Noname manuscript No.
(will be inserted by the editor)

Constrained Automated Mechanism Design for Infinite Games of Incomplete Information

Yevgeniy Vorobeychik · Daniel M. Reeves · Michael P. Wellman

Abstract We present a functional framework for automated Bayesian and worst-case mechanism design, based on a two-stage game model of strategic interaction between the designer and the mechanism participants. At the core of our framework is a black-box optimization algorithm which guides the process of evaluating candidate mechanisms. We apply the approach to several classes of two-player infinite games of incomplete information, producing optimal or nearly optimal mechanisms using various objective functions. By comparing our results with known optimal mechanisms, and in some cases improving on the best known mechanisms, we provide evidence that ours is a promising approach to parametrized mechanism design for infinite Bayesian games.

1 Motivation

The field of Mechanism Design provides a compelling general framework for incentive-centered design of resource allocation processes, and as such has earned a foundational place in economic theory. Its reach has recently extended to other disciplines concerned with decentralized resource allocation, including operations research (Gallien, 2006) and computer science (Nisan, 2007). In academic literature, typical mechanism design exercises (including the recently Nobel-awarded major advances) produce analytical results characterizing ideal mechanisms under specified conditions. In practice, the theory has often informed the design of actual

Y. Vorobeychik
Sandia National Laboratories, P.O. Box 969, Mailstop 9159, Livermore, CA 94551-0969
Tel.: 1-773-562-0148
Fax: 1-925-294-2234 E-mail: eug.vorobey@gmail.com

D.M. Reeves
Beeminder LLC, 2606 W Fountaindale Dr, Peoria, IL 61614-3732
E-mail: dreeves@beeminder.com

M.P. Wellman
Computer Science and Engineering, University of Michigan, 2260 Hayward St, Ann Arbor MI 48109-2121
E-mail: wellman@umich.edu

mechanisms, despite the deviation of the given real-world situation from theoretical conditions. For this reason and others, successful application of mechanism design principles needs to be embedded within a broader engineering perspective (Roth and Peranson, 1999).

A key difficulty in practical mechanism design is the presence of idiosyncratic objectives and constraints. For example, when the US government tried to set up a mechanism to sell radio spectrum licenses, it identified among its objectives promotion of rapid deployment of new technologies. Additionally, it imposed constraints such as ensuring that some licenses go to minority-owned and women-owned companies (McMillan, 1994).

Conitzer and Sandholm (2002, 2003a) introduced the phrase *automated mechanism design* (AMD) to refer to the approach of formulating and computationally solving specific instances of mechanism design, cast as optimization problems, given arbitrary objectives and constraints. They have studied various classes of AMD problems, generally focusing on solutions in the form of direct truthful mechanisms. This reliance has at its core the *revelation principle* (Myerson, 1981), which states that the outcome of any given mechanism can still be achieved if we restrict the design space to mechanisms that induce truthful revelation of agent preferences. Despite this result, there may be computational reasons not to adopt the prescriptions of this principle, as pointed out by Conitzer and Sandholm (2003b). Furthermore, the revelation principle may simply fail to hold in the face of mechanism constraints. For example, in a combinatorial setting, communication constraints may preclude the revelation of preferences over all possible bundles. While the computational criticisms can often be addressed to some degree within the spirit of direct mechanisms (e.g., by multi-stage mechanisms, such as ascending auctions, which implement partial revelation of agent preferences in a series of steps), idiosyncratic constraints on the design problem generally present a more difficult hurdle.

We introduce an approach to the design of general mechanisms (direct or indirect) given arbitrary designer objectives and arbitrary constraints on the design space, which we allow to be continuous. Our mechanisms induce games of incomplete information in which agents may have infinite sets of strategies and types. As in most mechanism design literature, we assume that the designer knows the set of all possible agent types and their distribution, but not the actual type realizations. Our methods build on our previous work on empirical mechanism design (Vorobeychik et al, 2006), as well as related work on parametric evolutionary auction design (Phelps et al, 2002, 2003; Byde, 2006). The approach of Guo and Conitzer (2009, 2010) to automated design of linear redistribution mechanisms is also similar in spirit. In place of the linear programming formulation of that work, we adopt a more general search framework, albeit restricted in our implementation to a particular class of infinite games of incomplete information. We further simplify the mechanism design domain by restricting search to some subset of an n-dimensional Euclidean space, rather than in an arbitrary function space, as would be required in a completely general setting. Our premise is that many practical design problems involve search for the optimal or nearly optimal setting of parameters within an existing infrastructure. For example, it is more likely that policymakers will seek an appropriate tax rate to achieve their objective than overhaul the entire tax system.

In the following sections, we present our framework for automated mechanism design and test it in several application domains. We specifically look at two settings: Bayesian and worst-case. In both settings, we assume that the designer knows the probability distribution over agent types. The difference is in the designer's optimization criterion. In the Bayesian setting, the designer simply maximizes expected value of the objective function. In the worst-case setting, the designer maximizes the value of the worst outcome. Since it is impossible to guarantee computationally that a particular mechanism is robust with respect to every realization of agent types, we introduce the notion of *probably approximately robust mechanism design*, which instead aims to probabilistically ensure that very few type profiles can result in poor outcomes for the designer. Our results suggest that this framework has much promise: most of the designs that we discover automatically are nearly as good as or better than the best known hand-built designs in the literature.

This paper makes four contributions. The first is conceptual: we offer a general framework for designing mechanisms entirely computationally, either from scratch, or starting with a known acceptable mechanism. Our framework can be applied to truthful mechanism design (as illustrated in Section 6), but is not restricted to this case. It can easily incorporate external information (e.g., a characterization of truthful mechanisms in the specified design space) when available. The framework is composed of two main pieces: a stochastic search algorithm, which performs the actual search in the design space, and a game solver, which obtains a solution or a set of solutions for any game induced by a design choice. These pieces, as well as other elements of the framework, such as the designer's objective and constraints, can be independently implemented. Our second contribution is an approach for probabilistic relaxation of classes of mechanism design constraints, as well as of robust optimization, which allows for sensitivity analysis of results. Our third contribution is an actual implementation of our framework, operationalized for a restricted class of games with two players, a set of constraints commonly used in mechanism design, and several objective functions of general interest. Our fourth contribution is verification of practical feasibility and generality of our approach, performed via a series of examples of varying degree of complexity. We demonstrate generality by moving with ease between different problem settings (i.e., different objectives, constraints, and problem specifications such as Bayesian and worst-case mechanism design) while staying within the same framework and, indeed, within the same basic implementation.

2 Game Notation

We restrict our attention to *one-shot games of incomplete information*, denoted by $[I, \{A_i\}, \{T_i\}, F(\cdot), \{u_i(a,t)\}]$, where I refers to the set of players and $m = |I|$ is the number of players. A_i is the set of actions available to player $i \in I$, and $A = A_1 \times \cdots \times A_m$ is the joint action space. T_i is the set of types (private information) of player i, with $T = T_1 \times \cdots \times T_m$ representing the joint type space. Since players know their own types prior to taking an action, but do not know types of others, we allow them to condition their actions on own type. Thus, we define a strategy of a player i to be a function $s_i : T_i \to A_i$, and S_i the space of such strategies. For a joint strategy $s \in S = S_1 \times \cdots \times S_m$, $s(t)$ denotes the vector $(s_1(t_1), \ldots, s_m(t_m))$. $F(\cdot)$ is the distribution over the joint type space.

It is often convenient to refer to a strategy of player i separately from that of the remaining players. To accommodate this, we use s_{-i} to denote the joint strategy of all players other than player i. Similarly, t_{-i} designates the joint type of all players other than i.

We define the payoff (utility) function of each player i by $u_i : A \times T \to \mathbb{R}$, where $u_i(a_i, t_i, a_{-i}, t_{-i})$ indicates the payoff to player i with type t_i for playing action $a_i \in A_i$ when the remaining players with joint types t_{-i} play a_{-i}. Given a strategy profile $s \in S$, the expected payoff of player i is $\tilde{u}_i(s) = E_t[u_i(s(t), t)]$.

Faced with such a game, we assume that players play optimally against each other.

Definition 1 A strategy profile $s = (s_1, \ldots, s_m)$ constitutes a *Bayes-Nash equilibrium* of game $[I, \{A_i\}, \{T_i\}, F(\cdot), \{u_i(a, t)\}]$ if for every $i \in I$ and $s_i' \in S_i$, $\tilde{u}_i(s_i, s_{-i}) \geq \tilde{u}_i(s_i', s_{-i})$.

While our focus here is on pure strategy equilibria, we note that our framework naturally incorporates mixed strategies, as long as a solver is available to compute (or approximate) mixed strategy equilibria.

3 Mechanism Design on Bayesian Games

We can model the strategic interactions between the designer of the mechanism and its participants as a two-stage game (Vorobeychik et al, 2006). The designer moves first by selecting a value θ from a set of allowable mechanism settings, Θ. All the participant agents observe the mechanism parameter θ and move simultaneously thereafter. For example, the designer could be deciding between a first-price and a second-price sealed-bid auction mechanism, with the presumption that after the choice has been made, the bidders participate with full awareness of the auction rules.

Since the participants know the mechanism parameter, we define a game between them in the second stage as

$$\Gamma_\theta = [I, \{A_i\}, \{T_i\}, F(\cdot), \{u_i(a, t, \theta)\}].$$

We refer to Γ_θ as the game *induced* by θ.

The design objective takes the form $W(s(t, \theta), t, \theta)$, where $s(t, \theta)$ is a *solution* or a prediction of outcome of agent play. As is common in the mechanism design literature, we evaluate mechanisms with respect to a specific Bayes-Nash equilibrium solution, $s(t, \theta)$.[1] Significantly, the objective may be specified *algorithmically* by a procedure that outputs a real number representing the objective value for any combination of mechanism parameter, solution, and joint type.

Note that an equilibrium solution $s(t, \theta)$ is a function of player types, since each player is presumed to observe its type prior to making a strategic choice. Below, we also use the short notation $s(\theta)$ to denote the equilibrium strategy profile, which

[1] Focus on a specific equilibrium is typically justified by allowing the designer to suggest the equilibrium to participants, presuming that no agent will subsequently have an incentive to deviate.

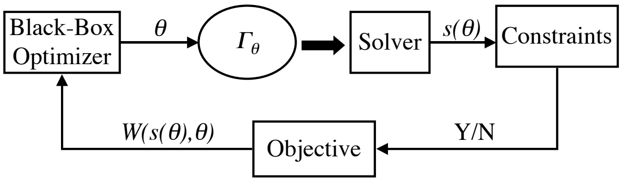

Fig. 1 Automated mechanism design procedure based on black-box optimization.

in the Bayesian setting is a profile of functions of player types. Since the designer's objective depends on player types, either indirectly due to its dependence on the player strategies, or directly through the type argument, we need to transform the type-dependent specification of the objective, $W(s(t,\theta),t,\theta)$, into $W(s(\theta),\theta)$. That is, we need to summarize the objective value by some transformation with respect to the type distribution (e.g., expectation). We refer to this transformation as *objective evaluation*. In Section 3.4 we present two principled approaches for evaluating the objective with respect to a distribution of player types.

In addition to the objective function, the designer may specify a collection of constraints on the outcomes (solutions) induced by the corresponding design choices. Let the constraints be specified as $\mathcal{C} = \{\mathcal{C}_i(s(t,\theta),t,\theta)\}$, although these may likewise be provided in the form of an algorithm that returns *true* if the constraint is satisfied and *false* otherwise for the given setting of the specified arguments.

Observe that if we *knew* $s(t,\theta)$ as a function of θ, the designer would simply be faced with an optimization problem. This follows by backwards induction, which would have us find $s(t,\theta)$ first for every θ and then compute an optimal mechanism with respect to these equilibria. If the design space were small, backwards induction applied to our model would thus yield an algorithm for optimal mechanism design. Indeed, if additionally the games Γ_θ featured small sets of players, strategies, and types, we would say little more about the subject. Our goal, however, is to develop a mechanism design tool for settings in which it is infeasible to obtain a solution of Γ_θ for every $\theta \in \Theta$, either because the space of possible mechanisms is large, or because solving (or approximating solutions to) Γ_θ is computationally daunting. Additionally, we try to avoid making assumptions about the objective function or constraints on the design problem or the agent type distributions. In our computational studies below, we do restrict the games to two players with piecewise-linear utility functions, but allow them to have infinite strategy and type sets.

In short, we propose the following high-level procedure for finding optimal mechanisms:

1. Select a candidate mechanism, θ.
2. Find (approximate) solutions to Γ_θ.
3. Evaluate the objective and constraints given solutions to Γ_θ.
4. Repeat this procedure for a specified number of steps.
5. Return an approximately optimal design based on the resulting optimization path.

We visually represent this procedure by a diagram in Figure 1. Below, we instantiate this procedure using a concrete black-box optimization routine and elucidate its first three steps, thereby presenting a full parametrized mechanism design framework for Bayesian games.

3.1 Designer's Optimization Problem

We begin by treating the designer's problem as black-box optimization, where the black box produces a noisy evaluation of the designer's objective, $W(s(\theta),\theta)$. Once we frame the problem as black-box optimization, we can draw on a wealth of literature devoted to algorithms for this setting (Spall, 2003). Whereas we can in principle select any one of these, we adopt simulated annealing for this study, as it has proved quite effective for a great variety of simulation optimization problems in noisy settings with many local optima (Corana et al, 1987; Fleischer, 1995; Siarry et al, 1997). By instantiating the high-level procedure above with simulated annealing, we obtain the following procedure, to which we refer below as the *AMD framework*:

1. Begin with an arbitrarily selected design $\theta_0 \in \Theta$.
2. (iteration k, $k = 0, 1, \dots$) *Evaluate* θ_k, obtaining W_k as follows:
   (a) Compute an exact or approximate solution $s(t, \theta_k)$ of Γ_{θ_k}.
   (b) Apply every constraint $\mathcal{C}_i(s(t,\theta_k), t, \theta_k) \in \mathcal{C}$ to the solution $s(t, \theta_k)$; return that θ_k is infeasible if any constraint fails (in our implementation, set $W(s(\theta_k), \theta_k) \leftarrow -\infty$).[2]
   (c) If all the constraints are satisfied, evaluate the objective value $W(s(\theta_k), \theta_k)$ as described in Section 3.4.
3. (if $k \geq 1$) Set $\theta_k \leftarrow \theta_{k-1}$ with probability $1 - p_k(W_{k-1}, W_k)$.
4. Select the next candidate mechanism, $\theta_{k+1} \mid \theta_k$ from a probability distribution $G_k(\theta_k)$.
5. Repeat steps 2–4 until termination criterion reached.
6. Return the best design found.

In this procedure, $p_k(W_{k-1}, W_k)$ is the *Metropolis acceptance probability* (Spall, 2003), defined by:

$$p_k(W_{k-1}, W_k) = \begin{cases} \exp\left[-\frac{W_{k-1}-W_k}{M_k}\right] & \text{if } W_k < W_{k-1} \\ 1 & \text{otherwise,} \end{cases}$$

where M_k is a schedule of "temperatures" which govern the degree of exploration of inferior candidate neighborhoods performed by the algorithm. We opt for a relatively simple adaptive implementation of simulated annealing, with normally distributed random perturbations applied to the solution candidate θ_k in every iteration to obtain the candidate mechanism θ_{k+1}. That is, $G_k(\theta_k) = N(\theta_k, \sigma_k^2)$ for a specified variance sequence σ_k^2. We use an exponentially decreasing sequence of variances in our implementation of the algorithm.[3]

[2] A more refined approach would use a penalty or barrier method (Nocedal and Wright, 2006), which would account for the extent of constraint violation. We note, however, that we wish to demonstrate a general applicability of our framework to Boolean black-box constraints, as well as more traditional real-valued constraints, and penalty and barrier methods are primarily targeted at the latter. Consequently, while we apply a penalty method in several instances below, we stay with this basic setup in most cases.

[3] These choices seem to be common in simulated annealing implementations (Spall, 2003).

To complete the algorithmic specification of the mechanism design problem, we allow the designer to specify the distribution of player types as a black box from which *samples* of type profiles can be drawn. Thus, we must use numerical techniques to evaluate the objective with respect to player types, thereby introducing sampling noise into the process.

As an application of black-box optimization, the mechanism design problem in our formulation is just one of many problems that can be addressed with one of a selection of methods. What makes it special is the subproblem of evaluating the objective function for a given mechanism choice, and the particular nature of mechanism design constraints which are evaluated based on Nash equilibrium outcomes and agent types.

3.2 Computing Nash Equilibria

As implied by the backwards induction process, we must obtain solutions (Bayes-Nash equilibria in the current setting) of the games induced by the design choice, θ, in order to evaluate the objective function. In general, this is simply not possible, since Bayes-Nash equilibria may not even exist in an arbitrary game, nor is there a general-purpose tool to find them when they do. However, there are a number of tools that can find or approximate solutions in *specific* settings. For example, GAMBIT (McKelvey et al, 2005) is a general-purpose toolbox of solvers that can find Nash equilibria in finite games, although the runtime is often prohibitive for even moderately sized games. Vorobeychik and Wellman (2008) offer general-purpose methods for approximating Nash equilibria in infinite games specified using stochastic simulations; one of these is provably convergent (in probability) to a Nash equilibrium (if one exists). Vorobeychik (2009) performs simulation-based Bayes-Nash equilibrium and mechanism design analysis of a mathematically intractable class of sponsored-search auctions. Ganzfried and Sandholm (2010) recently introduced a mixed-integer linear programming formulation for infinite games, which computes equilibria given a specified threshold-strategy form known to cover all solutions.

In this work, we employ a best-response finder introduced by Reeves and Wellman (2004) (henceforth, RW), applying it iteratively to obtain sample Bayes-Nash equilibria for a restricted class of infinite two-player games of incomplete information. Whereas RW is often effective in converging to a sample Bayes-Nash equilibrium, it does not do so always. To deal with non-convergent cases, we take the conservative approach of discarding any design choices for which the solver does not produce an answer. In any case, RW does usually return a nearly exact equilibrium solution, which allows us to focus on the mechanism design problem rather than equilibrium computation itself.

3.3 Dealing with Constraints

Mechanism design can feature any of the following three classes of constraints: *ex ante* (constraints evaluated with respect to the joint distribution of types), *ex interim* (evaluated separately for each player and type with respect to the joint type distribution of other players), and *ex post* (evaluated for every joint type profile).

When the type space is infinite we of course cannot numerically evaluate any expression for every type. We therefore replace these constraints with probabilistic constraints that must hold for a set of types which has a large probability measure. For example, an ex post individual rationality (IR) constraint would need to hold only for a set of type profiles that occurs with probability greater than 0.95.[4]

Intuitively, it is unlikely to matter if a constraint fails on a set of types which occurs with probability zero. We conjecture, further, that in most practical design problems, violation of a constraint on a low-measure set of types will also be of little consequence, either because the resulting design is easy to fix, or because the other types will likely not have very beneficial deviations even if they account in their decisions for the effect of these unlikely types on the game dynamics. We support this conjecture via a series of applications of our framework; in none of these did our constraint relaxation lead the designer much astray.

To verify probabilistic constraints over types, we evaluate the constraints on samples drawn from the type distribution. Since we can take only a finite number of samples, we verify a probabilistic constraint only at some level of confidence. The following theorem provides a lower bound on the number of samples required to achieve a given confidence level.

Theorem 1 *Let B denote a set on which a probabilistic constraint is violated, and suppose that we have a uniform prior over the interval $[0,1]$ on the probability measure of B. Then, we need at least $\frac{\log \alpha}{\log(1-p)} - 1$ samples to verify with probability at least $1-\alpha$ that the measure of B is at most p.*

Note that this result applies only for a given mechanism. Since we are running the optimization routine for a number of iterations, verifying the constraint in each, we may wish to know the probability that the constraint holds *every time* it is evaluated during the optimization run. Suppose that the constraint has been found to hold L times. It is then easy to see that if the constraint is p-satisfied in each instance with probability $1-\alpha$, then the probability it is satisfied in all L instances is $(1-\alpha)^L$.

Evaluation of an ex ante and ex interim constraint requires one to compute the value of an expectation which in general can only be done numerically. If we use Monte Carlo sampling (as we do in our implementation) to estimate the expectation, we introduce noise into constraint evaluation which the above theorem does not account for. Such noise is especially problematic when many samples are taken: if the constraint holds relatively tightly, it is quite likely that at least one estimate will be especially unfavorable, and the constraint will *appear* to be violated.[5] In our current implementation, we address this problem heuristically by declaring a constraint violated only if some evaluation exceeds an allowed tolerance level (which we set to 0.01 in most cases and 0.07 in several others).

We next describe three specific constraints employed in our applications.

Equilibrium Convergence Constraint Given that the game solutions are produced by a heuristic (iterative best-response) algorithm, they are not inherently guaranteed

[4] In fact, in all of the applications that we consider below, one may verify such constraints analytically. Our purpose, however, was to keep the framework as general as possible. As such, we wanted to demonstrate that even when we assume nothing about the specific setting, nevertheless our approach offers good results.

[5] Such artificial shrinking of the feasible space makes an already hard problem practically unsolvable.

to represent equilibria of the candidate mechanism. We can instead enforce this property through an explicit constraint. The purpose of this constraint is to ensure that every mechanism is indeed evaluated with respect to a true equilibrium (or near-equilibrium) strategy profile, given our assumption that a Bayes-Nash equilibrium is a relevant predictor of agent play. For example, best-response dynamics using RW need not converge at all.

Definition 2 Let $s(t)$ be the last strategy profile in a sequence of best-response iterations, and let $s'(t)$ immediately precede $s(t)$ in this sequence. Then the *equilibrium convergence constraint* is satisfied if for every joint type profile of players, $|s(t) - s'(t)| < \delta$ for some a priori fixed tolerance level δ.[6]

The problem that we cannot in practice evaluate this constraint for every joint type profile is resolved by making it probabilistic, as described above.

Definition 3 Let $s(t)$ be the last strategy profile produced in a sequence of solver iterations, and let $s'(t)$ immediately precede $s(t)$ in this sequence. Then the $(1-p)$-*strong equilibrium convergence constraint* is satisfied if for a set of type profiles t with probability measure no less than $1 - p$, $|s(t) - s'(t)| < \delta$ for some a priori fixed tolerance level δ.

Ex Interim Individual Rationality Ex-Interim-IR specifies that for every agent and type, the agent's expected utility conditional on its type is greater than its opportunity cost of participating in the mechanism.

Definition 4 The *ex interim IR constraint* is satisfied when for every agent $i \in I$, and for every type $t_i \in T_i$, $E_{t_{-i}}[u_i(t, s(t) \mid t_i)] \geq c_i(t_i)$, where $c_i(t_i)$ is the opportunity cost to agent i with type t_i of participating in the mechanism.

Again, in the automated mechanism design framework, we must modify the classical definition of Ex-Interim-IR to a probabilistic constraint as described above.

Definition 5 $(1 - p)$-*strong ex interim IR* is satisfied when for every agent $i \in I$, and for a set of types $t_i \in T_i$ with probability measure no less than $1 - p$: $E_{t_{-i}}[u_i(t, s(t) \mid t_i)] \geq c_i(t_i) - \delta$, where $c_i(t_i)$ is the opportunity cost of agent i with type t_i of participating in the mechanism, and δ is some a priori fixed tolerance level.

Commonly in the mechanism design literature the opportunity cost of participation, $c_i(t_i)$, is taken to be zero.

While we attempt here a very general framework, there is a special opportunity offered by the nature of individual rationality constraints that cannot be ignored: these can always be satisfied if the designer offers a sufficiently large payment to agents for participating in an auction.[7] This, of course, has a consequence for the designer's revenue, and therefore there is intimate interplay between individual rationality and revenue. To keep things general, we implement the individual rationality constraint below equivalently to all other constraints, but once a final mechanism is identified, we apply a "fix" to ensure that it holds as close to opportunity cost as possible, either lowering or raising the designer's revenue accordingly.

[6] Note that if the payoff functions are Lipschitz continuous with a Lipschitz constant L, the condition above implies that $s(t)$ is an $L\delta$-Bayes-Nash equilibrium.

[7] Observe that such constant transfers will not affect agent incentives.

Minimum Revenue Constraint The final constraint that we consider ensures that the designer will obtain some minimal amount of revenue (or bound its loss) in attaining a non-revenue-related objective.

Definition 6 The *minimum revenue constraint* is satisfied if $E_t[k(s(t), t)] \geq C$, where $k(s(t), t)$ is the total payment made to the designer by agents with joint strategy $s(t)$ and joint type profile t, and C is the lower bound on revenue.

3.4 Evaluating the Objective

As we mention above, if any constraint fails, the corresponding objective function value $W(s(\theta), \theta)$ is evaluated to $-\infty$. If all the constraints are satisfied, however, the objective must be evaluated with respect to the distribution of player types. Below, we present two approaches for doing this. The first is traditional *Bayesian mechanism design*, whereas the second is in the spirit of robust optimization, and we term it *worst-case mechanism design*.

Bayesian Mechanism Design In Bayesian mechanism design, the designer is presumed to have a belief about the distribution of agents' types. The designer's objective value for a mechanism $\theta \in \Theta$ is evaluated by taking the expectation of $W(s(t, \theta), t, \theta)$ with respect to the distribution of player types,

$$W(s(\theta), \theta) = E_t[W(s(t, \theta), t, \theta)].$$

We assume for convenience that the designer has the same belief about agent types as the agents themselves, although this assumption could be straightforwardly relaxed.

Worst-Case Mechanism Design We address the problem of worst-case mechanism design by appealing to the analogous problem in the optimization literature (Ben-Tal and Nemirovski, 2002). Robust optimization treats uncertain parameters of an optimization program as though they are selected by an adversary aiming to produce the worst outcome for the problem at hand. The analogy here comes from allowing the adversary to select a profile of player types.

Formally, we can express the robust objective of the designer as

$$W(s(\theta), \theta) = \inf_{t \in T} W(s(t, \theta), t, \theta).$$

Note that this change is syntactically minor and has no effect on the rest of the framework (replacing the expectation operator with the infimum). However, it entails a computationally infeasible problem of ensuring robustness for every joint type of a possibly infinite type space; anything short of that is no longer really worst-case. To address this problem, we relax the pure robustness criterion to probabilistic robustness.[8] Our relaxation is that the designer is not worried about the worst subset of outcomes of the type space if that subset has very small

[8] To clarify, the critical issue is not so much the impossibility of computing the objective value exactly: this problem obtains even in the Bayesian mechanism design setting. Rather, the relaxation is necessary in order to enable us to speak in a meaningful way about objective estimation and to obtain probabilistic bounds, such as the one we present below.

measure. For example, if the set of types that has probability measure of 0.0001 are extremely unfavorable, their appearance is deemed sufficiently unlikely not to worry the designer. Furthermore, we can probabilistically ascertain that the worst outcome based on a finite number of samples from the type distribution is no better than a large measure of the type space. We refer to the resulting mechanism as *probably approximately robust*.

To formalize this, suppose that in every exploration step using our framework one takes n samples from the type distribution, $T^n = \{T_1, \ldots, T_n\}$, and then selects the worst value of the objective over these n types:

$$\hat{W}(s(t,\theta),t,\theta) = \inf_{t \in T^n} W(s(t,\theta),t,\theta).$$

One would like to select a sufficiently high number of samples n, in order to attain high enough confidence, $1-\alpha$, that the best objective value that he can obtain via L explorations using this framework is approximately robust. The following theorem gives such an n.

Theorem 2 *Suppose we select the best design of L candidates, using n samples from the type distribution for each to estimate the value of $\inf_{t \in T \setminus T_A} W(s(t,\theta),t,\theta)$, where T_A is the set of types with value of $W(s(t,\theta),t,\theta)$ below $\hat{W}(s(t,\theta),t,\theta)$. To attain a confidence of at least $1-\alpha$ that the measure of T_A is at most p, we need*

$$n \geq \frac{\log(1-(1-\alpha)^{\frac{1}{L}})}{\log(1-p)}$$

samples.

In all of our automated mechanism design examples and applications below, we use 500 samples to evaluate either the Bayesian or robust objectives, and run 50 iterations of an optimization routine. Theorem 2 then tells us that $\Pr T_A \leq 0.02$ with confidence $1-\alpha \geq 0.99$.

4 Extended Example: Shared-Good Auction

4.1 Setup

Consider the problem of two people trying to decide how to allocate a shared good. Unless both players prefer the same allocation, no standard voting mechanism (with either straight votes or a ranking of the alternatives) can help with this problem. We propose a simple *shared-good auction* (SGA): each player submits a bid and the player with the higher bid wins the good, paying some function of the bids to the loser in compensation. Reeves (2005) considered a special case of this auction and gave the example of two roommates using it to decide who should get the bigger bedroom and for how much more rent. Cramton et al (1987) and McAfee (1992) considered this problem in the context of dissolving partnerships.

We define a space of mechanisms for this problem that are all budget-balanced, individually rational, and (assuming monotone strategies) socially efficient. We then search the mechanism space for games that satisfy additional properties. The following is a payoff function defining a space of games parametrized by a payment function f.

$$u(t,a,t',a') = \begin{cases} t - f(a,a') & \text{if } a > a' \\ f(a',a) & \text{if } a < a', \end{cases} \tag{1}$$

where $u(\cdot)$ gives the utility for an agent who has a value t for winning and bids a against an agent who has value t' and bids a'. The semantics are that the winner (i.e., the player with the higher bid) pays $f(a,a')$ to the loser, where a in this case is the winning and a' the losing bid. In the tie-breaking case (which occurs with probability zero for many classes of strategies) the payoff is the average of the two other cases because the winner is chosen by a coin flip.

We now consider a restriction of the class of mechanisms defined above.

Definition 7 SGA(h,k) is the mechanism defined by Equation (1) with $f(a,a') = ha + ka'$, $h,k \in [0,1]$.

For example, in SGA$(1/2,0)$ the winner pays half its own bid to the loser; in SGA$(0,1)$ the winner pays the loser's bid to the loser. More generally, h and k will be the relative proportions of winner's and loser's bids that will be transferred from the winner to the loser. We now give Bayes-Nash equilibria for such games when types are uniformly distributed.

Theorem 3 *For $h,k \geq 0$ and types $U[A,B]$ with $B \geq A+1$ the following is a symmetric Bayes-Nash equilibrium of* SGA(h,k)*:*

$$s(t) = \frac{t}{3(h+k)} + \frac{hA+kB}{6(h+k)^2}.$$

For the following discussion, we need to define the notion of truthfulness, or Bayes-Nash incentive compatibility.

Definition 8 (BNIC) A mechanism is *Bayes-Nash incentive compatible* (truthful) if bidding $s(t) = t$ constitutes a Bayes-Nash equilibrium of the game induced by the mechanism.

For example, it follows directly from Theorem 3 that SGA$(1/3,0)$ is BNIC for $U[0,B]$ types. We now show that this is the *only* truthful mechanism in the SGA(h,k) design space.

Theorem 4 *With $U[0,B]$ types ($B \geq 1$),* SGA(h,k) *is BNIC if and only if $h = 1/3$ and $k = 0$.*

Below, we use this characterization to present concrete examples of the failure of the revelation principle for several sensible designer objectives.[9] Since SGA$(1/3,0)$ is the only truthful mechanism in our design space, we can directly compare the objective value obtained from this mechanism and the best untruthful mechanism in the sections that follow. From here on we focus on the case of $U[0,1]$ types.

[9] We emphasize that our parametric restriction on the design space was not introduced in order to doom the revelation principle. Rather, the requirement that payment functions be linear in player bids was motivated in part by tractability of best-response calculation and in part by the simplicity of the resulting mechanism.

4.2 Automated Design Problems

4.2.1 Bayesian Mechanism Design Problems

Minimize Difference in Expected Utility First, we consider *fairness*, or negative difference between the expected utility of winner and loser, as the objective. Formally, the goal is to minimize

$$|E_{t \geq t'}[u(t, s(t), t', s(t'), k, h) - u(t', s(t'), t, s(t), k, h)]| \tag{2}$$

We first use the equilibrium bid derived above to analytically characterize optimal mechanisms.

Theorem 5 *The difference in expected utility* (2) *for* SGA(h, k) *is*

$$\frac{2h + k}{9(h + k)}.$$

Furthermore, SGA$(0, k)$*, for any* $k > 0$*, minimizes this objective, and the optimal value is* $1/9$.

By comparison, the objective value for the truthful mechanism, SGA$(1/3, 0)$, is $2/9$, twice as high as the minimum produced by an untruthful mechanism. Thus, the revelation principle does not hold for this objective function in the specified design space. We can use Theorem 5 to find that the objective value for SGA$(1/2, 0)$, the mechanism described by Reeves (2005), is also $2/9$.

Now, suppose we do not know about the above analytic derivations, including the characterization of Bayes-Nash equilibrium. To evaluate the automated mechanism design framework, we run the AMD procedure (recall from Section 3.1) in "black-box" mode. Table 1 presents results of AMD for two methods of initializing h and k values. Since the objective function turns out to be fairly simple, it is not surprising that we obtain the optimal mechanism for specific and random starting points.

Parameters	Initial Design	Final Design
h	0.5	0
k	0	1
objective	2/9	1/9
h	random	0
k	random	1
objective	N/A	1/9

Table 1 Design that approximately minimizes difference in expected utility between utility of winner and loser (maximizes fairness) when the optimization search starts at a fixed starting point ($h = 0.5$ and $k = 0$), and the best mechanism from five random restarts.

Minimize Expected (Ex Ante) Difference in Utility Here we modify the objective function slightly as compared to the previous section, and instead aim to minimize the expected ex ante difference in utility:

$$E_{t \geq t'}[|u(t, s(t), t', s(t'), k, h) - u(t', s(t'), t, s(t), k, h)|]. \tag{3}$$

Although the only difference from the previous section is the placement of the absolute value sign inside the expectation, this difference complicates the analytic derivation of the optimal design considerably. Therefore, we cannot present the optimum design values in closed form.

Parameters	Initial Design	Final Design
h	0.5	0.49
k	0	1
objective	0.22	0.176
h	random	0.29
k	random	0.83
objective	N/A	0.176

Table 2 Design that approximately minimizes expected ex ante difference between utility of winner and loser when the optimization search starts at a random and a fixed starting points.

The results of application of our AMD framework are presented in Table 2. Though the objective function in this example appears somewhat complex, it turns out (as we discovered through additional exploration) that there are many mechanisms that yield nearly optimal objective values.[10] Thus, both random restarts as well as a fixed starting point produced essentially the same near-optima. By comparison, the truthful design ($\mathrm{SGA}(1/3, 0)$) yields the objective value of about 0.22, which is considerably worse.

4.2.2 Worst-Case Mechanism Design Problems

Minimize Nearly-Maximal Difference in Utility Here, we study the problem of probably approximately robust design to minimize maximal difference in players' utility (that is, to maximize a notion of robust fairness). The robust formulation of this problem is to minimize

$$\sup_{t \geq t'} |u(t, s(t), t', s(t'), k, h) - u(t', s(t'), t, s(t), k, h)|.$$

Theorem 6 *The maximal difference in expected utility in* $\mathrm{SGA}(h, k)$ *(i.e., worst case with respect to agent types) is*

$$\frac{h + 2k}{3(h + k)}.$$

Thus, $k = 0$ *is robust optimal for any* $h > 0$*, and the robust optimal value is* $1/3$.

As one can see from the results in Table 3, the mechanism produced via the automated framework is optimally robust, as the optimum corresponds to one of the robust designs in Theorem 6.

Of the examples we considered so far, most turned out to be analytic, and one we could approach only numerically. Nevertheless, even in the analytic cases, the objective function forms were not trivial, particularly from a blind optimization perspective. Furthermore, one must take into account that even the simple cases

[10] We carried out a far more intensive exploration of the search space given the analytic expression for the Bayes-Nash equilibrium to ascertain that the values reported are close to actual optima.

Parameters	Initial Design	Final Design
h	random	0.01
k	random	0
objective	N/A	1/3

Table 3 Design that approximately minimizes the maximum difference in utility.

are somewhat complicated by the presence of noise, and thus one need not arrive at global optima even in the simplest of settings without a very large number of samples.

Having found success in the simple shared-good auction setting, we now turn our attention to a series of considerably more difficult problems.

5 Applications

We present results from several applications of our automated mechanism design framework to specific two-player problems. One of these problems, finding auctions that yield maximum revenue to the designer, has been studied in a seminal paper by Myerson (1981) in a much more general setting than the one we consider. Another, which seeks to find auctions that maximize social welfare, has also been studied more generally. For these, and other instances we were able to solve analytically, we can compare the AMD results to a known benchmark. Others have no known optimal design.

An important consideration in any optimization routine is the choice of a starting point. This could be especially relevant where AMD is used as a tool to enhance an already working mechanism through parametrized search. We explore this possibility in one of our applications, using a previously studied design as a starting point. Additionally, we apply our framework to every application with completely randomly seeded optimization runs, taking the best result of five randomly seeded runs in order to mitigate the problem of local optima. Furthermore, we enhance the optimization procedure by using a *guided* restart, that is, by running the optimization procedure once using the current best mechanism as a new starting point. Each optimization run lasted up to several hours: the actual running time was determined in part by the running time of RW solver in computing Bayes-Nash equilibria.

In all of our applications, player types are independently distributed with uniform distribution on the unit interval. We used 50 samples from the type distribution to verify Ex-Interim-IR throughout the run of the AMD framework. If all samples satisfy the constraint, this gives us 0.95 probability that 94% of types lose no more than the opportunity cost plus a specified tolerance we add to ensure that the presence of noise does not overconstrain the problem. If we choose to fully account for the fact that we are verifying this constraint in each of 50 optimization iterations, we get a very pessimistic bound of 0.08 on the probability that it is satisfied for the mechanism ultimately found by the framework. However, note that most iterations consider mechanisms found to be infeasible. Thus, for example, if we instead suppose that the "real" choice was between 10 feasible, high-quality mechanisms, we obtain a much more palatable bound of 0.6. It turns out that every application that we consider produces a mechanism that is individually rational for all types *with respect to the tolerance level that was set*. Thus, our choice of parameters here is ultimately empirically justified. Once the final mechanism is produced, we "fix" Ex-Interim-IR by computing the difference between expected value and payment of the least fortunate type both analytically and by sampling over types. The computational verification of Ex-Interim-IR uses 10 times as many samples to estimate expected value and payment to each type as during the standard run of the framework. We report both the actual and approximated Ex-Interim-IR adjustment below.

One problem with choosing a mechanism that is empirically maximal is that a mediocre or poor mechanism could be chosen merely because of a lucky sample, or, perhaps, because by some fortunate coincidence it had passed the empirical constraint test even while in reality violating it. Noting that this problem is only important with mechanisms that are "current best" (that is, those that are better than any that were previously encountered during the optimization run), we periodically reevaluate the objective of the current best mechanism (specifically, we do so with probability 0.2 in every iteration).

5.1 Myerson Auctions

The seminal paper by Myerson (1981) presented a theoretical derivation of revenue maximizing auctions in a relatively general setting. Here, our aim is to find a mechanism with a nearly optimal value of some given objective function, of which revenue is an example. However, we restrict ourselves to a considerably less general setting than did Myerson,[11] constraining the design space to that described by the parameters in (4).

$$u(t,a,t',a') = \begin{cases} qt - k_1 a - k_2 a' - K_1 & \text{if } a > a' \\ 0.5(t - (k_1+k_3)a - (k_2+k_4)a' - K_1 - K_2) & \text{if } a = a' \\ (1-q)t - k_3 a - k_4 a' - K_2 & \text{if } a < a' \end{cases} \tag{4}$$

We further constrain all the design parameters to be in the interval [0,1]. In standard terminology, the designer specifies the probability q that the winner (i.e., agent with the larger bid) gets the good, along with a schedule of transfers that are linear in agents' bids.

5.1.1 Bayesian Mechanism Design Problems

Maximize Revenue We begin by seeking approximately revenue-maximizing designs in our parametrized design space. Based on Myerson's feasibility constraints, we derive in the following theorem that an optimal incentive compatible mechanism in this design space yields revenue of 1/3 to the designer,[12] as compared to 0.425 in the general two-player case.[13]

[11] Conitzer and Sandholm (2003a) also tackled a restricted version of Myerson's problem, constrained to finite type and strategy spaces of agents, as well as a finite design space.

[12] For example, Vickrey auction will yield this revenue.

[13] The optimal mechanism prescribed by Myerson is not implementable in our design space, since the designer is in effect not allowed to introduce a positive reserve price for the good.

Lemma 1 *The mechanism in the design space described by the parameters in* (4) *is BNIC and Ex-Interim-IR if and only if* $k_3 = k_4 = K_1 = K_2 = 0$ *and* $q - k_1 - 0.5k_2 = 0.5$.

Theorem 7 *An optimal incentive compatible mechanism in our setting yields revenue of* $1/3$*, which can be achieved by selecting* $q = 1$*,* $k_1 \in [0, 0.5]$*, and* $k_2 \in [0, 1]$*, respecting the constraint that* $k_1 + 0.5k_2 = 0.5$.

Parameters	Initial Design	Final Design
q	random	0.96
k_1	random	0.95
k_2	random	0.84
K_1	random	0.78
k_3	random	0.73
k_4	random	0
K_2	random	0.53
objective	N/A	0.3

Table 4 Design that approximately maximizes the designer's revenue.

The automated mechanism design procedure produced the design in Table 4. We now verify the Ex-Interim-IR and revenue properties of this design.

Theorem 8 *The design described in Table 4 is Ex-Interim-IR and yields expected revenue of approximately* 0.3*. Furthermore, the designer could gain an additional* 0.0058 *in expected revenue without effect on incentives while maintaining the individual rationality constraint.*

By verifying Ex-Interim-IR computationally, we find that the designer would gain an additional 0.0057 using participation fees, only slightly below an "optimal" setting of such fees described in the theorem. Thus, our AMD framework produced a design near to the best known. It is an open question what the actual global optimum is.

Maximize Welfare It is well known that the Vickrey auction is welfare-optimal. Thus, we know that the welfare optimum is attainable in the specified design space. Before proceeding with search, however, we must make one observation. While we are interested in welfare, it would be inadvisable in general to completely ignore the designer's revenue, since the designer is unlikely to be persuaded to run a mechanism at a disproportionate loss. To remedy this problem, we use a minimum revenue constraint, ensuring that no mechanism that is too costly will be selected as optimal.

First, we present a general result that characterizes welfare-optimal mechanisms in our setting.

Theorem 9 *Welfare is maximized if either the equilibrium bid function is strictly increasing and* $q = 1$ *or the equilibrium bid function is strictly decreasing and* $q = 0$*. Furthermore, the maximum expected welfare in the specified design space is* $2/3$.

Parameters	Initial Design	Final Design
q	random	1
k_1	random	0.88
k_2	random	0.23
K_1	random	0.28
k_3	random	0.06
k_4	random	0.32
K_2	random	0
objective	N/A	2/3

Table 5 Design that approximately maximizes welfare.

Thus, for example, both first- and second-price sealed bid auctions are welfare maximizing (as is well known).

In Table 5 we present the result of our search for optimal design. We verified using the RW solver that the bid function $s(t) = 0.645t - 0.44$ is an equilibrium given this design. Since it is strictly increasing in t, we can conclude based on Theorem 9 that *this design is welfare-optimal.* We need only to verify that both the minimum revenue and the individual rationality constraints hold.

Theorem 10 *The design described in Table 5 is Ex-Interim-IR, welfare optimal, and yields the revenue of approximately* 0.2*. Furthermore, the designer could gain an additional* 0.128 *in revenue (for a total of about* 0.33*) without affecting agent incentives or compromising individual rationality and optimality.*

Computationally verifying the Ex-Interim-IR gap we find that the designer could gain 0.128, exactly as described in the theorem. It is interesting that this auction, besides being welfare-optimal, also yields a slightly higher revenue to the designer than our mechanism in the previous section if we implement the modification proposed in Theorem 10. Thus, there appears to be some synergy between optimal welfare and optimal revenue in our design setting.

5.1.2 Worst-Case Mechanism Design Problems

Maximize Nearly-Minimal Revenue The objective in this section is to maximize minimal revenue to the designer over the entire joint type space. Formally, the objective function is

$$\begin{aligned}\inf_{t,t' \in T | s(t) > s(t')} & [k_1 s(t) + k_2 s(t') + k_3 s(t') + k_4 s(t)] + \\ & \inf_{t,t' \in T | s(t) < s(t')} [k_1 s(t') + k_2 s(t) + k_3 s(t) + k_4 s(t')] + \\ & \inf_{t,t' \in T | s(t) = s(t')} [(k_1 + k_2 + k_3 + k_4) s(t)] + K_1 + K_2.\end{aligned} \tag{5}$$

Assuming symmetry, here is a simple result about the set of mechanisms that yields 0 for the objective in (5).

Theorem 11 *Any auction with* $K_1 = K_2 = 0$ *which induces equilibrium strategies in the form* $s(t) = mt$ *with* $m > 0$ *yields* 0 *as the value of the objective in* (5)*.*

Thus, both first-price and second-price sealed-bid auctions result in the value of 0 for the worst-case objective. Furthermore, by Lemma 1 it follows that the same is true for *any* BNIC and ex interim individually rational mechanism in the specified design space.

Since it is far from clear what the actual optimum is for this problem or for its probably approximately robust equivalent, we ran our automated framework to obtain an approximately optimal design, shown in Table 6.

Parameters	Initial Design	Final Design
q	random	1
k_1	random	1
k_2	random	0.34
K_1	random	0.69
k_3	random	0
k_4	random	0
K_2	random	0
objective	N/A	0.0066

Table 6 Design that approximately robustly maximizes revenue.

Theorem 12 *The mechanism in Table 6 yields the value of* 0.0066 *for the worst-case objective. While it is not Ex-Interim-IR, it can be made so by paying each agent a fixed* 0.000022*, resulting in the adjusted worst-case objective value above* 0.0065*.*

Computing the "fix" automatically, we would pay each agent slightly more, and would then lose 0.000067 in revenue. Thus, we confirm that while not precisely individually rational, our mechanism is very nearly so, and with a small adjustment becomes individually rational with little cost to the designer. Furthermore, the designer is able to make a positive (albeit small) profit no matter what the joint type of the agents.

5.2 Auctions with Anti-Social Bidders

In this section we study a mechanism design problem motivated by the *vicious Vickrey* auction (Brandt and Weiß, 2001; Brandt et al, 2007; Morgan et al, 2003; Reeves, 2005). Auction games with anti-social bidders capture a notion of spite, where each player gets disutility from the surplus of the other, to a degree modeled by parameter l. For example, the standard Vickrey auction is a special case of the vicious Vickrey auction with $l = 0$.

We further generalize auctions with anti-social bidders beyond Vickrey to cover the Myerson-inspired family of mechanisms discussed in Section 5.1 above. Formally, the auction utility of an anti-social bidder is described by the following parametrized form.

$$u(t, a, t', a') = \begin{cases} U_1 & \text{if } a > a' \\ 0.5(U_1 + U_2) & \text{if } a = a' \\ U_2 & \text{if } a < a' \end{cases} \tag{6}$$

$$\begin{aligned}
U_1 &= q(1-l)t - (k_1(q(1-l)+(1-q)) - (1-q)l)a \\
&\qquad - ((1-q)l)t' - k_2(q(1-l)+(1-q))a' - K_1 \\
U_2 &= (1-q)(1-l)t - (k_3((1-q)(1-l)+q) - ql)a \\
&\qquad - qlt' - k_4((1-q)(1-l)+q)a' - K_2.
\end{aligned}$$

The vicious Vickrey auction is a special case of (6) with $q = k_2 = 1$ and $k_1 = k_3 = k_4 = K_1 = K_2 = 0$. The Myerson auction utility function (4) analyzed in the previous section is likewise a special case with $l = 0$.

For all the analysis below, we fix $l = 2/7$. Reeves (2005) reports an equilibrium for vicious Vickrey with this value of l to be $s(t) = (7/9)t + 2/9$. Thus, we can see that we are no longer assured incentive compatibility even in the second-price auction case. In general, it is unclear whether there exist incentive compatible mechanisms in this design space, particularly because we constrain all the parameters to be in the interval $[0, 1]$.

For this setting, we adopt a slightly modified definition of individual rationality: every agent can earn nonnegative expected value less expected payment (that is, expected surplus).[14] To formalize, we constrain that

$$\mathrm{EU}(t) = v(t) - m(t) \geq 0,$$

where

$$v(t) = qE_{s(t)>s(t')}[t] + (1-q)E_{s(t)<s(t')}[t] + 0.5E_{s(t)=s(t')}[t]$$

is the expected net value (type less cost, in the "traditional sense") to agent with type t and

$$\begin{aligned}
m(t) &= E_{s(t)>s(t')}[k_1 s(t) + k_2 s(t') + K_1] + E_{s(t)<s(t')}[k_3 s(t) + k_4 s(t') + K_2] \\
&\quad + 0.5E_{s(t)=s(t')}[(k_1+k_3)s(t) + (k_2+k_4)s(t') + K_1 + K_2]
\end{aligned}$$

is the expected payment to the auctioneer by the agent with type t.

5.2.1 Bayesian Mechanism Design Problems

Maximize Revenue The first objective is to (nearly) maximize revenue. The results of automated mechanism design in two distinct cases are presented in Table 7. The top part of Table 7 presents the results of simulated annealing search that uses the previously studied vicious Vickrey as a starting point. Our purpose for doing so is twofold. First, we would like to see if we can easily (i.e., via an automated process) do better than the previously studied mechanism. Second, we want to suggest automated mechanism design as a framework not only for finding good mechanisms from scratch, but also for improving mechanisms that are initially designed by hand. The latter could become especially useful in practice when applications are extremely complex and we can use theory and intuition to give us good starting mechanisms.

First, we determine the expected revenue and individual rationality properties of the vicious Vickrey auction in the following theorem.

[14] To contrast, a usual definition would guarantee the agent nonnegative expected *utility*, which we feel is too strong a requirement for this setting.

Parameters	Initial Design	Final Design
q	1	1
k_1	0	0
k_2	1	0.98
K_1	0	0.09
k_3	0	0.33
k_4	0	0
K_2	0	0
objective	0.48	0.49
q	random	1
k_1	random	1
k_2	random	0.33
K_1	random	0.22
k_3	random	0.22
k_4	random	0.12
K_2	random	0
objective	N/A	0.44

Table 7 Design that approximately maximizes revenue.

Theorem 13 *The expected revenue from vicious Vickrey auction with $l = 2/7$ is approximately 0.480. This auction is not Ex-Interim-IR, but can be adjusted by awarding each agent 0.021. The adjusted revenue would become 0.438.*

We now give the individual rationality and revenue properties of the auction that AMD obtains with vicious Vickrey as the starting point.

Theorem 14 *The expected revenue from the auction $\langle 1, 0, 0.98, 0.09, 0.33, 0, 0\rangle$ in Table 7 is approximately 0.49. This auction is Ex-Interim-IR, and will remain so if the designer charges a fixed entry fee of 0.0027, giving itself a total revenue of approximately 0.4932.*

Verifying Ex-Interim-IR computationally, we would charge each agent 0.000685, giving us 0.00137 in gain, slightly below the "optimal" prescription in the theorem. Thus, we found a design which yields more revenue than the design previously studied in the literature (adjusted to be individually rational).

Next, we performed search from a random starting point. The results are shown in the lower section of Table 7. Properties of the resulting auction are explored in Theorem 15.

Theorem 15 *The expected revenue from the auction $\langle 1, 1, 0.33, 0.22, 0.22, 0.12, 0\rangle$ in Table 7 is approximately 0.44. This auction is Ex-Interim-IR, and can remain so if the designer charges all agents an additional fixed participation fee of 0.0199. This design change would increase the expected revenue to 0.4798.*

Computationally verifying Ex-Interim-IR in this case yields a revenue increase of 0.38, only a little bit below the optimal adjustment. Thus, the design we obtained from a completely random starting point yields revenue that is not far below that of vicious Vickrey (or the design that we found using vicious Vickrey as a starting point), and is better than vicious Vickrey if the latter is adjusted to be individually rational. Furthermore, this design can be improved considerably via a participation tax without sacrificing individual rationality.

Parameters	Initial Design	Final Design
q	random	0.37
k_1	random	0.8
k_2	random	1
K_1	random	0.49
k_3	random	0.29
k_4	random	0.67
K_2	random	0.48
objective	N/A	0.54

Table 8 Design that approximately maximizes welfare.

Maximize Welfare In Table 8 we present an outcome of the automated mechanism design process with the goal of maximizing welfare. In the optimization, we utilized both the Ex-Interim-IR and minimum revenue constraints. In the following theorem we establish the welfare, revenue, and individual rationality properties of this mechanism.

Theorem 16 *The expected welfare of the mechanism in Table 8 is approximately 0.54 and expected revenue is approximately 0.225. It is Ex-Interim-IR for all types in [0.17,1] and can be made Ex-Interim-IR for every type at an additional loss of 0.13 in revenue.*

By computing the amount of violation in Ex-Interim-IR in an automated mode, we would sacrifice 0.132 in revenue, slightly more than the amount prescribed based on an exact derivation. Thus, while individual rationality does not hold for almost 80% of types, this failure is easy to remedy at some additional loss in revenue (importantly, the adjusted expected revenue is positive).

Nevertheless, after a sequence of successful applications of AMD, we stand before an evident failure: the mechanism we found is quite a bit below the known optimum of 2/3. Interestingly, recall that the optimal revenue mechanism in the anti-social auction setting had a strictly increasing bid function and $q = 1$, and consequently was also welfare-optimal by Theorem 9.

We hypothesize that the most important reason for the poor results is that we introduced nonnegative revenue as a hard constraint. From observing the optimization runs in general, we notice that the optimization problem both in the Myerson auctions and the anti-social auctions design space seems to be rife with islands of local optima in the sea of infeasibility. Thus, the problem was difficult for black-box optimization already, and we made it considerably more difficult by adding more infeasible regions. In general, we would expect such optimization techniques to work best when the objective function varies smoothly and most of the space is feasible. Hard constraints make it more difficult by introducing (at least in our implementation) spikes in the objective value.[15]

We have seen some evidence to the correctness of our hypothesis already, since our revenue-optimal design also happens to maximize social utility. To test our hypothesis directly, we remove minimum revenue as a hard constraint in the next section, and instead try to maximize the weighted sum of welfare and revenue.

[15] Recall that we implemented hard constraints as a very low value of the objective. Thus, adding hard constraints increases nonlinearity of the objective function, and the increase could be quite dramatic.

Maximize Weighted Sum of Revenue and Welfare In this section, we present results of AMD with the goal of maximizing the weighted sum of revenue and welfare.[16] For simplicity (and having no reason for doing otherwise), we set weights to be equal. A design that our framework found from a random starting point is presented

Parameters	Initial Design	Final Design
q	random	1
k_1	random	0.51
k_2	random	1
K_1	random	0.09
k_3	random	0.34
k_4	random	0.26
K_2	random	0
objective	N/A	0.6372

Table 9 Design that approximately maximizes the average of welfare and revenue.

in Table 9. We verified using RW that $s(t) = 0.935t - 0.18$ is an (approximate) symmetric equilibrium bid function. Thus, by Theorem 9 this auction is welfare-optimal.

Theorem 17 *The expected revenue from the auction in Table 9 is 0.6078. However, it is not Ex-Interim-IR, and the least fortunate type loses nearly 0.044. However, by compensating the agents the designer can induce individual rationality without affecting incentives, at a revenue loss of 0.088. This would leave it with an adjusted expected revenue of 0.5198.*

In computational mode, we find the revenue loss to be 0.088, exactly the same as in the theorem. Interestingly, we were much more successful in both revenue and welfare objectives by eliminating the hard minimum revenue constraint and instead making it a part of the objective. Indeed, we found here the best mechanism so far for *both* objectives we considered, suggesting that there is substantial synergy between the two objectives.

5.2.2 Worst-Case Mechanism Design Problems

Maximize Nearly-Minimal Revenue We now apply our framework to the problem of maximizing worst-case revenue of the designer. First, we present the result for the previously studied vicious Vickrey auction.

Theorem 18 *By running the vicious Vickrey auction, the designer can obtain at least* $2/9$ *(*≈ 0.22*) in revenue for any joint type profile. By adjusting to make the auction individually rational, minimum revenue falls to* $220/1089$ *(*≈ 0.20*).*

The results from running our automated design framework from a random starting point are shown in Table 10. We now verify the revenue and individual rationality properties of this mechanism.

[16] This can alternatively be viewed as applying a penalty method to the problem of welfare maximization under a revenue constraint (Nocedal and Wright, 2006). We explore the use of a penalty method in our framework explicitly below.

Parameters	Initial Design	Final Design
q	random	0.86
k_1	random	1
k_2	random	0.71
K_1	random	0.14
k_3	random	0
k_4	random	0.09
K_2	random	0
objective	N/A	0.059

Table 10 Design that approximately maximizes minimum revenue.

Theorem 19 *The design in Table 10 yields revenue of at least 0.059 to the designer for any agent type profile, but is not ex interim individually rational. It can be made such if the designer awards each agent 0.0095 for participation, yielding the adjusted revenue of 0.04.*

Computationally we find that each agent should be paid 0.01 for participation, yielding 0.039 in adjusted revenue, or very near that given the optimal adjustment. As we can see, the randomly generated design is considerably worse than the adjusted vicious Vickrey. However, adjusted vicious Vickrey requires negative settings of several of the design parameters. Since the parameters are initially constrained to be nonnegative, it is unclear whether a better solution is indeed attainable in the specified constrained design space, even at a slight (< 0.02) sacrifice in individual rationality.

6 Truthful Mechanisms

While this paper comes, in part, to address a need for designing general—that is, not necessarily truthful (or even direct)—mechanisms, we now revisit incentive compatible mechanism design. A natural question that may arise from the above examples is whether we can, perhaps, do better by only searching through a space of (Bayes-Nash) incentive compatible mechanisms. Recall, for example, that our automatically generated mechanism in the context of "Myerson" revenue maximization (Section 5.1.1) failed to match the optimal truthful auction (although came relatively close).

Operationalizing the idea of searching in the space of truthful mechanisms in general is not immediate. One way we can do this is to constrain the parameter space to guarantee truthfulness, per Lemma 1. However, this would require us to derive characterizations of Bayes-Nash incentive compatibility in every context. An approach more in the spirit of automated mechanism design is to include incentive compatibility as a constraint (Conitzer and Sandholm, 2002). The danger, however, is that this additional constraint would prove too much for our implementation, since the space of incentive compatible mechanisms is quite small. As we shall see, the problem with overconstraining optimization can be solved by moving constraints into the objective, that is, by the use of penalty methods (Nocedal and Wright, 2006).

6.1 Bayes-Nash Incentive Compatibility Constraint

Our first step is to formally define the Bayes-Nash incentive compatibility (BNIC) constraint.[17]

Definition 9 The *BNIC constraint* is satisfied when for every agent $i \in I$, and for every type $t_i \in T_i$, $E_{t_{-i}}[u_i(t, s(t) \mid t_i)] \geq \max_{t'_i \in T_i} E_{t_{-i}}[u_i(t, t'_i, s_{-i}(t_{-i}) \mid t_i)]$.

Again, in the automated mechanism design framework, we must modify the classical definition of BNIC to a probabilistic constraint.

Definition 10 The $(1-p)$*-strong BNIC constraint* is satisfied when for every agent $i \in I$, and for a set of types $t_i \in T_i$ with probability measure no less than $1-p$, $E_{t_{-i}}[u_i(t, s(t) \mid t_i)] \geq \max_{t'_i \in T_i} E_{t_{-i}}[u_i(t, t'_i, s_{-i}(t_{-i}) \mid t_i)] - \delta$, where δ is some a priori fixed tolerance level.

Since the maximization in the above definition is still over the possibly infinite set of deviations T_i, we can repeat the exercise and focus on making statements about a "large measure" of deviations in T_i, and take the maximum over a finite sample of types. We do not make the extension formally since it is relatively direct and entirely analogous to the p-strong relaxation of constraints, as well as our relaxation of worst-case mechanism design above.

6.2 Application to Revenue Maximization in Myerson Auctions

We now search for revenue-optimal Myerson auctions in the parameter space identified in Section 5.1, with the additional BNIC constraint. Recall that an optimal BNIC mechanism in this setting yields revenue of 1/3, so we have a precise bar for which to strive.

Above, we suspected that perhaps the addition of BNIC constraint would prove too much for the current AMD implementation, in which a constraint failure returns negative infinity as the objective value. These fears proved true: we failed to find a feasible mechanism over a series of runs. We consequently made a modification to the basic framework as regards constraints: all constraints are moved into the objective and their failure magnitude is penalized, as is standard in *penalty methods* (Nocedal and Wright, 2006). It turns out that there is a natural way to measure the magnitude of constraint failure for both the Ex-Interim-IR and the BNIC constraints. In the former case, the magnitude of failure for a particular player i and type t_i is (upon setting the opportunity cost to zero):

$$-\min\{E_{t_{-i}}[u_i(t, s(t) \mid t_i)], 0\}.$$

We then take the maximum constraint failure over all types and players. Similarly, the magnitude of failure of the BNIC constraint for player i and type t_i is

$$\max_{t'_i \in T_i} E_{t_{-i}}[u_i(t, t'_i, s_{-i}(t_{-i}) \mid t_i)] - E_{t_{-i}}[u_i(t, s(t) \mid t_i)],$$

[17] We already defined it above, but for the purposes of our computational framework it will pay to be a bit more precise in the definition here, particularly as we extend it to a p-strong variant.

which, upon maximizing over players and types, yields the well-known notion of *game-theoretic regret*. Naturally, these are approximated in our implementation by taking the maxima over a finite set of player types (and a finite set of deviations t'_i in the case of the BNIC constraint). In the experiments we checked 50 player types and 11 deviations (we also restricted deviations to the unit interval, since the designer would commonly constrain bids to be positive, and it seems reasonable to also prevent bids above the higest possible valuation).

The mechanism obtained by running the search procedure is presented in Table 11. We used penalty values of 3 for the initial five runs and 4 for the final run, identical for both constraints. To begin, let us compare the results above to

Parameters	Initial Design	Final Design
q	random	1
k_1	random	0.62
k_2	random	0.3655
K_1	random	0
k_3	random	0
k_4	random	0
K_2	random	0
objective	N/A	0.533

Table 11 Design that approximately maximizes average revenue under Ex-Interim-IR and BNIC.

Lemma 1 and Theorem 7. We can readily observe that some of the preconditions of BNIC and Ex-Interim-IR hold, specifically, that $k_3 = k_4 = K_1 = K_2 = 0$. Indeed, it is easy to verify that Ex-Interim-IR holds, since $k_1 + k_2 < 1$. As Lemma 1 gives necessary and sufficient conditions for BNIC, we can also easily confirm that BNIC in our mechanism is not satisfied: the condition that $q - k_1 - 0.5k_2 = 0.5$ fails. Nevertheless, the mechanism may well be sufficiently close to truthful for practical purposes if players do not have very much to gain by deviating. To compute the gain from deviation for a fixed player i with type t_i, first note that expected utility to this player for playing t'_i is

$$\begin{aligned} E_{t_{-i}}[u_i(t, t'_i, s_{-i}(t_{-i}) \mid t_i)] &= \int_0^{t'_i} t_i - 0.62t'_i - 0.3655T dT \\ &= t_i t'_i - 0.62t'^2_i - 0.18t'^2_i = t_i t'_i - 0.8t'^2_i, \end{aligned}$$

which is maximized at $t'_i = s_i(t_i) \approx 0.62t_i$. The gain to optimal deviation from truthful bidding is then

$$0.62t_i^2 - 0.8(0.62)^2 t_i^2 - (t_i^2 - 0.8t_i^2) \approx 0.11t_i^2.$$

Thus, while regret is non-zero, it is a relatively small proportion of a player's type, even in the worst case.[18]

We can verify that the revenue of this mechanism, under the assumption of truthful bidding, is approximately 0.533, much higher than the optimal revenue of 1/3 in a perfectly truthful auction. Therein, we have a tradeoff: bidders now have some regret, but the designer can obtain considerably higher revenue.

[18] Indeed, computing expected regret with respect to the joint type distribution gives 0.006.

By applying the RW solver, we can compute the actual equilibrium of this auction as well, which turns out to be $s_i^*(t_i) \approx 0.62t_i$.[19] If players play this equilibrium, the expected revenue falls to 0.332, or very nearly the BNIC optimal revenue of 1/3. Since players shade their bids relative to true valuations, the Ex-Interim-IR constraint is still satisfied. Consequently, even if players take full advantage of deviation opportunities and end up in equilibrium, the designer obtains a payoff nearly as high as the best mechanisms known.

In claiming near incentive compatibility above, we do so by calibrating regret with respect to the magnitude of a player's type (valuation). While this is reasonable, an alternative way to calibrate regret is by comparing it to a player's payoff. In this metric, our mechanism does not do so well: a player can gain as much as 55% of payoff by deviating. Naturally, it is easy in our framework to redefine regret to consider gains relative to a player's payoff. We have implemented this alternative definition of regret, but our framework was unable to find a good mechanism in this case.[20] If we consider Lemma 1, this failure should be predictable: forcing both BNIC and Ex-Interim-IR to be satisfied restricts the space of feasible mechanisms greatly, making the optimization problem nearly impossible if done blindly. An alternative is to use the structural characteristics obtained in the lemma to enforce the desired constraints. Running the AMD framework while enforcing the constraints on parameters prescribed by the lemma, we obtain a mechanism with $q = 0.994$, $k_1 = 0.342$ and $k_2 = 0.303$ (with other parameters forced to 0 by the conditions of the lemma). Since BNIC and Ex-Interim-IR have been artificially satisfied, we need only check the expected revenue, which can be verified to be 0.329, or nearly the BNIC optimum of 1/3.

7 Conclusion

We presented a framework for automated mechanism design using the Bayes-Nash equilibrium solver for infinite games developed by Reeves and Wellman (2004). Results from applying this framework to several auction domains demonstrate the value of our approach for parametrized mechanism design. The mechanisms that we found were typically either close to the best known mechanisms, or better.

Whereas in principle it is not surprising that we can find mechanisms by searching the design space—as long as we have an equilibrium finding tool—it remains to establish that any such system would have practical merit. We presented evidence that mechanism design in a constrained space can indeed be effectively automated on somewhat realistic design problems that yield infinite games of incomplete information. Undoubtedly, real design problems can be vastly more complicated than any that we considered (or any that can be solved theoretically). In such cases, we believe that our approach could offer considerable benefit if used in conjunction with other techniques, either to provide a starting point for design, or to tune a mechanism produced via theoretical analysis and computational experiments.

---

[19] In particular, the expected utility of a player is maximized at this value for any $s_i(t_i) = mt_i$ strategy profile.

[20] There is actually some subtlety about how payoff calibration is implemented, since payoffs may be negative. We implemented several variations, all with essentially similar results.

Acknowledgments Much of this work was performed while the first author was at the University of Michigan. This research was supported in part by Grant CCF-0905139 from the U.S. National Science Foundation. Sandia is a multiprogram laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy under contract DE-AC04-94AL85000.

References

Ben-Tal A, Nemirovski A (2002) Robust optimization: Methodology and applications. Mathematical Programming 92:453–480

Brandt F, WeißG (2001) Antisocial agents and Vickrey auctions. In: Eighth International Workshop on Agent Theories, Architectures, and Languages, Springer, Seattle, Lecture Notes in Computer Science, vol 2333, pp 335–347

Brandt F, Sandholm T, Shoham Y (2007) Spiteful bidding in sealed-bid auctions. In: Twentieth International Joint Conference in Artificial Intelligence, pp 1207–1214

Byde A (2006) Applying evolutionary search to a parametric family of auction mechanisms. Australian Journal of Management 31(1):1–16

Conitzer V, Sandholm T (2002) Complexity of mechanism design. In: Eighteenth Conference on Uncertainty in Artificial Intelligence, pp 103–110

Conitzer V, Sandholm T (2003a) Applications of automated mechanism design. In: UAI-03 Bayesian Modeling Applications Workshop

Conitzer V, Sandholm T (2003b) Computational criticisms of the revelation principle. In: Workshop on Agent Mediated Electronic Commerce-V

Corana A, Marchesi M, Martini C, Ridella S (1987) Minimizing multimodal functions of continuous variables with simulated annealing algorithm. ACM Transactions on Mathematical Software 13(3):262–280

Cramton P, Gibbons R, Klemperer P (1987) Dissolving a partnership efficiently. Econometrica 55(3):615–632

Fleischer M (1995) Simulated Annealing: Past, present, and future. In: Winter Simulation Conference, pp 155–161

Gallien J (2006) Dynamic mechanism design for online commerce. Operations Research 54:291–310

Ganzfried S, Sandholm T (2010) Computing equilibria by incorporating qualitative models. In: Ninth International Conference on Autonomous Agents and Multi-Agent Systems, Toronto, pp 183–190

Guo M, Conitzer V (2009) Worst-case optimal redistribution of VCG payments in multi-unit auctions. Games and Economic Behavior 67(1):69–98

Guo M, Conitzer V (2010) Optimal-in-expectation redistribution mechanisms. Artificial Intelligence 174(5-6):363–381

McAfee RP (1992) Amicable divorce: Dissolving a partnership with simple mechanisms. Journal of Economic Theory 56(2):266–293

McKelvey RD, McLennan AM, Turocy TL (2005) Gambit: Software tools for game theory, version 0.2005.06.13. URL http://econweb.tamu.edu/gambit

McMillan J (1994) Selling spectrum rights. The Journal of Economic Perspectives 8(3):145–162

Morgan J, Steiglitz K, Reis G (2003) The spite motive and equilibrium behavior in auctions. Contributions to Economic Analysis and Policy 2(1)

Myerson RB (1981) Optimal auction design. Mathematics of Operations Research 6(1):58–73

Nisan N (2007) Introduction to mechanism design (for computer scientists). In: Nisan N, Roughgarden T, Tardos E, Vazirani VV (eds) Algorithmic Game Theory, Cambridge University Press, pp 209–241

Nocedal J, Wright S (2006) Numerical Optimization. Springer

Phelps S, Parsons S, McBurney P, Sklar E (2002) Co-evolutionary mechanism design: A preliminary report. In: Workshop on Agent-Mediated Electronic Commerce, pp 123–142

Phelps S, Parsons S, Sklar E, McBurney P (2003) Using genetic programming to optimise pricing rules for a double-auction market. In: Workshop on Agents for Electronic Commerce

Reeves DM (2005) Generating trading agent strategies: Analytic and empirical methods for infinite and large games. PhD thesis, University of Michigan

Reeves DM, Wellman MP (2004) Computing best-response strategies in infinite games of incomplete information. In: Twentieth Conference on Uncertainty in Artificial Intelligence, pp 470–478

Roth AE, Peranson E (1999) The redesign of the matching market for American physicians: Some engineering aspects of economic design. American Economic Review 89:748–780

Siarry P, Berthiau G, Durbin F, Haussy J (1997) Enhanced simulated annealing for globally minimizing functions of many continuous variables. ACM Transactions on Mathematical Software 23(2):209–228

Spall JC (2003) Introduction to Stochastic Search and Optimization. John Wiley and Sons

Vorobeychik Y (2009) Simulation-based game theoretic analysis of keyword auctions with low-dimensional bidding strategies. In: Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp 583–590

Vorobeychik Y, Wellman MP (2008) Stochastic search methods for Nash equilibrium approximation in simulation-based games. In: Seventh International Conference on Autonomous Agents and Multiagent Systems, pp 1055–1062

Vorobeychik Y, Kiekintveld C, Wellman MP (2006) Empirical mechanism design: Methods, with application to a supply-chain scenario. In: Seventh ACM Conference on Electronic Commerce, pp 306–315

Appendix

8 Proofs

8.1 Proof of Theorem 1

In the most favorable case, none of the n i.i.d. samples $X = \{X_1, \ldots, X_n\}$ from the type distribution violated the constraint. Thus, we take α to be the probability that the actual measure r of set B is above p given that best case:

$$\alpha = \Pr\{r \geq p \mid X \cap B = \emptyset\} = \frac{\Pr\{X \cap B = \emptyset \wedge r \geq p\}}{\Pr\{X \cap B = \emptyset\}}.$$

Since the samples are i.i.d.,

$$\Pr\{X \cap B = \emptyset \mid r\} = (1-r)^n,$$

and since we assumed a uniform prior on r, we get

$$\Pr\{X \cap B = \emptyset\} = \int_0^1 (1-r)^n dr = \frac{1}{n+1}$$

and

$$\Pr\{X \cap B = \emptyset \wedge r \geq p\} = \int_p^1 (1-r)^n dr = \frac{(1-p)^{n+1}}{n+1}.$$

Consequently, we obtain the following relationship between α, p, and n:

$$\alpha = (1-p)^{n+1}.$$

Solving for n, we get

$$n = \frac{\log \alpha}{\log(1-p)} - 1.$$

8.2 Proof of Theorem 2

Suppose p is the probability measure of T_A and suppose we select the best θ_i of $\{\theta_1, \ldots, \theta_L\}$. Suppose further that we take n samples for each θ_j, and let T^n be the set of n type realizations. We also use the notation $\theta \in G$ to indicate an event that for a particular θ, $\min_{t \in T^n} W(r,t,\theta) > \inf_{t \in T \setminus T_A} W(r,t,\theta)$.

We would like to compute the number of samples n for each of these samples such that $\Pr\{\theta_i \notin G\} \geq 1-\alpha$.

Note that

$$\Pr\{\theta_i \notin G\} \geq \Pr\{\theta_1 \notin G \wedge \cdots \wedge \theta_L \notin G\} = \Pr\{\theta_j \notin G\}^L.$$

Now,

$$\Pr\{\theta_j \in G\} = \Pr\{t_1 \notin T_A \wedge \cdots \wedge t_n \notin T_A\} = \Pr\{t_i \notin T_A\}^n = (1-p)^n.$$

Thus,

$$\Pr\{\theta_i \notin G\} \geq (1-(1-p)^n)^L = 1-\alpha.$$

Solving for n, we obtain the desired answer.

8.3 Proof of Theorem 3

We show that for the two-player game with types $U[A, B]$ and payoff function

$$u(t, a, t', a') = \begin{cases} t - ha - ka' & \text{if } a > a' \\ \frac{t-ha-ka'+ha'+ka}{2} & \text{if } a = a' \\ ha' + ka & \text{if } a < a', \end{cases}$$

with $h, k \geq 0$ and $B \geq A + 1$ that the following is a symmetric Bayes-Nash equilibrium strategy:

$$\frac{t}{3(h+k)} + \frac{hA + kB}{6(h+k)^2}. \tag{7}$$

Consider first the special case that $h = k = 0$. Equation (7) prescribes a strategy of bidding ∞ and it is clear that this is a dominant strategy in a game where the winner is the high bidder with no payments required.[21] We now assume that $h + k > 0$.

Define $m \equiv \frac{1}{3(h+k)}$ and $c \equiv \frac{hA+kB}{6(h+k)^2}$ and let T be a random $U[A, B]$ variable giving the opponent's type. Noting that the tie-breaking case ($a = a'$) happens with zero probability given that (7) is a continuous function of a uniform random variable, we write the expected utility for an agent of type t playing action a as

$$\begin{aligned} \mathrm{EU}(t, a) &= E_T[u(t, a, T, mT + c)] \\ &= E[t - ha - k(mT + c) \mid a > mT + c] \Pr(a > mT + c)] \\ &\quad + E[h(mT + c) + ka \mid a < mT + c] \Pr(a < mT + c) \\ &= E\left[t - ha - kmT - kc \;\middle|\; T < \frac{a-c}{m}\right] \Pr\left(T < \frac{a-c}{m}\right) \\ &\quad + E\left[hmT + hc + ka \;\middle|\; T > \frac{a-c}{m}\right] \Pr\left(T > \frac{a-c}{m}\right) \end{aligned} \tag{8}$$

We consider three cases on the range of a and find the optimal action a_i^* for each case i.

Case 1: $a \leq Am + c$. $\quad (\implies \frac{a-c}{m} \leq A)$

The probabilities in (8) are zero and one, respectively, and so the expected utility is:

$$\mathrm{EU}(t, a) = hm\frac{A+B}{2} + hc + ka.$$

This is an increasing function in a, implying an optimal action at the right boundary: $a_1^* = Am + c$. Thus the best expected utility for case 1 is

$$\mathrm{EU}(t, a_1^*) = \frac{2A + B}{6}.$$

Case 2: $a \geq Bm + c$. $\quad (\implies \frac{a-c}{m} \leq B)$

The probabilities in (8) are one and zero, respectively, and so the expected utility is:

$$\mathrm{EU}(t, a) = t - ha - km\frac{A+B}{2} - kc.$$

[21] This assumes that the space of possible bids includes ∞. More generally, the dominant strategy is the supremum of the bid space but if this is not itself a member of the bid space (as is the case if the bid space is $\mathbb{R}$) then there is in fact no Nash equilibrium of the game.

This is a decreasing function in a, implying an optimal action at the left boundary: $a_2^* = Bm + c$. Thus the best expected utility for case 2 is

$$\mathrm{EU}(t, a_2^*) = t - \frac{A + 2B}{6}.$$

Case 3: $Am + c < a < Bm + c$.

Knowing that $\frac{a-c}{m}$ is between A and B it is straightforward to compute the probabilities in (8) and the conditional expectation of T. So we write $\mathrm{EU}(t, a)$ as:

$$\begin{aligned}
&\left(t - ha - km\frac{A + \frac{a-c}{m}}{2} - kc\right)\left(\frac{a-c}{m} - A\right) \\
&+ \left(hm\frac{B + \frac{a-c}{m}}{2} + hc + ka\right)\left(B - \frac{a-c}{m}\right) \\
=&(-108a^2h^4 - 432a^2kh^3 - 648a^2k^2h^2 - 432a^2k^3h - \\
&- 108a^2k^4 + 36aAh^3 + 72ath^3 + A^2h^2 + 4B^2h^2 + \\
&+ 4ABh^2 + 72aAkh^2 + 36aBkh^2 - 36Ath^2 + 216akth^2 + \\
&+ 36aAk^2h + 72aBk^2h + 8A^2kh + 8B^2kh + 2ABkh + \\
&+ 216ak^2th - 60Akth - 12Bkth + 36aBk^3 + 4A^2k^2 + B^2k^2 \\
&+ 4ABk^2 + 72ak^3t - 24Ak^2t - 12Bk^2t)/(24(h+k)^2).
\end{aligned}$$

Since this is a concave function of a the maximum is where the derivative with respect to a is zero, that is (skipping the tedious algebra for which we used Mathematica):

$$\begin{aligned}
&\frac{\partial\,\mathrm{EU}(t,a)}{\partial a} = 0 \\
\Longrightarrow a_3^* = &\frac{t}{3(h+k)} + \frac{hA + kB}{6(h+k)^2}.
\end{aligned}$$

Since $A \leq t \leq B \implies Am + c \leq a_3^* \leq Bm + c$, a_3^* is in fact in the allowable range for case 3. The expected utility for case 3 is then

$$\mathrm{EU}(t, a_3^*) = \frac{3t^2 + A^2 + B^2 + A(B - 6t)}{6}.$$

It now remains to show that neither $\mathrm{EU}(t, a_1^*)$ nor $\mathrm{EU}(t, a_2^*)$ is greater than $\mathrm{EU}(t, a_3^*)$ for any t.

Since $t \geq A$ there exists a $\delta \geq 0$ such that $t = A + \delta$. And since $B \geq A + 1$ there exists an $\varepsilon \geq 0$ such that $B = A + 1 + \varepsilon$. First, $\mathrm{EU}(t, a_3^*) \geq \mathrm{EU}(t, a_2^*)$ because

$$
\begin{aligned}
& (\delta - 1)^2 \geq 0 \\
\Longrightarrow\ & \delta^2 - 2\delta + 1 \geq 0 \\
\Longrightarrow\ & \delta^2 + 1 \geq 2\delta \\
\Longrightarrow\ & (A + \delta - A)^2 + 2A + 1 \geq 2A + 2\delta \\
\Longrightarrow\ & (t - A)^2 + 2A + 1 \geq 2t \\
\Longrightarrow\ & t^2 + A^2 + 2A + 1 \geq 2At + 2t \\
\Longrightarrow\ & 3t^2 + 3A^2 + 6A + 3 + (3A\varepsilon + \varepsilon^2 + 4\varepsilon) \geq 6At + 6t \\
\Longrightarrow\ & 3t^2 + A^2 + (A^2 + 2A + 2A\varepsilon + \varepsilon^2 + 2\varepsilon + 1) + \\
& \quad + (A^2 + A + A\varepsilon) - 6At \geq 6t - A - 2A - 2 - 2\varepsilon \\
\Longrightarrow\ & 3t^2 + A^2 + (A + 1 + \varepsilon)^2 + A(A + 1 + \varepsilon) - 6At \\
& \quad \geq 6t - A - 2(A + 1 + \varepsilon) \\
\Longrightarrow\ & 3t^2 + A^2 + B^2 + AB - 6At \geq 6t - A - 2B.
\end{aligned}
$$

Finally, $\mathrm{EU}(t, a_3^*) \geq \mathrm{EU}(t, a_1^*)$ because

$$
\begin{aligned}
& (t - A)^2 \geq 0 \\
\Longrightarrow\ & t^2 - 2At + A^2 \geq 0 \\
\Longrightarrow\ & t^2 + A^2 \geq 2At \\
\Longrightarrow\ & 3t^2 + 3A^2 \geq 6At \\
\Longrightarrow\ & 3t^2 + 3A^2 + (3A\varepsilon + \varepsilon^2 + \varepsilon) \geq 6At \\
\Longrightarrow\ & 3t^2 + 3A^2 + 3A + 3A\varepsilon + \varepsilon^2 + \varepsilon - 6At \geq 3A \\
\Longrightarrow\ & 3t^2 + (A^2 + A + \varepsilon) - 6At + \\
& \quad + (A^2 + 2A + 2A\varepsilon + \varepsilon^2 + 2\varepsilon + 1) + A^2 \geq 3A + \varepsilon + 1 \\
\Longrightarrow\ & 3t^2 + A(A + 1 + \varepsilon) - 6At + A^2 + (A + 1 + \varepsilon)^2 \\
& \quad \geq 2A + (A + \varepsilon + 1) \\
\Longrightarrow\ & 3t^2 + AB - 6At + A^2 + B^2 \geq 2A + B.
\end{aligned}
$$

8.4 Proof of Theorem 4

It is direct from Theorem 3 that setting $h = 1/3$ and $k = 0$ yields a symmetric Bayes-Nash equilibrium $s(t) = t$ when $A = 0$. We now show that the best response to truthful bidding is only truthful under this parameter setting—i.e., that SGA$(1/3, 0)$ is the only BNIC game in the SGA family, for $U[0, B]$ types.

Suppose that the opponent bids truthfully (i.e., $s(t) = t$ for one of the agents). First, assume that $a \in [0, B]$. The expected utility of an agent with type t from bidding a is then

$$\begin{aligned}\mathrm{EU}(t,a) &= \int_0^a (t - ha - kT)dT + \int_a^1 (hT + ka)dT = \\ &= \frac{1}{2}\left(-3(h+k)a^2 + 2(Bk+t)a + B^2h\right).\end{aligned}$$

Since this function is strictly concave in a, we can use the first-order condition to find the optimum bid:

$$\frac{\partial\,\mathrm{EU}(t,a)}{\partial a} = t - 3(h+k)a + Bk = 0$$

yielding

$$a = \frac{t + Bk}{3(h+k)}, \tag{9}$$

which is truthful for every type t only when $h = 1/3$ and $k = 0$.

Now, if $a \leq 0$, it will always lose, and the expected utility is

$$\mathrm{EU}(t,a) = \int_0^B (hT + ka)dT = B^2h/2 + kBa,$$

which is maximized when $a = 0$. Consequently, there is no incentive to ever bid below 0. Similarly, if $a \geq B$, the agent will never lose, and

$$\mathrm{EU}(t,a) = \int_0^B (t - ha - kT)dT = -\frac{1}{2}B(2ah + Bk - 2t),$$

which is maximized when $a = B$. Thus, there is no incentive to ever bid above B. All incentive compatible mechanisms will thus induce bidding according to (9). It follows, then, that SGA$(1/3, 0)$ is the only truthful mechanism for $U[0,B]$ $(B > 0)$ types. The extension to $A > 0$ is straightforward.

8.5 Proof of Theorem 5

Define t_w and t_l to be the "winner's" (one who ultimately gets the item) and "loser's" types respectively. The objective function in terms of h and k is

$$\begin{aligned}\min_{h,k} |E[t_w - 2h(\frac{t_w}{3(h+k)} + \frac{k}{6(h+k)^2}) - 2k(\frac{t_l}{3(h+k)} + \\ \frac{k}{6(h+k)^2}) \mid t_w > t_l]|.\end{aligned}$$

Since $E[t_w \mid t_w > t_l]$ is the expectation of the first order statistic of two $U[0,1]$ random variables, it is $2/3$ (and $1/3$ for t_l). Thus, the objective function above reduces to

$$\min_{h,k} \left| \frac{2h + k}{9(h+k)} \right|.$$

We now show that this expression cannot be less than 1/9:

$$
\begin{aligned}
&h \geq 0 \\
\Longrightarrow\ &2h \geq h \\
\Longrightarrow\ &2h + k \geq h + k \\
\Longrightarrow\ &\frac{2h+k}{h+k} \geq 1 \\
\Longrightarrow\ &\frac{2h+k}{9(h+k)} \geq \frac{1}{9}.
\end{aligned}
$$

Since setting $h = 0$ yields the minimum of 1/9 for any $k > 0$ we conclude that all mechanisms SGA$(0, k)$ minimize the objective function.

8.6 Proof of Theorem 6

First, we obtain the expression to be minimized.

$$
\begin{aligned}
&\sup_{t>t'} |t - 2h(\frac{t}{3(h+k)} + \frac{k}{6(h+k)^2}) - 2k(\frac{t'}{3(h+k)} + \frac{k}{6(h+k)^2})| \\
&= \sup_{t>t'} |t - \frac{2ht + 2kt'}{3(h+k)} - \frac{k}{3(h+k)}| \\
&= \sup_{t>t'} |\frac{ht + 3kt - 2kt' - k}{3(h+k)}|.
\end{aligned}
$$

Clearly, this is minimized when $t = 1$ and $t' = 0$, yielding

$$\frac{h+3k-k}{3(h+k)} = \frac{h+2k}{3(h+k)}.$$

Now, note that since $h, k \geq 0$,

$$\frac{h+2k}{3(h+k)} \geq \frac{h+k}{3(h+k)} = \frac{1}{3}.$$

Thus, the expression cannot be less than 1/3. Consequently, since setting $k = 0$ for any $h > 0$ results in the objective function value of 1/3, it describes a subset of optimal values.

8.7 Proof of Lemma 1

First, let us derive $Q(q, t)$ and $U(q, x, t)$, where q is the probability that player with the higher type wins the good and $x(t)$ is the expected payment by players (Myerson, 1981).

$$Q(q, t) = \int_0^t q dT + \int_t^1 (1 - q) dT = t(2q - 1) - q + 1.$$

$$\begin{aligned}U(q,x,t) = &\int_0^t (tq - k_1 t - k_2 T - K_1)dT + \\ &+ \int_t^1 ((1-q)t - k_3 t - k_4 T - K_2)dT = \\ &= (2q - k_1 - 0.5k_2 + k_3 + 0.5k_4 - 1)t^2 + \\ &+ (1 - q - K_1 - k_3 + K_2)t - (0.5k_4 + K_2).\end{aligned}$$

The first constraint that must be satisfied according to Myerson (1981) is if $s \leq t$ then $Q(q,s) \leq Q(q,t)$. This constraint is always satisfied in our design space by inspection of the form of $Q(q,t)$ above.

Individual rationality constraint requires that $U(q,x,0) \geq 0$, implying in our setting that $0.5k_4 + K_2 \leq 0$. Since all design parameters are constrained to be nonnegative, this implies that $k_4 = K_2 = 0$, and, consequently, $U(q,x,0) = 0$.

The version of the final constraint in Myerson (1981) in our setting

$$U(q,x,t) = \int_0^1 Q(q,s)ds = (q - 0.5)t^2 + (1-q)t$$

implies that $K_1 = k_3 = 0$ and $q - k_1 - 0.5k_2 - 0.5 = 0$, completing the proof.

8.8 Proof of Theorem 7

The expected revenue to the designer is

$$U_0(q,x) = \int_0^1 \int_0^1 (x_1(t,T) + x_2(t,T))dtdT$$

which by symmetry and Lemma 1 is equivalent to

$$U_0(q,x) = 2\int_0^1 \int_0^t (k_1 t + k_2 T)dTdt = \frac{2}{3}k_1 + \frac{1}{3}k_2.$$

Rewriting the constraint from Lemma 1 to be $k_1 + 0.5k_2 = q - 0.5$, it is clear that the revenue is maximal when $q = 1$. Now, if we let $k = k_1$ and $k_2 = 1 - 2k$, the expected revenue becomes $(2/3)k + (1/3)(1-2k) = 1/3$. Thus, we can set any $k_1 \in [0, 0.5]$ and $k_2 \in [0,1]$, respecting the constraint, to achieve optimal revenue of 1/3.

8.9 Proof of Theorem 8

We use the equilibrium bids of $s(t) = 0.72t - 0.73$ in this proof. First, let us derive the expected payment of an agent with type t, which we designate by $m(t)$. We simplify our task by taking advantage of strict monotonicity of the equilibrium bid function in t.

$$\begin{aligned}m(t) &= \int_0^t (0.95s(t) + 0.84s(T) + 0.78)dT + \\ &+ \int_t^1 (0.73s(t) + 0.53)dT = \\ &= 0.95t(0.72t - 0.73) + 0.84(0.36t^2 - 0.73t) + \\ &+ 0.78t + 0.73(0.72t - 0.73)(1 - t) + 0.53(1 - t) = \\ &= 0.4604t^2 + 0.0018t - 0.0029.\end{aligned}$$

By symmetry, the expected revenue is twice the expectation of $m(t)$:

$$R = 2\int_0^1 m(t)dt = 2\int_0^1 (0.4604t^2 + 0.0018t - 0.0029)dt > 0.3.$$

To confirm individual rationality, we need to compute the expected value to an agent with type t from this auction, which we label $v(t)$:

$$v(t) = \int_0^t 0.96tdT + \int_t^1 0.04tdT = 0.92t^2 + 0.04t.$$

The expected utility to an agent with type t is its expected value less expected payment:

$$\mathrm{EU}(t) = v(t) - m(t) = 0.4596t^2 + 0.0382t + 0.0029.$$

Clearly, this is always positive. Furthermore, the designer can charge each agent an additional participation fee of 0.0029 and maintain individual rationality. Since this uniform fee will not affect agents' incentives, the designer will gain an additional 0.0058 in revenue without compromising the individual rationality constraint.

8.10 Proof of Theorem 9

The intuition for the proof is straightforward. Suppose that the equilibrium bid function is strictly increasing and $q = 1$. Then, since the high bidder always gets the good, and the higher type is always the high bidder, the good always goes to the agent that values it more. Consequently, this design yields optimal welfare. The reverse argument works in the other case.

Formally, expected welfare is

$$pE_{t,T}[t \mid t > T] + (1 - p)E_{t,T}[t \mid t < T] + 0.5E_{t,T}[t \mid t = T],$$

where p is the probability that the high type gets the good. Since the probability that types of both agents are equal is 0, the third term is 0. Furthermore, $E_{t,T}[t \mid t > T] = 2/3$, since this is just the first order statistic of the type distribution, and $)E_{t,T}[t \mid t < T] = 1/3$ since it is the second order statistic of the type distribution. Consequently, expected welfare is $(2/3)p + (1/3)(1 - p)$. This is maximized when $p = 1$, and the maximal value is 2/3. Now, if bid function is increasing in t, then $q = p = 1$ ensures optimality. If bid function is decreasing in t, on the other hand, $q = (1 - p) = 0$ ensures optimality.

8.11 Proof of Theorem 10

We work with the symmetric equilibrium bid of $s(t) = 0.645t - 0.44$. Since we have already shown the optimality of this mechanism, we just need to confirm individual rationality and compute the revenue from this auction.

As before, we start with computing the payment of an agent with type t:

$$\begin{aligned}
m(t) = &\int_0^t (0.88s(t) + 0.23s(T) + 0.28)dT + \\
&+ \int_t^1 (0.06s(t) + 0.32s(T))dT = \\
&= 0.88t(0.645t - 0.44) + 0.23(0.3225t^2 - 0.44t) + \\
&+ 0.28t + 0.06(0.645t - 0.44)(1 - t) + \\
&+ 0.32(-0.3225t^2 + 0.44t - 0.1175) = \\
&= 0.499875t^2 - 0.0025t - 0.064.
\end{aligned}$$

By symmetry, the expected revenue is twice the expectation of $m(t)$:

$$R = 2\int_0^1 (0.499875t^2 - 0.0025t - 0.064)dt = 0.20275.$$

The expected value of an agent, $v(t)$ is just t^2, since the high type always gets the good. Consequently, expected utility to an agent is

$$\mathrm{EU}(t) = v(t) - m(t) = 0.50012t^2 + 0.0025t + 0.064.$$

Since this is always nonnegative when $t \in [0, 1]$, ex interim individual rationality constraint holds. Note also that it will hold weakly if we charge each participant 0.064 for entering the auction. Thus, the designer could gain an additional 0.128 in revenue without affecting incentives, welfare optimality, and individual rationality.

8.12 Proof of Theorem 11

Since we are assuming symmetry and the equilibrium bid function is increasing in t, the objective is equivalent to

$$\begin{aligned}
&\inf_{t>T}[k_1 s(t) + k_2 s(T) + k_3 s(T) + k_4 s(t)] = \\
&\quad \inf_{t>T}[k_1 mt + k_2 mT + k_3 mT + k_4 mt] = \\
&\quad m \inf_{t>T}[(k_1 + k_4)t + (k_2 + k_3)T] = 0.
\end{aligned}$$

8.13 Proof of Theorem 12

We use the symmetric equilibrium bid of (approximately) $s(t) = 0.43t - 0.51$.

First we establish the worst-case revenue properties of the design. By symmetry, the worst-case objective is equivalent to

$$\inf_{t>T}(s(t) + 0.34s(T) + 0.69) = \inf_{t>T}(0.43t + 0.1462T + 0.0066) = 0.0066.$$

The expected utility of type t is

$$\int_0^t (t - s(t) - 0.34s(T) - 0.69)dT = 0.4969t^2 - 0.0066t,$$

which attains a minimum at $t = 0.0066412$, with the minimum value of just above -0.000022.

8.14 Proof of Theorem 13

We use the symmetric equilibrium bid of $s(t) = (7/9)t+2/9$. The expected payment of type t is

$$m(t) = \int_0^t (\frac{7}{9}T + \frac{2}{9})dT = \frac{7}{18}t^2 + \frac{2}{9}t.$$

The expected revenue is then

$$R = 2\int_0^1 (\frac{7}{18}t^2 + \frac{2}{9}t)dt = \frac{13}{27}$$

which is approximately 0.480.

Since the high bidder always gets the good, $v(t) = t^2$. The expected utility of an agent with type t is then

$$eu = \frac{11}{18}t^2 - \frac{2}{9}t,$$

which attains its minimum when $t = 2/11$, with the minimum value of $-44/2178$ (just under –0.02). Thus, it is not individually rational. To fix the mechanism, the designer could afford each agent 0.021 for participation, reducing his revenue to 0.438.

8.15 Proof of Theorem 14

We use the symmetric equilibrium bid of $s(t) = 1.613t - 0.234$. First, we compute expected payment of type t:

$$\begin{aligned} m(t) &= \int_0^t (0.98s(T) + 0.09)dT + \int_t^1 0.33s(t)dT \\ &= 0.98(0.8065t^2 - 0.234t) + 0.09t + \\ &\quad + 0.33(1.613t - 0.234)(1 - t) \\ &= 0.25808t^2 + 0.47019t - 0.07722. \end{aligned}$$

The expected revenue is then

$$R = 2\int_0^1 (0.25808t^2 + 0.47019t - 0.07722)dt = 0.4878.$$

Since the high bidder always gets the good, $v(t) = t^2$, and the expected utility of type t is then

$$\mathrm{EU}(t) = 0.74192t^2 - 0.47019t + 0.07722.$$

The function $\mathrm{EU}(t)$ is always positive, and the minimum gain for any agent type is 0.00273. Thus, the designer could charge an entry fee of 0.0027 and gain an additional 0.0054 in revenue, for a total of 0.4932.

8.16 Proof of Theorem 15

In this case, we use the symmetric equilibrium bid of $s(t) = 0.595t - 0.2$. The expected payment of type t is

$$\begin{aligned} m(t) = &\int_0^t (s(t) + 0.33s(T) + 0.22)dT + \\ &+ \int_t^t (0.22s(t) + 0.12s(T))dT \\ = &\, 0.595t^2 - 0.2t + 0.33(0.2975t^2 - 0.2t) + 0.22t + \\ &+ 0.22(0.595t - 0.2)(1 - t) + \\ &+ 0.12(-0.2975t^2 + 0.2t + 0.0975) \\ = &\, 0.526575t^2 + 0.1529t - 0.0323. \end{aligned}$$

The expected revenue is then

$$R = 2\int_0^1 (0.526575t^2 + 0.1529t - 0.0323) \approx 0.44.$$

Since $q = 1$, $v(t) = t^2$, and, therefore

$$\mathrm{EU}(t) = 0.473425t^2 - 0.1529t + 0.0323,$$

which we can verify is always positive. Thus, this design is ex interim individually rational. Since its minimum value is slightly above 0.0199, we can bill this amount to each agent for participating in the auction without affecting incentives or ex interim individual rationality. This adjustment will give the designer 0.0398 of additional revenue, for a total of about 0.4798.

8.17 Proof of Theorem 16

We use the symmetric equilibrium bid function $s(t) = -0.22t - 0.175$ here.

Since the bids are strictly decreasing in types, the expected value of type t is

$$v(t) = \int_0^t 0.63t\, dT + \int_t^1 0.37t\, dT = 0.26t^2 + 0.37t.$$

By symmetry, the expected welfare is then

$$W = 2\int_0^1 v(t)\, dt = 0.543.$$

The expected payment of type t is

$$\begin{aligned}
m(t) = &\int_0^t (0.29s(t) + 0.67s(T) + 0.48)dT + \\
&+ \int_t^1 (0.8s(t) + s(T) + 0.49)dT \\
= &-0.29(0.22t + 0.175)t - 0.67(0.11t^2 + 0.175t) + \\
&+ 0.48t + 0.8(-0.22t - 0.175)(1-t) - 0.11t^2 + \\
&+ 0.175t - 0.285 + 0.49(1-t) \\
= &\ 0.1485t^2 - 0.004t + 0.065.
\end{aligned}$$

Thus, we can compute the expected revenue:

$$R = \int_0^1 (0.1485t^2 - 0.004t + 0.065)dt = 0.225.$$

The expected utility of type t is

$$\mathrm{EU}(t) = v(t) - m(t) = 0.1115t^2 + 0.374t - 0.065,$$

which attains its minimum at the lower type boundary of 0, with the minimum value of –0.065, and is negative over the range of types $[0, 0.17]$. Thus, the designer could make the mechanism completely ex interim IR at a loss of an additional 0.013 in revenue by offering each agent a participation gift of 0.065. With this gift, the revenue would fall to 0.095.

8.18 Proof of Theorem 17

We use the symmetric equilibrium bid function $s(t) = 0.935t - 0.18$ here.

The expected payment of an agent with type t is

$$\begin{aligned}
m(t) = &\int_0^t (0.51s(t) + s(T) + 0.09)dT + \\
&+ \int_t^1 (0.34s(t) + 0.26s(T))dT \\
&= 0.51(0.935t^2 - 0.18t) + 0.4675t^2 - 0.18t + 0.09t + \\
&\quad + 0.34(0.935t - 0.18)(1 - t) + \\
&\quad + 0.26(-0.4675t^2 + 0.18t + 0.2875) \\
&= 0.5049t^2 + 0.2441t + 0.01355.
\end{aligned}$$

The expected revenue is thus

$$R = 2\int_0^1 (0.5049t^2 + 0.2441t + 0.01355)dt = 0.6078.$$

The expected utility of an agent with type t is

$$\mathrm{EU}(t) = v(t) - m(t) = 0.4951t^2 - 0.2441t - 0.01355,$$

which is negative for a fairly broad range of types (although always above the tolerance level that we set). Type $t^* = 0.24652$ fairs the worst, incurring a loss of nearly 0.044. However, by compensating both agents this amount, we ensure ex interim individual rationality without affecting incentives. As a result, the designer will lose 0.088 in expected revenue, which will fall to 0.5198.

8.19 Proof of Theorem 18

By symmetry, the objective value is equivalent to

$$\inf_{t>T} s(T) = \inf_{t>T} (\frac{7}{9}T + \frac{2}{9}) = 2/9.$$

The rest follows by Theorem 13.

8.20 Proof of Theorem 19

The objective value is equivalent to

$$\begin{aligned}
&\inf_{t>T} (s(t) + 0.71s(T) + 0.14 + 0.09s(t)) = \\
&\quad = \inf_{t>T} (0.3t - 0.045 + 0.71(0.3T - 0.045) + 0.14 + 0.09(0.3t - 0.045)) = 0.059.
\end{aligned}$$

The expected utility of an agent is

$$\begin{aligned}
eu(t) = &\int_0^t (0.86t - 0.3t + 0.045 - 0.71(0.3T - 0.045) - 0.14)dT + \\
&+ \int_t^1 (0.14t - 0.09(0.3T - 0.045))dT = \\
= &\ 0.56t^2 + 0.07695t - 0.1065t^2 - 0.14t + 0.14t - 0.14t^2 + 0.00405 - \\
&- 0.00405t - 0.0135(1 - t^2) = \\
= &\ 0.327t^2 + 0.0729t - 0.00945.
\end{aligned}$$

which attains a minimum value of −0.00945. Thus, the participation award of 0.00945 to each agent is necessary to make this design individually rational, with the resulting worst-case revenue of 0.04.