Socially Optimal Non-Discriminatory Restrictions for Continuous-Action Games

Michael Oesterle, [1] **Guni Sharon** [2]

[1]Institute for Enterprise Systems, University of Mannheim, Germany
[2]Department of Computer Science & Engineering, Texas A&M University
michael.oesterle@uni-mannheim.de, guni@tamu.edu

Abstract

We address the following mechanism design problem: Given a multi-player Normal-Form Game (NFG) with a continuous action space, find a non-discriminatory (i.e., identical for all players) restriction of the action space which maximizes the resulting Nash Equilibrium with respect to a fixed social utility function. First, we propose a formal model of a *Restricted Game* and the corresponding restriction optimization problem. We then present an algorithm to find optimal non-discriminatory restrictions under some assumptions. Our experimental results with Braess' Paradox and the Cournot Game show that this method leads to an optimized social utility of the Nash Equilibria, even when the assumptions are not guaranteed to hold. Finally, we outline a generalization of our approach to the much wider scope of Stochastic Games.

1 Introduction

Consider a multi-player game with an additional social utility function over the joint actions. Assuming that the players are self-interested and learn independently, they might converge to joint actions ("user equilibria") which are suboptimal, both from their own perspective (e.g., with respect to Pareto efficiency) and from the viewpoint of social welfare (Cigler and Faltings 2011). This can be demonstrated in minimal setups (see the examples in Section 3), but it is also common in real-world settings (Ding and Song 2012; Acemoglu et al. 2016; Memarzadeh, Moura, and Horvath 2020).

While the challenge of reconciling selfish optimization and overall social utility in multi-player settings has long been known (Roughgarden and Tardos 2002; Andelman, Feldman, and Mansour 2009), it has become increasingly relevant with the rise of ubiquitous autonomous agents and automated decision-making in recent years as advancements in deep reinforcement learning have enabled agents to learn very effective (but still selfish) policies not only in well-defined games but also in multi-agent systems with large, complex, and unknown environments (Du and Ding 2021; Gronauer and Diepold 2021).

A common solution method for this problem involves *reward shaping*, where players' utility functions are altered by giving them additional positive rewards for socially desirable behavior and negative rewards (i.e., sanctions) for undesirable behavior. Normative Systems (Andrighetto et al. 2013) derive such rewards and sanctions from norms, while Vickrey–Clarke–Groves (VCG) mechanisms (Nisan and Ronen 2004) attribute to each player the marginal social cost of its actions.

Reward-shaping methods generally make two assumptions which limit their applicability:

1. *Rewards can be changed at will, and players simply accept the new reward function.* This assumption is feasible in stylized settings, but involves an arbitrary amount of additional incentives (money) when applied in real-world settings.

2. *It is both possible and ethically justifiable to discriminate between players by shaping their reward functions differently.* On top of ethical issues, this approach might not be applicable whenever players are not identifiable.

We propose a novel solution for closing the gap between user equilibrium and social optimum, based on *shaping the action space* available to the players at any given time (as commonly done by regulating governmental entities). Therefore, players continue to optimize their own objective function over the restricted action space. This motivates the problem of finding an *optimal non-discriminatory restriction* of the players' action space, i.e., a restriction which is identical for all players and maximizes the social utility of a stable joint action.

In this paper, we analyze the problem of finding socially optimal restrictions for Normal-Form Games (NFG): We define the concept of a Restricted Game (RG) and present a novel algorithm denoted *Socially Optimal Action-Space Restrictor* (SOAR) which finds optimal restrictions via an exhaustive Breadth-First Search over the restriction space, assuming that (a) there is always a Nash Equilibrium, and (b) there is an oracle function which produces such a Nash Equilibrium for a given restriction. We then demonstrate the algorithm's performance using two well-known game-theoretic problems—Braess' Paradox and the Cournot Game. Our experiments show that applying SOAR can find favorable outcomes even when we relax the assumptions. Finally, in Section 7, we outline how the approach developed for (stateless) multi-player Normal-Form Games is also applicable to Stochastic Games with state transitions.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

2 Preliminaries

Unless indicated otherwise, we follow the standard Game Theory notation as used, for instance, in Leyton-Brown and Shoham (2008).

2.1 Model

Let $G = (N, A, \boldsymbol{u})$ be a Normal-Form Game with player set $N = \{1, ..., n\}$ and action space A which applies to all players ("uniform" NFG). Writing product sets and vectors of variables in bold face, a joint action is given by $\boldsymbol{a} \in \boldsymbol{A} := A^N$. The players' utility functions are $\boldsymbol{u} = (u_i)_{i \in N}$, where $u_i : \boldsymbol{A} \to \mathbb{R}$. Moreover, let $\mathfrak{u} : \boldsymbol{A} \to \mathbb{R}$ be a *social utility function*. We call $G = (N, A, \boldsymbol{u}, \mathfrak{u})$ a *social game*.

It is important to note that the notion of "optimality" refers to two functions: First, there is the individual utility u_i of each player which defines the player's optimization goal (the reward in multi-agent systems). Second, the social utility $\mathfrak{u}$ represents the view of the governance (i.e., the entity imposing restrictions on the system), which is not necessarily linked to the player utilities. In practice, however, we often use functions like $\mathfrak{u} = \sum_{i \in N} u_i$ to measure overall social welfare (we will do so for the remainder of this paper). As usual in multi-player games, all utility values depend on the *joint action*, such that players cannot simply maximize their utility without taking into account their competitors' actions.

Definition 1 (Restriction)**.** *A* restriction *is any subset $R \subseteq A$ of the action space, denoting the set of* allowed actions.

Definition 2 (Restricted Game)**.** *For a social game $G = (N, A, \boldsymbol{u}, \mathfrak{u})$ and a* restriction *$R \subseteq A$, we define the Restricted Game (RG), $G|_R = (N, R, \boldsymbol{u})$ such that the players are only allowed to use actions in R instead of the full action space A. Equivalently, the domain of the utility functions is restricted to $\boldsymbol{R} := R^N$.*

2.2 Equilibria and Optima

Definition 3 (Best Response)**.** *Let $\boldsymbol{a} \in \boldsymbol{A}$ be a joint action, and let $\boldsymbol{a}_{-i} := (a_1, ..., a_{i-1}, a_{i+1}, ..., a_n)$ denote the vector obtained by removing a_i (thus, we can write $\boldsymbol{a} = (a_i, \boldsymbol{a}_{-i})$). Then*

$$\mathcal{B}_i(\boldsymbol{a}_{-i}) := \arg\max_{a \in A} u_i(a, \boldsymbol{a}_{-i}) \subseteq A$$

denotes the set of all best responses *(BRs) of player i to the other players' given actions $\boldsymbol{a}_{-i}$.*

Definition 4 (Nash Equilibrium)**.** *A joint action $\boldsymbol{a} \in \boldsymbol{A}$ is a (pure)* Nash Equilibrium *(NE) of G if each individual action in $\boldsymbol{a}$ is a best response to the other players' actions. $\mathcal{N}$ denotes the set of all Nash Equilibria of G:*

$$\mathcal{N} := \{\boldsymbol{a} \in \boldsymbol{A} : \ a_i \in \mathcal{B}_i(\boldsymbol{a}_{-i}) \, \forall i\} \ .$$

Definition 5 (Minimum Nash Equilibrium)**.** *A* minimum Nash Equilibrium *$\mathcal{N}^- := \arg\min_{\boldsymbol{a} \in \mathcal{N}} \mathfrak{u}(\boldsymbol{a})$ is an equilibrium with the lowest social utility (the worst NE from the governance's perspective).*

Definitions 3 and 4 can be applied to restricted games, and are denoted as $\mathcal{B}_i|_R$, $\mathcal{N}|_R$, and $\mathcal{N}^-|_R$, respectively. It is noteworthy that they can, in general, change arbitrarily (for better or worse) by restricting a game.

Definition 6 (Minimum Equilibrium Social Utility (MESU))**.** *Let R be a restriction of A. For the RG $G|_R$,*

$$\mathcal{S}(R) := \min_{\boldsymbol{a} \in \mathcal{N}|_R} \mathfrak{u}(\boldsymbol{a}) = \mathfrak{u}(\mathcal{N}^-|_R)$$

denotes the minimum equilibrium social utility*, given the restriction R.*

We focus on the minimum NE in this definition, since the governance cannot decide which one of the equilibria the players converge to in an RG $G|_R$. Thus, $\mathcal{S}(R)$ provides a lower bound for the resulting social utility of a restricted game.

2.3 Restrictions

Although the model does not specify any structure for the action space, A, we limit our discussion here to real-valued intervals. By doing so, we can define restrictions that are finite unions of half-open intervals in A.

Assumption 1 (Interval-Union Restrictions)**.** *We assume in this work that the action space, A, is a one-dimensional interval $[a, b)$ (using $\pm\infty$ for unbounded spaces), and that the governance can define restrictions of A which are finite unions of intervals:*

$$R = \bigcup_i [l_i, u_i) \tag{1}$$

with interval bounds $l_i, u_i \in A \, \forall i$.

A joint action $\boldsymbol{a} \in \boldsymbol{A}$ is *allowed* if all components of $\boldsymbol{a}$ are in R, or equivalently, if $\boldsymbol{a} \in \boldsymbol{R}$, since the restriction R applies equally to all players (it is "non-discriminatory").

It is important to note that adding or removing an interval $[l, u) \subseteq A$ to or from R does not violate Equation 1 since this family of restrictions is closed under finite unions and set differences. We call a restriction R' *more constrained* than R if $R' \subset R$. Finally, for a restriction R of form (1), let $|R| := \sum_i (u_i - l_i)$ denote the *size* of R.

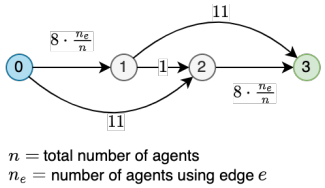
The limitation of one-dimensional action spaces is discussed in Section 7.1.

3 Braess' Paradox and the Cournot Game

The approach of governing via action space restrictions is best illustrated in the discrete case of Braess' Paradox (Braess 1968). Nonetheless, our main contribution applies to the more general case of continuous actions.

Example 1 (Braess' Paradox)**.** *Braess' Paradox can be translated from its original domain of traffic routing (Figure 1a) into the two-player matrix game shown in Figure 1b, where the row action is controlled by player 1, and the column action by player 2. By convention, both players want to maximize their respective payoff.*

The best response for both players is always b. Selfish players will converge to the user equilibrium (b, b) and therefore end up with a payoff of 1. Let us now forbid action b, i.e., restrict the action space to $\{a, c\}$. The user equilibria become (a, c) and (c, a) with a payoff of 2 for both players.



(a) as a routing problem with n players

	a	b	c
a	$(0,0)$	$(0,5)$	$(2,2)$
b	$(5,0)$	$(1,1)$	$(5,0)$
c	$(2,2)$	$(0,5)$	$(0,0)$

(b) as an equivalent two-player Matrix Game

Figure 1: Braess' Paradox

It is known that Braess-like scenarios are not only technical cases, but appear often in random networks (Valiant and Roughgarden 2006; Chung and Young 2010).

Apart from illustrating the efficacy of a restriction-based governance approach, this example also shows the "meta challenge" of restrictions: if we allowed for *individual* restrictions of the players' action spaces, it would be straightforward for the governance to achieve any possible outcome (i.e., combination of actions) by allowing each player to use exactly one action. This procedure reduces the (multi-player) game to a (single-player) optimization problem, where the governance computes the socially optimal matrix cell $\max_{\boldsymbol{a} \in \boldsymbol{A}} \mathfrak{u}(\boldsymbol{a})$ with $\boldsymbol{A} = \prod_{i \in N} A_i$, and then simply assigns the respective actions to the players.

Things become more challenging when only considering *non-discriminatory* restrictions, as we have done in Example 1. This also satisfies an extremely desirable property for any form of governance: All players are treated fairly by having the same space of allowed actions. In the example, the governance could enforce the (socially optimal) solutions (a, b), (b, a), (b, c), or (c, b) with social utility 5 by using *individual* restrictions. Still, with uniform restrictions, we can improve the game's MESU from 2 to 4.

Let us now consider a game with a *continuous* action space where rewards are given as individual utility functions over the joint action space: The *Cournot Game* is a classical example of an NFG with one-dimensional continuous action spaces, and one of the fundamental economic models for establishing produced quantities and prices on a market.

Example 2 (Cournot Game)**.** *Let two players decide on the produced quantities* $\boldsymbol{q} = (q_1, q_2) \in \mathbb{R}^2$ *of a good whose price is defined as* $p(\boldsymbol{q}) = \max(p_{max} - q_1 - q_2, 0)$ *with* $p_{max} > 0$*. Both players produce at a constant cost of* $c \geq 0$ *per unit. The players' utilities (i.e., their profit) are therefore given as* $u_i(\boldsymbol{q}) = q_i \cdot (p(\boldsymbol{q}) - c)$*.*

Choosing $p_{max} = 120$ *and* $c = 12$*, the BR of player* i *to action* q_j *is* $\mathcal{B}_i(q_j) = 54 - \frac{q_j}{2}$*, which leads to a unique NE of* $\boldsymbol{q}^* = (36, 36)$ *and a payoff of* $u_1(\boldsymbol{q}^*) = u_2(\boldsymbol{q}^*) = 1296$*. By restricting the quantities produced by each player to the range* $q_i \leq 27$*, it would be possible to improve the equilibrium payoff to* 1458 *per player.*

In these examples, we have the particular situation that the restriction improves the utility of *all players*, which makes a very strong case for using such restrictions. In general, it is not the case that all players will be better off, so the governance's goal is simply to maximize the social utility $\mathfrak{u}$.

We revisit the examples in Section 6, using the SOAR algorithm (see Section 5.2) to find optimal restrictions.

4 Related Work

The literature on mechanism design through action space restrictions is sparse. Apart from Pernpeintner, Bartelt, and Stuckenschmidt (2021), who use end-to-end Reinforcement learning to design restrictions of discrete action spaces, there is, to our best knowledge, no prior work on action space shaping as a means of aligning user equilibria and social optima in a multi-agent setting.

Mittelmann et al. (2022) propose a logic-based method (*Automated Synthesis of Mechanisms*) for automated mechanism design, but focus on optimizing the transition function of a game while keep the action space. This is, in a way, a complementary approach to our Restricted Games. Kanervisto, Scheller, and Hautamäki (2020) use action space shaping to improve learning, focusing on the observation and action spaces of a single agent in video games such as Atari, StarCraft and Dota. Kalweit et al. (2021) shape the action space of a DQN agent in the domain of autonomous driving by defining a cost function for actions and then restricting the action space to those actions whose cost is below a fixed threshold. A similar approach is used by Achiam et al. (2017) to directly shape the policy space of an RL agent. Tang (2017) and Cai et al. (2018), on the other hand, use *Reinforcement Mechanism Design* to automate the design of e-auctions, restricting bidders' actions based on past behavior. Therefore, their restrictions are imposed from an outside entity (as in our approach), but they are not optimized over a social utility function.

As an alternative to the action space shaping approach, reward shaping (Mataric 1994) addresses the agents' rewards to change their behavior, relying on the fact that optimizing the expected new reward will result in a different action policy. Centralized reward shaping can follow the structure of, for example, a Vickrey–Clarke–Groves (VCG) mechanism (Nisan and Ronen 2004). However, instead of letting the agents optimize their policies, a VCG mechanism performs the outcome selection itself, computing concrete best actions to the agents. This leads to well-known computability issues, for example when solving the NP-hard problem of optimal allocation in a combinatorial auction. The VCG-based method of Marginal-Cost Pricing (MCP) (Turvey 1969) has been successfully applied to Braess' Paradox (Ding and Song 2012) and real-world traffic networks (Sharon et al. 2017a,b, 2018, 2019; Hanna et al. 2019). While presenting promising theoretical and experimental results, these solutions rely on the assumption that agents' utility functions can be manipulated in a discriminatory way.

Normative Systems (Andrighetto et al. 2013; Chopra, van der Torre, and Verhagen 2018) go one step further by defining norms from which rewards and sanctions (i.e., negative rewards) are then derived by the agents. Whether this normative reward is imposed onto the agents from an outside entity (Morales et al. 2013; Neufeld et al. 2021) or emerges from within the agent community (Morris-Martin, De Vos,

and Padget 2021), there is a need for the agents to be *norm-aware* and to use normative capabilities in their action policy (Cramton 2006).

5 Finding Optimal Restrictions

Next, we present the Socially Optimal Action-Space Restrictor (SOAR) algorithm for continuous-action games with a finite action space A. SOAR defines a search tree of increasingly constrained restrictions by identifying and testing subsets of existing restrictions, starting from the unrestricted action space. The theoretical results and conclusions in this section hold for arbitrary social utility functions $\mathfrak{u}$.

5.1 Restricting the Action Space

For a given joint action, $\boldsymbol{a}$, we say that a restriction, R, *invalidates* $\boldsymbol{a}$ if $\boldsymbol{a} \notin \boldsymbol{R}$, i.e., at least one individual action is not allowed. In general, a restriction R which invalidates an existing NE does not simply cause a new NE to appear at the boundary of R (i.e., as close to the old NE as allowed by R)—instead, a new NE might appear anywhere else in the joint action space, or the restriction might not allow for an NE at all. However, a restriction that does not invalidate any existing NE (we call such a restriction *irrelevant*), leaves the existence of those NE unchanged. More formally:

Proposition 1. *Given some $x \in \mathbb{R}$, let $\mathcal{U}_\epsilon(x) := [x-\epsilon, x+\epsilon)$ denote the half-open ϵ-neighborhood of x, and for a vector $\boldsymbol{x} \in \mathbb{R}^N$, let $\mathcal{U}_\epsilon(\boldsymbol{x}) := \cup_{i\in N}\mathcal{U}_\epsilon(x_i) \subseteq \mathbb{R}$ (note that this neighborhood is still one-dimensional!). Assume that $\boldsymbol{a} \in \boldsymbol{A}$ is a joint action such that $\mathcal{N} \subseteq \boldsymbol{R}$ with $R := A \backslash \mathcal{U}_\epsilon(\boldsymbol{a})$. Then*

$$\mathcal{N} \subseteq \mathcal{N}|_R \ ,$$

which means that invalidating actions within the ϵ-neighborhood of $\boldsymbol{a}$ removes none of the Nash Equilibria from G.

Proof. Let $\boldsymbol{x} \in \mathcal{N}$ be an NE over the action space A, and let R be defined as in the statement of the proposition. Then

$$\begin{aligned} & x_i \in \mathcal{B}_i(x_{-i}) \ \forall i \in N \\ \implies & u_i(\boldsymbol{x}) \geq u_i(a', \boldsymbol{x}_{-i}) \ \forall a' \in A \ \forall i \in N \\ \overset{R\subseteq A}{\implies} & u_i(\boldsymbol{x}) \geq u_i(a', \boldsymbol{x}_{-i}) \ \forall a' \in R \ \forall i \in N \\ \overset{\boldsymbol{x}\in\boldsymbol{R}}{\implies} & x_i \in \mathcal{B}_i|_R(x_{-i}) \forall i \in N \quad \implies \quad \boldsymbol{x} \in \mathcal{N}|_R \ . \end{aligned}$$

□

As a direct consequence, any restriction that improves the MESU of a game must invalidate all existing minimum Nash Equilibria.

5.2 The SOAR Algorithm

Starting from an unrestricted action space, the idea is to define successively more constrained restrictions and then search for the best of those restrictions in terms of their MESU (see Algorithm 1). Basically, we can check every possible restriction of the form shown in Equation 1 (see Assumption 1), starting from A and ending with maximally constrained restrictions. Of course, this brute-force method is not practical, since it requires computing the MESU of infinitely many restrictions.

Given a current restriction R, we propose the following improvement: Calculate the minimum NE, $\boldsymbol{a}^* := \mathcal{N}^-|_R$, and derive all *relevant actions*, i.e., the set $\Omega := \bigcup_{i\in N} \boldsymbol{a}_i^*$ of all (individual) actions that are used in $\mathcal{N}^-|_R$. For each $\omega \in \Omega$, define a new restriction by removing an ϵ-neighborhood $\mathcal{U}_\epsilon(\omega)$ from R (see Figure 2).

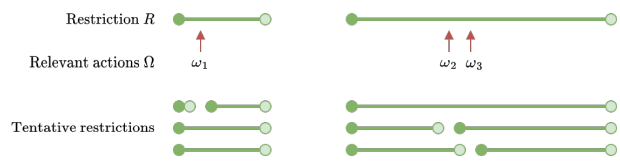


Figure 2: Tentative restrictions for a set Ω of relevant actions

It follows from Proposition 1 that, in the setting of Figure 2, any restriction $R' \subset R$ which includes ω_1, ω_2 and ω_3, would not eliminate $\mathcal{N}^-|_R$, and therefore cannot have a higher MESU than R. Hence, it is not necessary to check those restrictions, effectively pruning them from the search tree. As we show experimentally in Section 6, this can lead to a significant reduction in the number of NE calculations required compared to uniformly checking all restrictions.

For each of the tentative restrictions, we repeat the process of computing the NE and relevant actions and subsequently restrict them further until the action space is empty. Of all those restrictions, we then select the one which gives the highest MESU, resulting in a breadth-first search over the restriction space. To ensure that restrictions are not considered multiple times, we keep a set (a closed/duplicate list) of already explored restrictions. Moreover, the state space size can be controlled via the hyperparameter ϵ (the *resolution* of SOAR) which defines the size of the interval around a relevant action that is removed for tentative restrictions.

5.3 Equilibrium Oracle

To add restrictions purposefully, we need to know where the current equilibria are. For now, we assume that there is an oracle function μ which, for a given RG $G|_R$, returns a joint action $\boldsymbol{a} \in \boldsymbol{R}$ which is an equilibrium of $G|_R$ with minimum social utility. In Section A of the appendix, we show how to implement such an oracle for quadratic utility functions.

5.4 Complexity and Correctness

Proposition 2. *Let $A = [a, b)$ and $\epsilon > 0$. Then any restriction chain $A = R_0 \sqsupset R_1 \sqsupset \ldots \sqsupset R_\delta = \varnothing$ consists of at most $\lceil \frac{b-a}{\epsilon} \rceil$ elements, where $R \sqsupset R'$ means that R' is a tentative restriction over R as created by Algorithm 1. In other words, the depth of the search tree is bounded by $\lceil \frac{b-a}{\epsilon} \rceil$.*

Proof. For any subsequent pair $R_i \sqsupset R_{i+1}$ in the restriction chain, let us denote by ω_i the action whose ϵ-neighborhood was removed at this step. We see that $|\omega_i - \omega_j| \geq \epsilon \ \forall i < j$ (otherwise, ω_j would have been already forbidden before its removal). There cannot be more than $\lceil \frac{b-a}{\epsilon} \rceil$ points with pairwise distance $\geq \epsilon$ on the interval A which has length $(b-a)$. □

Algorithm 1: Socially Optimal Action-Space Restrictor (SOAR)

Data: Social Game $G = (N, A, \boldsymbol{u}, \mathfrak{u})$, equilibrium oracle μ, resolution ϵ

Result: Socially optimal restriction $\hat{R}^* \subseteq A$

1 $(\hat{R}^*, \hat{u}^*) \leftarrow (A, \mathfrak{u}(\mu(A)))$
2 $Q \leftarrow$ Queue with content A
3 **while** *Q is not empty* **do**
4 $\quad R \leftarrow Q.dequeue()$
$\quad$ `// Loop through relevant actions`
5 $\quad$ **for** $\omega \in \Omega(\mu(R))$ **do**
6 $\quad\quad R' \leftarrow R.remove(\mathcal{U}_\epsilon(\omega))$ `// Tentative restriction`
7 $\quad\quad$ **if** *R' is not empty and has not been explored before* **then**
8 $\quad\quad\quad Q.enqueue(R')$
9 $\quad\quad\quad$ **if** $\mathfrak{u}(\mu(R')) > \hat{u}^*$ **then**
10 $\quad\quad\quad\quad (\hat{R}^*, \hat{u}^*) \leftarrow (R', \mathfrak{u}(\mu(R')))$
11 $\quad\quad\quad$ **end**
12 $\quad\quad$ **end**
13 $\quad$ **end**
14 **end**
15 **return** $\hat{R}^*$

As a result of Proposition 2, we can bound the runtime of SOAR by $|\Omega_{max}|^d$, where $d = \lceil \frac{b-a}{\epsilon} \rceil$, and Ω_{max} is the largest set $\Omega(\mu(R))$ we encounter in the **for** loop (line 5).

Definition 7. *A restriction R^* is called* optimal *for G if $\mathcal{S}_G(R^*) \geq \mathcal{S}_G(R)\ \forall R \subseteq A$, and an optimal restriction R^* is called* minimally restrictive *if no proper superset of R^* is an optimal restriction.*

Assumption 2. *We assume that there is always a Nash Equilibrium for a restricted game, i.e., $\Omega(R) \neq \varnothing\ \forall R \subseteq A$.*

Proposition 3. *Let R^* be an optimal restriction for a game G. Then, under Assumption 2,*

$$R^* \subset R \ \Rightarrow\ \exists \omega \in \Omega(R) : \omega \notin R^* \ .$$

Proof. Assume that $\forall \omega \in \Omega(R) :\ \omega \in R^*$. Then $\mathcal{N}^-|_R \in \boldsymbol{R^*}$, and since $R^* \subset R$, $\mathcal{N}^-|_R \in \mathcal{N}|_{R^*}$ according to Proposition 1. Therefore, $\mathcal{S}(R^*) \leq \mathcal{S}(R)$, which, together with $R^* \subset R$, contradicts the optimality of R^*. □

Proposition 4. *Throughout Algorithm 1, (at least) one of the following two conditions holds true:*

(i) The restriction queue Q contains a restriction R which is a superset of an optimal restriction R^*

(ii) $\hat{R}^*$ is already set to an optimal restriction

Proof. After the initialization step, condition (i) holds since any optimal restriction R^* is a subset of A, which is in Q.

From Definition 7, we see immediately that the update step $(\hat{R}^*, \hat{u}^*) \leftarrow (R', \mathfrak{u}(\mu(R')))$ in line 10 satisfies two properties: A non-optimal restriction never replaces an optimal one, and an optimal restriction always replaces a non-optimal one. Therefore, once condition (ii) is satisfied, it stays satisfied until SOAR terminates. Let us therefore assume that (ii) does not hold yet.

Whenever a restriction R is dequeued from Q, (i) either still holds (this is the case if there is another such restriction still in Q), or R is a superset of an optimal restriction R^*. Since (ii) is not satisfied, we know that R itself is not optimal. Proposition 3 asserts that there is a relevant action $\omega \in \Omega(R)$ which is not in R^*. Hence, at the respective pass of the **for** loop, we will have $R' := R \setminus \mathcal{U}_\epsilon(\omega)$, and, for a sufficiently small ϵ, $R' \supseteq R^*$.

If R' has been explored before, it was enqueued then, meaning that (i) still holds. Otherwise, R' is enqueued now. If $R' \supset R^*$, (i) holds, and if not, R' is optimal, such that (ii) becomes true. □

Theorem 1. *Let $G = (N, A, \boldsymbol{u}, \mathfrak{u})$ be a social game. If Assumption 2 holds, and for a sufficiently small $\epsilon > 0$, Algorithm 1 finds an optimal restriction R^*.*

Proof. SOAR terminates after finitely many steps: Any tentative restriction R' produced by a reduction of some $R \in Q$ continues a chain of increasingly constrained restrictions, as in Proposition 2, and the length of such a chain is bounded by $\lceil \frac{b-a}{\epsilon} \rceil$.

At the point of termination, Q is empty. Condition (i) in Proposition 4 does not hold anymore, which means that $\hat{R}^*$ is indeed an optimal restriction. □

6 Experiments

We have shown that the SOAR algorithm finds an optimal restriction for a given NFG under some assumptions. However, these assumptions are not always satisfied in real-world settings. Our experimental study is therefore set to address the following open questions:

Q1 If Assumption 2 is not guaranteed to hold, does SOAR still find (close to) optimal restrictions?

Q2 Does the state-space pruning technique used by SOAR allow for reasonable run-times, despite the fact that the size of the search tree is exponential in $\frac{b-a}{\epsilon}$?

To answer these questions, we examine parameterized continuous-action versions of the Cournot Game (CG) and Braess' Paradox (BP). First, we use domain knowledge about both games to establish theoretical results for their social optimum and optimal restriction (see appendix). Afterwards, we compare these findings with the results of SOAR for a range of parameters to obtain insights into SOAR's scaling behavior. The values of ϵ were empirically chosen to provide a good balance between run-time and accuracy, but the results are actually reasonably insensitive to this choice: Varying ϵ by a factor of 5 caused the MESU to change by less than 1% in both games.

6.1 Quadratic Utility Functions

Let us start by observing that many interesting problems, including the continuous Braess Paradox, the Cournot Game,

and the continuous version of any 2x2 Matrix Game, can be represented as NFGs with *quadratic utility functions*. They have the convenient property of being convex or concave (or both, i.e., linear) in each variable x_i, depending on the sign of the coefficient of x_i^2. They allow for efficient computation of BR and NE, and therefore lend themselves well to the examination of RGs and the optimization of restrictions.

Definition 8 (Quadratic Utility Function)**.** *A utility function $u : \boldsymbol{A} \to \mathbb{R}$ is called* quadratic *if it is polynomial in the players' actions x_i and has a maximum degree of 2. This means that, for n players,*

$$u(\boldsymbol{x}) = \sum_{\alpha \in \mathbb{N}^n} c_\alpha \cdot x_1^{\alpha_1} \cdots x_n^{\alpha_n}$$

with $c_\alpha \in \mathbb{R}$ and $\max_{c_\alpha \neq 0} \left(\sum_{i=1}^n \alpha_i\right) \leq 2$.

For two players, quadratic utility functions have the form

$$u(x_1, x_2) = ax_1^2 + bx_2^2 + cx_1x_2 + dx_1 + ex_2 + f$$

with real coefficients $a, b, c, d, e, f \in \mathbb{R}$.

Quadratic utility functions allow us to construct an equilibrium oracle μ for SOAR without any specific knowledge about the game (see Section A in the appendix).

6.2 Definition of Parameterized Games

Definition 9 (Cournot Game)**.** *A parameterized Cournot Game (CG) with parameter $\lambda := p_{max} - c$ is defined by $N = \{1, 2\}$, $A = [0, \lambda]$, $u_1(x_1, x_2) = -x_1^2 - x_1x_2 + \lambda x_1$ and $u_2(x_1, x_2) = -x_2^2 - x_1x_2 + \lambda x_2$.*

In the continuous version of Braess' Paradox, players do not choose one of the available routes, but decide which fraction of their flow they send through each route (see Section D of the appendix for the derivation of the utility functions):

Definition 10 (Continuous Braess Paradox)**.** *A parameterized continuous Braess Paradox (BP) with parameter $b \geq 0$ is defined by $N = \{1, 2\}$, $A = [0, 1]$, $u_1(\boldsymbol{x}) = -4x_1^2 + (b-5)x_1 - 4x_2 + 17$ and $u_2(\boldsymbol{x}) = -4x_2^2 - 4x_1 + (b-5)x_2 + 17$.*

Varying b changes the attractiveness of taking the "social" routes, compared to the "selfish" route. This degree of freedom is sufficient to change the structure of the game and its equilibria.

6.3 Metrics

To measure the performance of SOAR, we use the following metrics:

Definition 11. *For an action space A and a restriction $R \subseteq A$, the* degree of restriction *is defined as $\mathfrak{r}(R) := 1 - \frac{|R|}{|A|}$.*

Definition 12. *The* relative improvement *of a restriction R is*

$$\Delta(R) := \frac{\min_{x \in \mathcal{N}|_R} \mathfrak{u}(x) - \min_{x \in \mathcal{N}} \mathfrak{u}(x)}{|\min_{x \in \mathcal{N}} \mathfrak{u}(x)|} .$$

Moreover, we measure the number of oracle calls in SOAR as a proxy for the cost of finding an optimal restriction, implying that μ is assumed to have constant run-time (see also the *oracle complexity* of an algorithm as defined by Nemirovsky and Yudin (1983)).

6.4 Theoretical Expectation

Cournot Game The optimal restriction R^* for the CG with parameter λ is $R^* = [0, \frac{\lambda}{4}) \cup [\frac{\lambda}{2}, \lambda)$ with a constant degree of restriction $\mathfrak{r}(R^*) = 25\%$ (see Section B in the appendix for details). We expect the result of SOAR to fluctuate around these values, depending on the size of ϵ. The value of λ does not change the structure of the game, merely scaling the action space size, the equilibria and the restrictions.

Braess' Paradox The unique unrestricted NE (user equilibrium) is $\left(\frac{b-5}{8}, \frac{b-5}{8}\right)$, while the social optimum is $\left(\frac{b-9}{8}, \frac{b-9}{8}\right)$. This means that for $b \notin [5, 17]$, both joint actions coincide, and restricting the action space cannot improve the MESU. Within the interval $[5, 17]$, however, the players' actions need to be pushed down (toward 0) to match the social optimum, giving the optimal restriction $R^* = A \setminus \left[\frac{b-9}{8}, \frac{b-5}{4}\right]$ with a degree of restriction of $\mathfrak{r}(R) = \frac{b-5}{4}$ on $[5, 9]$ and $\mathfrak{r}(R) = \frac{17-b}{8}$ on $[9, 17]$ (see Section E of the appendix for the formal analysis).

6.5 Experimental Results

Cournot Game Figure 3 shows the results of SOAR for $\lambda \in \{10, 11, ..., 200\}$ with $\epsilon = 0.1$. The MESU of the restrictions found by SOAR is consistently $\approx 12.5\%$ larger than the unrestricted MESU, which matches the theoretical prediction. Together with a degree of restriction of $\approx 25\%$, this answers Q1 affirmatively for this setting. The number of oracle calls (i.e., tentative restrictions) increases quadratically in $|A|$ (see Section C of the appendix), as opposed to the exponential bound shown in Section 5.4. Regarding Q2, this indicates that the pruning technique can eliminate a large part of the possible restrictions.

Braess' Paradox Figure 4 shows the results of SOAR for $b \in [4, 18]$ in steps of 0.1 with $\epsilon = 0.001$. Let us have a look at $b \in [5, 9]$ first: While the user equilibrium decreases when b exceeds 5 (players find it increasingly advantageous to take the center route, causing more and more congestion), this effect can be completely eliminated using restrictions (as we see, the restricted MESU stays at 34). For $b > 9$, the optimal restriction stops pushing the players to choose action 0 but allows an interval of $[0, \frac{b-9}{8}]$. Hence, both social optimum and user equilibrium have increasing social utility, eventually joining at $b = 17$. Again, the degree of restriction and the restricted MESU approximately match the theoretical optimum (Q1). Since the action space has a constant size, the number of oracle calls is asymptotically constant, only impacted by the required degree of restriction and the subsequent pruning (Q2).

6.6 Reproducibility

The experiments were designed and executed in a Google Colaboratory notebook and are fully reproducible with zero configuration. The notebook is publicly available on the authors' GitHub repository at https://github.com/michoest/aaai-2023.

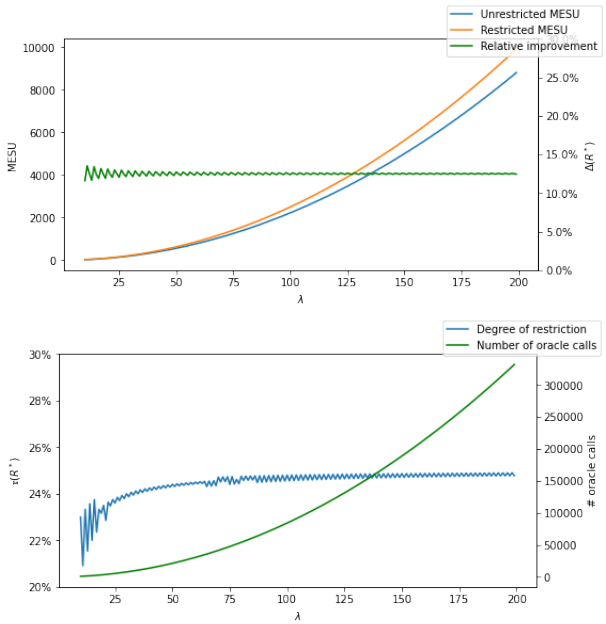


Figure 3: Unrestricted and restricted MESU, relative improvement, degree of restriction and oracle calls for the CG

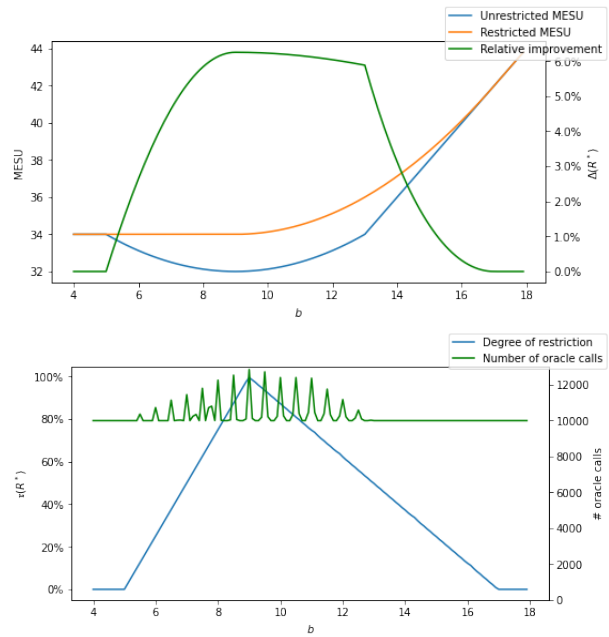


Figure 4: Unrestricted and restricted MESU, relative improvement, degree of restriction and oracle calls for the BP

7 Discussion

After showing experimentally that the SOAR algorithm reaches the theoretical expectation in practice and is robust against relaxed assumptions, we discuss three further aspects: Limitations of SOAR, restriction learning, and Stochastic Games.

7.1 Limitations of SOAR

We distinguish two types of limitations: *Fundamental limitations* (inherent boundaries of our approach), and *practical limitations* due to the implementation of the oracle.

Fundamental Limitations (a) In *coordination games*, social welfare is maximized when players choose the same action. *Contribution games*, in contrast, require the players to choose different actions to maximize social welfare, which means that uniform restrictions are unlikely to suffice for socially optimal outcomes; (b) Action spaces with multiple dimensions require a different approach for defining tentative restrictions, which respects the space's topology and the correlation between dimensions; (c) Lifting the assumption of finite interval-union restrictions could allow for better restrictions, but they might not have a compact representation.

Practical Limitations (a) SOAR currently only deals with pure strategies and equilibria. Mixed equilibria can be handled similarly, but the set of relevant actions will generally be infinite, such that sampling or discretizing actions will be necessary; (b) Other types of stable actions can be used instead of NE. In fact, NE are not the only concept for stable actions in multi-player interactions, and they do not always coincide with experimental results taken from RL agents (Nowé, Vrancx, and De Hauwere 2012).

7.2 Restriction Learning

The mechanism design problem of finding an optimal restriction for a given NFG with static utility functions requires upfront knowledge about the game. If, on the other hand, the utilities are unknown, it is necessary to derive them from observations and find restrictions through on-line learning. This represents a promising line of future work on SOAR.

7.3 Stochastic Games

Multi-agent systems are usually modeled as sequential decision processes with a changing environmental state. The presented version of SOAR explicitly calculates an optimal restriction for a given NFG; when generalizing to Stochastic Games, the utility (i.e., reward) functions depend on the state. As a consequence, the optimal restriction needs to be re-calculated for each time step. For a future extension of SOAR, we propose a *restriction policy* $\pi : \mathcal{S} \rightarrow \mathcal{R}$ which maps states to restrictions, and which can be trained to maximize the expected social utility.

8 Summary

In this paper, we have introduced the problem of designing optimal restrictions for Normal-Form Games with continuous action spaces. The SOAR algorithm can significantly improve a game's minimum equilibrium social utility by aligning user equilibrium and social optimum. Therefore, this work sets the scene for future work on more general restriction-based mechanism design approaches (e.g., Restricted Stochastic Games), which we conjecture to be a crucial step to building powerful governance entities for an emergent multi-agent society.

Acknowledgements

This work is supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) and the German-American Fulbright Commission.

References

Acemoglu, D.; Makhdoumi, A.; Malekian, A.; and Ozdaglar, A. 2016. Informational Braess' Paradox: The Effect of Information on Traffic Congestion. *Operations Research*, 66.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained Policy Optimization. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 22–31. JMLR.org.

Andelman, N.; Feldman, M.; and Mansour, Y. 2009. Strong Price of Anarchy. *Games and Economic Behavior*, 65(2): 289–317.

Andrighetto, G.; Governatori, G.; Noriega, P.; and van der Torre, L., eds. 2013. *Normative Multi-Agent Systems*. Dagstuhl Follow-Ups.

Braess, D. 1968. Über Ein Paradoxon Aus Der Verkehrsplanung. *Unternehmensforschung*, 12(1): 258–268.

Cai, Q.; Filos-Ratsikas, A.; Tang, P.; and Zhang, Y. 2018. Reinforcement Mechanism Design for E-commerce. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, 1339–1348. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-5639-8.

Chopra, A.; van der Torre, L.; and Verhagen, H., eds. 2018. *Handbook of Normative Multiagent Systems*. College Publications. ISBN 978-1-84890-285-5.

Chung, F.; and Young, S. J. 2010. Braess's Paradox in Large Sparse Graphs. In Saberi, A., ed., *Internet and Network Economics*, 194–208. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN 978-3-642-17572-5.

Cigler, L.; and Faltings, B. 2011. *Reaching Correlated Equilibria through Multi-Agent Learning*, volume 1. International Foundation for Autonomous Agents and Multiagent Systems.

Cramton, P. 2006. Combinatorial Auctions. In *European Economic Review*. MIT Press.

Ding, C.; and Song, S. 2012. Traffic Paradoxes and Economic Solutions. *Journal of Urban Management*, 1(1): 63–76.

Du, W.; and Ding, S. 2021. A Survey on Multi-Agent Deep Reinforcement Learning: From the Perspective of Challenges and Applications. *Artificial Intelligence Review*, 54(5): 3215–3238.

Gronauer, S.; and Diepold, K. 2021. Multi-Agent Deep Reinforcement Learning: A Survey. *Artificial Intelligence Review*.

Hanna, J. P.; Sharon, G.; Boyles, S. D.; and Stone, P. 2019. Selecting Compliant Agents for Opt-in Micro-Tolling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 565–572.

Kalweit, G.; Huegle, M.; Werling, M.; and Boedecker, J. 2021. Q-Learning with Long-term Action-space Shaping to Model Complex Behavior for Autonomous Lane Changes. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5641–5648. ISBN 2153-0866.

Kanervisto, A.; Scheller, C.; and Hautamäki, V. 2020. *Action Space Shaping in Deep Reinforcement Learning*. IEEE.

Leyton-Brown, K.; and Shoham, Y. 2008. *Essentials of Game Theory: A Concise Multidisciplinary Introduction*, volume 2. Morgan and Claypool Publishers. ISBN 978-1-59829-593-1.

Mataric, M. J. 1994. Reward Functions for Accelerated Learning. In Cohen, W. W.; and Hirsh, H., eds., *Machine Learning Proceedings 1994*, 181–189. San Francisco (CA): Morgan Kaufmann. ISBN 978-1-55860-335-6.

Memarzadeh, M.; Moura, S.; and Horvath, A. 2020. Multi-Agent Management of Integrated Food-Energy-Water Systems Using Stochastic Games: From Nash Equilibrium to the Social Optimum. *Environmental Research Letters*, 15(9): 0940a4.

Mittelmann, M.; Maubert, B.; Murano, A.; and Perrussel, L. 2022. Automated Synthesis of Mechanisms. In Raedt, L. D., ed., *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, 426–432. International Joint Conferences on Artificial Intelligence Organization.

Morales, J.; López-Sánchez, M.; Rodríguez-Aguilar, J.; Wooldridge, M.; and Vasconcelos, W. 2013. Automated Synthesis of Normative Systems. In *AAMAS*, volume 1. International Foundation for Autonomous Agents and Multiagent Systems.

Morris-Martin, A.; De Vos, M.; and Padget, J. 2021. A Norm Emergence Framework for Normative MAS – Position Paper. In Aler Tubella, A.; Cranefield, S.; Frantz, C.; Meneguzzi, F.; and Vasconcelos, W., eds., *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIII*, 156–174. Cham: Springer International Publishing. ISBN 978-3-030-72376-7.

Nemirovsky, A. S.; and Yudin, D. B. 1983. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons.

Neufeld, E.; Bartocci, E.; Ciabattoni, A.; and Governatori, G. 2021. A Normative Supervisor for Reinforcement Learning Agents. In Platzer, A.; and Sutcliffe, G., eds., *Automated Deduction – CADE 28*. Cham: Springer International Publishing.

Nisan, N.; and Ronen, A. 2004. Computationally Feasible VCG Mechanisms. *Journal of Artificial Intelligence Research*, 29.

Nowé, A.; Vrancx, P.; and De Hauwere, Y.-M. 2012. Game Theory and Multi-agent Reinforcement Learning. In Wiering, M.; and van Otterlo, M., eds., *Reinforcement Learning: State-of-the-Art*. Berlin, Heidelberg: Springer Berlin Heidelberg.

Pernpeintner, M.; Bartelt, C.; and Stuckenschmidt, H. 2021. Governing Black-Box Agents in Competitive Multi-Agent

Systems. In *Multi-Agent Systems: 18th European Conference, EUMAS 2021, Virtual Event, June 28–29, 2021, Revised Selected Papers*, 19–36. Berlin, Heidelberg: Springer-Verlag. ISBN 978-3-030-82253-8.

Roughgarden, T.; and Tardos, É. 2002. How Bad Is Selfish Routing? *Journal of The Acm*, 49(2): 236–259.

Sharon, G.; Albert, M.; Rambha, T.; Boyles, S.; and Stone, P. 2018. Traffic Optimization for a Mixture of Self-Interested and Compliant Agents. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Sharon, G.; Boyles, S. D.; Alkoby, S.; and Stone, P. 2019. Marginal Cost Pricing with a Fixed Error Factor in Traffic Networks. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, 1539–1546. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-1-4503-6309-9.

Sharon, G.; Hanna, J. P.; Rambha, T.; Levin, M. W.; Albert, M.; Boyles, S. D.; and Stone, P. 2017a. Real-Time Adaptive Tolling Scheme for Optimized Social Welfare in Traffic Networks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.

Sharon, G.; Levin, M. W.; Hanna, J. P.; Rambha, T.; Boyles, S. D.; and Stone, P. 2017b. Network-Wide Adaptive Tolling for Connected and Automated Vehicles. *Transportation Research Part C: Emerging Technologies*, 84: 142–157.

Tang, P. 2017. Reinforcement Mechanism Design. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 5146–5150.

Turvey, R. 1969. Marginal Cost. *The Economic Journal*, 79(314): 282–299.

Valiant, G.; and Roughgarden, T. 2006. Braess's Paradox in Large Random Graphs. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, 296–305. New York, NY, USA: Association for Computing Machinery. ISBN 1-59593-236-4.

Socially Optimal Non-Discriminatory Restrictions for Continuous-Action Games (Appendix)

Main Track, Paper ID: 6770

A Equilibrium Oracle for Quadratic Utilities

For a quadratic utility function u and a restriction R, the best response $\mathcal{B}_1|_R(x_2)$ can be found by a straight-forward case analysis: Let u be defined as

$$u(x_1, x_2) = ax_1^2 + bx_2^2 + cx_1x_2 + dx_1 + ex_2 + f \ ,$$

and define five "candidate points" $x_l := \min_{x\in R} x$, $x_u := \max_{x\in R} x$, $x^* := \frac{cx_2+d}{-2a}$, $x_- := \max_{x\in R, x<x^*} x$, and $x_+ := \min_{x\in R, x>x^*} x$. Then

- if u is constant in x_1 (i.e., $a = 0$ and $cx_2 + d = 0$), $\mathcal{B}_1(x_2) = R$
- if u is linear in x_1 with positive slope (i.e., $a = 0$ and $cx_2 + d > 0$), $\mathcal{B}_1(x_2) = \{x_u\}$
- if u is linear in x_1 with negative slope (i.e., $a = 0$ and $cx_2 + d < 0$), $\mathcal{B}_1(x_2) = \{x_l\}$
- if u is convex in x_1 (i.e., $a > 0$), $\mathcal{B}_1(x_2) = \arg\max_{x\in\{x_l,x_u\}} u(x)$
- if u is concave in x_1 (i.e., $a < 0$) and $x^* \in R$, $\mathcal{B}_1(x_2) = \{x^*\}$
- if u is concave in x_1 (i.e., $a < 0$) and $x^* \notin R$, $\mathcal{B}_1(x_2) = \arg\max_{x\in\{x_-,x_+\}} u(x)$.

Note that $\mathcal{B}_1(x_2)$ is not necessarily unique (or even a finite set). To find the NE, observe that the unrestricted best response functions $\mathcal{B}_1(x_2) = -\frac{c_1x_2+d_1}{2a_1}$ and $\mathcal{B}_2(x_1) = -\frac{c_2x_1+e_2}{2b_2}$ lead to the unique unrestricted NE

$$\boldsymbol{x}^* = (x_1^*, x_2^*) = \left(\frac{c_1e_2 - 2d_1b_2}{4a_1b_2 - c_1c_2}, \frac{c_2d_1 - 2e_2a_1}{4a_1b_2 - c_1c_2}\right) \ .$$

If this point exists and is allowed by R, i.e., $4a_1b_2 - c_1c_2 \neq 0$ and $\boldsymbol{x}^* \in R^2$, then $\mathcal{N}|_R = \{\boldsymbol{x}^*\}$. Otherwise, we use fictitious play (i.e., successive mutual best responses) to find the fixed points, repeatedly calling the restricted best response functions while maintaining a list of candidate solutions.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

B Expected Results for the Cournot Game

For the (unrestricted) PCG, the unique best responses are $\mathcal{B}_1(q_2) = \frac{\lambda-q_2}{2}$ and $\mathcal{B}_2(q_1) = \frac{\lambda-q_1}{2}$. Therefore, we get

$$\begin{aligned} D(\boldsymbol{q}) &= \left(q_1 - \frac{\lambda - q_2}{2}\right)^2 + \left(q_2 - \frac{\lambda - q_1}{2}\right)^2 \\ &= \frac{5}{4}(a^2 + b^2) + 2ab - \frac{3}{2}\lambda(a + b) + \frac{1}{2}\lambda^2 \end{aligned}$$

which has a unique global minimum $\boldsymbol{q}^* = (\frac{\lambda}{3}, \frac{\lambda}{3})$ with $D(\boldsymbol{q}^*) = 0$.

If we allow interval union restrictions for the PCG, best responses are not unique anymore, but still follow a simple pattern: If the unrestricted best response q^* is not part of an allowed interval, the restricted best responses are the closest allowed actions on either one or both sides of q^*.

More formally: Let $q^* := \frac{\lambda}{3}$ be the unrestricted optimal quantity, and define, for a given restriction $R \subseteq [0, \lambda]$, the two closest allowed quantities $q^+ := \min_{q\in R}(\{q > q^*\})$ and $q^- := \max_{q\in R}(\{q < q^*\})$. Setting $\Delta^+ := q^+ - q^*$ and $\Delta^- := q^* - q^-$, the Nash Equilibria $\mathcal{N}|_R$ of the restricted PCG are:

$$\mathcal{N}|_R = \begin{cases} \{(q^+, q^+)\} & \text{if } \Delta^+ < \frac{1}{2}\Delta^- \\ \{(q^+, q^+), (q^+, q^-), (q^-, q^+)\} & \text{if } \Delta^+ = \frac{1}{2}\Delta^- \\ \{(q^+, q^-), (q^-, q^+)\} & \text{if } \frac{1}{2}\Delta^- < \Delta^+ < 2\Delta^- \\ \{(q^-, q^-), (q^+, q^-), (q^-, q^+)\} & \text{if } \Delta^+ = 2\Delta^- \\ \{(q^-, q^-)\} & \text{if } \Delta^+ > 2\Delta^- \end{cases} .$$

This suggests the following sequence of successive restrictions for the SOAR algorithm:

- Identify $\frac{\lambda}{3}$ as the unique relevant action of the unrestricted game and therefore exclude $\overline{R} := \left[\frac{\lambda}{3} - \epsilon, \frac{\lambda}{3} + \epsilon\right)$ from the action space
- Identify both boundary actions as relevant and exclude one of them, increasing the excluded region $\overline{R}$ around $\frac{\lambda}{3}$
- Whenever $\overline{R}$ becomes imbalanced by a factor of > 2 around $\frac{\lambda}{3}$, a symmetric equilibrium appears at one end of it
- Finally, $\overline{R}$ is large enough to produce the symmetric equilibrium $(\frac{\lambda}{4}, \frac{\lambda}{4})$
- This occurs when $\overline{R} = [\frac{\lambda}{4}, \frac{\lambda}{2})$, and therefore $R = [0, \frac{\lambda}{4}) \cup [\frac{\lambda}{2}, \lambda)$

- The algorithm goes on to enlarge $\overline{R}$ until the set of allowed actions becomes empty
- Since no further restriction produces a socially better stable solution, the largest (i.e., least restrictive) R with $(\frac{\lambda}{4}, \frac{\lambda}{4}) \in \mathcal{N}|_R$ is finally returned as the optimal restriction R^*
- The resulting degree of restriction is $\mathfrak{r}(R^*) = 25\%$

The optimal restriction R^* has the unique equilibrium $(\frac{\lambda}{4}, \frac{\lambda}{4})$ which gives the SSU $\mathcal{S}(R^*) = \mathfrak{u}(\frac{\lambda}{4}, \frac{\lambda}{4}) = \frac{1}{4}\lambda^2$. In contrast, the unrestricted game produces a unique equilibrium of $(\frac{\lambda}{3}, \frac{\lambda}{3})$, such that $\mathcal{S}(A) = \mathfrak{u}(\frac{\lambda}{3}, \frac{\lambda}{3}) = \frac{2}{9}\lambda^2$. The resulting relative improvement is $\Delta_{rel} = \frac{1}{8}$.

C Number of Oracle Calls in the Cournot Game

To show that the number of oracle calls for SOAR's solution of the Cournot Game grows quadratically rather than exponentially, let us fit the two curves $f_1(\lambda) = ae^{b\lambda} + c$ and $f_2(\lambda) = a\lambda^2 + b\lambda + c$ to the data and check their deviation. Recall that the experimental data is $f(10) = 912$, $f(11) = 1095$, $f(12) = 1294$, $f(13) = 1513$, and so on (the full data set can be reproduced using the Colab notebook in the supplementary material).

As can be seen from Figure 1, the quadratic interpolation polynomial f_2 gives a close-to-perfect fit with parameters $a = 8.33$, $b = 8.00$, and $c = -1.45$. In contrast, the exponential fit with f_1 produces the degenerate parameter values $a = 0.00$, $b = 1.00$, and $c = 1.12 \cdot 10^{62}$.

D Continuous Braess' Paradox

In the original (discrete) version of Braess' Paradox (see Figure 1a in the main paper), each agent has three route options, of which they choose exactly one. The travel time from 0 to 3 is then used as their cost function (i.e., it is to be minimized).

When transforming this into a one-dimensional continuous NFG, we have to address two points: (a) There has to be a continuum of actions, and (b) we need utility functions instead of cost functions. Therefore, we define the action space as $A = [0, 1]$ and give it the following meaning: Agent 1 routes a flow of x_1 through route 0-1-2-3, and the remaining flow of $(1 - x_1)$ through route 0-2-3. Similarly, agent 2 routes a flow of x_2 through route 0-1-2-3, and the remaining flow of $(1 - x_2)$ through route 0-1-3. This means that, for both agents, 0 is the "cooperative" action, while 1 is the "competitive" action. The edge weights are adjusted such that full utilization (which is now a flow of 2 along an edge) gives the same travel time as utilization of 1 in the original setting (see Figure 2).

We calculate the expected travel time $c_i(\boldsymbol{x})$ for both agents and subtract them from a virtual baseline of 32 in order to get the utility functions $u_i(\boldsymbol{x})$. The expected travel time along a route is the flow on the route, multiplied by the sum of the edge weights $w_i(\boldsymbol{x})$, given this flow. For agent 1,

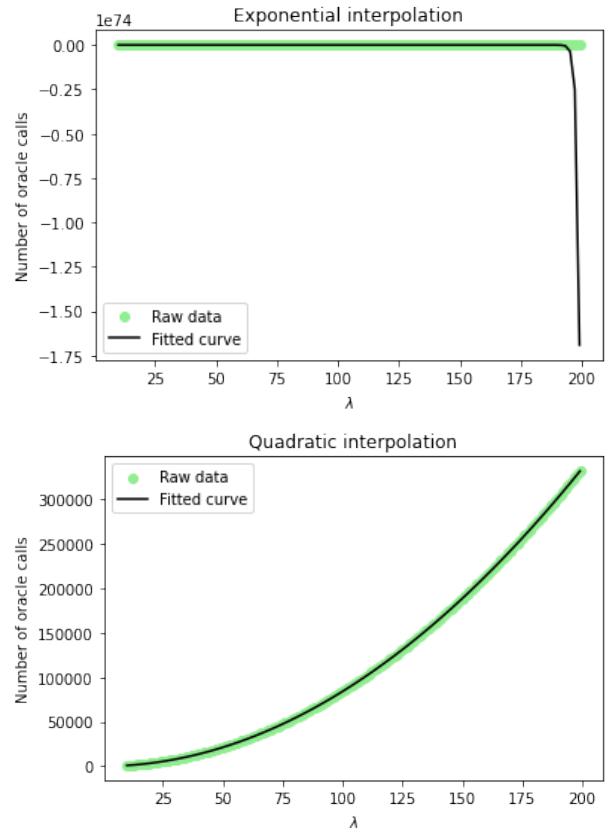


Figure 1: Exponential and quadratic interpolation of the number of oracle calls in the Cournot Game

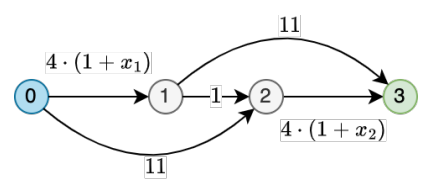


Figure 2: Continuous version of Braess' Paradox

this calculation is:

$$\begin{aligned} c_1(\boldsymbol{x}) &= x_1(w_{01} + w_{12} + w_{23}) + (1 - x_1)(w_{02} + w_{23}) \\ &= x_1(w_{01} + w_{12}) + (1 - x_1) \cdot w_{02} + w_{23} \\ &= x_1(4(1 + x_1) + 1) + 11(1 - x_1) + 4(1 + x_2) \\ &= 4x_1^2 - 6x_1 + 4x_2 + 15 \ , \end{aligned}$$

and the corresponding utility function is

$$u_1(x_1, x_2) = -4x_1^2 + 6x_1 - 4x_2 + 17 \ .$$

In the same way, we get

$$u_2(x_1, x_2) = -4x_2^2 - 4x_1 + 6x_2 + 17 \ .$$

To generalize this setting, let us assume affine the weight functions $w_{0\text{-}2}(\boldsymbol{x}) = w_{1\text{-}3}(\boldsymbol{x}) = a(x_1 + x_2) + b$ and

$w_{0\text{-}1}(\boldsymbol{x}) = w_{2\text{-}3}(\boldsymbol{x}) = c(x_1 + x_2) + d$, while leaving the constant weight $w_{1\text{-}2}(\boldsymbol{x}) = 1$ unchanged. This gives the parameterized utility functions

$$u_1(\boldsymbol{x}) = -(a+c)x_1^2 + (2a+b-c-1)x_1 - cx_2 + (4c+d+1)$$

and

$$u_2(\boldsymbol{x}) = -(a+c)x_2^2 - cx_1 + (2a+b-c-1)x_2 + (4c+d+1)\ .$$

To obtain a one-dimensional range of experiments, we fix $a = 0$, $c = 4$ and $d = 0$ and vary b (intuitively, we vary the attractiveness of taking the social routes, compared to the selfish route). The parameterized utility functions $u_i(\boldsymbol{x})$ are therefore

$$u_1(\boldsymbol{x}) = -4x_1^2 + (b-5)x_1 - 4x_2 + 17$$

and

$$u_2(\boldsymbol{x}) = -4x_2^2 - 4x_1 + (b-5)x_2 + 17\ .$$

E Expected Results for the Braess Paradox

From u_1 and u_2 as defined above, we can immediately derive the best response functions $\mathcal{B}_i(x_j) = \frac{b-5}{8}$, resulting in $\mathcal{N} = \left\{\left(\frac{b-5}{8}, \frac{b-5}{8}\right)\right\}$. Moreover, since $\mathfrak{u}(\boldsymbol{x}) = u_1(\boldsymbol{x}) + u_2(\boldsymbol{x}) = -4x_1^2 - 4x_2^2 + (b-9)x_1 + (b-9)x_2 + 34$, we get the social optimum $\boldsymbol{x}^* = \left(\frac{b-9}{8}, \frac{b-9}{8}\right)$.

Finally, we conclude from $A = [0,1]$ that, for $b \leq 5$ and $b \geq 17$, $\mathcal{N} = \{\boldsymbol{x}^*\}$ such that the unrestricted and the restricted MESU are equal. For $b \in (5,17)$, however, the two values differ, such that restricting A can improve the MESU.

Let us first assume that $b \in (5,9]$. To make $\frac{b-9}{8}$ a best response for a player, we have to exclude any action from A that this player would prefer over $\frac{b-9}{8}$. It is easy to see that the range of actions that needs to be excluded is $(0, \frac{b-5}{4})$, giving the unique optimal restriction $R^* = \{0\} \cup [\frac{b-5}{4}, 1]$.

For $b \in [9,17)$, a similar analysis yields that $(\frac{b-9}{8}, 1)$ needs to be excluded, and therefore $R^* = [0, \frac{b-9}{8}]$.

From the optimal restriction R^*, we can calculate the unrestricted and restricted MESU as well as the degree of restriction:

$$\mathcal{S}(A) = \begin{cases} 34 & \text{for } b \leq 5 \\ \frac{1}{8}\left(b^2 - 18b + 337\right) & \text{for } b \in [5,13] \\ 2b + 8 & \text{for } b \geq 13 \end{cases},$$

$$\mathcal{S}(R^*) = \begin{cases} 34 & \text{for } b \leq 9 \\ \frac{1}{8}(b-9)^2 + 34 & \text{for } b \in [9,17] \\ 2b + 8 & \text{for } b \geq 17 \end{cases},$$

and

$$\mathfrak{r}(R^*) = \begin{cases} 0 & \text{for } b \leq 5 \\ \frac{b-5}{4} & \text{for } b \in [5,9] \\ \frac{17-b}{8} & \text{for } b \in [9,17] \\ 0 & \text{for } b \geq 17 \end{cases}.$$