Black-Box Randomized Reductions in Algorithmic Mechanism Design

Shaddin Dughmi *
Department of Computer Science
Stanford University
shaddin@cs.stanford.edu

Tim Roughgarden †
Department of Computer Science
Stanford University
tim@cs.stanford.edu

***Abstract*—We give the first black-box reduction from arbitrary approximation algorithms to truthful approximation mechanisms for a non-trivial class of multi-parameter problems. Specifically, we prove that every packing problem that admits an FPTAS also admits a truthful-in-expectation randomized mechanism that is an FPTAS. Our reduction makes novel use of smoothed analysis, by employing small perturbations as a tool in algorithmic mechanism design. We develop a "duality" between linear perturbations of the objective function of an optimization problem and of its feasible set, and use the "primal" and "dual" viewpoints to prove the running time bound and the truthfulness guarantee, respectively, for our mechanism.**

***Keywords*-Mechanism Design; Truthful Approximation Algorithms; Smoothed Analysis**

I. Introduction

Algorithmic mechanism design studies optimization problems where the underlying data — such as a value of a good or a cost of performing a task — is a priori unknown to the algorithm designer, and must be elicited from self-interested participants (e.g., via a bid). The high-level goal of mechanism design is to design a protocol, or "mechanism", that interacts with participants so that self-interested behavior yields a desirable outcome. *Algorithmic* mechanism design adopts computational tractability as an equally important requirement.

An important research agenda, suggested roughly ten years ago [1], is to understand rigorously what can and cannot be efficiently computed when the problem data is held by selfish agents, thereby reconciling strategic concerns with the computational requirements customary in computer science. The central question in the field is:

> *To what extent is "incentive-compatible" efficient computation fundamentally less powerful than "classical" efficient computation?*

This question remains poorly understood, despite some recent positive results for single-parameter problems[1] and negative results for deterministic mechanisms (discussed further below). A starry-eyed mechanism designer might hope for the best-possible answer:

> *Not at all: If an optimization problem Π admits a polynomial-time α-approximation algorithm $\mathcal{A}$, then it admits a polynomial-time α-approximate incentive-compatible mechanism.*

Since such a result makes no hypotheses about the algorithm $\mathcal{A}$ beyond those on its running time and approximation factor, it would presumably be proved via a "black-box reduction" — a generic method that invokes $\mathcal{A}$ at most polynomially many times, and restores incentive-compatibility without degrading $\mathcal{A}$'s approximation factor.

> *The primary contribution of this paper is the first such black-box reduction for a non-trivial class of multi-parameter problems.*

In this paper, by "incentive compatible" we mean a (possibly randomized) mechanism such that every participant maximizes its expected payoff by truthfully revealing its information to the mechanism, no matter how the other participants behave. Such mechanisms are called *truthful in expectation*, and are defined formally in Section II. Our main result can be summarized as follows.

Main Result (Informal): *If a packing problem Π admits an FPTAS, then it admits a truthful-in-expectation randomized mechanism that is an FPTAS.*

Recall that a *fully polynomial-time approximation scheme (FPTAS)* for a maximization problem takes as

*Supported by NSF Grant CCF-0448664 and a Siebel Foundation Scholarship.
†Supported in part by NSF CAREER Award CCF-0448664, an ONR Young Investigator Award, an ONR PECASE Award, an AFOSR MURI grant, and an Alfred P. Sloan Fellowship.

[1] Informally, a mechanism design problem is *single-parameter* if every participant's utility function can be described naturally using a single real number, and is *multi-parameter* otherwise.

input an instance and an approximation parameter ϵ, and returns a feasible solution with objective function value at least $1 - \epsilon$ times that of an optimal solution, in time polynomial in the size of the instance and in $1/\epsilon$.

Thus the requirement of (randomized) incentive-compatibility *imposes no loss in performance* in packing problems that admit an FPTAS. Our main result is arguably the first to suggest the intriguing possibility of very general black-box (randomized) reductions in algorithmic mechanism design.

A. Executive Summary of Results and Techniques

We follow the most general approach known for designing (randomized) truthful multi-parameter mechanisms, via *maximal-in-distributional range (MIDR)* algorithms [2]. An MIDR algorithm fixes a set of distributions over feasible solutions — the *distributional range* — independently of the reported player utilities, and outputs a random sample from the distribution that maximizes expected welfare. These algorithms are randomized analogues of *maximal-in-range* algorithms (see e.g. [1], [3]). Since the VCG payment scheme renders an MIDR algorithm truthful in expectation, we can focus on the purely algorithmic problem of designing an MIDR FPTAS.

Our primary and most sweeping result concerns *binary packing problems of polynomial dimension*, instances of which are described by a feasible set $\mathcal{S} \subseteq \{0,1\}^d$ and an objective function $v \in \mathbb{R}^d_+$, where d is polynomial in the description of $\mathcal{S}$ and $\mathcal{S}$ is downward-closed (i.e., if $x \in \mathcal{S}$ and $y \leq x$ component-wise, then $y \in \mathcal{S}$). The goal is to maximize $v^T x$ over $x \in \mathcal{S}$. In a mechanism design context, the objective function v is the sum $\sum_i u_i$ of several players' utility functions. (See Sections IV and V for several concrete examples.) Consider such a problem Π that admits an FPTAS, and hence — via a recent result of Röglin and Teng [4] — admits an exact algorithm $\mathcal{A}$ with polynomial smoothed complexity. (See Section II for precise definitions.)

As a naive starting point, suppose we apply a perturbation to a given instance of Π and then invoke the smoothed polynomial-time algorithm $\mathcal{A}$ to compute an optimal solution to the perturbed instance. The good news is that this solution will be near-optimal for the unperturbed instance provided the perturbation is not too large. The two-fold bad news is that an algorithm with smoothed polynomial running time has polynomial expected running time only when the magnitude of perturbations is commensurate with that of the input numbers (to within a polynomial factor, say); and, moreover, exact optimization using perturbed valuations does not generally yield a truthful mechanism. On the first point, simultaneously learning the scale of players' utility functions and using this knowledge to compute an outcome seems incompatible with the design of truthful mechanisms, particularly for multi-parameter problems where essentially only minor variations on the VCG mechanism are known to be truthful. Is there a way to apply truthfully perturbations of the necessary magnitude? Since we use perturbations only as an algorithmic tool internal to our algorithm, we bear no burden of ensuring that the perturbations are "natural" in any sense (unlike in traditional smoothed analysis).

We provide an affirmative answer to the above question by developing a simple "duality theory" for perturbations of the following form: for a random $d \times d$ matrix P and a given objective function v, the perturbed objective function is defined as Pv. We observe that exact maximization of the perturbed objective function Pv over the feasible solutions of an instance is equivalent to exact maximization of the true objective function v over a set of "perturbed solutions" with the "adjoint" perturbation matrix P^T. When P satisfies certain conditions, each such perturbed solution can be expressed as a probability distribution over solutions. In this case, the "adjoint problem" can be solved truthfully via an MIDR algorithm. Moreover, a valuation-independent perturbation of the feasible solutions is necessarily "scale free", and we show that if it designed appropriately, it dualizes to a perturbation of the valuations at the correct scale. Thus the "dual perspective" and the use of perturbed solutions allow us to argue truthfulness for perturbation schemes that seem, at first blush, fundamentally incompatible with truthful mechanisms. Blending these ideas together, we design a perturbation scheme that, in effect, learns the scale of the objective function v and applies perturbations of the appropriate magnitude, thereby obtaining simultaneously expected polynomial running time, an approximation factor of $(1 - \epsilon)$ for arbitrary $\epsilon > 0$, and an MIDR (and hence truthful-in-expectation) implementation.

We also extend our main result in various ways: to binary covering problems in Section V-A; to non-packing binary maximization problems in Section V-B; and to certain problems that do not have polynomial dimension in the full version of the paper (including multi-unit auctions, thereby recovering the main result of [2]).

B. Comparison to Previous Work

There are three known black-box reductions from approximation algorithms to truthful approximation mechanisms for *single-parameter* mechanism design problems, where outcomes can be encoded as vectors in $\mathbb{R}^n$

(where n is the number of players) and the utility of a player i for an outcome x is $u_i x_i$, where u_i is a parameter privately known to i (the value per allocation unit). The space of truthful mechanisms for single-parameter problems is well understood and reasonably forgiving: an approximation algorithm can be used in a truthful mechanism if and only if it is monotone, meaning that the computed allocation x_i for player i is non-decreasing in the reported utility u_i (holding other players' reported utilities fixed). See [5] for precise definitions and many examples of monotone approximation algorithms. The first black-box reduction is due to Briest et al. [6], who proved that every single-parameter binary optimization problem with polynomial dimension that admits an FPTAS also admits a truthful mechanism that is an FPTAS. Their black-box reduction is also deterministic. Second, Babaioff et al. [7] exhibit a black-box reduction that converts an approximation algorithm for a single-parameter problem to a truthful mechanism. However, their reduction degrades the approximation factor by a super-constant factor. Finally, Hartline and Lucier [8] consider the weaker goal of implementation in Bayes-Nash equilibria — as opposed to in dominant strategies, the notion considered here and in most of the algorithmic mechanism design literature — and show that for *every* single-parameter welfare maximization problem, every non-monotone approximation algorithm can be made monotone without degrading the expected approximation factor. All three of these black-box reductions rely heavily on the richness of the monotone algorithm design space, and do not admit obvious extensions to multi-parameter problems.[2]

For multi-parameter problems, the result of Lavi and Swamy [9] is in the spirit of black-box reductions. They show how to convert certain approximation algorithms to truthful in expectation mechanisms without degrading the approximation ratio. However, their result imposes non-trivial extra requirements on the approximation algorithm that is to be converted into a truthful approximation mechanism. For many problems, it is not clear if there are near-optimal approximation algorithms that meet these extra requirements.

On the negative side, there is no general and lossless black-box reduction from approximation algorithms to *deterministic* truthful approximation mechanisms for multi-parameter problems. This fact was first established by Lavi et al. [10], and Papadimitriou et al. [11] gave a quantitatively much stronger version of this lower bound. Additional evidence of the difficulty of multi-parameter mechanism design was provided in [3] and [12], in the context of combinatorial auctions. These negative results do not apply to randomized mechanisms, however, and Dobzinski and Dughmi [2] showed that, for a variant of multi-unit auctions, truthful-in-expectation mechanisms are strictly more powerful than deterministic ones.

Finally, we know of only one previous application of smoothed analysis techniques to the design of new algorithms: Kelner and Spielman [13] used an iterative perturbation approach to design a randomized simplex-type algorithm that has (weakly) polynomial expected running time.

[2]For example, the black-box reduction in [6] uses a simple truncation trick that preserves monotonicity but violates the weak monotonicity condition needed for truthfulness in multi-parameter problems; it also uses a monotonicity-preserving MAX operator to effectively learn the scale of the valuations, which again appears possible only in a single-parameter context.

II. Preliminaries

A. Binary Packing Problems

An instance of a *binary maximization problem* Π is given by a *feasible set* $\mathcal{S}$ encoded — perhaps implicitly — as vectors in $\{0,1\}^d$, as well as a non-negative vector $v \in \mathbb{R}^d_+$ of coefficients. The goal is to compute a feasible solution $x \in \mathcal{S}$ that maximizes the linear objective $v^T x$. Many natural maximization problems are *packing problems*, meaning that if x belongs to the feasible set $\mathcal{S}$ and $y_i \leq x_i$ for all i, then $y \in \mathcal{S}$ as well. (Binary covering problems can be defined analogously; see Section V-A.)

We are interested in binary packing mechanism design problems, where the objective function $v^T x$ is the welfare of self-interested players with private utility functions. Consider a feasible set $\mathcal{S} \subseteq \{0,1\}^d$ and n players, where player i has utility $\sum_{j=1}^d u_{ij} x_j$ for each $x \in \mathcal{S}$. The corresponding welfare maximization problem — computing the outcome x that maximizes the sum of players' utilities — is then the binary maximization problem with $v_j = \sum_{i=1}^n u_{ij}$ for each $j = 1, 2, \ldots, d$. We next give a simple example to make these definitions concrete for the reader; see Sections IV and V for several more examples.

Example II.1 (Multi-Parameter Knapsack) In the *multi-parameter knapsack* problem, there are m projects and n players. Each project j has a publicly known cost s_j, and the feasible sets correspond to subsets of projects that have total cost at most a publicly known budget C. Each player i has a private utility u_{ij} for each project j. Welfare maximization for multi-parameter knapsack instances is a binary packing problem: the feasible set is naturally encoded as the vectors x in $\{0,1\}^m$ with $\sum_j s_j x_j \leq C$, and the coefficient v_j is defined as the total utility $\sum_i u_{ij}$ to all players of selecting the project j.

The binary packing problem in Example II.1 has *polynomial dimension*, meaning that the number d of decision variables has size polynomial in the description of the feasible set. Our most sweeping results (Section IV) are for problems with polynomial dimension, but our techniques also extend to some interesting problems with exponential dimension – we defer details to the full version of the paper.

B. Mechanism Design Basics

We consider direct-revelation mechanisms for binary optimization mechanism design problems. Such a mechanism comprises an *allocation rule*, which is a function from (hopefully truthfully) reported utility functions $u_1, \ldots, u_n$ to an outcome $x \in \mathcal{S}$, and a *payment rule*, which is a function from reported utility functions to a required payment from each player. We allow the allocation and payment rules to be randomized.

A mechanism with allocation and payment rules $\mathcal{A}$ and p is *truthful in expectation* if every player always maximizes its expected payoff by truthfully reporting its utility function, meaning that

$$\mathbf{E}[u_i(\mathcal{A}(u)) - p_i(u)] \geq \mathbf{E}[u_i(\mathcal{A}(u_i', u_{-i})) - p_i(u_i', u_{-i})] \tag{1}$$

for every player i, (true) utility function u_i, (reported) utility function u_i', and (reported) utility functions u_{-i} of the other players. The expectation in (1) is over the coin flips of the mechanism. If the mechanism is deterministic and satisfies this condition, then it is simply called *truthful*.

The mechanisms that we design can be thought of as randomized variations on the classical VCG mechanism, as we explain next. Recall that the *VCG mechanism* is defined by the (generally intractable) allocation rule that selects the welfare-maximizing outcome with respect to the reported utility functions, and the payment rule that charges each player i a bid-independent "pivot term" minus the reported welfare earned by other players in the selected outcome. This (deterministic) mechanism is truthful; see e.g. [14].

Now let $dist(\mathcal{S})$ denote the probability distributions over a feasible set $\mathcal{S}$, and let $\mathcal{R} \subseteq dist(\mathcal{S})$ be a compact subset of them. The corresponding *Maximal in Distribution Range (MIDR)* mechanism has the following (randomized) allocation rule: given reported utility functions $u_1, \ldots, u_n$, return an outcome that is sampled randomly from a distribution $D^* \in \mathcal{R}$ that maximizes the expected welfare $\mathbf{E}_{x\sim D}[\sum_{i,j} u_{ij} x_j]$ over all distributions $D \in \mathcal{R}$. Analogous to the VCG mechanism, there is a (randomized) payment rule that can be coupled with this allocation rule to yield a truthful-in-expectation mechanism (see [2]).

We will need the following fact, that probability distributions over MIDR allocation rules are again MIDR allocation rules.

Lemma II.2 *An allocation rule that chooses an MIDR allocation rule randomly from an arbitrary distribution over such rules is also an MIDR allocation rule.*

Proof: We fix a feasible set $\mathcal{S}$ and consider an allocation rule $\mathcal{A}$ that randomly picks an MIDR allocation rule to run. We assume that $\mathcal{A}$ runs the MIDR allocation rule $\mathcal{A}_k$ with probability p_k, and use $\mathcal{R}_k$ to denote the range of $\mathcal{A}_k$. We let D_k^v be the distribution over outcomes sampled from by $\mathcal{A}_k$ given the valuations v. (As usual, v_j denotes $\sum_i u_{ij}$, where u_i is the private utility function of player i.) By definition, $\mathcal{D}_k^v \in \operatorname{argmax}_{D\in\mathcal{R}_k}\{\mathbf{E}_{x\sim D}[v^T x]\}$. Now, the induced distribution over outcomes in the allocation rule $\mathcal{A}$ for v can be written as $D^v = \sum_k p_k D_k^v$. Similarly, the range of $\mathcal{A}$ is a subset of

$$\mathcal{R} = \left\{ \sum_k p_k D_k : D_k \in \mathcal{R}_k \right\}.$$

Since D_k^v maximizes welfare over all elements of $\mathcal{R}_k$ for every v and k, D^v maximizes expected welfare over $\mathcal{R}$ — and hence also over the range of $\mathcal{A}$ — for every v. Therefore, $\mathcal{A}$ is an MIDR allocation rule. ■

C. Smoothed Complexity Basics

Smoothed complexity was defined by Spielman and Teng [15]; our formalism is similar to that in Beier and Vöcking [16] and Röglin and Teng [4]. A *perturbed instance* of a binary packing problem Π consists of a fixed feasible set $\mathcal{S} \subseteq \{0,1\}^d$ and d random variables $v_1, \ldots, v_d$, where each v_i is drawn independently from a distribution with support in $[0, v_{\max}]$ and a density function that is bounded above everywhere by $\phi/v_{\max}$. The parameter ϕ measures the maximum concentration of the distributions of the v_i's. We say that an algorithm $\mathcal{A}$ for a binary packing problem Π runs in *smoothed polynomial time* if its expected running time is polynomial in the description length of $\mathcal{S}$ and ϕ for every perturbed instance.

Our work relies on the fact that every FPTAS for a binary optimization problem with polynomial dimension can be converted into an algorithm that runs in smoothed polynomial time. This is a special case of a result of Röglin and Teng [4], who strengthen a result of Beier and Vöcking [16].

Proposition II.3 ([16], [4]) *For every FPTAS $\mathcal{F}$ for a binary maximization problem Π of polynomial dimension, there is an exact algorithm $\mathcal{A}^{\mathcal{F}}$ for Π that runs in smoothed polynomial time.*

Moreover, the quite natural algorithm $\mathcal{A}^{\mathcal{F}}$ in Proposition II.3 treats $\mathcal{F}$ as an "oracle" or "black box", meaning that its behavior depends only on the outputs of $\mathcal{F}$ and not on the actual description of $\mathcal{F}$.[3]

III. Perturbation Schemes That Yield Truthful FPTASes

A. Perturbation Schemes

A *perturbation scheme* for a binary packing problem Π is a randomized algorithm Ψ that takes as input an instance $(\mathcal{S}, v)$ of Π and an approximation parameter ϵ and outputs another objective function $\widehat{v} = \Psi(v, \mathcal{S}, \epsilon)$ of the same dimension as v. Such a scheme is *approximation preserving* if for every instance $(\mathcal{S}, v)$ of Π and parameter $\epsilon > 0$,

$$\mathbf{E}[v^T \operatorname*{argmax}_{x \in \mathcal{S}} (\widehat{v}^T x)] \geq (1 - \epsilon) \max_{x \in \mathcal{S}} v^T x,$$

where the expectation is over the random coin flips of the scheme.

B. An Overly Simplistic Approach

Suppose we design an exact algorithm $\mathcal{A}$ and an approximation-preserving perturbation scheme Ψ for a binary packing problem Π such that, for every instance $(\mathcal{S}, v)$ and $\epsilon > 0$, algorithm $\mathcal{A}$ has expected running time polynomial in the instance size and $1/\epsilon$ when the instance is perturbed by Ψ. Then, we immediately get an FPTAS for Π: given an instance of Π and ϵ, use the scheme Ψ to perturb the instance and the algorithm $\mathcal{A}$ to efficiently solve the perturbed instance. Since Ψ is approximation preserving, this algorithm gives a $(1-\epsilon)$-approximation (in expectation).

Can we design such a perturbation scheme so that the resulting FPTAS can be used in a truthful-in-expectation mechanism? We face two quandaries. First, the perturbations have to be at the same "scale" as the largest coefficient $v_{\max}$ of the objective function (recall Section II); but truthfulness seems to preclude explicitly learning and subsequently using this scale in a mechanism. Second, exactly optimizing a randomly perturbed objective function does not generally yield a truthful mechanism. To address both of these issues, we require another idea.

C. Adjoint Perturbations

We now narrow the discussion to *linear* perturbation schemes, where $\Psi(\mathcal{S}, v, \epsilon) = Pv$ for a (random) matrix P whose distribution is independent of v. We next develop a "duality" for such schemes. We will need both the "primal" and "dual" viewpoints to prove the running time bound and the truthfulness guarantee, respectively, of our final mechanism.

Here is a trivial observation: for every fixed perturbation matrix P, objective function v, and feasible solution $x \in \mathcal{S}$, the value $(Pv)^T x$ of the solution x with respect to the perturbed objective Pv equals the value $v^T(P^T x)$ of the "perturbed solution" $P^T x$ with respect to the true objective v. We say that the perturbation P^T is *adjoint* to P. Taking this alternative adjoint viewpoint, solving a linearly perturbed instance $(\mathcal{S}, Pv)$ of a binary packing problem is the same as solving the optimization problem

$$\begin{array}{ll} \text{maximize} & v^T \widetilde{x} \\ \text{subject to} & \widetilde{x} \in P^T \mathcal{S}, \end{array} \quad (2)$$

where $\widetilde{x} = P^T x$ and $P^T \mathcal{S} = \{\widetilde{x} : x \in \mathcal{S}\}$. See Figures 1 and 2 for an illustration of this relationship.

The adjoint problem (2) is meaningful when we can associate every $\widetilde{x} \in P^T \mathcal{S}$ with a probability distribution over the feasible solutions $\mathcal{S}$ that has expectation $\widetilde{x}$. This is possible if and only if $P^T \mathcal{S} \subseteq convexhull(\mathcal{S})$. Assume that we have designed P to possess this property, and for every $x \in \mathcal{S}$ let D_x be an arbitrary distribution over $\mathcal{S}$ with expectation $\widetilde{x} = P^T x$. Let $\mathcal{R} = \{D_x\}_{x \in \mathcal{S}}$ denote the corresponding distributional range. By linearity, the adjoint problem (2) is then equivalent to the problem of maximizing the expected objective function value over $\mathcal{R}$:

$$\begin{array}{ll} \text{maximize} & \mathbf{E}_{y \sim D_x}[v^T y] \\ \text{subject to} & D_x \in \mathcal{R}. \end{array} \quad (3)$$

The key point is that *this is precisely the type of optimization problem that can be solved — truthfully — using an MIDR allocation rule and the corresponding payment rule* (recall Section II).

D. Structure of the Black-Box Reduction

The next theorem formalizes our progress so far: designing truthful-in-expectation mechanisms reduces

[3] The results in [16], [4] are stated as conversions from randomized pseudopolynomial-time algorithms to smoothed polynomial-time algorithms. Proposition II.3 follows since every FPTAS for a binary optimization problem of polynomial dimension can be converted easily to a pseudopolynomial-time exact algorithm in a black-box manner.

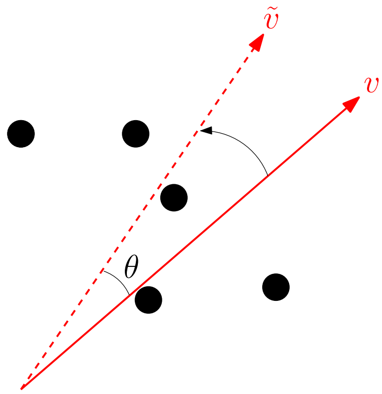

Figure 1. Perturbation P rotates v by an angle θ to $\widetilde{v}$.

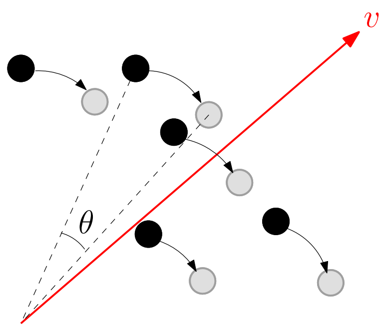

Figure 2. This is the same, relatively speaking, as rotating each feasible solution by an angle of $-\theta$.

to designing perturbation schemes that meet a number of requirements.

For a linear perturbation scheme Ψ for a binary packing problem Π, we say that Ψ is *feasible* if, for every instance $(\mathcal{S}, v)$ of Π, Ψ's random perturbation matrix P satisfies $P^T\mathcal{S} \subseteq convexhull(\mathcal{S})$ with probability 1. Such a scheme is *tractable* if it runs (i.e., outputs the matrix P) in polynomial time; and if for every instance $(\mathcal{S}, v)$, feasible solution $x \in \mathcal{S}$, and possible perturbation matrix P, the distribution D_x with expectation $P^T x$ can be sampled from in polynomial time. A *FLAT* perturbation scheme is one that is feasible, linear, approximation preserving, and tractable. The outline of our black-box reduction is displayed below as Algorithm 1.

Theorem III.1 *For every binary packing problem* Π *and FLAT perturbation scheme* Ψ*, the corresponding perturbation-based (PB) allocation rule (Algorithm 1) satisfies the following properties:*

(a) *it is MIDR and hence defines a truthful-in-expectation mechanism;*
(b) *for every instance of* Π *and* $\epsilon > 0$*, it outputs a feasible solution with expected objective function value at least* $(1-\epsilon)$ *times the maximum possible;*
(c) *its worst-case expected running time is bounded by a polynomial plus that of the exact algorithm* $\mathcal{A}$ *on a perturbed instance* $(\mathcal{S}, Pv)$.

Algorithm 1 Perturbation-Based (PB) Allocation Rule for a Binary Packing Problem Π.

Parameter: Approximation parameter $\epsilon > 0$.
Parameter: Exact algorithm $\mathcal{A}$ for Π.
Parameter: FLAT perturbation scheme Ψ for Π.
Input: Instance $(\mathcal{S}, v)$.
Output: Solution $y \in \mathcal{S}$

1: Draw $P \sim \Psi(\mathcal{S}, \epsilon)$.
2: Let $x = \mathcal{A}(\mathcal{S}, Pv)$.
3: Let D_x be a distribution over $\mathcal{S}$ with expectation $P^T x$, chosen independently of v.
4: Return a sample $y \sim D_x$.

The key point of Theorem III.1 is part (a), which guarantees truthfulness while permitting remarkable freedom in designing perturbation schemes.

Proof of Theorem III.1: First, we note that the choice of P in Step 1 is independent of v by the definition of a linear scheme and Step 3 is well defined because Ψ is feasible. Part (c) follows immediately from the assumption that Ψ is tractable. Part (b) follows from the definition of an approximation-preserving scheme, the fact that $\mathcal{A}$ is an exact algorithm, and the fact that the expected value of the solution y returned by the PB allocation rule equals $\mathbf{E}_{y\sim D_x}[v^T y] = v^T(P^T x) = (Pv)^T x$, which is the objective function value (with respect to the perturbed objective Pv) of the solution returned by $\mathcal{A}$.

To prove part (a), consider an instance $(\mathcal{S}, v)$ and approximation parameter ϵ. To begin, condition on the choice of P by $\Psi(S, \epsilon)$ in Step 1 of the PB allocation rule. Let D_x be the distribution over $\mathcal{S}$ with expectation $P^T x$ that the allocation rule chooses in Step 3 in the event that $x = \mathcal{A}(\mathcal{S}, Pv)$, and set $\mathcal{R} = \{D_x : x \in \mathcal{S}\}$. By the definition of this step, the range $\mathcal{R}$ depends only on $\mathcal{S}$ and ϵ and is independent of the valuations v. Since the allocation rule explicitly computes the solution x^* that maximizes $(Pv)^T x$ and then samples an outcome from the corresponding distribution D_{x^*}, and this x^* is the same solution that maximizes $\mathbf{E}_{y\sim D_x}[v^T y]$ over $x \in \mathcal{S}$ (i.e., over D_x in $\mathcal{R}$), the output of the allocation rule is the same (for each v) as that of the MIDR allocation rule with distributional range $\mathcal{R}$.

We have established that for each fixed choice of P, the PB allocation rule is an MIDR rule. Since the random choice of P is independent of the valuations v, the PB allocation rule is a probability distribution over

MIDR rules. By Lemma II.2, it is an MIDR allocation rule. ■

IV. The Main Result

A. The Random Singleton Scheme

We now describe a FLAT perturbation scheme that leads to our main result: every binary packing problem with polynomial dimension that admits an FPTAS also admits a truthful-in-expectation mechanism that is an FPTAS.

We call our FLAT scheme the *Random Singleton (RS)* perturbation scheme, and we first describe it via its adjoint. Let $(\mathcal{S}, v)$ be an instance of a binary packing problem Π with polynomial dimension, with $\mathcal{S} \subseteq \{0,1\}^d$. Since Π is a packing problem, the all-zero vector lies in $\mathcal{S}$, and we can assume without loss of generality that each basis vector $e_1, \ldots, e_d$ lies in $\mathcal{S}$ (if $e_i \notin \mathcal{S}$ then we can ignore coordinate i). Given $x \in \mathcal{S}$ and a parameter $\epsilon > 0$, we consider the following randomized algorithm:

(1) for each $i = 1, 2, \ldots, d$, draw δ_i uniformly from the interval $[0, \epsilon/d]$;
(2) output a random solution $y \in \mathcal{S}$ according to the following distribution: output the given solution x with probability $1-\epsilon$, the "singleton" e_j with probability $(\sum_{i=1}^d \delta_i x_i)/d$ (for each $j = 1, \ldots, d$); and the all-zero solution with the remaining probability.

The motivation of the random choices in the first step is to ensure that the distribution defined by the perturbation is diffuse enough to permit algorithms with polynomial smoothed complexity (cf., the parameter ϕ in Section II). The motivation of the random choices in the second step is to reward a solution $x \in \mathcal{S}$ with a "bonus" of a random singleton with probability δ_i for each coordinate i with $x_i = 1$. Since there exists a singleton e_j with value v_j that is at least a $1/d$ fraction of the optimal value $\max_{x \in \mathcal{S}} v^T x$, these bonuses effectively ensure that the perturbations occur at the correct "scale."

We now make this vague intuition precise. After conditioning on the random choices in step (1), the expectation $\widetilde{x}$ of the distribution D_x over solutions $\mathcal{S}$ defined by step (2) can be expressed via the adjoint perturbation P^T given by

$$\widetilde{x} = P^T x = (1-\epsilon)x + \left(\sum_{i=1}^{d} \delta_i x_i\right)\left(\sum_{j=1}^{d} \frac{e_j}{d}\right)$$

Let δ denote the d-vector of δ_i's. Since P^T can be written as $(1-\epsilon)I + \frac{1}{d}\vec{\mathbf{1}}\delta^T$, dualizing gives the following formal definition of the RS perturbation scheme for Π, given $(\mathcal{S}, v)$ and ϵ and conditioned on the random choices of the δ_i's:

$$P = (1-\epsilon)I + \frac{\delta\vec{\mathbf{1}}^T}{d}$$

This corresponds to the perturbation

$$v_i \mapsto (1-\epsilon)v_i + \frac{\delta_i}{d}\sum_{j=1}^{d} v_j \qquad (4)$$

for each coefficient i. This perturbation depends on the v_i's and might appear unsuitable for deployment in a truthful mechanism. But its use is justified by our development of adjoint perturbations.

Lemma IV.1 *For every binary packing problem Π of polynomial dimension, the RS perturbation scheme is FLAT.*

Proof: Since the choice of the perturbation matrix P depends only on the feasible set $\mathcal{S}$, the approximation parameter ϵ, and the (valuation-independent) choices of the δ_i's, the RS scheme is linear. It is feasible because it is defined explicitly via the adjoint P^T and the distributions D_x over solutions whose expectations agree with $P^T x$ (for each $x \in \mathcal{S}$). It is clearly tractable. To argue that it is approximation preserving, we observe from equation (4) that for every possible perturbation matrix P and feasible solution $x \in \mathcal{S}$, $(Pv)^T x \geq (1-\epsilon)v^T x$. It follows that, with probability 1 over the choice of P, $\max_{x \in S}(Pv)^T x \geq (1-\epsilon)\max_{x \in \mathcal{S}} v^T x$. ■

B. Putting It All Together

We are now prepared to prove our main result.

Theorem IV.2 (Main Result) *Every binary packing problem of polynomial dimension that admits an FPTAS also admits a truthful-in-expectation mechanism that is an FPTAS.*

Proof: Let Π be a binary packing problem of polynomial dimension and $\mathcal{F}$ and arbitrary FPTAS for it. By Proposition II.3, there is an exact algorithm $\mathcal{A}^{\mathcal{F}}$ for Π that runs in smoothed polynomial time in the sense of Section II. Let Ψ denote the RS perturbation scheme for Π, and instantiate the PB allocation rule with the scheme Ψ and algorithm $\mathcal{A}^{\mathcal{F}}$. Since Ψ is FLAT (Lemma IV.1), Theorem III.1 implies that this allocation rule is MIDR, has an approximation guarantee of $1-\epsilon$ in expectation (for an arbitrary supplied parameter ϵ), and has expected running time bounded by a polynomial plus that of $\mathcal{A}^{\mathcal{F}}$ on the perturbed instance $(\mathcal{S}, Pv)$.

To analyze the expected running time of $\mathcal{A}^{\mathcal{F}}$ on $(\mathcal{S}, Pv)$ and complete the proof, recall the perturbation formula (4). Let $v_{\max}$ denote $\max_{i=1}^{d} v_i$. Every coordinate of Pv is bounded above by $v_{\max}$ with probability 1, and these coordinates are independent random variables (since the δ_i's are independent). Since $\sum_{j=1}^{d} v_j \geq v_{\max}$ and δ_i is drawn uniformly from $[0, \epsilon/d]$, the density of the random variable $(Pv)_i$ is bounded above everywhere by $\frac{d^2}{\epsilon v_{\max}}$. Thus the concentration parameter ϕ from Section II is bounded by d^2/ϵ. Since Π has polynomial dimension and $\mathcal{A}^{\mathcal{F}}$ has polynomial smoothed complexity, the expected running time of $\mathcal{A}^{\mathcal{F}}$ on $(\mathcal{S}, Pv)$ is polynomial in the input size and $1/\epsilon$. ■

C. Examples

We feel that the primary point of Theorem IV.2 is conceptual: it shows that requiring (randomized) incentive-compatibility requires no sacrifice in performance for a non-trivial class of multi-parameter problems, and suggests that even more general "black-box randomized reductions" might be possible. Of course, a general result like Theorem IV.2 can be instantiated for various concrete problems, and we conclude the section by listing a few examples. Numerous single-parameter examples are given in Briest et al. [6]. Below we present some multi-parameter examples, which are beyond the reach of the results in [6].

MULTI-PARAMETER KNAPSACK: From a purely algorithmic perspective, the problem in Example II.1 is equivalent to the Knapsack problem and hence admits a (non-truthful) FPTAS.

ARBORESCENT MULTI-PARAMETER KNAPSACK: This is a generalization of the multi-parameter Knapsack problem, where additional constraints are placed on the feasible solutions $\mathcal{S} \subseteq \{0,1\}^m$. Namely, the projects $[m]$ are the ground set of a laminar[4] set system $\mathcal{L} \subseteq 2^{[m]}$, and there is a budget C_T for each $T \in \mathcal{L}$. The feasible set $\mathcal{S}$ is constrained so that $\sum_{j \in T} s_j \leq C_T$ for each $T \in \mathcal{L}$. A (non-truthful) FPTAS for this problem was given in [17].

TREE-ORDERED MULTI-PARAMETER KNAPSACK: This is another generalization of the multi-parameter Knapsack problem, where precedence constraints are placed on the projects $[m]$. Namely, a directed acyclic graph G with vertices $[m]$ encodes precedence constraints, and the feasible set $\mathcal{S} \subseteq \{0,1\}^m$ is constrained so that $x_j \geq x_k$ whenever $(j,k) \in E(G)$ for every $x \in \mathcal{S}$. When G is a directed-out tree, a (non-truthful) FPTAS for this problem was given in [18]. Observe, however, that this is no longer a binary packing problem. Fortunately, our proof of Theorem IV.2 relied very little on the packing assumption: we argue in Section V-B that we only require $\vec{\mathbf{0}} \in \mathcal{S}$, which is certainly the case here.

MAXIMUM JOB SEQUENCING WITH DEADLINES: In this problem, m jobs are to be scheduled on a single machine. Job $j \in [m]$ has processing time p_j and deadline d_j. There are n players, and player i has private utility u_{ij} for each job j that completes before its deadline d_j. The goal is to find the welfare-maximizing subset of the jobs that can be scheduled so that each finishes before its deadline. Converting such a set of jobs to a schedule can be done via the obvious greedy algorithm. This yields a binary packing problem with a welfare objective. A (non-truthful) FPTAS for this problem was given in [19].

[4] A set system $\mathcal{L} \subseteq 2^{[m]}$ is *laminar* if for each $T, T' \in \mathcal{L}$ either $T \bigcap T' = \emptyset$, or $T \subseteq T'$, or $T' \subseteq T$.

V. EXTENSIONS

A. Binary Covering Problems

We next use the results of Sections III and IV to derive truthful-in-expectation approximation schemes for *binary covering problems* of polynomial dimension. Such problems are defined analogously to binary packing problems, except that the feasible $\mathcal{S}$ is upward closed and the goal is to minimize $v^T x$ over $x \in \mathcal{S}$.

We assume that $v_j = \sum_{i=1}^{n} c_{ij}$ for each $j = 1, 2, \ldots, d$, where c_i denotes the private cost function of player i.

We show how to use Theorem IV.2 to design an "additive FPTAS" for binary covering problems. We will show that this is the best we can hope for by an MIDR mechanism — and MIDR mechanisms are essentially the only general technique we know for designing truthful mechanisms for multi-parameter problems — as no polynomial-time MIDR mechanism obtains a finite approximation of an NP-hard binary covering problem (assuming $P \neq NP$).

Given a binary covering problem Π, we can define the following complementary binary packing problem $\overline{\Pi}$. For $x \in \{0,1\}^d$, let $\overline{x} = \vec{1} - x$. Moreover, for $\mathcal{S} \subseteq \{0,1\}^d$ let $\overline{\mathcal{S}} = \{\overline{x} : x \in \mathcal{S}\}$. $\overline{\Pi} = \left\{(\overline{\mathcal{S}}, v) : (\mathcal{S}, v) \in \Pi\right\}$ is the problem of maximizing $v^T \overline{x}$ for $\overline{x} \in \overline{\mathcal{S}}$. It is easy to see that if $\overline{x}$ is an optimal solution to $\overline{\Pi}$, then $x = \vec{1} - \overline{x}$ is an optimal solution to Π. We use this complementary relationship in both directions: First, observe that a (non-truthful) FPTAS $\mathcal{B}$ for Π can be converted, in a black box fashion, to a (non-truthful) FPTAS $\overline{\mathcal{B}}$ for $\overline{\Pi}$ using the obvious reduction $x \to 1 - \overline{x}$.[5] Applying Theorem IV.2, we derive an MIDR FPTAS $\overline{\mathcal{A}}$ for $\overline{\Pi}$. To establish the following theorem, it remains to show that $\overline{\mathcal{A}}$ can be converted to an MIDR additive FPTAS $\mathcal{A}$ for Π – we defer the simple proof of this fact to the full version of the paper.

Theorem V.1 *Let Π be a binary covering problem of polynomial dimension that admits an FPTAS. There exists an MIDR algorithm $\mathcal{A}$ for Π with approximation parameter ϵ such that the following holds. On input $(\mathcal{S}, v)$, $\mathcal{A}$ runs in expected time polynomial in $1/\epsilon$ and the length of the description of the input, and outputs a solution $x \in \mathcal{S}$ such that*

$$\mathbf{E}[v^T x] \leq \min_{y \in \mathcal{S}} v^T y + \epsilon v_{max}$$

In the above expression, v_{max} denotes $\max_i v_i$, and the expectation is taken over the internal random coins of the algorithm.

The bound in Theorem V.1 becomes an FPTAS in the multiplicative sense for instances where the value of the optimal solution can be bounded below by an inverse polynomial fraction of $v_{\max}$. In general, however, this additive loss is inevitable if we restrict ourselves to MIDR algorithms.

Lemma V.2 *Let Π be a binary minimization problem. If an MIDR algorithm $\mathcal{A}$ provides a finite approximation ratio for Π, then $\mathcal{A}$ is optimal.*

Proof: Assume $\mathcal{A}$ is MIDR, and provides a finite approximation ratio for Π. Fix a feasible set $\mathcal{S}$ of Π, and let $\mathcal{R}$ be the corresponding distributional range of $\mathcal{A}$. We say a feasible solution $x \in \mathcal{S}$ is *minimal* if there does not exist $y \neq x$ in $\mathcal{S}$ with $y_i \leq x_i$ for all i. It is clear that for every objective $v \in \mathbb{R}^d_+$, there exists an optimal solution that is minimal. Since $\mathcal{A}$ is MIDR, it then suffices to show that $\mathcal{R}$ contains all point distributions corresponding to minimal feasible solutions.

Consider a minimal $x \in \mathcal{S}$, and let the objective function v be such that $v_i = 0$ when $x_i = 1$, and $v_i = 1$ when $x_i = 0$. By definition we have $v^T x = 0$. Moreover, since x is minimal, $v^T y > 0$ for every $y \in \mathcal{S}$ with $y \neq x$. Therefore, the only distribution over $\mathcal{S}$ providing a finite approximation ratio for v is the point distribution corresponding to x. Thus, $\mathcal{R}$ contains all point distributions of minimal feasible solutions, as needed. ■

[5]In more detail, let OPT denote the optimal objective function value of the covering problem. Invoking the FPTAS with approximation parameter ϵ/d yields a solution with additive error at most $(\epsilon/d)OPT \leq (\epsilon/d) \cdot dv_{\max} = \epsilon v_{\max}$, where $v_{\max} = \max_{i=1}^n v_i$. Since the optimal objective function value of the complementary packing problem is at least $v_{\max}$, the complement of the computed approximate solution is a $(1-\epsilon)$-approximation for the packing problem.

Our negative result for binary covering problems follows immediately from Lemma V.2.

Theorem V.3 *Let Π be an NP-hard binary minimization problem. No polynomial-time MIDR allocation rule provides a finite approximation ratio for Π unless $P = NP$.*

Theorem V.3 and its proof also extend to the slightly more general class of *distributional affine maximizers* (see [2]), and hence to all known types of VCG-based mechanisms.

Examples: We conclude the section with a few multi-parameter problems to which Theorem V.1, and the complementary negative result in Theorem V.3, apply. Again, for numerous single-parameter examples see Briest et al. [6].

MINIMUM JOB SEQUENCING WITH DEADLINES: This is the minimization variant of Maximum Job Sequencing with Deadlines. Here, player i incurs a cost c_{ij} for every job j that completes past its deadline d_j. The goal is to minimize social cost. This is a binary covering problem. A (non-truthful) FPTAS for this problem was given in [20].

CONSTRAINED SHORTEST PATH: We are given a graph $G = (V, E)$, and two terminals $s, t \in V$. Additionally, there is latency l_j for each $j \in E$. The mechanism is interested in selecting a path from s to t of total latency at most L. There are n players, and player i incurs private cost c_{ij} if $j \in E$ is selected. We consider a covering variant of this problem, where the mechanism may select any subgraph of G connecting s to t via a path of latency at most L, and the goal is to minimize social cost. A (non-truthful) FPTAS for this problem was given in [21].

CONSTRAINED MINIMUM SPANNING TREE ON TREEWIDTH BOUNDED GRAPHS: We are given a graph $G = (V, E)$ with bounded treewidth. Additionally, there is a weight w_j for each $j \in E$. The mechanism is interested in selecting a spanning tree of G with total weight at most W. There are n players, and player i incurs private cost c_{ij} if $j \in E$ is selected. We consider the covering variant of this problem, where the mechanism may select any spanning subgraph of G containing a spanning tree of total weight at most W, and the goal is to minimize social cost. A (non-truthful) FPTAS for this problem was given in [22].

B. Non-Packing Binary Maximization Problems

We observe that the packing assumption of Theorem IV.2 can be relaxed. In particular, if Π is a binary maximization problem, it suffices that $\vec{\mathbf{0}} \in \mathcal{S}$ for every feasible set $\mathcal{S}$ of Π. To see this, observe that the only other property of packing problems that was used in the proof of Theorem IV.2 was that $e_j \in \mathcal{S}$ for each $j = 1, \ldots, d$. It is straightforward to modify the proof to use the following, relaxed, assumption: For each $j = 1, \ldots, d$, there exists $y^j \in \mathcal{S}$ such that $y^j_j = 1$ (and y^j can be identified in polynomial time). Letting $y = \sum_j y^j$, we then modify Ψ as follows: δ is drawn as before, and we let $P = (1-\epsilon)I + \frac{\delta y^T}{d}$. The proof proceeds in a similar fashion. Similarly, Theorem V.1 extends to binary minimization problems where $\vec{\mathbf{1}} \in \mathcal{S}$ for every feasible set $\mathcal{S}$.

C. Beyond Polynomial Dimension

In the full version of the paper, we show that we can sometimes exploit the structure of a problem in order to remove the requirements of polynomial dimension and unconstrained valuations. We adapt our framework to the important problem of welfare maximization in multi-unit auctions, thereby recovering the main result in [2].

Acknowledgments

We thank Amos Fiat and Brendan Lucier for helpful discussions and comments.

References

[1] N. Nisan and A. Ronen, "Algorithmic mechanism design," in *STOC*, 1999.

[2] S. Dobzinski and S. Dughmi, "On the power of randomization in algorithmic mechanism design," in FOCS'09.

[3] S. Dobzinski and N. Nisan, "Limitations of vcg-based mechanisms," in STOC'07.

[4] H. Röglin and S.-H. Teng, "Smoothed analysis of multiobjective optimization," in *FOCS*, 2009, pp. 681–690.

[5] R. Lavi, "Computationally efficient approximation mechanisms," in Algorithmic Game Theory, edited by Noam Nisan and Tim Roughgarden and Eva Tardos and Vijay Vazirani.

[6] P. Briest, P. Krysta, and B. Vöcking, "Approximation techniques for utilitarian mechanism design." in *STOC*, 2005.

[7] M. Babaioff, R. Lavi, and E. Pavlov, "Single-value combinatorial auctions and algorithmic implementation in undominated strategies," *J. ACM*, vol. 56, no. 1, 2009.

[8] J. D. Hartline and B. Lucier, "Bayesian algorithmic mechanism design," in *STOC*, 2010.

[9] R. Lavi and C. Swamy, "Truthful and near-optimal mechanism design via linear programming," in FOCS 2005.

[10] R. Lavi, A. Mu'alem, and N. Nisan, "Towards a characterization of truthful combinatorial auctions," in *FOCS'03*.

[11] C. Papadimitriou, M. Schapira, and Y. Singer, "On the hardness of being truthful," in *FOCS*, 2008.

[12] D. Buchfuhrer, S. Dughmi, H. Fu, R. Kleinberg, E. Mossel, C. Papadimitriou, M. Schapira, Y. Singer, and C. Umans, "Inapproximability for vcg-based combinatorial auctions," in SODA'10.

[13] J. A. Kelner and D. A. Spielman, "A randomized polynomial-time simplex algorithm for linear programming," in *STOC*, 2006, pp. 51–60.

[14] N. Nisan, 2007, introduction to Mechanism Design (for Computer Scientists). In "Algorithmic Game Theory", N. Nisan, T. Roughgarden, E. Tardos and V. Vazirani, editors.

[15] D. A. Spielman and S.-H. Teng, "Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time," in *STOC*, 2001, pp. 296–305.

[16] R. Beier and B. Vöcking, "Typical properties of winners and losers in discrete optimization," *SIAM J. Comput.*, vol. 35, no. 4, pp. 855–881, 2006.

[17] G. V. Gens and E. V. Levner, "Fast approximation algorithms for knapsack type problems," in *Optimization Techniques*, 1980, pp. 185–194.

[18] D. S. Johnson and K. A. Niemi, "On knapsacks, partitions, and a new dynamic programming technique for trees," *Mathematics of Operations Research*, vol. 8, no. 1, pp. 1–14, 1983.

[19] S. Sahni, "General techniques for combinatorial approximation," *Operations Research*, vol. 25, no. 6, pp. 920–936, 1977.

[20] G. V. Gens and E. V. Levner, "Approximate algorithms for certain universal problems in scheduling theory," in *Soviet Journal of Computers and System System Sciences*, 1978.

[21] R. Hassin, "Approximation schemes for the restricted shortest path problem," *Math. Oper. Res.*, vol. 17, no. 1, pp. 36–42, 1992.

[22] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, "Bicriteria network design problems,," *Journal of Algorithms*, vol. 28, no. 1, pp. 142 – 171, 1998.