x_2 - 34x_1$ s.t.: $x_1 + x_2 \leq 15, 0 \leq x_1 \leq 10$ $P_2(x_1) : \min_{x_2} x_2^2 + \frac{5}{4}x_1x_2 - 24.25x_2$ s.t.: $x_1 + x_2 \leq 15, 0 \leq x_2 \leq 10$	$\left\{ \binom{5}{9} \cup \binom{t}{15-t}, 9 \leq t \leq 10 \right\}$
Ex. 3 Wild (2017)	
$P_1 : \min_{x,y} x^2 + xy + y^2 + (x+y)z - 25x - 38y$ s.t.: $x + 2y - z \leq 14, 3x + 2y + z \leq 30, x, y \geq 0$ $P_2 : \min_z z^2 + (x+y)z - 25z$ s.t.: $x + 2y - z \leq 14, 3x + 2y + z \leq 30, z \geq 0$	$\left\{ \binom{t}{11-t}, 0 \leq t \leq 2 \right\}$
Ex. 4 Krawczyk and Uryasev (2000)	
$P_1 : \min_{x_1} (\alpha_1 x_1 + \beta(x_1 + x_2 + x_3) - \chi_1)x_1$ s.t.: $3.25x_1 + 1.25x_2 + 4.125x_3 \leq 100$ $2.29115x_1 + 1.5625x_2 + 2.8125x_3 \leq 100, x_1 \geq 0$ $P_2 : \min_{x_2} (\alpha_2 x_2 + \beta(x_1 + x_2 + x_3) - \chi_2)x_2$ s.t.: $3.25x_1 + 1.25x_2 + 4.125x_3 \leq 100$ $2.29115x_1 + 1.5625x_2 + 2.8125x_3 \leq 100, x_2 \geq 0$ $P_3 : \min_{x_3} (\alpha_3 x_3 + \beta(x_1 + x_2 + x_3) - \chi_3)x_3$ s.t.: $3.25x_1 + 1.25x_2 + 4.125x_3 \leq 100$ $2.29115x_1 + 1.5625x_2 + 2.8125x_3 \leq 100, x_3 \geq 0$ $\alpha_1 = 0.001, \alpha_2 = 0.05, \alpha_3 = 0.01$ $\beta = 0.01, \chi_1 = 2.9, \chi_2 = 2.88, \chi_3 = 2.85$	

Table 3: Performance comparison of ARES and ARES-BR based on 1,500 and 300 independent runs, respectively.

Example	ARES		ARES-BR	
	PDP	SR (%)	PDP	SR (%)
Ex. 1	60.8	100.0	96.3	100.0
Ex. 2	86.7	52.3	43.6	100.0
Ex. 3	90.0	100.0	67.3	100.0
Ex. 4	98.6	100.0	100.0	100.0

a more diverse set of GNEs, with a PDP of 87.0 vs 43.6 for ARES-BR. This lower SR in Ex. 2 suggests a trade-off between solution robustness and solution diversity: although ARES does not find an equilibrium in every run, the equilibria it identifies

cover a more diverse set of solutions. ARES also achieves a higher PDP in Ex. 3, meaning it can obtain a more diverse set of GNEs.

Overall, these results show that when ARES with best-response direction behaves well, ARES can generally find equilibria too, but its randomization introduces a trade-off between robustness and solution exploration. In particular, while the ARES best-response method can converge more consistently in some cases, ARES can identify a more diverse set of GNEs through its randomized search.

4.4 Real-world jointly-convex GNEPs

For real-world jointly-convex GNEPs, both algorithms consistently obtain GNEs across all applications, though ARES fails in 2 out of 10 autonomous racing runs. However, while ARES with the best-response direction identifies only a single GNE in autonomous racing, ARES with best-response direction discovers some distinct solutions.

4.4.1 Internet switching model

We consider the N -user internet switching proposed by Alex Kesselman [Kesselman et al. \(2005\)](#) with a buffer capacity of B . The goal of each user is to determine the amount of buffer capacity allocated to user ν by maximizing the utility:

$$\theta_\nu(x) = \begin{cases} \frac{x_\nu}{\sum_{\mu=1}^N x_\mu} \left(1 - \frac{\sum_{\mu=1}^N x_\mu}{B} \right), & \sum_{\mu=1}^N x_\mu > 0, \\ 0, & \sum_{\mu=1}^N x_\mu = 0, \end{cases}$$

Adding shared buffer capacity constraints and private bound constraints, Player's ν problem is

$$\begin{aligned} \min_{x_\nu \in \mathbb{R}} \quad & -\theta_\nu(x) \\ \text{s.t.} \quad & x_\nu \geq 0, \quad \nu = 2, \dots, N, \\ & \sum_{\nu=1}^N x_\nu \leq 1, \end{aligned}$$

where $0 \leq x_1 \leq \frac{1}{2}$ is the private bound constraint for Player 1. To run our proposed algorithms, we initialize 10 feasible starting points and run the algorithms independently from each starting point.

In this case, we have $N = 5$ users and $B = 1$, as used in [Wang and Jia \(2025\)](#). This instance was considered in [Wang and Jia \(2025\)](#), where SMA and CWSMA were shown to converge to the GNE $x^* = (0.16, \dots, 0.16)$. We use it here to assess whether ARES recovers the same equilibrium structure. Table 4 shows that both methods converge to a unique GNE that is also discovered in both methods converge from all tested initial points to the same approximate GNE, consistent with the equilibrium

reported in Wang and Jia (2025). The computed GNEs are close to the reference equilibrium in both cases.

Table 4: Summary of convergence performance for the internet switching problem over 10 independent runs initialized from different starting points.

Method	SR (%)	Mean k^*	$\max_{r=1,\dots,10} \ x^{(r,\star)} - x^*\ _2$
ARES	100.0	13	0.00714
ARES-BR	100.0	12	0.00700

We omit the fitness value from the table, as it is consistently close to zero across all instances for both methods.

4.4.2 Oligopoly model

This problem is a 5-player oligopoly model Facchinei and Kanzow (2009); Dreves and Kanzow (2011).

$$\begin{aligned} P_\nu(x_{-\nu}) : \quad & \min_{x_\nu} \theta_\nu(x_\nu, x_{-\nu}) = f_\nu(x_\nu) - 5000^{1/\mu} x_\nu (x_1 + \dots + x_N)^{-1/\mu} \\ \text{s.t.} \quad & x_1 + \dots + x_N \leq P, \quad x_\nu \geq 0 \end{aligned}$$

with, $f_\nu(x_\nu) := c_\nu x_\nu + \frac{\eta_\nu}{1 + \eta_\nu} K_\nu^{-1/\eta_\nu} (x_\nu)^{\frac{(1 + \eta_\nu)}{\eta_\nu}}$ for $\nu = 1, \dots, N$. The parameter P represents the upper bound for the total production activity for all players, $P \in \{75, 100, 150, 200\}$, and $\mu = 1.1$. The remaining parameters are given in Table 5.

Table 5: Oligopoly model parameters

ν	1	2	3	4	5
c_ν	10	8	6	4	2
K_ν	5	5	5	5	5
η_ν	1.2	1.1	1.0	0.9	0.8

For each value of P , we run ARES on two starting points:

$$x^{1,0} = (10.0, 10.0, 10.0, 10.0, 10.0) \quad \text{and} \quad x^{2,0} = (0.0, 5.0, 10.0, 15.0, 20.0)$$

Results in Table 6 show that both the ARES and ARES-BR find GNEs. The different GNEs found by the two algorithms demonstrate that both can find distinct GNEs. Furthermore, the GNEs found by both ARES and ARES-BR make the common constraint active (see Dreves and Kanzow (2011)).

Table 6: Comparison of convergence results for the oligopoly model using ARES and ARES-BR.

P, x^0	ARES	k^*	ARES-BR	k^*
	GNE		GNE	
75, $x^{1,0}$	(9.99, 9.96, 34.98, 9.95, 10.09)	17	(10.00, 10.00, 10.00, 35.00, 10.00)	2
75, $x^{2,0}$	(0.02, 4.97, 10.03, 39.96, 19.99)	22	(25.00, 5.00, 10.00, 15.00, 20.00)	2
100, $x^{1,0}$	(9.99, 34.93, 10.06, 10.01, 34.99)	20	(10.00, 56.19, 10.00, 10.00, 13.81)	3
100, $x^{2,0}$	(0.00, 54.98, 10.00, 15.02, 19.98)	18	(50.00, 5.00, 10.00, 15.00, 20.00)	2
150, $x^{1,0}$	(35.02, 35.21, 34.81, 10.01, 34.92)	21	(20.00, 52.66, 52.59, 14.20, 10.56)	13
150, $x^{2,0}$	(24.98, 29.87, 35.00, 40.15, 19.99)	18	(50.32, 49.55, 10.12, 15.71, 24.30)	9
200, $x^{1,0}$	(34.90, 42.74, 44.44, 40.97, 36.93)	24	(37.89, 42.75, 44.47, 43.25, 31.64)	25
200, $x^{2,0}$	(38.10, 41.98, 37.56, 42.73, 39.64)	26	(38.08, 42.75, 44.45, 43.18, 31.54)	28

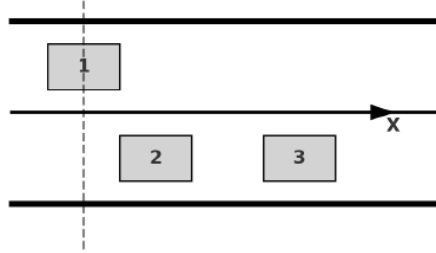


Fig. 2: Illustration of the 1D racing example.

4.4.3 Autonomous racing

Autonomous racing provides an application-inspired benchmark involving strategic interaction and collision-avoidance constraints. This problem is described in [Pustilnik and Borrelli \(2025\)](#); [Pustilnik et al. \(2025\)](#).

Consider a single-step, finite-horizon, discrete-time, general-sum, open-loop dynamic game in a racing scenario involving three cars on a two-lane track. Each car acts as a player in the GNEP, controlling its velocity along a one-dimensional lane. Car 1 occupies the first lane, while Cars 2 and 3 share the second lane; see Figure 2. Let x_ν and v_ν denote the final position and velocity of car ν , respectively. The 3-player game is given by

$$\begin{aligned}
 \text{Player 1: } & \min_{x_1, v_1} -x_1 + x_2 + \frac{1}{2}v_1^2, \quad \text{s.t. } x_1 = x_1(0) + \Delta t \cdot v_1, \quad x_2 \leq x_3, \quad x_1(0) = 0 \\
 \text{Player 2: } & \min_{x_2, v_2} -x_2 + x_1 + \frac{1}{2}v_2^2, \quad \text{s.t. } x_2 = x_2(0) + \Delta t \cdot v_2, \quad x_2 \leq x_3, \quad x_2(0) = 0.5 \\
 \text{Player 3: } & \min_{x_3, v_3} -x_1 + x_2 + \frac{1}{2}v_3^2, \quad \text{s.t. } x_3 = x_3(0) + \Delta t \cdot v_3, \quad x_2 \leq x_3, \quad x_3(0) = 0.75
 \end{aligned}$$

In this game, the three vehicles have the following goals: car 1 aims to outdistance car 2 and minimize its control effort; car 2 aims to outdistance car 1 and minimize its control effort. Car 3, on the other hand, tries to help car 1 win the race while minimizing its own control effort. Collision avoidance between cars 2 and 3 gives car 3 an opportunity to affect the motion of car 2 in its favor.

Table 7: Performance of ARES on the autonomous racing problem over 10 independent runs initialized from the same starting point.

Run	GNE	k^*
1	(1.000, 0.764, 0.764)	46
2	No solution	500
3	(0.999, 1.040, 1.040)	34
4	(1.000, 1.379, 1.379)	33
5	(0.999, 0.782, 0.782)	37
6	(1.000, 0.749, 0.749)	31
7	(0.999, 0.751, 0.751)	33
8	(1.000, 0.896, 0.896)	33
9	No solution	500
10	(1.000, 0.789, 0.789)	35

The initial positions and velocities are set as $x^0 = (x_1^0, x_2^0, x_3^0) = (0.0, 0.5, 0.75)$ and $v^0 = (v_1^0, v_2^0, v_3^0) = (1.0, 1.0, 0.0)$, respectively.

The variational GNE of this game is (1, 1.125, 1.125); see [Pustilnik et al. \(2025\)](#) for details. Surprisingly, this GNE exhibits counterintuitive behavior: rather than blocking car 2, car 3 starts moving, even though this action increases its cost.

From Table 7, it can be seen that ARES finds different GNEs across runs from the same initial point, indicating that the randomized search can explore different regions of the solution space. For example, one of the GNEs found by the algorithm is (1.000, 1.379, 1.379). In contrast, ARES with best-response direction converges to the same equilibrium in every run from this initialization, namely (0.990, 0.750, 0.750). This illustrates that the random actions in ARES enable exploration of multiple equilibrium solutions from an identical initialization, resulting in greater diversity among the GNEs discovered.

4.5 GNEPs with non-shared constraints

We next evaluate ARES on ten GNEPs with non-shared constraints described in Table 8. As discussed earlier, unilateral player updates may violate the constraints of other players, making direct best-response updates particularly challenging. The results can be found in Table 9.

As shown in Table 9, ARES achieves a higher SR than ARES-BR on seven of the ten test problems and a higher PDP on six of the ten test problems. In particular, ARES-BR fails to find any GNE in Ex. 4 and Ex. 5 (SR=0%) and performs poorly on Ex. 6 (SR=28.3%). In contrast, ARES achieves SRs of 99.8%, 47.2%, and 100.0% on these three examples, respectively. The largest differences occur in Examples 4–6,

Table 8: General (non-shared constraints) GNEPs.

Problems	GNEs set
Ex. 1 $P_1 : \min_{(x,y) \in \mathbb{R}^2} \frac{1}{2}(x-1)^2 + \frac{1}{2}(y-1)^2$ s.t.: $x + y + z \leq 3, x \geq 0.1, y \geq 0.1$ $P_2 : \min_{z \in \mathbb{R}} \frac{1}{2}xyz$ s.t.: $x^2 z^2 \geq \frac{1}{2}, y^2 z^2 \geq \frac{1}{2}, z \geq 0.1$	$\left\{ \left(\alpha, \alpha, \frac{1}{\alpha\sqrt{2}} \right) : \alpha = 1, 1 - \frac{1}{\sqrt{2}} \right\}$
Ex. 2 Migot and Cojocaru (2021) $P_1 : \min_{x \in \mathbb{R}} (x-1)^2$ s.t.: $x^2 + y^2 \leq 1, x + y \leq 1$ $P_2 : \min_{y \in \mathbb{R}} (y - \frac{1}{2})^2$ s.t.: $x + y \leq 1$	$\left\{ (t, 1-t) : 0.5 \leq t \leq 1 \right\}$
Ex. 3 $P_1 : \min_{(x_1, x_2) \in \mathbb{R}^2} x_1 + x_2$ s.t.: $x_1 + x_2 \geq x_3, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1$ $P_2 : \min_{x_3 \in \mathbb{R}} (x_3 - 1)^2$	$\left\{ (t, 1-t, 1) : 0 \leq t \leq 1 \right\}$
Ex. 4 Facchinei and Kanzow (2010a) $P_1 : \min_{x_1 \in \mathbb{R}} \frac{1}{2}(x_1 - 1)^2 - x_1 x_2$ s.t.: $x_1 + x_2 + x_3 \leq 1$ $P_2 : \min_{x_2 \in \mathbb{R}} \frac{1}{2}(x_2 - 1)^2 + x_1 x_2$ s.t.: $x_1 + x_2 + x_3 \leq 1$ $P_3 : \min_{x_3 \in \mathbb{R}} \frac{1}{2}(x_3 - 1)^2$ s.t.: $0 \leq x_3 \leq x_1 + x_2$	$\left\{ \left(\alpha, \frac{1}{2} - \alpha, \frac{1}{2} \right) : \alpha \leq \frac{3}{4} \right\}$
Ex. 5 $P_1 : \min_{x_1 \in \mathbb{R}} \frac{1}{5}x_1^2 + \frac{32}{5}x_1$ s.t.: $\frac{1}{6}x_1^2 + x_2 - \frac{5}{2} \leq 0$ $P_2 : \min_{x_2 \in \mathbb{R}} \frac{1}{2}x_2^2 + x_1 x_2 - \frac{4}{5}x_2$	$x^* = (-1.3817, 2.1817)$
Ex. 6 $P_1 : \min_{x_1 \in \mathbb{R}} x_1$ s.t.: $x_1^2 + x_2 \leq 1$ $P_2 : \min_{x_2 \in \mathbb{R}} \frac{1}{2}x_2^2$ s.t.: $x_2 \in \mathbb{R}$	$x^* = (-1.0, 0.0)$
Ex. 7 Dreves et al. (2012) $P_1 : \min_{x_1 \in \mathbb{R}} (x_1 - 2)^2$ s.t.: $x_1 + x_2 \leq 1$ $P_2 : \min_{x_2 \in \mathbb{R}} \frac{1}{2}(x_2 - 2)^2$ s.t.: $x_1 + x_2 \leq 1, x_2 - x_1 \leq 0$	$\left\{ (\lambda, 1-\lambda) : \lambda \in \left[\frac{1}{2}, 2 \right] \right\}$
Ex. 8 Dreves et al. (2012) $P_1 : \min_{x_1 \in \mathbb{R}} \frac{1}{2}x_1^2 + 3x_1 x_2$ s.t.: $x_1 - 3x_2 \leq 2, -3x_1 + x_2 \leq 2, x_1 + x_2 \leq 1$ $P_2 : \min_{x_2 \in \mathbb{R}} \frac{1}{2}x_2^2 + 3x_1 x_2$ s.t.: $x_1 - 3x_2 \leq 2, -3x_1 + x_2 \leq 2$	$\left\{ (0, 0), (-1, -1) \right\}$ $\bigcup \left\{ \left(\lambda, -\frac{2}{3} + \frac{1}{3}\lambda \right) : \lambda \in \left[\frac{1}{5}, 1 \right] \right\}$ $\bigcup \left\{ (\lambda, 2 + 3\lambda) : \lambda \in \left[-\frac{3}{5}, -\frac{1}{3} \right] \right\}$
Ex. 9 Dreves et al. (2012) $P_1 : \min_{x_1 \in \mathbb{R}} (x_1 - 1)^2$ s.t.: $x_1^2 - x_2^2 \leq 0$ $P_2 : \min_{x_2 \in \mathbb{R}} (x_2 - x_1)^2$ s.t.: $x_1^2 + x_2^2 \leq 1$	$\left\{ (\lambda, \lambda) : \lambda \in \left[0, \frac{1}{\sqrt{2}} \right] \right\}$
Ex. 10 Dreves et al. (2012) $P_1 : \min_{x_1 \in \mathbb{R}} (x_1 - 1)^2$ s.t.: $x_1^2 - x_2^2 \leq 0, x_1^2 + x_2^2 \leq 1$ $P_2 : \min_{x_2 \in \mathbb{R}} (x_2 - \frac{1}{2}x_1 - 2)^2$ s.t.: $x_1^2 + x_2^2 \leq 1$	$\left\{ (\lambda, \sqrt{1-\lambda^2}) : \lambda \in \left[0, \frac{1}{\sqrt{2}} \right] \right\}$

Table 9: Performance comparison of ARES and ARES-BR based on 1,500 and 300 independent runs, respectively.

Example	ARES		ARES-BR	
	PDP	SR (%)	PDP	SR (%)
Ex. 1	76.9	99.4	4.2	95.0
Ex. 2	79.2	100.0	62.0	100.0
Ex. 3	96.4	100.0	97.4	91.7
Ex. 4	91.6	99.8	—	0.0
Ex. 5	16.2	47.2	—	0.0
Ex. 6	13.9	100.0	2.3	28.3
Ex. 7	72.8	100.0	14.0	100.0
Ex. 8	93.4	74.2	99.2	75.6
Ex. 9	92.1	100.0	97.1	59.3
Ex. 10	92.4	50.6	94.4	48.3

where the ARES with best-response frequently fails to produce an acceptable update. This is consistent with the feasibility difficulty illustrated in Subsection 3.3 and suggests that randomized partial moves can substantially improve robustness when a full best-response step interacts poorly with non-shared constraints. Example 7 isolates the effect on solution exploration: both algorithms converge in every run, but ARES obtains a PDP of 72.8%, compared with 14.0% for the ARES-BR.

5 Conclusions and Future Work

This paper introduces the novel Adaptive Randomized Equilibrium Search (ARES), a simple and practical local search heuristic for GNEPs. ARES uses best-response information to generate randomized partial moves, combined with Gauss–Southwell-inspired player selection and an adaptive search radius. Computational results on academic GNEPs and practical jointly-convex GNEPs demonstrate the capability of ARES to find GNEs in different problems.

The numerical experiments illustrate different effects of this randomization. On jointly-convex GNEPs, both ARES and ARES with best-response direction (ARES-BR) generally identify GNEs, and ARES-BR can be more efficient or more reliable when they already behave well. ARES nevertheless provides additional exploration in some problems; in particular, repeated ARES runs from the same initialization in the autonomous-racing example identify different GNEs, whereas the deterministic variant follows a single trajectory. The strongest benefit is observed for GNEPs with non-shared constraints: ARES achieves higher success rates than ARES-BR on seven of the ten test examples, substantially improves robustness on several problems, and maintains high solution diversity. These results indicate that randomized partial best-response moves are particularly useful when full best-response updates interact poorly with coupled feasibility constraints.

The illustrative examples further show that partial randomized updates can avoid cycling or stagnation encountered by classical best-response or Gauss–Seidel updates on selected games. However, the experiments also indicate that randomization is not

uniformly advantageous, motivating further investigation of when partial versus full best-response steps should be preferred.

Future work includes studying the scalability of ARES to larger GNEPs, adaptive selection of the sampling and search-radius parameters, and alternative player-selection mechanisms. Extending the framework to nonsmooth, piecewise-linear, or black-box games is another promising direction. In settings where exact best-response information is difficult to obtain, surrogate or learning-based models could also be used to approximate the required best-response information.

Acknowledgements. Funding for this work was provided by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (No. 400486) awarded to the third author.

Declarations

- Funding for this works was provided by a Natural Sciences and Engineering Research Council Discovery Grant 400486 of the third author.
- Conflict of interest/Competing interests: The authors have no relevant financial or non-financial interests to disclose.
- Data availability: There is no data associated with this manuscript.
- Code availability: The code used for all algorithms and experiments is publicly available at <https://github.com/benjaminbenteke/AdaptiveRandomizedEquilibriumSearch>.

References

- Amiet, B., Collevocchio, A., Hamza, K.: When “better” is better than “best”. *Operations Research Letters* **49**(2), 260–264 (2021)
- Arrow, K.J., Debreu, G.: Existence of an equilibrium for a competitive economy. *Econometrica* **22**(3), 265–290 (1954) <https://doi.org/10.2307/1907353>
- Benteke, B., Cojocaru, M.G., Fields, R., Nica, M., Tarasuk, K.: Two heuristic methods for solving generalized nash equilibrium problems using a novel penalty function. In: *Mathematical Analysis, Differential Equations and Applications*, pp. 111–138. World Scientific, ??? (2024)
- Benenati, E., Grammatico, S.: Probabilistic game-theoretic traffic routing. *IEEE Transactions on Intelligent Transportation Systems* **25**(10), 13080–13090 (2024)
- Bassanini, A., La Bella, A., Nastasi, A.: Allocation of railroad capacity under competition: a game theoretic approach to track time pricing. In: *Transportation and Network Analysis: Current Trends: Miscellanea in Honor of Michael Florian*, pp. 1–17. Springer, ??? (2002)
- Benteke Longaou, B., Cojocaru, M.-G., Hoover, N.: Reinforcement learning for solving generalized nash equilibrium problem. In: *International Conference on Modelling,*

- Computation and Optimization in Information Systems and Management Sciences, pp. 296–308 (2025). Springer
- Ba, Q., Pang, J.-S.: Exact penalization of generalized nash equilibrium problems. *Operations Research* **70**(3), 1448–1464 (2022)
- Breton, M., Zaccour, G., Zahaf, M.: A game-theoretic formulation of joint implementation of environmental projects. *European Journal of Operational Research* **168**(1), 221–239 (2006)
- Cardellini, V., De Nitto Personé, V., Di Valerio, V., Facchinei, F., Grassi, V., Lo Presti, F., Piccialli, V.: A game-theoretic approach to computation offloading in mobile cloud computing. *Mathematical Programming* **157**(2), 421–449 (2016)
- Ceppi, S., Gatti, N., Patrini, G., Rocco, M.: Local search methods for finding a nash equilibrium in two-player games. In: 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 2, pp. 335–342 (2010). IEEE
- Contreras, J., Klusch, M., Krawczyk, J.B.: Numerical solutions to nash-cournot equilibria in coupled constraint electricity markets. *IEEE Transactions on Power Systems* **19**(1), 195–206 (2004)
- Cojocar, M.-G., Wild, E., Small, A.: On describing the solution sets of generalized nash games with shared constraints. *Optimization and Engineering* **19**, 845–870 (2018)
- Debreu, G.: A social equilibrium existence theorem. *Proceedings of the national academy of sciences* **38**(10), 886–893 (1952)
- Dreves, A., Gerdt, M.: A generalized nash equilibrium approach for optimal control problems of autonomous cars. *Optimal Control Applications and Methods* **39**(1), 326–342 (2018)
- Dreves, A., Kanzow, C.: Nonsmooth optimization reformulations characterizing all solutions of jointly convex generalized nash equilibrium problems. *Computational Optimization and Applications* **50**(1), 23–48 (2011)
- Dreves, A., Kanzow, C., Stein, O.: Nonsmooth optimization reformulations of player convex generalized nash equilibrium problems. *Journal of Global Optimization* **53**(4), 587–614 (2012)
- Dindoš, M., Mezzetti, C.: Better-reply dynamics and global convergence to nash equilibrium in aggregative games. *Games and Economic Behavior* **54**(2), 261–292 (2006)
- Dreves, A.: Globally convergent algorithms for the solution of generalized nash

- equilibrium problems. PhD thesis, Universität Würzburg (2011)
- Dreves, A.: Finding all solutions of affine generalized nash equilibrium problems with one-dimensional strategy sets. *Mathematical Methods of Operations Research* **80**(2), 139–159 (2014)
- Dreves, A.: Computing all solutions of linear generalized nash equilibrium problems. *Mathematical Methods of Operations Research* **85**(2), 207–221 (2017)
- Facchinei, F., Fischer, A., Piccialli, V.: On generalized nash games and variational inequalities. *Operations Research Letters* **35**(2), 159–164 (2007)
- Fischer, A., Herrich, M., Schönefeld, K.: Generalized nash equilibrium problems-recent advances and challenges. *Pesquisa Operacional* **34**, 521–558 (2014)
- Fields, R.: Two heuristic methods for solving generalized nash equilibrium problems using a novel penalty function. PhD thesis, University of Guelph (August 2021)
- Fabrikant, A., Jaggard, A.D., Schapira, M.: On the structure of weakly acyclic games. In: *International Symposium on Algorithmic Game Theory*, pp. 126–137 (2010). Springer
- Facchinei, F., Kanzow, C.: Penalty methods for the solution of generalized nash equilibrium problems (with complete test problems). *Sapienza University of Rome*, 15 (2009)
- Facchinei, F., Kanzow, C.: Generalized nash equilibrium problems. *Annals of Operations Research* **175**(1), 177–211 (2010)
- Facchinei, F., Kanzow, C.: Penalty methods for the solution of generalized nash equilibrium problems. *SIAM Journal on Optimization* **20**(5), 2228–2253 (2010)
- Facchinei, F., Lampariello, L.: Partial penalization for the solution of generalized nash equilibrium problems. *Journal of Global Optimization* **50**, 39–57 (2011)
- Friedman, J.W., Mezzetti, C.: Learning in games by random sampling. *Journal of Economic Theory* **98**(1), 55–84 (2001)
- Facchinei, F., Piccialli, V., Sciandrone, M.: Decomposition algorithms for generalized potential games. *Computational Optimization and Applications* **50**, 237–262 (2011)
- Fukushima, M.: Restricted generalized nash equilibria and controlled penalty algorithm. *Computational Management Science* **8**, 201–218 (2011)
- Graf, C., Zobernig, V., Schmidt, J., Kloeckl, C.: Computational performance of deep reinforcement learning to find nash equilibria. *Computational Economics* **63**(2), 529–576 (2024)

- Harker, P.T.: Generalized nash games and quasi-variational inequalities. *European journal of Operational research* **54**(1), 81–94 (1991)
- Hall, S., Guerrini, L., Dörfler, F., Liao-McPherson, D.: Receding horizon games for modeling competitive supply chains. *IFAC-PapersOnLine* **58**(18), 8–14 (2024)
- Hobbs, B.F., Pang, J.-S.: Nash-cournot equilibria in electric power markets with piecewise linear demand functions and joint constraints. *Operations Research* **55**(1), 113–127 (2007)
- Jing-Yuan, W., Smeers, Y.: Spatial oligopolistic electricity models with cournot generators and regulated transmission prices. *Operations Research* **47**(1), 102–112 (1999)
- Kim, J.G.: Equilibrium computation of generalized nash games: A new lagrangian-based approach. *arXiv preprint arXiv:2106.00109* (2021)
- Kesselman, A., Leonardi, S., Bonifaci, V.: Game-theoretic analysis of internet switching with selfish users. In: *International Workshop on Internet and Network Economics*, pp. 236–245 (2005). Springer
- Kelly, F.P., Maulloo, A.K., Tan, D.K.H.: Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research society* **49**(3), 237–252 (1998)
- Krawczyk, J.B.: Coupled constraint nash equilibria in environmental games. *Resource and Energy Economics* **27**(2), 157–181 (2005)
- Kulkarni, A.A., Shanbhag, U.V.: On the variational equilibrium as a refinement of the generalized nash equilibrium. *Automatica* **48**(1), 45–55 (2012)
- Kanzow, C., Steck, D.: Augmented lagrangian methods for the solution of generalized nash equilibrium problems. *SIAM Journal on Optimization* **26**(4), 2034–2058 (2016)
- Kanzow, C., Steck, D.: Augmented lagrangian and exact penalty methods for quasi-variational inequalities. *Computational Optimization and Applications* **69**, 801–824 (2018)
- Krawczyk, J.B., Uryasev, S.: Relaxation algorithms to find nash equilibria with economic applications. *Environmental Modeling & Assessment* **5**(1), 63–73 (2000)
- Le Cadre, H., Jacquot, P., Wan, C., Alasseur, C.: Peer-to-peer electricity market analysis: From variational to generalized nash equilibrium. *European Journal of Operational Research* **282**(2), 753–771 (2020)
- Le Cleac’h, S., Schwager, M., Manchester, Z.: Algames: a fast augmented lagrangian solver for constrained dynamic games. *Autonomous Robots* **46**(1), 201–215 (2022)

- Liu, L., Jia, W.: A new algorithm to solve the generalized nash equilibrium problem. *Mathematical Problems in Engineering* **2020**(1), 1073412 (2020)
- Lisboa, A.C., Santos, F.F., Vieira, D.A., Saldanha, R.R., Pereira, F.A.: An enhanced gradient algorithm for computing generalized nash equilibrium applied to electricity market games. *Energies* **18**(3), 727 (2025)
- Migot, T., Cojocaru, M.-G.: Nonsmooth dynamics of generalized nash games. *J. Nonlinear Var. Anal* **4**(1), 27–44 (2020)
- Migot, T., Cojocaru, M.-G.: A parametrized variational inequality approach to track the solution set of a generalized nash equilibrium problem. *European Journal of Operational Research* **283**(3), 1136–1147 (2020)
- Migot, T., Cojocaru, M.-G.: A decomposition method for a class of convex generalized nash equilibrium problems. *Optimization and Engineering* **22**(3), 1653–1679 (2021)
- Ma, Z., Callaway, D.S., Hiskens, I.A.: Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on control systems technology* **21**(1), 67–78 (2011)
- Monderer, D., Shapley, L.S.: Potential games. *Games and economic behavior* **14**(1), 124–143 (1996)
- Ma, Z., Zou, S., Ran, L., Shi, X., Hiskens, I.A.: Efficient decentralized coordination of large-scale plug-in electric vehicle charging. *Automatica* **69**, 35–47 (2016)
- Nesterov, Y.: Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization* **22**(2), 341–362 (2012)
- Nash Jr, J.F.: Equilibrium points in n-person games. *Proceedings of the national academy of sciences* **36**(1), 48–49 (1950)
- Neto, O.B., Mulas, M., Corona, F.: Sls-brd: A system-level approach to seeking generalised feedback nash equilibria. *IEEE Transactions on Automatic Control* (2025)
- Nutini, J., Schmidt, M., Laradji, I., Friedlander, M., Koepke, H.: Coordinate descent converges faster with the gauss-southwell rule than random selection. In: *International Conference on Machine Learning*, pp. 1632–1641 (2015). PMLR
- Nabetani, K., Tseng, P., Fukushima, M.: Parametrized variational inequality approaches to generalized nash equilibrium problems with shared constraints. *Computational optimization and applications* **48**(3), 423–452 (2011)
- Nie, J., Tang, X., Xu, L.: The gauss–seidel method for generalized nash equilibrium problems of polynomials. *Computational Optimization and Applications* **78**(2), 529–557 (2021)

- Pustilnik, M., Borrelli, F.: Generalized nash equilibrium solutions in dynamic games with shared constraints. arXiv preprint arXiv:2502.19569 (2025)
- Pang, J.-S., Fukushima, M.: Quasi-variational inequalities, generalized nash equilibria, and multi-leader-follower games. *Computational Management Science* **2**, 21–56 (2005)
- Pustilnik, M., Loquercio, A., Borrelli, F.: Non-normalized solutions of generalized nash equilibrium in autonomous racing. arXiv preprint arXiv:2503.12002 (2025)
- Porter, R., Nudelman, E., Shoham, Y.: Simple search methods for finding a nash equilibrium. *Games and Economic Behavior* **63**(2), 642–662 (2008)
- Stein, O., Sudermann-Merx, N.: The noncooperative transportation problem and linear generalized nash games. *European Journal of Operational Research* **266**(2), 543–553 (2018)
- Thiran, G., Stupia, I., Vandendorpe, L.: Best response dynamics convergence for generalized nash equilibrium problems: An opportunity for autonomous multiple access design in federated learning. *IEEE Internet of Things Journal* **11**(10), 18463–18482 (2024)
- Takahashi, S., Yamamori, T.: The pure nash equilibrium property and the quasi-acyclic condition. *Economics bulletin* **3**(22), 1–6 (2002)
- URYAs’ Ev, S., Rubinstein, R.Y.: On relaxation algorithms in computation of non-cooperative equilibria. *IEEE Transactions on Automatic Control* **39**(6), 1263–1267 (2002)
- Von Heusinger, A., Kanzow, C.: Relaxation methods for generalized nash equilibrium problems with inexact line search. *Journal of Optimization Theory and Applications* **143**(1), 159–183 (2009)
- Wang, X.: A computational approach to dynamic generalized nash equilibrium problem with time delay. *Communications in Nonlinear Science and Numerical Simulation* **117**, 106954 (2023)
- Wild, E.: A study of heuristic approaches for solving generalized nash equilibrium problems and related games. PhD thesis, University of Guelph (2017)
- Wang, K., Jia, W.: A new intelligent algorithm for solving generalized nash equilibrium problem. *Alexandria Engineering Journal* **123**, 17–28 (2025)