Automated Design of Affine Maximizer Mechanisms In Dynamic Settings

Anonymous Author(s)
Affiliation
Address
email

Abstract

Dynamic mechanism design is a challenging extension to ordinary mechanism design in which the mechanism designer must make a sequence of decisions over time, in the face of possibly untruthful reports of participating agents' preferences. Optimizing dynamic mechanisms for welfare is relatively well understood. There has been less work on optimizing for other goals (*e.g.*, revenue) and without restrictive assumptions on valuations it may be challenging to find optimal mechanisms. The situation is similar in static mechanism design. However, optimization/machine learning-based automated mechanism design techniques have been successful in finding high-revenue mechanisms in cases beyond the reach of analytical results. Our aim is to make such approaches dynamic. We extend the class of affine maximizer mechanisms to work on MDPs where agents may untruthfully report their rewards. We define a bilevel optimization problem in which the upper problem involves choosing optimal mechanism parameters, and the lower problem involves solving a resulting MDP. Our approach can find truthful dynamic mechanisms which achieve strong performance on goals other than welfare, and can be applied to essentially any problem setting – without restrictions on valuations – for which RL can learn (near-)optimal policies.

1 Introduction

Dynamic mechanism design studies sequential decision-making problems, where decisions are based on the self-reported preferences of agents. A typical model is that the environment consists of a *Markov decision process (MDP)*, and the mechanism controls the process given reported utilities by the agents.

Obviously, this includes static mechanism design as a special case, yet that is already a challenging problem. Of course, to maximize welfare while ensuring truthfulness, one can use the celebrated Vickrey-Clarke-Groves (VCG) mechanism [50, 15, 23]. For optimizing the goal of revenue, Myerson [36] completely settles the question under the important assumption that agents' types are single-parameter (essentially, that there is only one type of item up for sale).

Beyond these cases, analytic solutions have been hard to come by, motivating *automated mechanism design (AMD)* [16, 44, 19] – a data-driven search through some class of mechanisms in order to find one that performs well while satisfying the constraints of strategyproofness and individual rationality. More recently, Duetting et al. [22] has introduced the use of tools from modern deep learning to approximate strategyproof mechanisms, with many further extensions [20, 18, 26, 42, 41]. Other work has used similar learning techniques to search within restricted classes of mechanisms which are exactly strategyproof [19, 48, 21].

Submitted to 16th European Workshop on Reinforcement Learning (EWRL 2023). Do not distribute.

The situation is similar in dynamic mechanism design. There is work focusing on maximizing welfare, developing dynamic analogues to the VCG mechanism [6, 38, 33]. Some other work considers revenue [10, 27, 24, 25]—but (as in static revenue-maximizing mechanism design) needs to make restrictive assumptions about the space of agent types. The use of automated mechanism design for dynamic problems, though, has been much more limited: that is the focus of our present work.

We consider mechanism design problems on Markov decision processes. Our model works with general loss functions (including revenue but also other domain-specific loss functions), not just the easier goal of welfare, and we don't assume that agent types must be one-dimensional. In particular, our assumptions are sufficiently general to capture essentially all of static multi-parameter mechanism design as a special case.

Like previous work for static mechanisms, we focus on the restricted class of mechanisms known as affine maximizers (AMAs)[1]. We justify this choice in two ways. First, as mentioned, our problem assumptions capture multi-parameter mechanism design – but finding optimal mechanisms in this setting has proven extremely difficult, so it makes sense to search within a more tractable class of mechanisms. Second, our framing of the problem is broad enough that it captures problem instances where Roberts' theorem [43] applies, which states that in full generality, *only* affine maximizers can be truthful.

We frame the search for a high-performing **dynamic affine maximizer mechanism** as a **bilevel optimization problem**, where the outer problem consists of choosing weights and (possibly state-dependent) boosts – the AMA parameters – to minimise a given loss function. The inner problem consists of learning to control an MDP to maximise affinely-transformed social welfare, given the weights, boosts, and agents' type reports. (See equation 2 below.)

For our particular implementations, we solve the inner problem via (regularized) linear programming and differentiate through the solution with respect to the weights and boosts. In a number of mechanism design settings, including auctions and task scheduling, this approach results in truthful mechanisms that outpeform the VCG baseline.

2 Related Work

2.1 Maximizing Welfare In Dynamic Mechanism Design

Athey and Segal [6] consider a dynamic mechanism design setting where agents update their beliefs over time, and where the goal is an efficient and budget-balanced outcome. Parkes, in [38], describes a dynamic mechanism design setting where the focus is on agents who may arrive and depart at different periods. Both of these approaches simply assume an optimal policy is available. More recently, Lyu et al. [33] presents a model for dynamic mechanism design in an offline RL setting; another work focuses on the online case [32]. Bergemann and Välimäki [10] presents a VCG-like "dynamic pivot mechanism".

There are different modeling choices and goals in each of these approaches, but the common theme is that the allocation problem involves making decisions on some MDP after observing agent reports. In a dynamic analogue to VCG mechanisms, the mechanism designer acts according to the welfare-optimal policy for this MDP, and additionally charges agents their externalities to ensure incentive compatibility.

2.2 Dynamic Mechanism Design for Goals Other Than Welfare

One can also consider dynamic mechanism design with goals other than welfare, typically revenue. Pavan et al. [40] and Kakade et al. [27] both consider cases where the private information about the value of the item is 1-dimensional. This allows for a Myerson-style analysis of the actual profit-maximizing mechanism. Other work considers optimal selling of items to agents who arrive and depart over time from the perspective of optimal stopping, but again considers single-parameter

[1]The acronym AMA refers to "affine maximizer auctions". We consider affine maximizer *mechanisms*, but we stick to the AMA acronym because it is widely used and to avoid confusion with AMM used to refer to "automated market makers".

item valuations [24, 25, 28]. Bergemann and Välimäki [11] surveys other results that make similar assumptions for tractability. Pai and Vohra [39] considers a similar setting and finds the optimal Bayes-Nash incentive compatible mechanism. Still other work considers settings where the mechanism designer may update the mechanism online over time (e.g. by changing reserve prices) [47] and where bidders may even strategically attempt to manipulate the learning process [4].

2.3 Learned preference elicitation and multistage mechanisms

Another line of work considers iterative preference elicitation. Dynamic mechanism design is typically concerned with making a sequence of decisions over time; these mechanisms, by contrast, make a single final decision, but elicit agents' preferences in multiple stages. Often the preference elicitation models make use of machine learning [13, 30, 49, 52, 14].

In this context, Sandholm et al. [46] uses automated mechanism design to first find good mechanisms (for general type spaces) and then convert them into multistage mechanisms – in other words, the elicitation protocol itself is "learned".

2.4 Static automated mechanism design and differentiable economics

Due to the difficulty of analytically finding optimal mechanisms, a number of works have instead attempted to treat static mechanism design as a machine learning problem, and learn good mechanisms from samples. The success of such techniques in static settings is a motivation for our work.

One line of work makes use of static affine maximizers and achieves good performance in multi-item multi-bidder auctions [45, 19, 21]. There is also a line of learning theory research on choosing the AMA parameters given samples from the valuation distribution [9, 7, 8].

Another direction is to start with perhaps non-strategyproof mechanisms and iteratively modify to improve strategyproofness – this is known as incremental mechanism design [17]. One line of work in this direction makes use of rich function approximators to learn mechanisms – this broad class of research is sometimes referred to as "differentiable economics". Duetting et al. [22] presents one influential direction, which uses neural networks to optimize revenue and a penalized loss to approximately enforce strategyproofness. Shen et al. [48] presents an alternative approach for single bidder settings which can cope with broader classes of utility functions.

3 Our Approach

Below, we describe our mechanism design problem at a high level. The nature of the problem and our mechanism design desiderata motivate our choice to restrict attention to affine maximizer mechanisms. We then describe in more detail how these mechanisms work in a dynamic setting.

3.1 Problem Setting and Mechanism Design Desiderata

Environment and policy/allocation rule At the most general level, consider some "reward-free" MDP $\mathcal{M} = (S, A, T)$, where S is a set of states, A is a set of actions, and P is a transition function. Now suppose there are n agents, each with their own reward function $r_i : S \times A \to \mathbb{R}$. We emphasize that these agents are not themselves taking actions in the MDP – this is done by the mechanism. Their only choice will be which rewards to report to the mechanism. We assume the mechanism designer wants to maximize some objective function, which may depend on the agents' expected rewards over trajectories of the MDP. The MDP policy thus corresponds to the allocation rule in traditional mechanism design.

Incentive compatibility and payments In order to achieve this goal, the mechanism will need access to the true $r_i(s, a)$. But in general, we should expect the agents to misreport their reward function if they think it will benefit them. Thus, we will allow for some side payments to be made based on the MDP solution and the agents' reports, in order to ensure that there is no incentive to misreport – making the mechanism *incentive compatible* (IC). (Note that such payments only exist for some choices of π.) We assume agent utility is quasilinear – i.e. payments just correspond to negative reward.

The mechanism designer's goal may also relate to these payments. For example, a canonical mechanism design goal is to maximize revenue – in our setting, choose a policy π^* such that payments can be made as high as possible while still ensuring IC.

Individual rationality We also want to guarantee some notion of individual rationality (IR), meaning that agents should not be charged so much that they receive negative utility and would be better off not participating in the mechanism.

3.2 Affine Maximizers

As motivated above our goal is to choose some π such that IC/IR side payments can be constructed, while also performing as well as possible on the mechanism designer's higher-level objective.

Unlike prior work e.g. [27, 10], we do not make any assumptions about the structure of the reward functions. Our setting is therefore general enough to incorporate many hard problems, e.g. optimal multi-item mechanism design, and is thus at least as hard as those problems. So hoping to get the truly best-performing π, even only in an infinite-sample/asymptotic sense, is too much. Thus it is appropriate to restrict attention to some more tractable class of mechanisms.

Also, our problem setting is general enough to include situations where a result known as Roberts' theorem applies [43]. This result says that under certain conditions (arbitrary rewards, at least 3 outcomes), the only allocation rules that can possibly have IC payments take the form of affine maximizers.

Definition 3.1 (Affine maximizers)**.** *Given so-called weights $w \in R^n_+$, boosts $b \in \mathbb{R}^{|S|\times|A|}$, a dynamic affine maximiser mechanism (AMA) takes reported rewards $\boldsymbol{r} \in \mathbb{R}^{|S|\times|A|\times n}$ and returns a policy $\pi^*(w, b, \boldsymbol{r})$ on the MDP that maximizes the affine social welfare $asw^{(\pi)}(w, b, \boldsymbol{r})$ where*

$$asw^{(\pi)}(w, b, \boldsymbol{r}) = \mathbb{E}_\pi \left[\sum_{t=0}^{T} \left(\sum_{i=1}^{n} w_i r_i(s_t, a_t) \right) + b(s_t, a_t) \right]$$

Defining $\mathsf{asw}(w, b, \boldsymbol{r}) = asw^{(\pi^*(w,b,\boldsymbol{r}))}(w, b, \boldsymbol{r})$ as the maximum affine social welfare for reports $\boldsymbol{r}$, the resulting payment is then

$$\begin{aligned} \boldsymbol{p}_i(w, b, \boldsymbol{r}) &= \frac{1}{w_i} \left(\mathsf{asw}^{(-i)}(w, b, \boldsymbol{r}) - \left(\mathbb{E}_{\pi^*(w,b,\boldsymbol{r})} \left[\sum_{t=0}^{T} \left(\sum_{j \neq i} w_j r_j(s_t, a_t) \right) + b(s_t, a_t) \right] \right) \right) \\ &= \frac{1}{w_i} \left(\mathsf{asw}^{(-i)}(w, b, \boldsymbol{r}) - \mathsf{asw}(w, b, \boldsymbol{r}) + \sum_{t=0}^{T} w_i \mathbb{E}_{\pi^*(w,b,\boldsymbol{r})}[r_i(s_t, a_t)] \right) \end{aligned} \tag{1}$$

where $\mathsf{asw}^{(-i)}(w, b, \boldsymbol{r}) = \arg\max_\pi \mathbb{E}_\pi \left[\sum_{t=0}^{T} \left(\sum_{j=1, j\neq i}^{n} w_j r_j(s_t, a_t) \right) + b(s_t, a_t) \right]$ is the maximum affine social, when disregarding i.

No matter the choice of weights and boosts, every resulting AMA is strategyproof ex-post – after learning the rewards of the other agents, reporting truthfully is a dominant strategy – and IR in expectation over the MDP trajectories. The proof can be found in Appendix A.

4 Dynamic Mechanism Design as Bilevel Optimisation

The problem of searching for a performant mechanism within the class of AMAs can be naturally formulated as stochastic bilevel optimisation. Let $\mathcal{L}(\pi, \theta)$ be the loss function of the mechanism designer where $\theta = (w, b)$ denotes the weights and boosts parameterizing the AMA, then the problem can be formulated as follows:

$$\min_{w,b} \mathbb{E}_{\boldsymbol{r}\sim f}[\mathcal{L}(\pi^*, w, b)] \text{ s.t. } \pi^* \in \arg\max_\pi \mathbb{E}_{s_t, a_t \sim \pi} \left[\sum_{t=0}^{T} \left(\sum_{i=1}^{n} w_i r_i(s_t, a_t) \right) + b(s_t, a_t) \right] \tag{2}$$

Given the bilevel structure, we can think of the problem as a game between a leader and a follower.

- The leader knows only the distribution f from which the set of reward functions r_i are drawn according to $f_i \sim r_i$, and chooses weights w_i and boosts $b(s, a)$.
- For any draw of r_i and the weights and boosts, the follower acts optimally ("best responds") in the MDP according to the AMA objective.

To clarify (and to contrast with many models of mechanism design where the agents making reports are treated as followers): the leader and follower are only "fictional": in reality, there is only one mechanism designer.

It may seem strange for the mechanism designer to pretend to be two different agents, one of whom knows less information than the other. However, the reason for doing this is to ensure that the resulting mechanism is a valid AMA, and thus will truly be incentive compatible. If, for instance, weights or boosts were chosen after learning about drawn r_i, incentive compatibility would be destroyed.

In general, the problem above – bilevel optimization, with stochasticity in the leader's objective – is quite difficult. Indeed, the case of revenue-maximisation in one-round auctions with multiple goods, which is a very special case of our much more general problem, remains essentially unsolved. Therefore finding a globally optimal solution to the above problem is too much to hope for, but we show that derivatives do exist and we can differentiate through a smoothed LP whose local stationary points are close to those of the original problem.

4.1 Existence of derivatives

So far, we have not made any assumptions about the loss function. A very natural desideratum would be that $\mathcal{L}(\pi^*(\theta, \boldsymbol{r}), \theta)$ is differentiable. As we will see, this is unlikely to hold in general. However, we prove that the relaxed condition that $E_{\boldsymbol{r}}[\mathcal{L}(\pi^*(\theta, \boldsymbol{r}), \theta)]$ is differentiable holds for revenue – arguably the most important loss function in mechanism design literature and a challenging one to optimize. We further show that for expected revenue (and other loss functions whose expected value is differentiable) we can approach locally optimal solutions by differentiating through a regularized LP formulation of the MDP.

The expected revenue of a dynamic AMA mechanism is given by:

$$\begin{aligned}\mathsf{rev}(w, b) &= \mathbb{E}_{\boldsymbol{r}}\left[\sum_{i=1}^{n} \boldsymbol{p}_i(w, b, \boldsymbol{r})\right] \\ &= \mathbb{E}_{\boldsymbol{r}}\left[-\sum_{i=1}^{n}(\frac{1}{w_i}\mathsf{asw}(w, b, \boldsymbol{r})) + \mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r}) + \sum_{i=1}^{n}\frac{1}{w_i}\mathsf{asw}^{(-i)}(w, b, \boldsymbol{r})\right]\end{aligned}$$

(Here sw, the non-transformed social welfare, is just asw when weights are all 1 and boosts are all 0, i.e. $\mathsf{sw}^{(\pi)} = \mathbb{E}_{\pi}\left[\sum_{t=0}^{T}\sum_{i=1}^{n} r_i(s_t, a_t)\right]$.)

First we show that this is differentiable almost everywhere (a.e.).

Theorem 4.1. *The revenue in a dynamic AMA mechanism is continuous and a.e. differentiable.*

Proof Idea. The full proof can be found in Appendix B. In brief, we identify boosts and agent's rewards with vectors in $\mathbb{R}^{|S|\times|A|}$ and weights with vectors in $\mathbb{R}^n_+$. To prove the statement it suffices to show asw and sw are continuous and differentiable. Affine social welfare is the maximum over a set of functions which are linear in w and b, so bounding the change as a result of changing weights and boosts is straightforward – one gets a Lipschitz constant of $T\mathbb{E}\left[\|\boldsymbol{r}\|_\infty\right]$.

Showing local Lipschitzness of expected social welfare is rather more involved. One can argue that, for any w, b, the set of rewards for which two alternative policies give the same *affine* social welfare, lie on a hyperplane.

Changing the weights and boosts may change the optimal policy, and therefore may result in differing social welfare for some subset of the rewards. We would like to bound the volume of the set of rewards where this can happen. Indeed, the rewards where the optimal policy may change, under a change parameters, all lie in between the aforementioned hyperplanes. The distance between the hyperplanes depends on the change in w and b, so the volume can be bounded, which, assuming the probability density decays sufficiently fast, results in a local Lipschitz constant. □

We note that for $\mathsf{asw}(w, b, r)$ the gradients can be computed straightforwardly.[2] For this, rewrite asw using the state-action occupancy measure ν_π. [3] Let Π be the finite set of deterministic policies.[4] We have:

$$\mathsf{asw}(w, b, r) = \max_{\pi \in \Pi} \sum_{s,a} \nu_\pi(s, a)(\sum_{i=1}^{n} w_i r_i(s, a) + b(s, a))$$

As we are taking the maximum over a finite set of functions which are differentiable in w, b, we can apply a multi-dimensional version of the envelope theorem [35] to get

$$\nabla_{w,b}\mathsf{asw}(w, b, r) = \nabla_{w,b} \sum_{s,a} \nu_{\pi^*(w,b,r)}(s, a)(\sum_{i=1}^{n} w_i r_i(s, a) + b(s, a)) \quad a.e.$$

and hence a.e.

$$\nabla_{w_i}\mathsf{asw}(w, b, r) = \sum_{s,a} \nu_{\pi^*(w,b,r)}(s, a) r_i(s, a) = \mathbb{E}_{\pi^*(w,b,r)}[\sum_{t=0}^{T} r_i(s, a)] \quad \forall i \tag{3}$$

$$\nabla_{b(s,a)}\mathsf{asw}(w, b, r) = \nu_{\pi^*(w,b,r)}(s, a) \quad \forall s, a \tag{4}$$

Boosting a state increases asw in relation to how often the state is visited under the optimal policy. Similarly increasing the weight of an agent changes asw in proportion to the expected sum of rewards the agent gets. In fact, ν_{π^*} is the solution to the linear programming formulation of the MDP.

4.2 Linear programming formulation

We now describe the linear-programming formulation for an episodic MDP (where the time step is distinguishable in the state) [3]. In particular, we suppose the follower solves the following LP to find the optimal state-action occupancy measure $\nu_{\pi^*(w,b,\boldsymbol{r})}$[5]:

$$\begin{aligned} \max \sum_{s \in S, a \in A} & \left(\sum_i w_i r_i(s, a) + b(s, a) \right) \nu(s, a) \\ \text{s.t.} \sum_{a \in A} \nu(s, a) &= \sum_{s' \in S, a' \in A} P(s|s', a')\nu(s', a') + \mu_0(s) \quad \forall s \in S \\ \nu(s, a) &\geq 0 \quad \forall s \in S, a \in A \end{aligned} \tag{5}$$

As we argue in the Appendix, $\nabla_{w,b}\mathsf{sw}^{\pi^*(w,b,r)}(\boldsymbol{r}) = 0$ a.e. and – akin to asw – is undefined for all $\boldsymbol{r}$ where more than one policy is optimal. This prevents us from estimating $\nabla_{w,b}\mathsf{rev}(w, b)$ from samples as we cannot exchange the gradient and the expected value. Indeed it seems that variations of this problem are likely to arise for general loss functions other than revenue. When small changes in the weights and boosts lead to jumps from one deterministic policy to another, it can generally be expected that the loss function will exhibit these discontinuities as well.

While zeroth order methods have been proposed in certain related settings [12, 34] to cope with this type of problem, we instead propose to regularize our problem to get a smooth surrogate of the derivative.

We add a small entropy regularizer to the follower's objective. (A technique equivalent to such regularization has also been used to deal with a similar issue in first-order computation of equilibria of non-truthful static auctions [29]).

[2] The analysis holds equivalently for $\mathsf{asw}_{-i}(w, b, r)$

[3] In general finite horizon MDPs this would be defined as $\nu_\pi(s, a) = \sum_{t=1}^{T} \mathbb{P}_\pi(s_t = s, a_t = a)$. In our experiments we always assume the states contain the current timestep such that this simplifies to $\nu_\pi(s, a) = \mathbb{P}_\pi(s_t = s, a_t = a)$

[4] The optimal follower policy will always be deterministic.

[5] This corresponds to the optimal policy by $\pi^*(a|s) = \frac{\nu_{\pi^*(w,b,\boldsymbol{r})}(s,a)}{\sum_a \nu_{\pi^*(w,b,\boldsymbol{r})}(s,a)}$.

The result is now a convex program:

$$\begin{aligned}
\max \sum_{s\in S, a\in A} \left(\sum_i w_i r_i(s,a) + b(s,a) \right) \nu(s,a) - \alpha \sum_{s\in S, a\in A} \nu(s,a) \log \nu(s,a) \\
\text{s.t.} \sum_{a\in A} \nu(s,a) = \sum_{s'\in S, a'\in A} P(s|s',a')\nu(s',a') + \mu_0(s) \quad \forall s \in S \\
\nu(s,a) \geq 0 \quad \forall s \in S, a \in A
\end{aligned} \tag{6}$$

When the objective is strongly convex, the solution map $\nu_r^* : (w,b) \mapsto \nu_{\pi^*(w,b,\boldsymbol{r})}$ is smooth. Therefore, derivatives of revenue can now be estimated from the gradients at sampled type profiles, calculated using reverse-mode automatic differentiation [2, 1].

Adding a small amount of regularisation does not change the leader's (at least if the loss function is well-behaved) and follower's problem and solutions significantly. Indeed it has been shown in [51] that the distance between the solutions of Equations (5) and (6) decays exponentially fast in the regularisation constant α.

Along with Theorem 5 of [51] this implies exponential convergence of the follower's objective (asw) of the regularized and unregularized solutions, which, because revenue can be written as a linear combination of different affine social welfares, can be used to show exponential convergence of the leader's loss function (revenue).

5 Experiments

To test this regularized linear programming approach, we optimize AMAs in dynamic mechanism design settings on tabular MDPs. In most cases, we find that optimizing weights and boosts significantly improves revenue or other performance measures compared to a VCG baseline. Results are heavily initialization-dependent – for some initializations of boosts, there was little or no improvement. Given the non-convex loss landscape of expected revenue (or other performance measures) with respect to AMA parameters, this is to be expected, and there can be no guarantee that we find the best mechanism parameters. Thus we consider multiple random seeds and report average improvements as well as the best observed improvements across all random seeds.

Implementation details We compute derivatives with respect to social welfare using the regularized program 6, with smoothing parameter 10^{-2} except where mentioned. We solve the regularized program using MOSEK [5] and use the DiffOpt package within JuMP [31] to differentiate. Our experiments were run on CPU cluster nodes with 12 cores each; larger-scale experiments varied in runtime but took about 4-6 hours on one such node for all 9 random seeds per experiment. (More experiments than are reported here were run for development and debugging purposes.) As a computational heuristic, we do not compute full derivatives for affine social welfare terms when calculating revenue, instead simply computing exact derivatives in the form given above. For each stochastic gradient step, we sample 20 type profiles and optimize with learning rate 10^{-2}; when evaluating revenue, we sample 1000 type profiles and do not use regularization. Thus at evaluation time, the LP solution is exactly correct, thereby ensuring strategyproofness.

5.1 Sequential Sales

We consider a simple setting in which identical items are sold sequentially to unit-demand bidders. The states consist of a record of who has received the item; the allowed actions are to sell the item to some bidder, or to no one. The welfare maximizing mechanism thus involves giving the items to the highest bidder, but by altering the boosts, revenue can be increased by sometimes withholding the item. We consider a distribution type profiles drawn uniformly on $[0, 1]$, with results in table 1. For the 4 agent, 2 item case, we observed extremely poor performance until changing the learning rate to 0.05; this did not occur in other cases where we used 0.01. In this setting, we fix the weights to 1.

We observe that optimizing the boosts can improve performance compared to VCG, especially when there are not tight supply constraints. Intuitively, if there is a large supply of goods, VCG revenue should be low, as agents do not cause much externality on other agents. By setting boosts to effectively withhold goods (like a reserve price), revenue can be increased.

Table 1: Results for optimizing auction revenue in a sequential sales setting (n agents, m sales) with symmetric uniformly-distributed types.

n	m	Avg. VCG rev.	Avg. AMA rev	Avg. VCG Std. Err	Avg. AMA Std. Err.	Best VCG rev.	Best AMA rev.	Avg. % imp.	% Best Imp.
2	2	0	0.4160	0.00	0.01	0	**0.4275**	N/A	N/A
3	2	0.4937	0.6821	0.01	0.01	0.5084	**0.6953**	38.2%	36.8%
4	2	0.7986	0.8177	0.01	0.00	0.8191	**0.8652**	3.5%	5.6%
4	3	0.5898	0.8787	0.01	0.01	0.6163	**0.9029**	49.1%	46.5%
5	2	0.9944	0.9997	0.01	0.00	1.0100	**1.0153**	0.5%	0.5%

Table 2: Results for optimizing auction revenue in a sequential sales setting (n agents, m sales) with asymmetric uniformly-distributed types – here, both weights and boosts are allowed to vary.

n	m	Avg. VCG rev.	Avg. AMA rev	Avg. VCG Std. Err	Avg. AMA Std. Err.	Best VCG rev.	Best AMA rev.	Avg. % imp.	% Best Imp.
2	2	0	0.1288	0.00	0.00	0	**0.3348**	N/A	N/A
3	2	.2310	0.3782	0.01	0.01	0.2375	**0.4344**	63.9%	82.9%
4	2	0.3103	0.3713	0.01	0.01	0.3173	**0.4163**	19.7%	31.2%
4	3	0.2246	0.4208	0.01	0.01	0.2325	**0.4971**	87.4%	113.8%
5	2	0.3338	0.4151	0.00	0.01	0.3386	**0.4774**	24.3%	41.0%

We also consider a distribution where agent i's type is uniformly distributed on $[0, i]$ – in this setting, we also allow the bidder weights to vary, with results in2. Again, we observe improved performance by optimizing the AMA parameters.

5.2 Dynamic truthful task scheduling

Nisan and Ronen [37] present the static problem of truthful task scheduling, where workers report the time it takes them to complete certain tasks. A mechanism then has to assign the tasks and give payments to incentivize truthful reports with the goal of minimising the makespan of all jobs.

We formulate a dynamic version of the truthful task scheduling problem. There are n workers and T tasks. Each round, one of the tasks arrives and has to be assigned. Each worker has a cost vector $\theta_i = (t_{i,1}, \ldots, t_{i,T})$ distributed according to some prior f_i, which consists of the times they take to finish each task. Each round τ the mechanism designer takes an allocative action $\boldsymbol{x}_\tau \in \{0,1\}^n$, s.t. $\sum_{i=1}^n \boldsymbol{x}_{\tau,i} = 1$. At time τ, this causes agent i to receive reward $r_{i,\tau} = -\boldsymbol{x}_{\tau,i} t_{i,\tau}$.

The objective of the leader is to minimise total makespan. For this define $\tilde{t}_i$ as the time i has already worked on its jobs, when the last task has been assigned in round T. The leader's loss is then given by

$$\mathcal{L} = \max_{i \in \{1,\ldots,n\}} \left(\left(\sum_{\tau=1}^{T} \boldsymbol{x}_{\tau,i} t_{i,\tau} \right) - \tilde{t}_i \right)$$

Note, that in order to incentivize truthful reporting of the types, the agents have to be compensated for their work.

Table 3: We optimize AMA boosts in the truthful task scheduling setting with n agents and m tasks to minimize makespan, for 10 random seeds.

n	m	Avg. VCG	Avg. AMA	Avg. VCG Std. Err	Avg. AMA Std. Err.	Best VCG	Best AMA	% Best imp.	Avg. additional subsidy
2	4	1.0285	0.9446	0.03	0.03	0.9811	**0.9207**	6.16%	0.778
2	5	1.014	0.9298	0.03	0.03	0.9373	**0.895**	11.7%	0.853
3	4	0.602	0.589	0.02	0.02	0.582	**0.566**	2.8%	0.066

Table 4: We optimize AMA boosts in the truthful task scheduling setting with n agents and m tasks to minimize makespan, for 10 random seeds, with a type distribution which is asymmetric.

n	m	Avg. VCG	Avg. AMA	Avg. VCG Std. Err	Avg. AMA Std. Err.	Best VCG	Best AMA	% Best imp.	Avg. additional subsidy
2	4	1.8567	1.7317	0.04	0.04	1.7590	**1.5739**	10.5%	0.1084
2	5	1.9195	1.9375	0.05	0.04	1.8851	**1.6881**	10.4%	0.3296
3	4	1.3561	1.4557	0.04	0.03	1.3561	**1.3422**	1.0%	0.0590

We consider a VCG mechanism that chooses the solution which minimizes the total work done by the agents – an objective which is not the same as minimizing the makespan. (It is however not completely opposed to minimising makespan. Indeed, in the static problem, it has been shown that VCG is an n-approximation of the optimal mechanism.) We contrast this with affine maximizer mechanisms optimized to improve the makespan.

For a participant-symmetric valuation distribution (uniform on $[0, 3]$), results are in Table 3. In the more resource-constrained environments, we see a significant improvement in makespan for the optimized AMAs over the VCG mechanism. However, AMAs have to make larger compensating subsidies to the mechanism participants. We also consider an asymmetric distribution with disutilities distributed on $[0, 3i]$ for bidder i, with results in Table 4. In the more resource-constrained environments, we see a significant improvement in makespan for the best optimized AMAs over the VCG mechanism (although for some initializations only a poorly-performing AMA could be learned).

6 Conclusion

We optimize within the class of affine maximizer mechanisms to find high-performing dynamic mechanisms on MDPs. This optimization process amounts to a bilevel optimization problem, where the mechanism designer acting in the outer problem chooses parameters to maximize their objective in expectation over possible rewards, where the inner problem consists of optimally solving the MDP. In contrast to earlier work, this formulation allows for a wide array of possible loss functions (not just maximising social welfare) and works without strong restrictions on the type space. In principle, it captures essentially all problems of static mechanism design as a special case.

For the most prominent objective in mechanism design – revenue – we have proven differentiability, which allows for gradient-based optimisation approaches to converge to locally optimal mechanisms. Because we restrict to the class of AMAs, all these mechanisms are guaranteed to be exactly IC and IR. We further present a gradient-based algorithm to find local optima. Although the objective is discontinuous in the mechanism parameters, we can still estimate gradients from samples by adding a regularisation term to the LP formulation of the MDP. Our approach outperforms the benchmark dynamic VCG mechanism across a selection of some important dynamic mechanism design problems.

The affine maximizer approach is appropriate for any problem that can be formulated as controlling an MDP in the face of possibly untruthful preferences – this covers a wide range of interesting scenarios. In future work, we plan to replace our exact LP solver with approaches from deep reinforcement learning, allowing this method to be scaled to much larger and more realistic problems.

References

[1] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J. Zico Kolter. Differentiable convex optimization layers. In *Neural Information Processing Systems (NeurIPS)*, 2019.

[2] Akshay Agrawal, Shane Barratt, Stephen Boyd, Enzo Busseti, and Walaa M. Moursi. Differentiating Through a Cone Program, May 2020.

[3] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC press, 1999.

[4] Kareem Amin, Afshin Rostamizadeh, and Umar Syed. Repeated Contextual Auctions with Strategic Buyers. In *Neural Information Processing Systems (NeurIPS)*, 2014.

[5] MOSEK ApS. *The MOSEK optimization toolbox, version 10.0*, 2023.

[6] Susan Athey and Ilya Segal. An efficient dynamic mechanism. *Econometrica*, 81(6):2463–2485, 2013.

[7] Maria-Florina Balcan, Tuomas Sandholm, and Ellen Vitercik. A general theory of sample complexity for multi-item profit maximization. In *Economics and Computation (EC)*, 2018.

[8] Maria-Florina Balcan, Dan DeBlasio, Travis Dick, Carl Kingsford, Tuomas Sandholm, and Ellen Vitercik. How much data is sufficient to learn high-performing algorithms? Generalization guarantees for data-driven algorithm design. In *ACM Symposium on Theory of Computing (STOC)*, 2021.

[9] Maria-Florina F Balcan, Tuomas Sandholm, and Ellen Vitercik. Sample complexity of automated mechanism design. In *Neural Information Processing Systems (NeurIPS)*, 2016.

[10] Dirk Bergemann and Juuso Välimäki. The dynamic pivot mechanism. *Econometrica*, 78(2): 771–789, 2010.

[11] Dirk Bergemann and Juuso Välimäki. Dynamic Mechanism Design: An Introduction. *Journal of Economic Literature*, 57(2):235–274, June 2019.

[12] Martin Bichler, Maximilian Fichtl, Stefan Heidekrüger, Nils Kohring, and Paul Sutterer. Learning equilibria in symmetric auction games using artificial neural networks. *Nature Machine Intelligence*, 3(8):687–695, 2021.

[13] Avrim Blum, Jeffrey C. Jackson, Tuomas Sandholm, and Martin Zinkevich. Preference Elicitation and Query Learning. *Journal of Machine Learning Research*, 2004.

[14] Gianluca Brero, Benjamin Lubin, and Sven Seuken. Machine learning-powered iterative combinatorial auctions. *arXiv preprint arXiv:1911.08042*, 2019.

[15] Edward H Clarke. Multipart pricing of public goods. *Public Choice*, pages 17–33, 1971.

[16] Vincent Conitzer and Tuomas Sandholm. Complexity of mechanism design. In *Uncertainty in Artificial Intelligence (UAI)*, 2002.

[17] Vincent Conitzer and Tuomas Sandholm. Incremental mechanism design. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[18] Michael Curry, Ping-yeh Chiang, Tom Goldstein, and John P. Dickerson. Certifying strategyproof auction networks. In *Neural Information Processing Systems (NeurIPS)*, 2020.

[19] Michael Curry, Tuomas Sandholm, and John Dickerson. Differentiable economics for randomized affine maximizer auctions. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2023.

[20] Michael J Curry, Uro Lyi, Tom Goldstein, and John Dickerson. Learning revenue-maximizing auctions with differentiable matching. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[21] Zhijian Duan, Haoran Sun, Yurong Chen, and Xiaotie Deng. A Scalable Neural Network for DSIC Affine Maximizer Auction Design, May 2023.

[22] Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David C. Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In *International Conference on Machine Learning (ICML)*, 2019.

[23] Theodore Groves. Incentives in teams. *Econometrica: Journal of the Econometric Society*, pages 617–631, 1973.

[24] MohammadTaghi Hajiaghayi, Robert Kleinberg, and David C Parkes. Adaptive limited-supply online auctions. In *Electronic Commerce (EC)*, 2004.

[25] MohammadTaghi Hajiaghayi, Robert Kleinberg, and Tuomas Sandholm. Automated online mechanism design and prophet inequalities. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2007.

[26] Dmitry Ivanov, Iskander Safiulin, Ksenia Balabaeva, and Igor Filippov. Optimal-er auctions through attention. *arXiv preprint arXiv:2202.13110*, 2022.

[27] Sham M Kakade, Ilan Lobel, and Hamid Nazerzadeh. Optimal dynamic mechanism design and the virtual-pivot mechanism. *Operations Research*, 61(4):837–854, 2013.

[28] Robert Kleinberg. A Multiple-Choice Secretary Algorithm with Applications to Online Auctions. In *Symposium on Discrete Algorithms (SODA)*, 2005.

[29] Nils Kohring, Fabian R. Pieroth, and Martin Bichler. Enabling First-Order Gradient-Based Learning for Equilibrium Computation in Markets. *arXiv preprint arXiv:2303.09500*, March 2023.

[30] Sebastien M. Lahaie and David C. Parkes. Applying learning algorithms to preference elicitation. In *Proceedings of the 5th ACM Conference on Electronic Commerce*, EC '04, New York, NY, USA, May 2004. Association for Computing Machinery.

[31] Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. Jump 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023. In press.

[32] Boxiang Lyu, Qinglin Meng, Shuang Qiu, Zhaoran Wang, Zhuoran Yang, and Michael I. Jordan. Learning dynamic mechanisms in unknown environments: A reinforcement learning approach, 2022. URL `https://arxiv.org/abs/2202.12797`.

[33] Boxiang Lyu, Zhaoran Wang, Mladen Kolar, and Zhuoran Yang. Pessimism meets VCG: Learning dynamic mechanism design via offline reinforcement learning. *arXiv preprint arXiv:2205.02450*, 2022.

[34] Carlos Martin and Tuomas Sandholm. Finding mixed-strategy equilibria of continuous-action games without gradients using randomized policy networks. *arXiv preprint arXiv:2211.15936*, 2022.

[35] Paul Milgrom and Ilya Segal. Envelope Theorems for Arbitrary Choice Sets. *Econometrica*, 70 (2):583–601, 2002.

[36] Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981.

[37] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1-2):166–196, apr 2001.

[38] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007. doi: 10.1017/CBO9780511800481.

[39] Mallesh Pai and Rakesh V. Vohra. Optimal dynamic auctions. Working Paper 1461, Discussion Paper, 2008.

[40] Alessandro Pavan, Ilya Segal, and Juuso Toikka. Dynamic mechanism design: A Myersonian approach. *Econometrica*, 82(2):601–653, 2014.

[41] Jad Rahme, Samy Jelassi, Joan Bruna, and S Matthew Weinberg. A permutation-equivariant neural network architecture for auction design. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[42] Jad Rahme, Samy Jelassi, and S Matthew Weinberg. Auction learning as a two-player game. In *International Conference on Learning Representations (ICLR)*, 2021.

[43] Kevin Roberts. The characterization of implementable choice rules. *Aggregation and Revelation of Preferences*, 12(2):321–348, 1979.

[44] Tuomas Sandholm. Automated mechanism design: A new application area for search algorithms. In *International Conference on Principles and Practice of Constraint Programming*, 2003.

[45] Tuomas Sandholm and Anton Likhodedov. Automated design of revenue-maximizing combinatorial auctions. *Operations Research*, 63(5):1000–1025, 2015.

[46] Tuomas W Sandholm, Vincent Conitzer, and Craig Boutilier. Automated Design of Multistage Mechanisms. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[47] Weiran Shen, Binghui Peng, Hanpeng Liu, Michael Zhang, Ruohan Qian, Yan Hong, Zhi Guo, Zongyao Ding, Pengjun Lu, and Pingzhong Tang. Reinforcement mechanism design, with applications to dynamic pricing in sponsored search auctions. *CoRR*, abs/1711.10279, 2017.

[48] Weiran Shen, Pingzhong Tang, and Song Zuo. Automated mechanism design via neural networks. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, 2019.

[49] Ermis Soumalias, Behnoosh Zamanlooy, Jakob Weissteiner, and Sven Seuken. Machine Learning-powered Course Allocation, 2023.

[50] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.

[51] Jonathan Weed. An explicit analysis of the entropic penalty in linear programming. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 1841–1855. PMLR, 06–09 Jul 2018. URL `https://proceedings.mlr.press/v75/weed18a.html`.

[52] Jakob Weissteiner, Jakob Heiss, Julien Siems, and Sven Seuken. Bayesian Optimization-based Combinatorial Assignment. *arXiv preprint arXiv:2208.14698*, 2023.

A Affine maximizers are incentive compatible

Consider player i with true reward function r_i and reported rewards $\hat{r}_i$, where other players have rewards r_{-i}. Recall that the payments charged for each agent can be written as

$$p_i(\boldsymbol{r}) = \frac{1}{w_i}\left(\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r}) - \left(\mathbb{E}_{\pi^*(w,b,\boldsymbol{r})}\left[\sum_{t=0}^{T}\left(\sum_{j\neq i} w_j r_j(s_t,a_t)\right) + b(s_t,a_t)\right]\right)\right)$$
$$= \frac{1}{w_i}\left(\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r}) - \mathsf{asw}(w,b,\boldsymbol{r}) + \sum_{t=0}^{T} w_i \mathbb{E}_{\pi^*(w,b,\boldsymbol{r})}[r_i(s_t,a_t)]\right)$$

The true expected utility at the time they report, as a function of their true reward function and a possible misreport, is

$$U_i(r_i,\hat{r}_i) = \mathbb{E}\left[\sum_{t=0}^{T} r_i(s_t,a_t) - \frac{1}{w_i}\left(\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r}) - \left(\sum_{t=0}^{T}\left(\sum_{j\neq i} w_j r_j(s_t,a_t)\right) + b(s_t,a_t)\right)\right)\right]$$

where the expectation is over the randomness in the MDP (note that it does *not* need to be over the randomness in opponent types – IC should hold in dominant strategies considering the opponents).

In choosing a misreport $\hat{r}_i$, player i thus faces a maximization problem:

$$\arg\max_{\tilde{r}_i} U_i(r_i,\tilde{r}_i) = \arg\max_{\tilde{r}_i} \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{T} r_i(s_t,a_t) - \frac{1}{w_i}\left(\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r}) - \left(\sum_{t=0}^{T}\left(\sum_{j\neq i} w_j r_j(s_t,a_t)\right) + b(s_t,a_t)\right)\right)\right]$$
$$= \arg\max_{\tilde{r}_i} \mathbb{E}\left[\sum_{t=0}^{T} r_i(s_t,a_t) + \frac{1}{w_i}\left(\sum_{t=0}^{T}\sum_{j\neq i} w_j r_j(s_t,a_t) + b(s_t,a_t)\right)\right] - c$$
$$= \arg\max_{\tilde{r}_i} \mathbb{E}\left[\sum_{t=0}^{T} w_i r_i(s_t,a_t) + \sum_{t=0}^{T}\sum_{j\neq i} w_j r_j(s_t,a_t) + b(s_t,a_t)\right]$$

where c is a constant. This is exactly the objective that the mechanism attempts to maximize if the bidder reports truthfully, so the best choice they can make is to do so.

A.1 IR and IC guarantees

In mechanism design, there is often a distinction between:

- *Ex post* – properties that hold after all types have been reported and decisions have been made
- *Ex interim* – properties that hold after a bidder has observed their own type, but before seeing other types.
- *Ex ante* – properties that hold before types have been observed.

As far as the prior distribution of agent types is concerned, all our guarantees are *ex post* – i.e. we ensure dominant-strategy incentive compatibility and ex-post IR.

In our problem, we have an additional source of randomness, the inherent randomness in the MDP itself. We thus refer to "in expectation" to refer to properties that hold when averaging over the randomness in the MDP, but ex post in the types.

Given the above choices of terminology, and given a correct policy and value estimate, our chosen mechanism and payment rules can guarantee in-expectation incentive compatibility and individual rationality.

B Proofs of continuity and differentiability a.e. for revenue loss function.

In this proof we will assume that f is a continuous density function, according to which $\boldsymbol{r}$ is distributed and that it is sufficiently well-behaved, i.e. it is fastly-decreasing such that all moments exist, as well as the expected values of powers of norms of $\boldsymbol{r}$.

Recall that expected revenue is defined as

$$\begin{aligned}\mathsf{rev}(w,b) &= \mathbb{E}_{\boldsymbol{r}}\left[\sum_{i=1}^{n} p_i(\boldsymbol{r})\right] \\ &= \mathbb{E}_{\boldsymbol{r}}\left[-\sum_{i=1}^{n}(\frac{1}{w_i}\mathsf{asw}(w,b,\boldsymbol{r})) + \mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r}) + \sum_{i=1}^{n}\frac{1}{w_i}\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r})\right]\end{aligned}$$

In order to prove revenue is continuous and differentiable a.e., it suffices to show that $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{asw}(w,b,\boldsymbol{r})\right], \mathbb{E}_{\boldsymbol{r}}\left[\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})\right]$ and $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r})\right]$ are locally Lipschitz continuous in w, b. Differentiability a.e. then follows from Rademacher's theorem. Further note that the proofs for $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{asw}(w,b,\boldsymbol{r})\right]$ and $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{asw}^{(-i)}(w,b,\boldsymbol{r})\right]$ are identical, since one is equivalent to the other, when removing one agent. We thus restrict ourselves to proving $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{asw}(w,b,\boldsymbol{r})\right]$ and $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})\right]$ are locally Lipschitz continuous.

B.1 Affine social welfare is Locally Lipschitz continuous

Here we only show continuity in w, since the proof for b works analogous. Fix $\boldsymbol{r} \in \mathbb{R}^{|S|\times|A|\times n}$ and let $w, \hat{w} \in \mathbb{R}^n_+$, s.t. $\exists! j : w_j \neq \hat{w}_j$. We need to show that $\exists K_{\boldsymbol{r}}$, s.t.

$$\begin{aligned}|\sum_{i=1}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)] - \sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) \\ - \sum_{t=0}^{T}\hat{w}_j\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] - \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]| \leq K_{\boldsymbol{r}}|w_j - \hat{w}_j|\end{aligned} \tag{7}$$

where $\pi = \pi^*(w,b,\boldsymbol{r})$ and $\hat{\pi} = \pi^*(\hat{w},b,\boldsymbol{r})$. To perform our proof we need the following inequalities:

By optimality of π

$$\begin{aligned}&\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j\mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)] \\ \geq &\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]\end{aligned} \tag{8}$$

By optimality of $\hat{\pi}$

$$\begin{aligned}&\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j\mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)] \\ \leq &\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T}\mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]\end{aligned} \tag{9}$$

By assumption on $w, \hat{w}_j$

$$\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)]$$
$$-\left(\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)]\right)$$
$$\leq |w_j - \hat{w}_j| \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] \tag{10}$$

$$\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]$$
$$-\left(\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]\right)$$
$$\leq |w_j - \hat{w}_j| \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] \tag{11}$$

Now we have all the ingredients to show local Lipschitz continuity wrt to w_i. Let $K_{\boldsymbol{r}} = \|\boldsymbol{r}\|_\infty T \geq \max(\sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)], \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)])$. Then we have

$$\sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)] + K_{\boldsymbol{r}}|w_j - \hat{w}_j|$$
$$\underset{Eq.(8)}{\geq} \sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)] + K_{\boldsymbol{r}}|w_j - \hat{w}_j|$$
$$\underset{Eq.(11)}{\geq} \sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\hat{\pi}}[b(s_t,a_t)]$$
$$\underset{Eq.(9)}{\geq} \sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} \hat{w}_j \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)]$$
$$\underset{Eq.(10)}{\geq} \sum_{i=1,i\neq j}^{n}(\sum_{t=0}^{T} w_i \mathbb{E}_{s_t,a_t\sim\pi}[r_i(s_t,a_t)]) + \sum_{t=0}^{T} w_j \mathbb{E}_{s_t,a_t\sim\pi}[r_j(s_t,a_t)] + \sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi}[b(s_t,a_t)] - K_{\boldsymbol{r}}|w_j - \hat{w}_j|$$

To conclude this proof, we note

$$\mathbb{E}_{\boldsymbol{r}}[\mathsf{asw}(w,b,\boldsymbol{r}) - \mathsf{asw}(\hat{w},b,\boldsymbol{r})] \leq T\mathbb{E}_{\boldsymbol{r}\sim f}[\|\boldsymbol{r}\|_\infty]|w_j - \hat{w}_j|$$

and $\mathbb{E}_{\boldsymbol{r}\sim f}[\|\boldsymbol{r}\|_\infty]$ is finite by assumption. Differentiability a.e. now follows from Rademacher's theorem.

B.2 Expected social welfare is Locally Lipschitz continuous

Unlike for affine social welfare, we cannot argue that social welfare is Lipschitz continuous. Indeed changing the weights and boosts only slightly can cause a completely different policy to become optimal, leading to a discontinuous jump or drop in social welfare. However, as we will show,

when taking the expected value, these discontinuities get smoothed out, guaranteeing local Lipschitz continuity and thereby differentiability almost surely.

We will restrict to proving Lipschitz continuity of $\mathbb{E}_{\boldsymbol{r}}\left[\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})\right]$ with respect to b, as the proof with respect to w works similar.

In the proofs below we identify the reported reward functions of all agents with vectors $\boldsymbol{r} \in \mathbb{R}^d$, where $d = |S| \times |A| \times n$.

Our first observation is that all vectors $\boldsymbol{r}$ for which two policies π_1, π_2 give the same affine social welfare lie on a hyperplane. Indeed, let ν_i denote the induced state-action occupancy measure of policy π_i. Then

$$\sum_{s,a} \nu_1(s,a)(w_i r_i(s,a) + b(s,a)) = \sum_{s,a} \nu_2(s,a)(w_i r_i(s,a) + b(s,a))$$

is equivalent to

$$\sum_{s,a,i} (\nu_1(s,a) - \nu_2(s,a)) w_i r_i(s,a) = \sum_{s,a} (\nu_2(s,a) - \nu_1(s,a))(b(s,a))$$

Let $\boldsymbol{\nu}_1 = (w_1\nu_1(s_1,a_1), w_2\nu_1(s_1,a_1), \ldots, w_n\nu_1(s_1,a_1), w_1\nu_1(s_2,a_1), \ldots, w_n\nu_1(s_{|S|}, a_{|A|})$. Then the above is equivalent to the following hyperplane:

$$\mathcal{H}_{12}(w,b) = \{\boldsymbol{r} : (\boldsymbol{\nu_1} - \boldsymbol{\nu_2})^T \boldsymbol{r} = \sum_{s,a} (\nu_2(s,a) - \nu_1(s,a))(b(s,a))\}$$

Let $b, \tilde{b} \in \mathbb{R}^{|S|\times|A|}$, s.t. $\exists!(s',a') : b(s',a') \neq \tilde{b}(s',a')$. $\tilde{b}$ gives us another hyperplane of equivalence between π_1, π_2

$$\mathcal{H}_{12}(w,\tilde{b}) = \{\boldsymbol{r} : (\boldsymbol{\nu_1} - \boldsymbol{\nu_2})^T \boldsymbol{r} = \sum_{s,a} (\nu_2(s,a) - \nu_1(s,a))(\tilde{b}(s,a))\}$$

Note that $\mathcal{H}_{12}(w,\tilde{b})$ and $\mathcal{H}_{12}(w,b)$ are parallel with a distance $\frac{|(\nu_2(s',a') - \nu_1(s',a'))(b(s',a') - \tilde{b}(s',a'))|}{\|(\boldsymbol{\nu_1} - \boldsymbol{\nu_2})\|}$.

Moreover, for any given parameters (w,b) there are only $|\Pi|^2$ planes of equivalence, where $|\Pi|$ is the number of deterministic policies.

When we change $b(s',a')$ to $\tilde{b}(s',a')$, then there can be rewards $\boldsymbol{r}$, for which $\pi^*(w,\tilde{b},r) \neq \pi^*(w,b,r)$. For these rewards it follows that in general the social welfare $\mathsf{sw}^{(\pi^*(w,\tilde{b},\boldsymbol{r}))}(\boldsymbol{r})$ is different from $\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})$. Using the hyperplanes defined above, we know the set of all $\boldsymbol{r}$, where $\mathsf{sw}^{(\pi^*(w,\tilde{b},\boldsymbol{r}))}(\boldsymbol{r}) \neq \mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})$ is contained in the set

$$U(w,b,\tilde{b}) = \bigcup_{\pi_j,\pi_i \in \Pi} U(w,b,\tilde{b})_{ij}$$

where

$$U(w,b,\tilde{b})_{ij} = \{\boldsymbol{r} : |(\boldsymbol{\nu_i} - \boldsymbol{\nu_j})^T \boldsymbol{r} - \sum_{s,a} (\nu_j(s,a) - \nu_i(s,a))(b(s,a))| \leq |(\nu_j(s',a') - \nu_i(s',a'))(\tilde{b}(s',a') - b(s',a'))|\}$$

are the polytopes induced by the hyperplanes $\mathcal{H}_{ij}(w,\tilde{b})$ and $\mathcal{H}_{ij}(w,b)$.

With this in mind let us make a first naive analysis of the difference of expected social welfare, when changing b.

$$
\begin{aligned}
&|\mathbb{E}_{\boldsymbol{r}}[\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r})] - \mathbb{E}_{\boldsymbol{r}}[\mathsf{sw}^{(\pi^*(w,\tilde{b},\boldsymbol{r}))}(\boldsymbol{r})]| && (12)\\
&\leq E_{\boldsymbol{r}}[|\mathsf{sw}^{(\pi^*(w,b,\boldsymbol{r}))}(\boldsymbol{r}) - \mathsf{sw}^{(\pi^*(w,\tilde{b},\boldsymbol{r}))}(\boldsymbol{r})|] && (13)\\
&\leq 2Tn \int_{U(w,b,\tilde{b})} \|\boldsymbol{r}\|_\infty f(\boldsymbol{r}) d\boldsymbol{r} && (14)\\
&\leq \sum_{\pi_i,\pi_j \in |\Pi|} 2Tn \int_{U(w,b,\tilde{b})_{ij}} \|\boldsymbol{r}\|_\infty f(\boldsymbol{r}) d\boldsymbol{r} && (15)\\
& && (16)
\end{aligned}
$$

where for the second inequality we use $\mathsf{sw}^{(\pi)}(\boldsymbol{r}) = \sum_{i=1}^{n}(\sum_{t=0}^{T} \mathbb{E}_{s_t,a_t\sim\pi} r_i(s_t,a_t)) \leq Tn\|\boldsymbol{r}\|_\infty$. We want to show that the above can be bounded by $L|b(s',a') - \tilde{b}(s',a')|$ for some L. For this we need to get a better understanding of $\int_{U(w,b,\tilde{b})_{ij}} \|\boldsymbol{r}\|_\infty f(\boldsymbol{r}) d\boldsymbol{r}$.

For simplicity, we assume now the probability density f has compact support on R^d, i.e. there exists a K such that $\forall \boldsymbol{r} : \|\boldsymbol{r}\|_2 \geq K \implies f(r) = 0$.[6] Since the hyperplanes $\mathcal{H}_{ij}(w,\tilde{b})$ and $\mathcal{H}_{ij}(w,b)$ have dimension $d-1$ and are parallel with distance $\frac{|(\nu_2(s',a')-\nu_1(s',a'))(b(s',a')-\tilde{b}(s',a'))|}{\|(\boldsymbol{\nu_1}-\boldsymbol{\nu_2})\|}$, we can bound the integral by multiplying an upper bound of the volume of $U(w,b,\tilde{b})_{ij}$ with the maximum possible value of $\|\boldsymbol{r}\|_\infty f(\boldsymbol{r})$.

$$
\begin{aligned}
\int_{U(w,b,\tilde{b})_{ij}} \|\boldsymbol{r}\|_\infty f(\boldsymbol{r}) d\boldsymbol{r} \leq (2K)^{d-1} \frac{|(\nu_2(s',a') - \nu_1(s',a'))(b(s',a') - \tilde{b}(s',a'))|}{\|(\boldsymbol{\nu_1} - \boldsymbol{\nu_2})\|} K \max_{\boldsymbol{r}} f(\boldsymbol{r})\\
= L|(b(s',a') - \tilde{b}(s',a'))|
\end{aligned}
$$

for $L = (2K)^d \frac{|(\nu_2(s',a')-\nu_1(s',a'))|}{\|(\boldsymbol{\nu_1}-\boldsymbol{\nu_2})\|} \max_{\boldsymbol{r}} f(\boldsymbol{r})$, which proves Lipschitz continuity and thereby differentiability almost surely.

[6]This assumption is not necessary. As long as f decays sufficiently quickly for large $\boldsymbol{r}$, the proof still goes through with some minor changes.