Designing Core-selecting Payment Rules: A Computational Search Approach

Benjamin Lubin
Boston University

Benedikt Bünz
Stanford University

Sven Seuken
University of Zurich

First Version: February 13, 2017
This Version: February 15, 2018

Abstract

We study the design of core-selecting payment rules for combinatorial auctions (CAs), a challenging setting where no strategyproof rules exist. Unfortunately, under the rule most commonly used in practice, the *Quadratic* rule Day and Cramton (2012), the equilibrium strategies are far from truthful. In this paper, we present a computational approach for finding good core-selecting payment rules. We present a parameterized payment rule we call FRACTIONAL* that takes three parameters (reference point, weights, and amplification) as inputs. This way, we construct and analyze 610 rules across 30 different domains. To evaluate each rule in each domain, we employ a computational Bayes-Nash equilibrium solver. We first use our approach to study the well-known Local-Local-Global domain in detail, and identify a set of 20 "all-rounder rules" which beat Quadratic by a significant margin on efficiency, incentives, and revenue in all, or almost all domains. To demonstrate robustness of our findings, we take four of these all-rounder rules and evaluate them in the significantly larger LLLLGG domain (with six bidders and eight goods), where we show that all four rules also beat Quadratic. Overall, our results demonstrate the power of a computational search approach in a mechanism design space, and more specifically the large improvements that are possible over Quadratic.

1. Introduction

The spectrum auctions conducted by governments around the world over the last twenty years are a true success story for market design in general and auction design in particular. Sophisticated mechanisms have been used to sell resources worth billions of dollars, forming the basis for today's wireless industry (Cramton, 2013). Recent versions of these markets have used a combinatorial auction (CA) mechanism. The advantage of CAs (e.g., in contrast to running multiple single-item auctions) is that buyers can express complex valuation functions over bundles of goods, which avoids the exposure problem, and can increase efficiency.

Over the last 15 years, there has been a large literature on the design of bidding languages, clearing algorithms, and activity rules for use in combinatorial auctions (CAs) (Cramton, Shoham and Steinberg, 2006). However, finding optimal *payment rules* has turned out to be somewhat elusive. In this work, we focus on the payments that are charged after the auction closes and the winners have been determined (i.e., we treat the CA as a one-shot game). This allows us to focus on *direct* payment rules which take as input the bidders' value reports and compute payments for each of the winning bidders. A real-world application is the *Combinatorial Clock Auction (CCA)*, whose *supplementary round* is a sealed-bid package auction Ausubel and Baranov (2017).

Early proposals for CAs typically considered charging VCG prices (Varian and MacKie-Mason, 1994). At first sight, the VCG mechanism may seem like an appealing payment rule for a CA because it is strategyproof. Unfortunately, VCG is generally viewed as unsuitable in a CA domain where items can be complements because it often produces outcomes outside of the *core* Ausubel and Milgrom (2006). Informally, this means that payments may be so low that a coalition of bidders may be willing to pay more in total than what the seller receives from the current winners. From a revenue perspective, VCG is also often undesirable, because CAs can produce very low or even zero revenue, despite high competition for the goods in the auction. For these reasons, recent auction designs have employed *core*-selecting payment rules that are guaranteed to charge payments in the (revealed) *core* (Ausubel and Milgrom, 2002; Milgrom, 2007; Day and Milgrom, 2008).

1.1. The Quadratic Rule

Unfortunately, there exists no strategyproof core-selecting payment rule Goeree and Lien (2016), and thus, designing an "optimal" core-selecting rule is a challenging market design problem. Parkes, Kalagnanam and Eso (2001) were the first to introduce the idea of finding prices that minimize some distance metric to VCG, first for combinatorial exchanges, and later for CAs (Parkes, 2002). Since then, a few rules have been proposed that minimize some distance metric to VCG (e.g., (Day and Raghavan, 2007)). However, in each case, the weights or distance metrics were chosen "manually," typically without substantial justification.

Ultimately, Day and Cramton (2012) proposed the *Quadratic* rule, and this is also the rule most often used in practice today. The Quadratic rule selects prices that are (1) enforced to be in the *core* (with respect to the submitted bids), and within this (2) minimal in their total revenue to the seller, and then within this (3) minimal in the Euclidean distance to VCG prices.

The Quadratic rule has been used for more than 10 years by many governments around the world to allocate resources worth more than $20 billion Ausubel and Baranov (2017). Nevertheless, we still have an incomplete understanding of this rule. Only recently has the research community started to graple with the incentive properties of the Quadratic rule. For the so-called *Local-Local-Global (LLG)* setting, with 2 items and 3 bidders, Ausubel and Baranov (2013) as well as Goeree and Lien (2016) have independently derived the Bayes-Nash equilibrium of the Quadratic rule. It turns out that, even though the rule minimizes the Euclidean distance to VCG, the equilibrium strategies are far from truthful. This motivates the search for better payment rules in this paper.

1.2. Prior Work: A Manual Search for Better Payment Rules

The research community has already considered a number of alternative proposals for payment rules. Erdil and Klemperer (2010) argued that non-VCG reference points that are independent of the bidders' reports offer better incentives at the margin of truthful play. However, they do not offer a concrete payment rule, and they do not offer an argument about what happens when further deviations are necessary. Along the same lines, Day and Cramton (2012) studied the Quadratic rule with a ZERO reference point, instead of VCG. Using computational experiments (assuming truthful bidding), they found that, on average, the use of ZERO may tilt the payoff distribution in favor of those winning bidders with higher values. However, they did not study this payment rule in equilibrium, nor did they analyze the effect on efficiency, revenue or overall incentives.

Ausubel and Baranov (2013) provided an analytical study of three core-selecting payment rules (Quadratic rule, Proxy rule, and Nearest-Bid rule), varying the distributional assumptions and the degree of risk-aversion. They found that core-selecting payment rules perform better in terms of efficiency and revenue when bidders' values are more correlated, while VCG performs worse. However, they did not identify a new payment rule with superior properties to Quadratic.

Parkes, Kalagnanam and Eso (2001) already proposed three *weighted payment rules*, in particular, the use of the bidders' VCG payoff to influence which core point would be selected. Ausubel and Baranov (2017) reported that weighted versions of the Quadratic rule have been used in the most recent CCAs conducted in Australia and Canada. However, they did not use VCG-payoff, but rather well-chosen reserve prices, to power their weighted version of the Quadratic rule. The (intuitive) reason in favor of using reserve-price-weights is to make sure that small players in the auction are not disadvantaged compared to larger players. However, no theoretical analysis of the properties of this reserve-price-weighted version of Quadratic exists.

We would like to highlight one common theme among this prior work on core-selecting payment rules: it always required some experienced auction designer to come up with a new payment rule, or an improvement to an existing payment rule. Sometimes, we have a few theoretical results that support their arguments (in favor or against) the use of a particular rule. But in many cases, new rules were proposed using arguments that made intuitive sense, but without a comprehensive analysis. In this paper, we propose a radically different approach. Instead of relying on our own ingenuity of coming up with an even better payment rule, and having to analyze its properties in equilibrium by hand, we developed a *computational search approach* to automate this process.

1.3. A Computational Search for Better Payment Rules

The basic idea of our computational search approach is relatively simple: we construct a formal framework to describe and parameterize the space of core-selecting payment rules, and we then use an algorithm to systematically search through this space and identify the best-performing rules. The details, however, are quite intricate.

First, the design space for core-selecting payment rules is infinitely-large. To make it amenable to a computational search, we must thus choose a well-suited framework and parametrization. To this end, we introduce a *parameterized payment rule* we call FRACTIONAL*, that is parameterized via three parameters: a *reference point* R, a *weight* W, and an *amplification* A. FRACTIONAL* minimizes the W-weighted Euclidean distance to the reference point R, whereby the weights can be amplified (i.e., exponentiated) by the amplification A. This provides us with a rich framework that can capture all (minimum-revenue) core-selecting payment rules (if suitable parameters are chosen). For example, FRACTIONAL*(R=p_{VCG},W=EQUAL,A=1) is a complicated way to describe the Quadratic rule. FRACTIONAL*(R = ZERO,W = π_{VCG},A = 3) is a way to describe a VCG-payoff-weighted version of Quadratic with the ZERO reference point, but where the weights are also exponentiated by 3. In this paper, we consider 9 different reference points, 11 different weights, and 6 different amplifications, giving rise to a total of 610 payment rules that we have studied.

The second challenge relates to comparing the 610 rules in terms of their efficiency, incentives, and revenue. Because no core-selecting payment rule can be strategyproof, we must evaluate all of our rules in *Bayes-Nash equilibrium (BNE)*. Of course, nobody wants to derive 610 BNEs by hand (which typically involves solving a differential equation), which is why we use a recently developed *computational BNE solver* by Bosshard et al. (2017) to power our computational search approach. Concretely, this is an algorithm that takes as input a payment rule and produces as output an ε-BNE. Because the algorithm only produces a numerical result, and not a theoretical result, we only obtain an ε-BNE with $\varepsilon > 0$, instead of a true, theoretical BNE. However, the precision of the algorithm is extremely high (e.g., for the LLG domain, we report ε-BNEs with $\varepsilon < 0.1\%$). For the BNE solver, a key question is the representation of the bidders' value space and its action/strategy space. For this work, we use an approach that lets us model the bidders' full, continuous value space (i.e., no discretization is necessary). In terms of the action/strategy space, we employ a piece-wise linear approximation of the bidder's optimal strategy, using 80 control points, which allows for an extremely good approximation of even very non-linear functions.[1] The final result of our computational search is still "one (or multiple) optimal core-selecting payment rule(s)" that can be described very easily via three parameters, as a mathematical formula, or as a simple math program.[2] Thus, our resulting optimal rules could easily be adopted in real-world CAs.

1.4. Towards Robust All-Rounder Rules

A key question that arises when evaluating core-selecting payment rules (analytically or computationally) is what *domain* to choose for the evaluation. In this paper, we analyze two domains. In

[1] This is in contrast to some prior work that has employed computational BNE solvers, but with a restricted strategy space, for example using a simple multiplicative or additive shading strategy Lubin and Parkes (2009); Lubin, Bünz and Seuken (2015).

[2] This is in contrast to the *automated mechanism design* approach Sandholm (2003), where a mechanism is automatically created (by an algorithm) for each specific problem instance (at "run-time", so to say). Our approach is also different from *computational mechanism design/algorithmic mechanism design* Nisan and Ronen (2001), where the mechanism is still typically designed by hand, but where an additional goal is that the overall mechanism/algorithm is computationally tractable.

a first step, we perform an extensive analysis in the stylized but well-know LLG domain, with two goods and three bidders. In a second step, to check robust of our results and evaluate whether our findings generalize, we go to the larger LLLLGG domain, with eight goods and six bidders.

LLG models a CA with two goods A and B, and three bidders, where the two *local* bidders are only interested in A or B respectively, and the *global* bidder is interested in the bundle $\{A, B\}$. However, instead of just studying standard LLG (with uniform distributions and no correlation), we have created 29 different variations of the LLG domain (varying, e.g., the marginal distribution as well as the correlation between bidders). Thus, with 610 rules evaluated on 29 domains, we have studied 17690 rule-domain combinations. This allows us to identify a set of 20 very good *all-rounder* rules, i.e., payment rules that perform well across all 29 domains.

While these all-rounder rules perform well under different distributions and correlations, all settings we study in the first step are structurally the same, i.e., the LLG domain. This naturally raises the question how robust these rules are to structural changes in the domain, i.e., how well our findings generalize to larger domain sizes. To this end, we consider the larger LLLLGG domain. While LLG is still small enough such that systematically searching through 17690 rule-domain combinations with a fully expressive strategy space is computationally feasible, even evaluating a single rule in LLLLGG takes more than one day on a compute cluster. For this reason, we only evaluate a small subset of our rules in LLLLGG. Our main findings are that those rules which are among the best in the LLG also beat Quadratic by a large margin in LLLLGG. Furthermore, those rules which are among the worst in LLG are also worse than Quadratic in LLLLGG. This suggests that our results are robust, not only to changes in the distribution and correlation, but also to the structure and size of the domain.

1.5. Towards a Proposal for a new Rule

Our overarching goal in this paper is to provide a thorough evaluation of different core-selecting payment rules to inform the decision regarding which rules perform best. Eventually, we hope to make a proposal regarding the implementation of a new, better rule in practice. While we may not be there yet, we consider this paper an important step in this direction.

One interesting finding that falls out of our analysis of the best 20 all-rounder rules is that using the *Shapley* value turns out to be extremely attractive in the design of our rules. Many of our best-performing rules use the Shapley value, either as a reference point or as weights. This dovetails nicely with a recent result by LukeLindsay (2017), who also showed the benefits of using the Shapley value for the design of pricing rules for auctions and exchanges, albeit not for core-selecting payment rules.

A second interesting pattern that emerges from our analysis is that, in general, the rules that perform best tend to “favor” bidders with larger values (i.e., generate lower core prices for them). Parkes, Kalagnanam and Eso (2001) called these kinds of rules *Large* rules (in contrast to *Small* rules). It turns out that, for efficiency and revenue, it is most important to provide the bidders with the large values with good incentives to bid truthful. This is in stark contrast with the findings of Lubin and Parkes (2009), who found that in combinatorial exchanges (CEs), *Small* rules tended to perform better. This illustrates that CAs and CEs may be more different than one might think at first sight.

Interestingly, there are different ways to achieve the effect of a Large rule, and our search for good all-rounders reveals the different combinations of reference points, weights, and amplifications that lead to this effect. One of our rules is very similar to Quadratic, as it also uses VCG payments as a reference point, but it uses the inverse bids as weights, which has the effect that the higher the bid, the lower the core payment will be, which turns this rule into a Large rule. A similar effect can be achieved by using a "zero" reference point, which strongly favors bidders with large values. Our results show that using Quadratic with a Zero reference point would actually overshoot the necessary effect. Instead, some of our best rules that use the Zero reference point dampen the strong effect which the reference point introduces by also using a small weight that shifts the power a little bit towards the smaller players.

This last observation actually points towards a more general learning from this overall exercise which we cannot emphasize enough: the performance of a core-selecting payment rule is determined by the *combination of its three design features, i.e., reference point, weights, and amplification.* Prior work has focused on individual design dimensions (e.g., changing the reference point, or adding weights). However, the main finding of our work is that these "local" changes to the payment rule are most likely misguided. Instead, what really matters in the design of optimal core-selecting payment rules seems to be the *perfect combination* of its features. Therefore, the result of our research is *not* one best reference point, or one best set of weights. Instead, our analysis leads to a list of best-performing rules, where for each rule, the reference point, the weights, and the amplification are perfectly tuned to complement each other in the best possible way.

Obviously, we could not have obtained these insights regarding the perfect combination of the rule parameters using a manual design approach. Thus, our paper shows the power of a computational search approach in a mechanism design space. Furthermore, our results demonstrate the large improvements that are possible over Quadratic in terms of efficiency, incentives and revenue. Finally, we hope that this work will eventually help auction designers identify an attractive alternative to Quadratic to be used in practice as well.

2. Preliminaries

In a *combinatorial auction (CA)*, there is a set M of m distinct, indivisible items, and a set N of n bidders. Each bidder i has a *valuation function* v_i which, for every bundle of items $S \subseteq M$, defines bidder i's value $v_i(S) \in \mathbb{R}$, i.e., the maximum amount that bidder i would be willing to pay for S. To simplify notation, we assume that the seller has zero value for all items, although our setup extends to the case where the seller has non-zero value (see Day and Cramton (2012) for how to handle reserve prices).

We let $p = (p_1, \ldots, p_n)$ denote the payment vector, with p_i denoting bidder i's payment. We assume that bidders have quasi-linear utility functions, i.e., $u_i(S, p_i) = v_i(S) - p_i$. Bidders make reports about their values to the mechanism, denoted $\hat{v}_i(S)$, which may be non-truthful (i.e.,

$\hat{v}_i \neq v_i$). Following existing work in this area (Goeree and Lien, 2016; Ausubel and Baranov, 2013), we assume that bidders only bid on items for which they have a positive value. We define an *allocation* $X = (X_1, \ldots, X_n) \subseteq M^n$ as a vector of bundles, with $X_i \subseteq M$ being the bundle that i gets allocated. A mechanism's *allocation rule* maps the bidders' reports to an allocation. We only consider allocation rules that maximize reported *social welfare*, yielding an allocation $X^* = \arg\max_X \sum_{i \in N} \hat{v}_i(X_i)$, subject to X being feasible, i.e., $\bigcap X_i^* = \emptyset$. In addition to the allocation rule, a mechanism also specifies a *payment rule*, defining prices as a function of the bids. Together, these define the *outcome* $O = \langle X, p \rangle$. An outcome O is called *individually rational (IR)* if, $\forall i$: $u_i(X_i, p_i) \geq 0$.

2.1. VCG, Payoff, and VCG Payoff

The famed VCG mechanism (Vickrey, 1961; Clarke, 1971; Groves, 1973) generalizes the well-known second-price auction. Its allocation rule chooses the allocation X^* that maximizes the bidders' reported social welfare. Its payment rule is defined as follows:

Definition 1 (VCG Payment Rule) *Given an allocation X^*, and bidders' value reports $\hat{v}$, VCG charges each bidder i its marginal cost to the economy, i.e., the externality he imposes on all other bidders. Formally:*

$$p_{VCG,i} = \sum_{j \neq i} \hat{v}_j(X^{-i}) - \sum_{j \neq i} \hat{v}_j(X^*), \tag{1}$$

where X^{-i} is the welfare-maximizing allocation when all bidders except i are present.

The VCG mechanism is social-welfare maximizing and *strategyproof*, i.e., it is a dominant strategy for every bidder to report his true value v_i. Every mechanism other than VCG we consider in this paper will not be strategyproof, and thus we will in general have to differentiate between a bidder's true value v_i and his reported value $\hat{v}_i$.

Definition 2 (Payoff) *Given a payment p_i charged by a particular payment rule, a bidder's* payoff *is the bidder's profit evaluated at his true value, i.e.,*

$$\pi_i = v_i(X_i^*) - p_i. \tag{2}$$

The *payoff* achieved under a particular rule is what a bidder actually cares about and tries to maximize in equilibrium.

Note that at the reports of a non-strategyproof mechanism, we can still compute VCG payments, which are the payments the bidders *would have paid* if the VCG payment rule *had been used*, but applied to the value reports submitted to the current payment rule. We will use those *VCG payments* in the definition of some of our more sophisticated *core*-selecting payment rules. Analogously, we can also define a bidder's *VCG payoff*, which is the payoff the bidder *would have gotten*, given his reported value, if the VCG payment rule *had instead been* used in place of the actual payment rule.

Definition 3 (VCG Payoff) *A bidder's reported* VCG Payoff *is the difference between his reported value and his VCG payment:*

$$\pi_{VCG,i} = \hat{v}(X_i^*) - p_{VCG,i}. \tag{3}$$

2.2. Bayes-Nash Equilibrium

For the analysis of auction payment rules we assume that the bidders know their own value function, but do not have full information about the other bidders' value functions. Instead, bidders only have distributional information regarding the other bidders' value functions. Thus, the appropriate equilibrium concept is the *Bayes-Nash Equilibrium*. For the following definition, we let s_i denote bidder i's strategy, which is a mapping from his true value function v_i to a possibly non-truthful report $\hat{v}_i$. Given a value function and a strategy from each bidder, this determines the outcome of the auction. Thus, we can let $u_i(s_1(v_1), s_2(v_2), \ldots, s_n(v_n))$ denote bidder i's utility for the outcome of the auction. We use v_{-i} to denote the value functions of all bidders except i, and analogously for the strategies s_{-i}:

Definition 4 (Bayes-Nash Equilibrium) *A strategy profile $s^* = (s_1^*, \ldots, s_n^*)$ is a* Bayes-Nash equilibrium *(BNE) in a sealed-bid auction if, for all bidders i, and all values functions v_i*

$$\mathbb{E}_{v_{-i}}\left[u_i(s_i^*(v_i), s_{-i}^*(v_{-i}))\right] \geq \mathbb{E}_{v_{-i}}\left[u_i(\hat{v}_i, s_{-i}^*(v_{-i}))\right], \quad \textit{for all possible reports } \hat{v}_i, \tag{4}$$

where the expectation is taken with respect to the distribution over the other bidders' value functions.

In words, a bidder's BNE strategy is an optimal strategy (a mapping of true value functions to reported value functions) given his belief regarding the other bidders' value functions. With this background, we next describe the key desiderata for payment rules in the *core*-selecting CA setting.

3. Computational Search Approach

Next, we define the the space of rules we investigate using our computational approach. These rules are all *core*-selecting, so we begin by introducing this concept.

3.1. The *Core*

Informally, a payment rule is outside the *core* if a coalition of bidders is willing to pay more than what the seller receives in the mechism. Payment rules that avoid such outcomes are said to be *core*-selecting (Day and Milgrom, 2008). Formally we have:

Definition 5 (Core) *We let W denote the set of winners, X^* the welfare-maximizing allocation, $C \subseteq N$ denotes a coalition of bidders, and X^C is the allocation that would be chosen by the mechanism* *if only the bidders in the coalition C were be present. Then, a price vector p is in the* core*, if, in addition to individual rationality, the following set of* core *constraints hold:*

$$\sum_{i \in W \setminus C} p_i \geq \sum_{i \in C} v_i(X^C) - \sum_{i \in C} v_i(X^*) \quad \forall C \subseteq N \tag{5}$$

Enforcing prices to be in the *core* puts lower bounds (constraints) on the payments of the winners, where each coalition of bidders leads to one *core* constraint.[3] Intuitively, the winners' payments must be *sufficiently large*, such that there exists no coalition that is willing to pay more to the seller than the current winners' payments. In a CA with complements, VCG prices are often outside the core and, in the worst case, VCG may generate zero revenue despite high competition for the goods. Figure 1 illustrates the *core* and several reference points in the setting with two goods and three players considered in this paper (see Section 5.1). Bidder 1 bids 80 for good A, bidder 2 bids 90 for good B, and bidder 3 bids 100 for the bundle $\{A, B\}$.

3.2. Design Framework for New Core-Selecting Payment Rules

Given a bid vector, a *core*-selecting payment rule always selects a price vector in the *core* (the shaded area in Figure 1). Day and Raghavan (2007) proposed to only select prices from the so-called *minimum-revenue*-core *(MRC)*, which is indicated as the diagonal line in Figure 1.[4] The motivation for using this constraint is that this minimizes the total amount of deviation potential for all bidders. The QUADRATIC rule most commonly used in practice also employs the MRC constraint, and for this reason we also only consider payment rules which select prices from the minimum revenue *core*. Still, this leaves an infinite number of price vectors to choose from, and thus lots of room for the design of new payment rules, which will all have different properties in equilibrium.

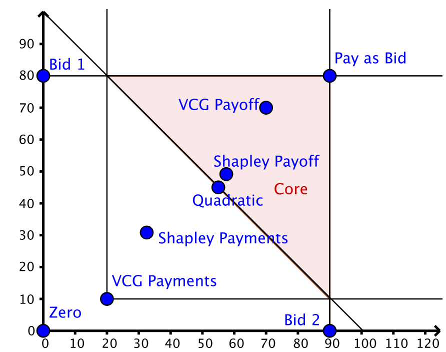


Figure 1: A graphical depiction of the *core*, as well as the price vectors corresponding to PAYASBID, VCG, QUADRATIC and the reference points π_{VCG}, p_{VCG}, p_{SHAPLEY},π_{SHAPLEY}, and ZERO.

[3]Note that the core is defined in terms of bidders' true values. However, given that no strategyproof core-selecting CA exists, we must expect bidders to be non-truthful. Goeree and Lien (2016) have recently shown that the outcome of a core-selecting CA can be outside the true core in a BNE. Thus, core-selecting CAs only guarantee to produce outcomes in the *revealed* core, i.e., in the core with respect to *reported values*. Going forward, whenever we talk about the *core*, or *core*-selecting rules, we always mean the *revealed* core, unless we state it otherwise. Note that for the auctioneer (e.g., the government), having prices in the *revealed* core is typically want she wants, because this protects the auctioneer against law-suits from losing bidders and from the appearance that an unacceptably large amount of revenue was left on the table.

[4]Day and Raghavan (2007) have shown the somewhat surprising result that this MRC constraint can end up being binding, even for the QUADRATIC rule, which minimizes the Euclidean distance to VCG.

As depicted in Figure 1, the QUADRATIC rule selects a price vector in the MRC which minimizes the Euclidean distance to p_{VCG} (computed at the reported values of the bidders). Here, p_{VCG} serves as the *reference point* of the payment rule; we note that p_{VCG} is not a fixed point but rather defined as a function of the bids, so formally reference points are functions, not single points. Of course p_{VCG} is only one possible reference point and bidders can, via their value reports, manipulate the reference point; this is true even for p_{VCG}. This motivates the use of alternative reference points, such as ZERO $= \vec{0}$, which as a constant obviously cannot be manipulated. We also consider weighted rules where a *weighting function* computes a weight per bidder based on the reported values. For QUADRATIC the weights are simply EQUAL $= \vec{1}$

To define our space of rules, we generalize the Euclidean distance minimization in the QUADRATIC rule, which is defined algorithmically as follows:

Definition 6 (Algorithmic Framework for *Core*-Selecting Payment Rules) *Given a* reference point, weights, *and* amplification, *the unique* core*-selecting price vector is chosen to be:*

1. *Within the* core.
2. *Within this, in the* minimal revenue *core (MRC).*
3. *Within this, minimal in the weighted and amplified distance metric to the reference point.*

In step (3) we generalize the Euclidean distance used by QUADRATIC to the following:

$$f_{w,a}(p, r^*) = \sqrt{\sum_{i=1}^{n} \frac{1}{w_i^a} |p_i - r_i^*|^2} \tag{6}$$

where p is the price vector being chosen, r^* is a *reference point*, w is a *weight* vector, and a is an *amplification* factor.

Proposition 1 *The payment rule framework described in Definition 6 is general and encompasses all possible MRC-selecting rules.*

Proof 1 *Let P^{MRC} be an arbitrary MRC selecting rule that outputs core payments p^{MRC}. Let $w = \vec{1}$ and $r^* = p^{MRC}$. Clearly $p = p^{MRC}$ minimizes $f_{\vec{1},1}(p, p^{MRC})$ and p^{MRC} is in MRC by definition.*

Working in a combinatorial exchange setting, Parkes, Kalagnanam and Eso (2001) introduced a FRACTIONAL rule (sometimes called the PROPORTIONAL rule) with this proportional charging property restricted to $r^* = p_{\text{VCG}}$ and $w = \pi_{\text{VCG}}$. Because we use a generalized form of this distance function, we call our payment rules FRACTIONAL*, and use a wide array of reference points and weights, described in the next section. Additionally, we also introduce the amplification factor. When $a = 0$, the weights are ignored and the QUADRATIC rule is recovered. As $a \to \infty$, the weights will dominate over the quadratic term, and payments will be pushed as far as possible in the direction of the weights consistent with the *core* and MRC constraints.

Within this framework, we also consider taking the *inverse* of the weights, which effectively reverses the prioritization they construct. And lastly, because we include in our examinations reference points that may be within/above the core, we consider *mirroring* those points within/above the core across the nearest MRC facet so as to put the points outside and below the core. This has the effect of ensuring that the direction (and thus the sign) of the distance to the MRC line is always consistent, which it otherwise would not be.

3.3. Overview of all Core-Selecting Payment Rules

Consistent with the above framework, in this paper, we structure the design space of MRC-*core*-selecting payment rules to include the cross product of the following parameters, resulting in 610 rules. This set includes all existing rules in this space we are aware of and many more:

1. **Reference Points:** Zero, Bid, p_{VCG}, π_{VCG}, $\pi^{\text{M}}_{\text{VCG}}$, p_{Shapley}, $p^{\text{M}}_{\text{Shapley}}$, π_{Shapley}, $\pi^{\text{M}}_{\text{Shapley}}$
2. **Weights:** Equal, Bid, Bid^{-1}, π_{VCG}, π^{-1}_{VCG}, p_{VCG}, p^{-1}_{VCG}, π_{Shapley}, $\pi^{-1}_{\text{Shapley}}$, p_{Shapley}, p^{-1}_{Shapley}
3. **Amplification:** We include $\{0.5, 1, 2, 3, 5, 10\}$

Here we use p for payments and π for payoffs, the M superscript denotes mirroring of reference points, and the -1 superscript denotes inversion of weights. π_{Shapley} denotes *Shapley values* which are a payoff structure for cooperative games with desirable properties (Shapley, 1953). The Shapley value is the average additional contribution that each agent brings to a coalition of agents. It is important that the seller is considered an agent as no coalition can produce welfare without him. p_{Shapley} are payments that lead to a payoff of π_{Shapley}, i.e. $p_{\text{Shapley}} := \hat{v} - \pi_{\text{Shapley}}$. For the design of our rules we use π_{Shapley} and p_{Shapley}. Like the core, the Shapley value is a concept from cooperative game theory. Unlike the core, though, the Shapley value, when applied to the coalition game including all bidders, can allocate surplus to losing bidders. However, because we restrict payments to the core, no surplus is allocated to losing bidders.[5] We will see that π_{Shapley} as well as p_{Shapley} will turn out to be very useful, as reference points and as weights.

3.4. Searching for Optimal Rules via a Computational BNE Solver

Given our framework for the design of core-selecting payment rules, the question arises how to find the *optimal* one. Note that for every *candidate rule* to be evaluated (in terms of efficiency, incentives and revenue) we must find the BNE of this rule. In this paper, instead of doing this analytically, we use a BNE search algorithm that was recently introduced by Bosshard et al. (2017). The algorithm represents the bidders' strategy space as a piecewise linear approximation with 80 control points. Its main idea is based on the basic approach of iterated best response dynamics via the fictitious play algorithm Brown (1951). The algorithm proceeds in rounds. In each round, each

[5]We also experimented with an alternative approach where the Shapley value is only applied to a coalition game among the winning bids, as in LukeLindsay (2017). However, the particular variant of the Shapley value did not seem to make a big different which is why we only report one set of results for simplicity.

bidder's strategy is moved towards its best response to the current strategies of the other players. This process repeats until a certain convergence criterion is reached. We provide a pseudo-code description of the high-level algorithm in Appendix H. For details, please see Bosshard et al. (2017).

We emphasize that the strategies we study here are not simple additive or multiplicative "shades," but rather full bidding functions. Because fictitious play algorithms are not guaranteed to converge in pure strategies (which we require), it is possible that we may only find an ϵ-BNE for a reasonably large ϵ. In our results for LLG, we only report rules for which the algorithm has achieved a 0.1%-BNE on all of our domains, i.e., a bidder can increase his utility by at most 0.1% (additively) by deviating from the BNE.[6] In our results for LLLLGG, we only report rules for which the algorithm has achieved a 2.5%-BNE.[7]

Equipped with the BNE algorithm, the next question is how to search through the (truly infinitely large) design space for new core-selecting payment rules. At first sight, some type of gradient decent in the parameter space may seem like a good approach. However, our results clearly show that the rule space is highly non-convex over the different parameters of the rule. Furthermore, we will show later that the outcome of the rules is highly sensitive to the exact combination of rule parameters. Perhaps even more importantly, some rule parameters are functions (e.g., Shapley), making numerical optimization extremely challenging. Even if such optimization were possible, it would not be generalizable to other contexts. Finally, the computational costs involved in computing each individual BNE (minutes to hours in LLG, days in LLLLGG) make gradient-based approaches infeasible, even for the continuous components of the parameters.

The approach we have chosen is effectively an exhaustive search over our design framework in the discretized parameter space. We compute the BNEs of every rule defined in the previous section for every domain we consider. We then compare these rules according to multiple design goals, which we detail in Section 4, to find the optimal rule. In addition to being computationally feasible, one distinct advantage of this approach is that all rules we evaluate are easily interpretable (i.e., the reference points, weights and amplification factors are general and human-readable). Thus, the rules can be lifted from the LLG and LLLLGG domain, to be directly applied in any CA, regardless of size or structure.

4. Design Dimensions

We strive for the following three objectives when designing core-selecting payment rules, in this order: (1) high efficiency, (2) good incentives, and (3) high revenue. In this section, we introduce

[6]In theory, it is possible that multiple BNEs exist. Practically, when we have run the solver multiple times, we have never found two significantly different ε-BNEs for the same rule. Consequently, in our search, we deterministically seek a single BNE (the one closest to truth).

[7] While such a filter is necessary to ensure we only examine rules for which a complete picture is known, it does mean that there may be rules from within our search space that we do not evaluate here because of the difficulty of finding a BNE for them. This yields the following interpretation to our results presented below: the rules we find are indeed good, but we can not preclude the possibility that there are even better rules, even in our parameter space, that we are forced to exclude.

our formal measures for these three dimensions. As there does not exist a strategyproof core-selecting payment rule Goeree and Lien (2016), all three dimensions are evaluated *in BNE*.

4.1. High Efficiency

From a social planner's perspective (e.g., a government auctioning off spectrum) it is desirable to maximize the *social welfare* of the mechanism (i.e., the sum of the winners' values for their allocations). The *efficiency* of a mechanism is defined as the fraction of the social welfare that the mechanism achieves. As we evaluate the efficiency of our rules in BNE, we consider the *expected efficiency* in BNE. Thus, our measure for efficiency is the expected welfare of the mechanism divided by the expected welfare of the optimal allocation. Formally, given an auction instance I, let $SW_{OPT}(I)$ denote the social welfare obtained under the optimal allocation given bidders' true values. Let $SW_M(I)$ denote the social welfare obtained by the mechanism M when all bidders play their BNE strategies. We define the efficiency of mechanism M as:

$$\text{Efficiency(M)} = \frac{\mathbb{E}_{\sim I}[SW_M(I)]}{\mathbb{E}_{\sim I}[SW_{OPT}(I)]} \tag{7}$$

where the expectation is taken over the auction instances in the domain being analyzed. This is the standard definition of efficiency used in prior work, e.g., by Goeree and Lien (2016).

4.2. Good Incentives

We next desire that our payment rules produce "good incentives." In the past, auction designers have frequently argued in favor of the Quadratic rule because of its property to "induce truthful bidding" Cramton (2013), or to "minimize the bidder's ability to benefit from strategic manipulation" Day and Raghavan (2007), even though Quadratic is not strategyproof . One argument is that, if the rule is "approximately strategyproof", then finding a beneficial deviation from truthful bidding may be so hard that many bidders may just report truthfully Day and Milgrom (2008). Of course, because there is no strategyproof core-selecting CA, there will always remain some strategic opportunities for the participants; however, we would like these opportunities in BNE to be as small as possible.

Of course, the manipulability of a payment rule depends on a bidder's value; in particular, some rules may be more manipulable for bidders with small values while others may be more manipulable for bidders with large values. To capture the incentive properties of a rule in one number, we define an *aggregate incentives measure* (which we just call "incentives" going forward) as the average Euclidean distance between bidders' truthful value v and their bid $\hat{v}$ in BNE. Formally, we have:

$$\text{Incentives(M)} = \sqrt{\int_v f(v) \cdot (v - \hat{v})^2 \, dv} \tag{8}$$

where $f(v)$ is the probability density function (PDF) in the domain we are analyzing, and the bid $\hat{v}$ is the optimal bid in BNE for that domain. Note that this is simply a PDF-weighted L^2 norm.

4.3. High revenue

Another motivation for using core-selecting payment rules is to achieve high revenue, in particular higher revenue than VCG Day and Milgrom (2008); Day and Raghavan (2007). Because the revenue achieved by a rule is very domain dependent, we measure the *fraction of the VCG revenue* which a rule achieves in a particular domain, defined analogously to efficiency (except that, in contrast to efficiency, a rule can achieve more than 100% of the revenue achieved by VCG). Thus, our measure for the *revenue* achieved by a mechanisms M is defined as:

$$\text{Revenue(M)} = \frac{\mathbb{E}_{\sim I}[Revenue_M(I)]}{\mathbb{E}_{\sim I}[Revenue_{VCG}(I)]} \tag{9}$$

where the expectation is taken over the auction instances in the domain being analyzed.

5. Results for LLG

In this section, we study the Local-Local-Global (LLG) domain (described below), and several novel variants. We first focus on LLG for several reasons: First, the existing theoretical results for simple rules provide a benchmark for our experiments. Second, solving for the BNE gets exponentially harder as the domain gets more complex. LLG is simple enough that we can solve for the *full* BNE strategies for a large number of rules. That said, as we shall show, both the design space and the resulting BNE structure is surprisingly subtle and intricate. Thus, it is important to understand the nature of the BNEs in a small domain before we move on to a larger domain.

5.1. LLG Uniform

In LLG, there are two items A and B, and three bidders. Two of the bidders are local bidders, each only interested in either item A or B respectively. The third bidder is the global bidder who wants both items simultaneously. It remains to define the distribution from which the bidders' values are drawn. Existing work has focused on the case we refer to as Uniform where each player is drawn independently with local bidders' values $\sim U[0,1]$ and the global bidder $\sim U[0,2]$.

BNEs of *core*-selecting payment rules are complex to study analytically, and consequently existing theoretical results are only available for LLG. Prior work has shown that the BNE strategies of the local bidders require an *additive* shading in this setting:[8]

[8]See Ausubel and Baranov (2013) for a similar result where Zero is used for the reference point.

Rule	Efficiency	Incentives	Revenue	
QUADRATIC	98.03%	16.19	91.30%	
Improvement Over QUADRATIC				**Avg.**
Best Efficiency FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=2)	0.31%	3.90%	1.62%	1.94%
Best Incentives FRACTIONAL*(R=π_{SHAPLEY},W=BID^{-1},A=10)	0.27%	5.13%	1.27%	2.22%
Best Revenue FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=2)	0.31%	3.90%	1.62%	1.94%

Table 1: The results for DOMAIN(MD = UNIFORM). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Proposition 2 (Goeree and Lien (2016); Ausubel and Baranov (2013)) *In UNIFORM, a Bayes-Nash equilibrium of the QUADRATIC rule is for the global bidder to be truthful, and for the local bidders to bid:*

$$\hat{v} = max(0, v - (3 - 2\sqrt{2})) \approx max(0, v - 0.17) \tag{10}$$

The results for our computational investigation of this domain are provided in Table 1. The way to read this (and all following tables) is as follows. The QUADRATIC results are always provided in the first row (and here they correspond to the BNE in Proposition 2). The next three rows of the table then show the top rules by each dimension (efficiency, incentives, revenue). The cell entries for these rules now represent the *(multiplicative) improvement* of the respective metric/column relative to QUADRATIC. In this case, FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=2) is best by both efficiency and revenue, providing evidence that the Shapley value can be useful in constructing these non-coperative mechanisms. The rule is able to increase the efficiency (relative to QUADRATIC) by 0.31% and revenue by 1.62%. In this domain, the best rule by incentives is the FRACTIONAL*(R=π_{SHAPLEY}, W=BID^{-1},A=10) rule, where the reference point is also Shapley, but where inverse bids are used as weights, with an extreme amplification. This combination of inverse bid-weights and a very high amplification produces a *Large*-style rule.

5.2. LLG with Correlation

We expand upon the basic LLG structure, by introducing correlation in the player values: instead of considering each player as drawn independently, we now draw the set of player values from a joint distribution. We use *Copulae* to define these joint distributions, a method which lets us separate the specification of the *marginal* distributions that each bidder obtains when viewing its distribution in isolation, from the *coupling* structure which describes the joint structure among these marginals. Formally, Sklar's Theorem (Sklar, 1959) states that *all* multivariate comulative distribution functions (CDFs) $F(x_1, \ldots, x_d) = \mathbb{P}(X_1 \leq x_1, \ldots, X_d \leq x_d)$ can be represented as $F(x_1, \ldots, x_d) = C(M_1(x_1), \ldots, M_d(x_d))$ where the M_i are the marginal CDFs in each dimension, e.g. $M_i(x) = \mathbb{P}(X_i \leq x)$, and C is a *copula* which is a joint CDF with uniform marginals. The theorem also provides that C will be unique if the F_i are continuous. The converse of the theorem

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	98.41%	10.91	95.19%	
		Improvement Over QUADRATIC			**Avg.**
Best Efficiency	FRACTIONAL*(R=ZERO,W=π_{VCG},A=2)	0.24%	6.77%	2.65%	3.22%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=BID,A=0.5)	0.20%	8.50%	2.61%	3.77%
Best Revenue	FRACTIONAL*(R=ZERO,W=$\pi^{-1}_{\text{SHAPLEY}}$)	0.04%	-0.61%	4.00%	1.14%

Table 2: The results for DOMAIN(MD = UNIFORM,Corr = SAME). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

lets us easily create multi-dimensional models by combining a set of marginal distributions M_i with a copula C to create a joint distribution $C(M_1(x_1), \cdot, M_d(x_d))$. In this section we consider several choices for C, in 5.3 we consider choices for M_i, and finally in 5.4 we consider the cross product of these choices.

To model correlation, we adopt standard Gaussian copulae, which use a multivariate normal distribution for the coupling function. We consider two correlation structures: (a) SAME which establishes correlation between both bidders interested in the *same* item and (b) CROSS which establishes correlation between the local players.[9] In both cases the correlation constant is 0.5.

Results for SAME side correlation are provided in Table 2. The third row of the table illustrates that a rule that is very good by one dimension, may be worse on others. We will seek to address this in Section 5.5 by finding good all-rounders. Moreover, each dimension yields a distinct best rule in this domain. We also investigated cross-side correlation and varying the intensity of the correlation, results of which we include in Appendix B.

5.3. LLG with BETA Marginals (Uncorrelated)

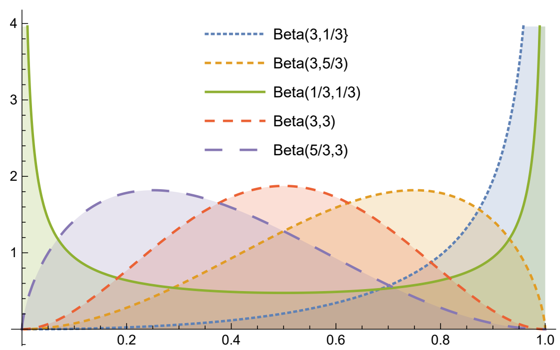


Figure 2: The various BETA marginal distributions we consider.

[9]Ausubel and Baranov (2013) consider a form of correlation where the local players are either exactly the same, or otherwise drawn independently. This approach is amenable to their form of theoretical analysis but is much less natural than approach taken here.

As is common practice, we employ a BETA distribution for our marginals as it will adopt a diverse set of shapes that are similar to many familiar distributions with just two parameters.[10] We consider the application of several parameterizations of BETA distributions to the local players as illustrated in Figure 2. We note that once one employs a skewed distribution, the relative bidder strength between the local and global players may no longer match that of the UNIFORM case. To address this, we linearly *calibrated* the distributions (unless explicitly mentioned) so as to ensure that the expected relative strength of the local and global players is identical to the UNIFORM case. We also experimented with uncalibrated domains, which we include in Appendices E and F.

	Rule	Efficiency	Incentives	Revenue
	QUADRATIC	97.79%	29.49	88.24%
	Improvement Over QUADRATIC			**Avg.**
Best Efficiency	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=2)	1.20%	24.80%	13.90%	13.30%
Best Incentives	FRACTIONAL*(R=p^{M}_{SHAPLEY},W=p^{-1}_{SHAPLEY},A=3)	1.10%	25.70%	12.30%	13.03%
Best Revenue	FRACTIONAL*(R=BID,W=p_{SHAPLEY},A=2)	1.00%	7.00%	15.80%	7.93%

Table 3: The results for DOMAIN(MD=BETA(3,1/3)). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Table 3 provides results for DOMAIN(MD=BETA(3,1/3)), which is an example of the types of results we see in domains with skewed marginals. We again observe that a different rule is optimal for each dimension. However, all three rules perform quite well in all dimensions, relative to QUADRATIC. Results for the other distributions are provided in Appendix C.

5.4. Maximum Improvements Relative to Quadratic

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	95.00%	29.73	143.59%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%
Best Incentives	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%
Best Revenue	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%

Table 4: The results for DOMAIN(MD=BETA(3,1/3),Corr=CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

As discussed in Section 5.2, modeling the joint distribution of value among the players using a copulae lets us mix-and-match between various marginal distributions and various types of

[10] Ausubel and Baranov (2013) consider a distribution very similar to BETA(a,0), but they are limited to Pareto-like shapes.

correlation among these distributions. We have investigated the full cross product of correlations we discussed in Section 5.2 with the set of marginal distributions discussed in Section 5.3. The full set of results is presented in Appendix D.

Here, we briefly point towards those rules that achieve the *largest improvement* over Quadratic, across all domains. In terms of efficiency, Fractional*(R = Bid,W = π_{Shapley}, A = 2.0) achieves a **4.00%** improvement over Quadratic in Domain(MD = Beta(3,1/3),Corr = Cross,Uncalibrated) (see Table 4). Considering the fact that many large-scale CAs allocate resources worth billions of dollars, an efficiency improvement of this magnitude is very significant. Note that the same rule, in the same domain, also achieves an incentive improvement of **47.2%**, demonstrating that core-selecting rules exist whose equilibrium strategies are significantly closer to truthful than Quadratic. This performance is even exceeded by Fractional*(R = Zero,W = Equal) in the domain Domain(MD=Uniform,Corr=CrossLarge), where this rule achieves an incentive improvement of **48.25%** over Quadratic. In terms of revenue, Fractional*(R=Bid$^{\text{M}}$,W=p_{Shapley},A=2) achieves a **23.4%** improvement over Quadratic in Domain(MD=Beta(3,1/3),Corr=Cross).

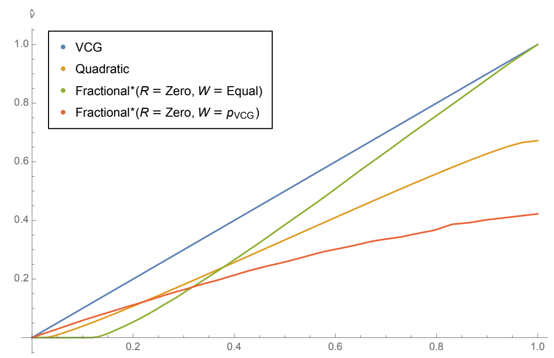


Figure 3: The BNE strategies in Domain(MD = Uniform,Corr = CrossLarge). Shown are the BNE strategies for Quadratic, Fractional*(R = Zero,W = Equal), and Fractional*(R=p_{VCG},W=Equal).

To illustrate the shape of the BNEs of different rules, Figure 3 shows the BNE strategies for four payment rules for Domain(MD=Uniform,Corr=CrossLarge). It is easy to see how much better Fractional*(R=Zero,W=Equal) is compared to Quadratic. Additionally, we also plot one of the worst-performing rules in this domain, i.e., Fractional*(R=Zero,W=p_{VCG}). Note how just changing one parameter of the rule (the weight) turns the rule from a top-performer into a very badly-performing rule. In the next section, we will search for good *all-rounder rules*, i.e., rules that perform very well across all 29 domains.

5.5. Best All-Rounder Rules

In the previous section, we have evaluated our rules one domain at a time. When auctioneers have good information about their domain structure this enables the selection of very high-performing rules, even if these rules perform poorly elsewhere in the domain space. However, in practice auctioneers may not know the exact structure of the domain in which they are operating. Accordingly, we may seek good "all-rounder" rules that are widely applicable.

Different auctioneers might reasonably place different emphasis on each of our evaluation dimensions. In the absence of such knowledge, we opt to take a simple average over all three dimensions and then rank our rules by this average. Table 5 shows the top 20. Here we see that

	Efficiency	Incentives	Revenue	
Quadratic	97.71%	17.66	63.72%	
Best All-Rounder Rules	**Avg. Improvement over Quadratic**			**Avg.**
Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG},A=2)	0.91%	17.83%	6.25%	8.33%
Fractional*(R=Zero,W=π_{VCG},A=0.5)	0.92%	17.17%	6.57%	8.22%
Fractional*(R=Zero,W=π_{Shapley},A=0.5)	0.86%	17.17%	5.90%	7.98%
Fractional*(R=π_{VCG},W=π_{Shapley})	0.86%	17.15%	5.86%	7.96%
Fractional*(R=π^{M}_{Shapley},W=$\pi^{-1}_{\text{Shapley}}$)	0.87%	17.10%	5.87%	7.94%
Fractional*(R=π^{M}_{VCG},W=π^{-1}_{VCG})	0.85%	16.92%	5.73%	7.83%
Fractional*(R=π_{VCG},W=π_{VCG})	0.85%	16.92%	5.73%	7.83%
Fractional*(R=π_{Shapley},W=π_{VCG},A=2)	0.84%	16.77%	5.47%	7.69%
Fractional*(R=p^{M}_{Shapley},W=Bid^{-1},A=3)	0.82%	16.74%	5.05%	7.54%
Fractional*(R=π_{VCG},W=Bid)	0.80%	16.30%	5.39%	7.50%
Fractional*(R=p_{VCG},W=p^{-1}_{Shapley})	0.82%	16.30%	5.34%	7.49%
Fractional*(R=p^{M}_{Shapley},W=p^{-1}_{Shapley},A=3)	0.81%	16.60%	5.00%	7.47%
Fractional*(R=Zero,W=Bid,A=0.5)	0.79%	16.23%	5.35%	7.46%
Fractional*(R=p^{M}_{Shapley},W=$\pi^{-1}_{\text{Shapley}}$,A=3)	0.81%	16.59%	4.96%	7.46%
Fractional*(R=p_{VCG},W=Bid^{-1})	0.80%	16.10%	5.16%	7.35%
Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG})	0.77%	15.69%	5.13%	7.20%
Fractional*(R=π_{VCG},W=p_{Shapley})	0.76%	15.64%	5.10%	7.17%
Fractional*(R=p_{VCG},W=π^{-1}_{VCG},A=2)	0.79%	15.72%	4.94%	7.15%
Fractional*(R=π^{M}_{Shapley},W=p^{-1}_{Shapley},A=0.5)	0.77%	15.42%	5.14%	7.11%
Fractional*(R=Zero,W=p_{Shapley},A=0.5)	0.75%	15.44%	5.01%	7.07%

Table 5: Results showing the top 20 all-rounder rules. The first row is the average performance of Quadratic over all 29 domains. The subsequent rows show the top rules by their average improvement over Quadratic. Rules that beat quadratic in every dimension in every domain are highlighted in grey.

the best rule, a π^{M}_{Shapley}-based rule, is able to achieve an 8.33% average improvement (across all three dimensions) over Quadratic, across all 29 domains.

Note that three of those 20 rules actually beat Quadratic in every dimension in every domain; those rules are highlighted in grey in Table 5. The other rules typically beat Quadratic in most (but not all) of the 29 domains (e.g., on 26, 27, or 28 domains). However, we do not consider it to be an exclusion criterion if a rule loses to Quadratic in one or multiple domains. In fact, we find that Quadratic actually performs quite well in some domains, where it is almost impossible to beat. It would not make sense to restrict our search for good all-rounder rules to only those that beat Quadratic everywhere.

Looking at Table 5 in more detail, we observe that Shapley-based rules are ubiquitous. Indeed, all three rules that beat Quadratic everywhere are Shapley-based. This is a fascinating finding, as Shapley-based reference points or weights had previously not been considered for the design of core-selecting payment rules. Looking into the calculation of the Shapley value in more detail, it is intuitive that the Shapley value captures a "fair" distribution of the surplus (and thus may naturally serve as a good reference point), while at the same time being somewhat robust against manipulations. However, understanding the exact way in which the Shapley value-based rules drive incentives (and then efficiency and revenue) requires a more detailed analysis of the individual rules, which is beyond the scope of this paper but which is an interesting subject for future work.

The Shapley-based rules are also particularly noteworthy from a computational point-of-view because computing the Shapley value is #P-hard in general games. Thus, we currently only have algorithms that have run-time exponential in the number of players. For the LLG domain, this is not a problem, but in larger domains, this computational complexity becomes prohibitively expensive, and consequently we must omit the Shapley-based rules from our analysis of a much larger domain, LLLLGG, in the next section.

Taking a broader view at the rules in Table 5 (and plotting and analyzing the corresponding BNEs), we observed a certain pattern: the rules that performed best tended to be *Large*-style rules, i.e., they favor bidders with large values. Of course, there are multiple ways to achieve this kind of behavior. One way is exemplified via the second best rule in Table 5, i.e., FRACTIONAL*(R=ZERO, W=π_{VCG},A=0.5). This is an interesting rule as it uses a ZERO reference point, which heavily tilts the payments in favor of the large players. Additionally, it uses π_{VCG} as weights (which tilts it back towards the small players) with a very small (0.5) amplification. Note that an amplification smaller than 1 de-emphasizes the weights, thus making the rule more like QUADRATIC, but not quite. Considering all parameters at once, the rule is a dampened version of QUADRATIC with ZERO reference point. A rule that is very similar is FRACTIONAL*(R=ZERO,W=BID,A=0.5) (in row 13), except that it uses BID as the weight instead of π_{VCG}. Given that π_{VCG} = BID $- p_{\text{VCG}}$, it is very intuitive that these two rules perform very similar.

An alternative way to construct a *Large*-style rule is to use a reference point with a more moderate impact/tilting, and instead create the effect via the weights. This is exemplified via the rule FRACTIONAL*(R = p_{VCG},W = BID^{-1}), in row 15 of Table 5. This rule uses the standard p_{VCG} reference point, but combines it with BID^{-1} as weights. The inverted bid weighting has the effect of heavily tilting the payments in favor of the large players, compared to the un-weighted version of QUADRATIC. Again, a similar rule can be found in row 18, namely FRACTIONAL*(R=p_{VCG},W=π^{-1}_{VCG}, A=2), which uses π^{-1}_{VCG} instead of BID^{-1} as the weighting (which has a similar effect). Additionally, this rule has a higher amplification, thus slightly increasing the tilting effect due to the weights.

These four non-Shapley-based rules which we have just discussed are the subset of rules from our LLG analysis which we will evaluate in the LLLLGG domain in the next section.

Remark 1 (Optimal Combination) *Note that there is no such thing as an "optimal reference point" or an "optimal set of weights." Consider Table 5, which shows the Top-20 all-rounder rules. Among those 20 rules, seven out of the nine reference points we consider show up. Similarly, eight out of the eleven weights we consider show up. No clear winner seems to emerge. In fact, our results sug-*

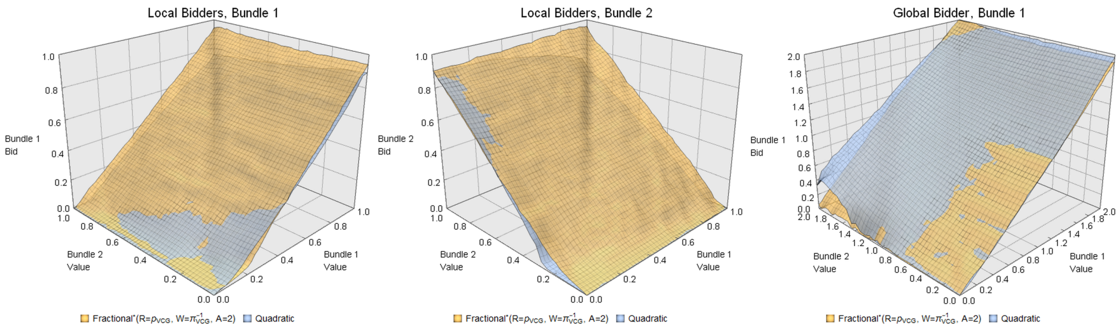
gest that a search for an optimal reference point or an optimal set of weights seems to be misguided, because it is really the combination *of the reference point, the weights and the amplification, that determine whether a payment rule performs well or not. This is very well illustrated by the pair of rules in Table 5 in Rows 6 and 7. As you can see, the two rules differ from each other in that one uses the mirrored reference point of the other as well as the inverted weight of the other. As our results show, the two rules have the exact same statistics - in fact, these two rules* are *exactly the same, at least in LLG. However, if we switch the weight of the first rule with the weight of the second rule, we obtain a new, very bad rule, whose average improvement over* QUADRATIC *is -3.4%. Note that this pattern we just described shows up throughout our results. This highlights the importance of choosing the right* combination *of reference point, weights, and amplification in the design of core-selecting payment rules.*

6. Results for LLLLGG

We next take the rules that worked well in the LLG domain and seek to find out if they work in a larger domain, specifically the LLLLGG domain introduced by Bosshard et al. (2017) as a generalization of LLG. This domain is significantly more complex, but numerical BNEs can just barely be computed for it using a powerful computational grid. Specifically, the LLLLGG domain has 8 goods, and 6 bidders, each of which is interested in two bundles. There are four local bidders, each interested in two distinct pairs of two goods. And there are two global bidders interested in two distinct sets of 4 goods. There are significant symmetries in the domain that reduce the complexity of the strategy space. Consequently, the strategies of the local players can

Rule	**Efficiency**	**Incentives**	**Revenue**	
QUADRATIC	99.2%	0.54	1.07%	
Improvement Over QUADRATIC				**Avg.**
FRACTIONAL*(R=p_{VCG},W=π^{-1}_{VCG},A=2)	0.61%	45.67%	3.65%	16.64%
FRACTIONAL*(R=p_{VCG},W=BID^{-1})	0.52%	42.95%	2.69%	15.39%
FRACTIONAL*(R=ZERO,W=π_{VCG},A=0.5)	0.37%	40.76%	2.00%	14.38%
FRACTIONAL*(R=ZERO,W=BID,A=0.5)	0.38%	40.77%	1.96%	14.37%
FRACTIONAL*(R=BID,W=p_{VCG},A=5)	0.24%	-22.44%	-2.54%	-8.25%
Proxy	-4.41%	-107.62%	15.26%	-32.26%
Proportional	-3.59%	-95.25%	-11.72%	-36.85%
First Price	-2.80%	-169.94%	-7.91%	-60.22%

Table 6: Shown are the results of running rules in the much larger LLLLGG domain. The first row is the performance of QUADRATIC relative to VCG. Next we show four of our best all-rounder rules from LLG, followed by one of the worst rules found in LLG. At the bottom we present the results for three commonly-studied non-MRC rules.



(a) The first strategy of the local player.

(b) The second strategy of the local player.

(c) The first strategy of the global player; her second strategy is symmetric.

Figure 4: BNE strategies for the LLLLGG domain.

be represented as two 2D surfaces, and the strategy of the global player can be represented as a pair of symmetric 2D surfaces (which unifies their computation).

Each local bidder in LLLLGG draws a value for each bundle from $U[0,1]$, while the global bidders draw their larger bundles from $U[0,2]$. In the standard version of the domain, none of the distributions are correlated or involve a complex marginal. We stick with this, because even the standard version requires tens of thousands of core-hours to solve for a high quality numerical BNE.

Accordingly, we have picked the four rules from LLG described in the previous section and evaluated them in this much more complex domain, as presented in Table 6. It is important to note that Quadratic is already almost efficient in LLLLGG. This is consistent with our LLG results, where correlation often co-occured with Quadratic being less than efficient. Even still, the selected high-performing rules from LLG did indeed generalize to the much harder LLLLGG domain, and were still able to significantly outperform Quadratic, especially on incentives and revenue.

For comparison, we also include one of the worst rules from LLG in the table, namely (Fractional*(R=Bid,W=p_{VCG},A=5)), and as expected, it performs poorly in LLLLGG as well. Lastly, we also present several other commonly-studied payment rules (Proxy, Proportional and First Price), even though they are not MRC-selecting payment rules. We see these rules perform significantly worse than both Quadratic, and our low performing MRC rule from LLG.

To obtain an intuition for how the rules operate in LLLLGG, we have plotted the BNE strategies in Figure 4. The figure illustrates the BNE strategy profile of Quadratic in blue, and the BNE strategy profile for one of the high performing rules, Fractional*(R=p_{VCG},W=π^{-1}_{VCG},A=2), in yellow. We see that our rule induces a far more truthful strategy than Quadratic. It is informative to observe how Quadratic induces players to both shade down a lot (especially for the local players) and overbid (especially for the global players, when their value for bundle 2 is large). Notably, our rule is significantly more truthful for the local players when they have a large value for either bundle. In Appendix G, we include a similar figure that depicts the poorly performing rule as well, which induces shading and over-bidding even more drastically than QUADRATIC. Overall, the rules and the intuitions we have obtained from LLG translate well to the far more complex LLLLGG setting.

7. Conclusion

In this paper, we have presented a computational search approach for finding good core-selecting payment rules. Using a suitable parametrization of the payment rule design space, we have been able to systematically search through this space and identify very good rules that outperform the commonly-used *Quadratic* rule on each dimension (efficiency, incentives, and revenue).

We have followed a two-step approach. First, we have studied the well-known but stylized LLG domain, which is amenable to an extensive search through our rule space. Within this domain, we have identified a set of 20 very good all-rounder rules which beat QUADRATIC by a significant margin across all domains (on average). Out of those, we have selected four rules for evaluation in the larger LLLLGG domain, and we have demonstrated that those rules still perform better than QUADRATIC. In contrast, other benchmark rules from the literature perform worse than QUADRATIC in LLLLG. Thus, we have shown a certain degree of generalizability of our overall design approach, and we have demonstrated the robustness of the new rules we have identified.

In terms of mechanism design, one of our important findings from this work is that previous approaches that have tried to optimize core-selecting payment rules by modifying only the reference point or the weights may have been misguided. According to our results, it is the *perfect combination* of reference points, weights, and amplification, that determine whether a rule is good or bad. In fact, we have shown that small local changes to a well-performing rule typically turns it into a badly-performing rule. Thus, our results demonstrate how complex and intricate it is to find good core-selecting payment rules.

In terms of our computational search approach, we believe that our work illustrates the power of an automated search for a good mechanism, rather than designing it by hand. The design of core-selecting rules lends itself to this approach, because the design space can be nicely parameterized, and then searched through. However, computational complexity is a serious concern, and future work should thus explore smart ways to scale this approach up to even larger settings.

Interestingly, computational complexity was also a concern for the design of the payment rules themselves. Some of our most promising rules were based on the Shapley value which is too computationally expensive to compute exactly in large domains. There exists some work on algorithms for approximating the Shapley value (e.g., exploiting some structure in the problem). Future work should evaluate the impact of using an approximate Shapley value instead of the "exact" Shapley value in our rules. Our estimate is that the impact should be small.

Going forward, we hope that other researchers may also consider using a computational search approach for similar auction design questions. Orthogonally, we hope that some of the very good all-rounder rules we have identified in our search may spur new theoretical research, new computational research, or that they may be considered for implementation in practice.

References

Ausubel, Lawrence, and Oleg Baranov. 2013. "Core-Selecting Auctions with Incomplete Information." Working Paper, University of Maryland.

Ausubel, Lawrence, and Oleg Baranov. 2017. "A Practical Guide to the Combinatorial Clock Auction." *Economic Journal.* Forthcoming.

Ausubel, Lawrence, and Paul Milgrom. 2002. "Ascending auctions with package bidding." *Frontier of Theoretical Economics*, 1(1): 1–42.

Ausubel, Lawrence, and Paul Milgrom. 2006. "The Lovely but Lonely Vickrey Auction." In In *Combinatorial Auctions.* , ed. Peter Cramton, Yoav Shoham and Richard Steinberg, 17–40. MIT Press.

Bosshard, Vitor, Benedikt Bünz, Benjamin Lubin, and Sven Seuken. 2017. "Computing Bayes-Nash Equilibria in Combinatorial Auctions with Continuous Value and Action Spaces." In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI).* Melbourne, Australia.

Brown, George W. 1951. "Iterative solution of games by fictitious play." *Activity Analysis of Production and Allocation*, 13(1): 374–376.

Clarke, Edward. 1971. "Multipart pricing of public goods." *Public Choice*, 11(1): 17–33.

Cramton, Peter. 2013. "Spectrum auction design." *Review of Industrial Organization*, 42(2): 161–190.

Cramton, Peter, Yoav Shoham, and Richard Steinberg. 2006. *Combinatorial Auctions.* MIT Press.

Day, Robert W., and Paul Milgrom. 2008. "Core-selecting Package Auctions." *International Journal of Game Theory*, 36(3): 393–407.

Day, Robert W., and Peter Cramton. 2012. "Quadratic Core-Selecting Payment Rules for Combinatorial Auctions." *Operations Research*, 60(3): 588–603.

Day, Robert W., and S. Raghavan. 2007. "Fair Payments for Efficient Allocations in Public Sector Combinatorial Auctions." *Management Science*, 53(9): 1389–1406.

Erdil, Aytek, and Paul Klemperer. 2010. "A New Payment Rule for Core-Selecting Package Auctions." *Journal of the European Economics Association*, 8(2–3): 537–547.

Goeree, Jacob, and Yuanchuan Lien. 2016. "On the Impossibility of Core-Selecting Auctions." *Theoretical Economics*, 11: 41–52.

Groves, T. 1973. "Incentives in Teams." *Econometrica*, 41(4): 617–631.

Lubin, Benjamin, and David Parkes. 2009. "Quantifying the strategyproofness of mechanisms via metrics on payoff distributions." In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Lubin, Benjamin, Benedikt Bünz, and Sven Seuken. 2015. "New Core-Selecting Payment Rules with Better Fairness and Incentive Properties [Extended Abstract]." In *Proceedings of the 3rd Conference on Auctions, Market Mechanisms and Applications (AMMA)*.

LukeLindsay. 2017. "Shapley value based pricing for auctions and exchanges." *Games and Economic Behavior.*

Milgrom, Paul. 2007. "Package Auctions and Exchanges." *Econometrica*, 75(4): 935–965.

Nisan, Noam, and Amir Ronen. 2001. "Algorithmic mechanism design." *Games and Economic Behavior*, 35: 166–196.

Parkes, David. 2002. "On Indirect and Direct Implementations of Core Outcomes in Combinatorial Auctions." Harvard University.

Parkes, David C., Jayant Kalagnanam, and Marta Eso. 2001. "Achieving Budget-Balance with Vickrey-Based Payment Schemes in Exchanges." In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)*. Seattle, WA.

Sandholm, Tuomas. 2003. "Automated mechanism design: A New Application Area for Search Algorithms." In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP)*.

Shapley, Lloyd S. 1953. "A value for n-person games." *Contributions to the Theory of Games*, 2(28): 307–317.

Sklar, M. 1959. *Fonctions de répartition à n dimensions et leurs marges.* Université Paris 8.

Varian, Hal, and Jeffrey MacKie-Mason. 1994. "Generalized Vickrey Auctions." University of Michigan.

Vickrey, William. 1961. "Counterspeculation, Auctions, and Competitive Sealed Tenders." *The Journal of Finance*, 16(1): 8–37.

Appendix

Results For All Distributions

In Section 5, we introduce variants of the LLG domain in which we analyze our payment rules. In the following sections we present the best rules for each of these domains.

We note the number of domains in this set, 29, is slightly smaller than the full cross product of all the domain features we have considered. This is because the BNE algorithm did not converge to a very high degree of tolerance (0.1%) on sufficiently many rules that we felt we could generalize from the results. Our threshold for this was that at least 1/3 of the rules (i.e., 203) had to converge. The threshold was not met for the domains DOMAIN(MD=BETA(3,1/3),Corr=SAME), DOMAIN(MD = BETA(5/3,3)), DOMAIN(MD = BETA(5/3,3)), DOMAIN(MD = BETA(3,5/3)), DOMAIN(MD = BETA(3,5/3)), DOMAIN(MD=BETA(3,3)). Additionally QUADRATIC failed to converge on DOMAIN(MD=BETA(1/3,1/3),Corr=SAME) so we could not compare the other rules to the benchmark rule. Finally, we also excluded all variations of DOMAIN(MD=BETA(1/3,3)), which is an exponentially left-skewed distribution. This is the reason why we only have five instead of six Beta distributions in Figure 2. The reason for excluding this domain is that it leads to pathological effects when evaluating core-selecting payment rules. In the uncalibrated setting, VCG is almost always in the core, which makes this an uninteresting case. In the calibrated setting, when a local bidder gets lucky and has a high value, then the other local bidder still almost certainly has a very low value. This implies that the high-valued bidder essentially has no ability to manipulate the core payment. In fact, shading his bid might risk losing to the global bidder. Thus, the distribution of this domain completely determines the incentives of any core-selecting rule (which are all very close to truthful), which renders comparing individual rules non-interesting.

A. UNIFORM Results

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	98.03%	16.19	91.30%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=2)	0.31%	3.90%	1.62%	1.94%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=BID^{-1},A=10)	0.27%	5.13%	1.27%	2.22%
Best Revenue	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=2)	0.31%	3.90%	1.62%	1.94%

Table 7: The results for DOMAIN(MD = UNIFORM). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

B. Uniform Marginals and Correlation Results

We investigated using both smaller, 0.25, and larger, 0.75, correlation constants in our copulae. Results on individual domains are provided below:

B.1. Same-Side

	Rule	Efficiency	Incentives	Revenue	
	Quadratic	98.13%	14.11	93.14%	
	Improvement Over Quadratic				**Avg.**
Best Efficiency	Fractional*(R=π_{Shapley},W=p^{-1}_{VCG},A=0.5)	0.34%	6.64%	3.09%	3.36%
Best Incentives	Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG},A=0.5)	0.31%	8.16%	2.50%	3.66%
Best Revenue	Fractional*(R=Zero,W=Equal)	0.24%	2.00%	3.29%	1.84%

Table 8: The results for Domain(MD=Uniform,Corr=SameSmall). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	Quadratic	98.41%	10.91	95.19%	
	Improvement Over Quadratic				**Avg.**
Best Efficiency	Fractional*(R=Zero,W=π_{VCG},A=2)	0.24%	6.77%	2.65%	3.22%
Best Incentives	Fractional*(R=π_{Shapley},W=Bid,A=0.5)	0.20%	8.50%	2.61%	3.77%
Best Revenue	Fractional*(R=Zero,W=$\pi^{-1}_{\text{Shapley}}$)	0.04%	-0.61%	4.00%	1.14%

Table 9: The results for Domain(MD = Uniform,Corr = Same). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	Quadratic	98.90%	6.16	98.37%	
	Improvement Over Quadratic				**Avg.**
Best Efficiency	Fractional*(R=p_{Shapley},W=p^{-1}_{VCG},A=10)	0.17%	8.72%	1.74%	3.54%
Best Incentives	Fractional*(R=π^{M}_{Shapley},W=π_{VCG},A=3)	0.14%	10.64%	2.36%	4.38%
Best Revenue	Fractional*(R=Zero,W=p^{-1}_{VCG},A=10)	-0.09%	4.14%	2.54%	2.20%

Table 10: The results for Domain(MD=Uniform,Corr=SameLarge). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

B.2. Cross-Side

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	97.87%	16.89	92.93%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π^{M}_{VCG},W=$\pi^{-1}_{\text{SHAPLEY}}$)	0.57%	11.40%	2.10%	4.69%
Best Incentives	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=$\pi^{-1}_{\text{SHAPLEY}}$)	0.55%	12.09%	2.13%	4.92%
Best Revenue	FRACTIONAL*(R=π_{VCG},W=π_{SHAPLEY})	0.56%	11.92%	2.17%	4.88%

Table 11: The results for DOMAIN(MD = UNIFORM,Corr = CROSSSMALL). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	97.69%	17.69	95.27%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.06%	23.31%	4.29%	9.55%
Best Incentives	FRACTIONAL*(R=ZERO,W=π_{VCG},A=0.5)	1.04%	23.77%	4.20%	9.67%
Best Revenue	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.06%	23.31%	4.29%	9.55%

Table 12: The results for DOMAIN(MD=UNIFORM,Corr=CROSS). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	97.35%	18.92	97.51%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=ZERO,W=EQUAL)	2.02%	48.25%	11.88%	20.72%
Best Incentives	FRACTIONAL*(R=ZERO,W=EQUAL)	2.02%	48.25%	11.88%	20.72%
Best Revenue	FRACTIONAL*(R=ZERO,W=EQUAL)	2.02%	48.25%	11.88%	20.72%

Table 13: The results for DOMAIN(MD = UNIFORM,Corr = CROSSLARGE). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

C. BETA Marginals and No Correlation Results

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	97.78%	21.53	88.63%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=p_{VCG},W=BID^{-1})	0.62%	9.71%	5.86%	5.40%
Best Incentives	FRACTIONAL*(R=π_{VCG},W=p_{SHAPLEY})	0.58%	11.04%	5.09%	5.57%
Best Revenue	FRACTIONAL*(R=p_{VCG},W=π^{-1}_{VCG},A=3)	0.55%	3.64%	6.02%	3.40%

Table 14: The results for DOMAIN(MD=BETA(3,5/3)). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	96.77%	21.53	101.20%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%
Best Incentives	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%
Best Revenue	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%

Table 15: The results for DOMAIN(MD=BETA(3,5/3),UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	98.39%	15.18	97.26%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=p_{SHAPLEY},W=π_{VCG},A=5)	0.43%	9.95%	2.55%	4.31%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=BID^{-1},A=3)	0.39%	10.47%	2.21%	4.36%
Best Revenue	FRACTIONAL*(R=π_{SHAPLEY},W=π_{VCG},A=5)	0.43%	7.21%	3.00%	3.55%

Table 16: The results for DOMAIN(MD = BETA(1/3,1/3),UNCALIBRATED*). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	Quadratic	97.80%	16.71	88.67%	
	Improvement Over Quadratic				
Best Efficiency	Fractional*(R=p_{VCG},W=$\pi^{-1}_{\text{Shapley}}$)	0.44%	6.69%	3.57%	3.57%
Best Incentives	Fractional*(R=p_{VCG},W=π^{-1}_{VCG})	0.39%	7.58%	2.97%	3.65%
Best Revenue	Fractional*(R=p_{VCG},W=$\pi^{-1}_{\text{Shapley}}$)	0.44%	6.69%	3.57%	3.57%

Table 17: The results for Domain(md=Beta(3,3),Uncalibrated). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	Quadratic	95.08%	29.39	141.89%	
	Improvement Over Quadratic				
Best Efficiency	Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG},A=2)	2.70%	23.60%	6.60%	10.97%
Best Incentives	Fractional*(R=p^{M}_{Shapley},W=p^{-1}_{Shapley},A=3)	2.60%	25.00%	6.30%	11.30%
Best Revenue	Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG},A=2)	2.70%	23.60%	6.60%	10.97%

Table 18: The results for Domain(md=Beta(3,1/3),Uncalibrated). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	Quadratic	97.79%	29.49	88.24%	
	Improvement Over Quadratic				
Best Efficiency	Fractional*(R=π^{M}_{Shapley},W=π^{-1}_{VCG},A=2)	1.20%	24.80%	13.90%	13.30%
Best Incentives	Fractional*(R=p^{M}_{Shapley},W=p^{-1}_{Shapley},A=3)	1.10%	25.70%	12.30%	13.03%
Best Revenue	Fractional*(R=Bid,W=p_{Shapley},A=2)	1.00%	7.00%	15.80%	7.93%

Table 19: The results for Domain(md=Beta(3,1/3)). The first row is the performance of Quadratic relative to VCG. The subsequent rows show the top rules by each dimension.

Rule	Efficiency	Incentives	Revenue
QUADRATIC	97.95%	11.71	89.93%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.27%	4.61%	1.08%	1.99%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.27%	4.61%	1.08%	1.99%
Best Revenue	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.27%	4.61%	1.08%	1.99%

Table 20: The results for DOMAIN(MD=BETA(5/3,3)). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Rule	Efficiency	Incentives	Revenue
QUADRATIC	98.83%	11.68	80.99%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%
Best Revenue	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%

Table 21: The results for DOMAIN(MD=BETA(5/3,3),UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

D. BETA Marginals and Correlation Results

D.1. Same-Side

Rule	Efficiency	Incentives	Revenue
QUADRATIC	97.96%	17.06	94.23%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=BID^{M},W=p^{-1}_{VCG})	0.55%	15.05%	10.24%	8.61%
Best Incentives	FRACTIONAL*(R=BID,W=p_{VCG},A=0.5)	0.52%	15.86%	8.41%	8.26%
Best Revenue	FRACTIONAL*(R=ZERO,W=$\pi^{-1}_{\text{SHAPLEY}}$,A=3)	0.01%	-7.38%	13.31%	1.98%

Table 22: The results for DOMAIN(MD=BETA(3,5/3),Corr=SAME). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	98.04%	12.69	94.21%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.42%	10.28%	7.64%	6.11%
Best Incentives	FRACTIONAL*(R=BID$^{\text{M}}$,W=p^{-1}_{VCG},A=0.5)	0.37%	13.29%	7.01%	6.89%
Best Revenue	FRACTIONAL*(R=ZERO,W=p^{-1}_{SHAPLEY},A=2)	0.11%	-0.34%	10.56%	3.44%

Table 23: The results for DOMAIN(MD = BETA(3,3),Corr = SAME,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

D.2. Cross-Side

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	97.65%	22.29	89.60%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=ZERO,W=EQUAL)	1.55%	36.92%	15.28%	17.92%
Best Incentives	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	1.53%	38.14%	14.17%	17.95%
Best Revenue	FRACTIONAL*(R=π_{VCG},W=BID,A=2)	1.50%	33.60%	15.50%	16.87%

Table 25: The results for DOMAIN(MD=BETA(3,5/3),Corr=CROSS). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	97.76%	29.78	88.39%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=BID,W=BID,A=2)	1.70%	45.10%	23.30%	23.37%
Best Incentives	FRACTIONAL*(R=$\pi^{\text{M}}_{\text{VCG}}$,W=$\pi^{-1}_{\text{SHAPLEY}}$)	1.70%	47.70%	20.80%	23.40%
Best Revenue	FRACTIONAL*(R=BID$^{\text{M}}$,W=p^{-1}_{SHAPLEY},A=2)	1.70%	41.80%	23.40%	22.30%

Table 24: The results for DOMAIN(MD=BETA(3,1/3),Corr=CROSS). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Rule	Efficiency	Incentives	Revenue
QUADRATIC	97.67%	18.22	102.48%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.29%	26.85%	4.39%	10.84%
Best Incentives	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.29%	26.85%	4.39%	10.84%
Best Revenue	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.29%	26.85%	4.39%	10.84%

Table 26: The results for DOMAIN(MD=BETA(1/3,1/3),Corr=CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Rule	Efficiency	Incentives	Revenue
QUADRATIC	97.64%	17.44	90.39%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=ZERO,W=EQUAL)	1.45%	34.24%	12.36%	16.02%
Best Incentives	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=3)	1.43%	35.11%	11.76%	16.10%
Best Revenue	FRACTIONAL*(R=ZERO,W=EQUAL)	1.45%	34.24%	12.36%	16.02%

Table 27: The results for DOMAIN(MD = BETA(3,3),Corr = CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

Rule	Efficiency	Incentives	Revenue
QUADRATIC	97.68%	12.66	94.26%

	Improvement Over QUADRATIC				Avg.
Best Efficiency	FRACTIONAL*(R=ZERO,W=π_{VCG},A=0.5)	1.01%	24.18%	4.03%	9.74%
Best Incentives	FRACTIONAL*(R=ZERO,W=π_{VCG},A=0.5)	1.01%	24.18%	4.03%	9.74%
Best Revenue	FRACTIONAL*(R=ZERO,W=π_{SHAPLEY},A=0.5)	0.97%	23.27%	4.33%	9.52%

Table 28: The results for DOMAIN(MD=BETA(5/3,3),Corr=CROSS). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

E. Uncalibrated BETA Marginals and No Correlation Results

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	95.08%	29.39	141.89%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=2)	2.70%	23.60%	6.60%	10.97%
Best Incentives	FRACTIONAL*(R=p^{M}_{SHAPLEY},W=p^{-1}_{SHAPLEY},A=3)	2.60%	25.00%	6.30%	11.30%
Best Revenue	FRACTIONAL*(R=π^{M}_{SHAPLEY},W=π^{-1}_{VCG},A=2)	2.70%	23.60%	6.60%	10.97%

Table 29: The results for DOMAIN(MD=BETA(3,1/3),UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	96.77%	21.53	101.20%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%
Best Incentives	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%
Best Revenue	FRACTIONAL*(R=π_{VCG},W=p_{VCG},A=2)	0.94%	11.13%	4.62%	5.56%

Table 30: The results for DOMAIN(MD=BETA(3,5/3),UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	98.83%	11.68	80.99%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%
Best Incentives	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%
Best Revenue	FRACTIONAL*(R=π_{SHAPLEY},W=p^{-1}_{VCG},A=10)	0.12%	3.24%	1.59%	1.65%

Table 31: The results for DOMAIN(MD=BETA(5/3,3),UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

F. Uncalibrated BETA Marginals and Correlation Results

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	95.00%	29.73	143.59%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%
Best Incentives	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%
Best Revenue	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	4.00%	47.20%	10.00%	20.40%

Table 32: The results for DOMAIN(MD=BETA(3,1/3),Corr=CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	96.64%	22.11	103.92%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=ZERO,W=EQUAL)	2.20%	35.84%	10.85%	16.29%
Best Incentives	FRACTIONAL*(R=BID,W=π_{SHAPLEY},A=2)	2.18%	37.44%	10.58%	16.73%
Best Revenue	FRACTIONAL*(R=ZERO,W=EQUAL)	2.20%	35.84%	10.85%	16.29%

Table 33: The results for DOMAIN(MD=BETA(3,5/3),Corr=CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	
	QUADRATIC	96.67%	17.93	107.77%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=ZERO,W=π_{VCG},A=0.5)	0.97%	10.98%	6.97%	6.30%
Best Incentives	FRACTIONAL*(R=BID$^{\text{M}}$,W=BID^{-1},A=2)	0.91%	12.88%	6.17%	6.65%
Best Revenue	FRACTIONAL*(R=ZERO,W=BID^{-1})	0.69%	1.55%	8.96%	3.73%

Table 34: The results for DOMAIN(MD=BETA(3,5/3),Corr=SAME,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	99.31%	7.14	88.33%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=π_{VCG},W=p^{-1}_{VCG},A=0.5)	0.05%	6.69%	2.05%	2.93%
Best Incentives	FRACTIONAL*(R=ZERO,W=p_{SHAPLEY},A=0.5)	0.01%	10.07%	4.39%	4.82%
Best Revenue	FRACTIONAL*(R=ZERO,W=BID^{-1},A=10)	-0.12%	-3.73%	6.93%	1.03%

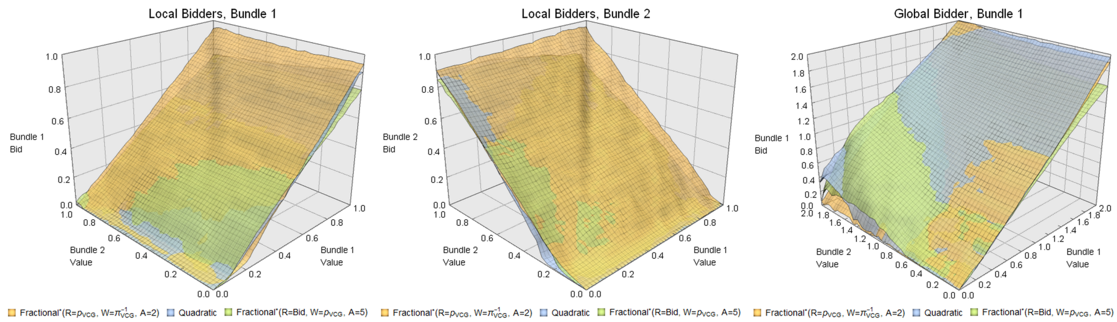
Table 35: The results for DOMAIN(MD=BETA(5/3,3),Corr=SAME,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

	Rule	Efficiency	Incentives	Revenue	Avg.
	QUADRATIC	98.65%	12.68	82.40%	
	Improvement Over QUADRATIC				**Avg.**
Best Efficiency	FRACTIONAL*(R=p_{VCG},W=π^{-1}_{VCG},A=3)	0.62%	25.24%	7.46%	11.10%
Best Incentives	FRACTIONAL*(R=p_{VCG},W=π^{-1}_{VCG},A=3)	0.62%	25.24%	7.46%	11.10%
Best Revenue	FRACTIONAL*(R=ZERO,W=EQUAL)	0.62%	23.84%	7.62%	10.69%

Table 36: The results for DOMAIN(MD=BETA(5/3,3),Corr=CROSS,UNCALIBRATED). The first row is the performance of QUADRATIC relative to VCG. The subsequent rows show the top rules by each dimension.

G. Robustness of the Generalization from LLG to LLLLGG

Below we present a figure the results of a running one of the worst MRC rules we found in LLG in the larger LLLLGG domain. From the figure it is evident that just as we can find rules that perform better than Quadratic in the MRC, we can also find rules that perform worse. And at least pointwise, rules that tend to perform well in LLG also perform well in LLLLGG.



(a) The first strategy of the local player.

(b) The second strategy of the local player.

(c) The first strategy of the global player; her second strategy is symmetric.

H. Pseudo-Code of BNE Algorithm

For details of the BNE algorithm, please see Bosshard et al. (2017). Here, we provide the pseudo-code for the main parts of the algorithm for convenience.

Algorithm 0: Fictitious Play

Data: bidder's value distributions V_i, utility functions u_i
Result: ϵ-BNE strategy profile
s := truthful strategies
while *not converged* **do**
  for *each bidder i* **do**
    s'_i := empty piecewise linear function
    for k = *1* **to** *number of control points* **do**
      `// Based on` s'_i
      choose a new value $v_i \in V_i$
      $br := \arg\max_{\hat{v}} \mathbf{E}_{v_{-i}} \left[u_i(v_i, \hat{v}, s_{-i}(v_{-i})) \right]$
      $\hat{v}_{old} := s_i(v_i)$ computed by interpolation
      $u_{old} := u_i(v_i, \hat{v}_{old}, s_{-i}(v_{-i}))$
      $u_{br} := u_i(v_i, br, s_{-i}(v_{-i}))$
      $\hat{v}_{new} := update(\hat{v}_{old}, br, u_{old}, u_{br})$
      add control point $(v_i, \hat{v}_{new})$ to s'_i
    end
  end
  s := s'
end
return s

Algorithm 1: Update Rule

Data: $w_{\min} \in \mathbb{R}$, $w_{\max} \in \mathbb{R}$,$\alpha \in \mathbb{R}$
Result: Merged strategy
Function *update*($S_c \in \mathbb{R}$, $\hat{S}_c \in \mathbb{R}$, $U_c \in \mathbb{R}$,$\hat{U}_c \in \mathbb{R}$)
  `// Adaptive weight`
  $w = \arctan(\alpha \cdot (\frac{\hat{U}_c}{U_c} - 1)) \cdot \frac{2}{\pi} \cdot (w_{\max} - w_{\min}) + w_{\min} \ \in (w_{\min}, w_{\max});$
  return $S_c \cdot (1 - w) + \hat{S}_c \cdot w;$
