

Stochastic Packing-Market Planning

Daniel Golovin
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA
dgolovin@cs.cmu.edu

ABSTRACT

Motivated by the problem of centralized market clearing in a market with probabilistic supply and demand, we introduce the Stochastic Packing-Market Planning problem (SPMP), which is a stochastic generalization of the Maximum k -Set Packing problem. We provide an $O(k)$ approximation algorithm for SPMP, as well as a $O(k)$ approximation mechanism that is truthful in expectation. This matches up to constants the best approximation ratio known for Maximum k -Set Packing. Along the way, we develop techniques for obtaining sparse subhypergraphs of an input hypergraph that preserves $\mathcal{E}_{\mathcal{P}}(\cdot)$ up to an $O(k)$ factor, where $\mathcal{E}_{\mathcal{P}}(G)$ measures the expected weight of a maximum weight set packing in a random subhypergraph of G . We also give a linear programming based approximation for $\mathcal{E}_{\mathcal{P}}(G)$. These techniques may be of independent interest.

Categories and Subject Descriptors

K.4.4 [Electronic Commerce]: Payment schemes; F.2.0 [Analysis of Algorithms and Problem Complexity]: General

General Terms

Algorithms, Theory, Economics

Keywords

Approximation Algorithms, Stochastic Optimization, Combinatorial Auctions, Mechanism Design, Probabilistic Demand

1. INTRODUCTION

For centuries, people have argued about the comparative merits of various economic systems. The resulting economic philosophies profoundly influenced the course of world history, and continue to do so. To a large extent, the world has rejected the policy of strictly centrally planned economies, and instead adopted a regime of free trade and largely unfettered markets. It is perhaps ironic then that research in planned markets is undergoing something of a renaissance, fueled by the tremendous growth in computing power and

emerging research in algorithmic mechanism design and auction theory. It is now widely recognized that when different agents in a market have different complementarities and/or substitution effects among items, there may be opportunities to exploit these to increase economic efficiency over what decentralized market mechanisms and single item auctions would give. In lieu of the above, the study of *combinatorial auctions* and *combinatorial exchanges* was initiated. In a combinatorial auction, buyers bid on subsets of goods offered up for sale by a single seller, whereas in combinatorial exchanges (also known as *double auctions*) multiple buyers and sellers may bid on subsets of items. A buyer's bid is interpreted as usual, whereas a seller's bid on a subset of items, called an *ask*, informs the mechanism of a desire to sell that subset of items at the asking price or higher. It has been demonstrated that cleverly designed centralized mechanisms can yield greater social surpluses than other previous mechanisms, both in theory and in practice (see [4, 5, 16, 19, 21] for broad treatments).

A key difficulty for the mechanism designer is how to decide which bids should win in an auction and which exchanges should be made in an exchange market. The well-studied Set Packing problem represents this problem neatly, and provides a clean abstraction.

In this paper we investigate special exchange markets, which we call *Stochastic Packing Markets* (SPM), that model uncertainty in the available supply and demand of items. We then model the problem of market clearing in such markets as a generalization of Set Packing, and investigate how to approximately solve it. Before we formally define an SPM, we will illustrate a potential SPM using an example from the airline industry. Suppose we wish to build a system to fill what would otherwise be empty seats on various flights. We solicit bids for flights to various locations on various dates. Ideally, our customers will exhibit different substitution effects among flights, and different complementarities between departing and returning flights. Airlines ask the selling price for each seat on each of their flights (which may vary by seat). This induces a set of potential deals between customers and airlines, each of which has a payoff if it is made, which may be the surplus of the deal or any other value. We let $w(S)$ denote the payoff of deal S . We then select a set of deals between customers and airlines and *notify* them of the deals. However, we do not commit the parties of each deal but make their commitment to any particular deal *voluntary*. This is done both to encourage user participation in the system, and to exploit any inherently probabilistic demand among customers. For an example of probabilistic demand, we can imagine a customer who receives vacation time probabilistically, and will commit to a deal only if the flights occur during vacation time. Thus we imagine each deal S is committed to by all relevant parties with some probability $p(S)$, which we can accurately estimate. Among the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

EC'07, June 13–16, 2007, San Diego, California, USA.

Copyright 2007 ACM 978-1-59593-653-0/07/0006 ...\$5.00.

deals that are committed to, we select a feasible subset which are made. From the customer’s perspective, there is a sequence of deals over time, and the customer need only commit to some small percentage of them. Now, given these probabilities $p(S)$, it will likely make sense to overbook flights in the notification stage. This will increase the chance that those seats are occupied, but may result in customers committing to a deal and then not being able to make it since their seat was given to someone else. We call such a deal, which is committed to by the customer but not made, a *lost deal*. We associate a cost, $c(S)$, with each lost deal S , which we can use to offset the disappointment of a customer who participates in the lost deal. Our system must then determine which deals to notify, wait to receive commitments from the agents, and finally decide which deals will be executed based on the received commitments. The goal will be to maximize the payoff that we receive from executed deals minus the costs we pay for lost deals.

We are now ready to formally define the Stochastic Packing-Market Problem. In our model, there is a ground set V of elements corresponding to demand and supply for various goods and services. There is also a set \mathcal{D} of potential deals that may be made: each potential deal is represented by a subset $S \subseteq V$. Each potential deal $S \in \mathcal{D}$ has a probability $p(S)$ of being realized, a *payoff* $w(S)$, and a *compensation* cost $c(S)$. The compensation cost models the cost due to lost deals, and we will assume that $c(S) \leq w(S)$ for all S . The output consists of a collection of sets $\mathcal{D}' \subseteq \mathcal{D}$. The objective is to maximize the expected profit of the following random process:

1. Construct a random subset \mathcal{F} of \mathcal{D}' by including each $S \in \mathcal{D}'$ in \mathcal{F} independently with probability $p(S)$.
2. Compute an (approximate) set packing on the sets of \mathcal{F} (i.e., find some $\mathcal{F}' \subseteq \mathcal{F}$ such that the sets of \mathcal{F}' are disjoint).
3. Receive profit $\sum_{S \in \mathcal{F}'} w(S) - \sum_{S \in \mathcal{F} \setminus \mathcal{F}'} c(S)$

We determine which deals to notify (denoted by \mathcal{D}'), as well as which deals to execute (denoted by \mathcal{F}'), but not which of the notified deals arrive (denoted by \mathcal{F}).

Our Contributions.

We introduce the Stochastic Packing-Market Planning (SPMP) problem to represent the problem of market clearing in a market with probabilistic supply and demand. We present an $O(k)$ approximation algorithm for SPMP, where k is the maximum size of any set in the collection of potential deals. This is within a constant factor of the best known algorithms for k -Set Packing [15, 2], which SPMP generalizes. We also devise a randomized mechanism for the SPMP problem that $O(k)$ approximates the social surplus and is truthful in expectation, assuming the compensation costs are not paid to the agents, but rather to some third party. In the course of our search for a good SPMP algorithm, we develop techniques for obtaining sparse subhypergraphs of an input hypergraph that preserves $\mathcal{E}_{\mathcal{P}}(\cdot)$ up to an $O(k)$ factor, where $\mathcal{E}_{\mathcal{P}}(G)$ is the expected weight of a maximum weight matching (aka set packing) in a random subhypergraph of G . We also give a linear programming based approximation for $\mathcal{E}_{\mathcal{P}}(G)$.

1.1 Related Work.

The Maximum Weight Set Packing problem is to find a maximum weight disjoint collection of sets in a set system. Formally, it can be defined as follows. Given a collection of sets \mathcal{D} over a ground set V , and a weight $w(S)$ for each set $S \in \mathcal{D}$, find a collection $\mathcal{D}' \subseteq \mathcal{D}$ maximizing $\sum_{S \in \mathcal{D}'} w(S)$ such that $S_i \cap S_j = \emptyset$

for all $S_i, S_j \in \mathcal{D}'$. If all sets $S \in \mathcal{D}$ have $|S| \leq k$, we say (V, \mathcal{D}, w) is an instance of Maximum k -Set Packing. Set Packing has been studied extensively. In the latest of a sequence of papers, Boppana and Halldórsson give an $O(|\mathcal{D}|/\log^2 |\mathcal{D}|)$ -approximation algorithm [1] and Halldórsson gives a $O(\sqrt{|V|})$ -approximation algorithm [12], but Håstad [13] has proved that unless $\mathbf{NP} \subseteq \mathbf{ZPP}$ it is hard to approximate within $O(|\mathcal{D}|^{1-\epsilon})$ for any $\epsilon > 0$, and Halldórsson, Kratochvíl, and Telle [11] build upon this work to show that unless $\mathbf{NP} \subseteq \mathbf{ZPP}$ it is hard to approximate within $O(|V|^{1/2-\epsilon})$ for any $\epsilon > 0$. For the k -Set Packing variant, Hurkens and Schrijver [15] give a $k/2$ -approximation for the unweighted case, Chandra and Halldórsson [2] give a $2(k+1)/3$ -approximation for the weighted case, and Hazan, Safra, and Schwartz [14] have recently proved that unless $\mathbf{P} = \mathbf{NP}$ it is hard to approximate within $O(k/\log k)$, even in the unweighted case.

The literature on combinatorial auctions and combinatorial exchanges is very extensive, and spans the fields of economics, operations research, computer science and even physics [10]. For a comprehensive introduction, we refer the reader to the excellent surveys of de Vries and Vohra [5], Pecek and Rothkopf [21], Kalagnanam and Parkes [16], and Narahari and Dayama [19], as well as the recently published book *Combinatorial Auctions* [4] edited by Crampton, Shoham, and Steinberg. Within the combinatorial auctions and exchanges literature, approximation algorithms play an important role in the quest for computationally efficient mechanisms. Lehmann, O’Callaghan and Shoham [18] investigated the effect on truthfulness by replacing exact algorithms with approximation algorithms, demonstrated that using approximation algorithms may render the mechanism non-truthful, and then proceeded to design a truthful, efficiently computable mechanism for single-minded bidders (i.e., bidders that have a positive bid for only one set of items) that achieves an approximation guarantee on social welfare of $O(\sqrt{m})$ where there are m items. This is nearly the best possible guarantee [11], since there is a lower bound of $\Omega(m^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$, assuming $\mathbf{NP} \not\subseteq \mathbf{ZPP}$. Nisan and Ronen [20] considered approximation algorithms for Vickrey-Clarke-Groves (VCG) mechanisms, introduced the concept of “feasible truthfulness,” and gave a way to convert any approximation algorithm for a VCG mechanism into one which is “feasibly truthful.” (A mechanism is feasibly truthful if agents that are computationally bounded in a particular, reasonable way cannot compute a better strategy than the truthful one.) Lavi and Swamy [17] gave a general procedure for converting certain linear programming based approximation algorithms into randomized mechanisms that are truthful in expectation (that is, an agent who seeks to maximize expected utility has no reason to deviate from the truthful strategy). More recently, these results have been improved by Dobzinski, Nisan, and Schapira [7] and Feige [8].

Previous papers on Stochastic Packing problems such as [6] consider a different model than ours, in which the goal is to find a maximum 0/1 vector x satisfying $Ax \leq b$ where the algorithm is given distributions for each column of A and sequentially selects columns i to include in the solution (i.e., by setting $x_i = 1$). The actual column values are sampled immediately after the algorithm chooses to include it. Our model is closer in spirit to the two-stage stochastic optimization problems with recourse. In this model, the algorithm is given some distributional information about the future, and some actions it can take in the *initial stage* to prepare. After these initial actions are taken, the future is sampled and the algorithm is allowed to take further actions during this *recourse stage* in order to address constraints it has not satisfied. The objective is then to minimize the total expected cost over all possible futures. An example is the Stochastic Set Cover problem, in which a set

system, a distribution over subsets of the ground set, and initial stage and recourse stage prices for each set are given. In the initial stage the algorithm can buy sets at initial stage prices. Then a subset U of the ground set is sampled from the distribution. Finally, in the recourse stage, the algorithm buys sets at recourse stage prices until all the sets purchased in both stages cover U . We refer the reader to the survey of Swamy and Shmoys [23] for results in this model. Of note is that in all the problems considered in this model, actions taken in the initial stage do not constrain the algorithm's choice of action in the recourse stage. That is, problems in this model tend to have covering constraints. Our problem represents a 2-stage stochastic optimization problem with *packing* constraints: In the initial stage, the algorithm selects a subset of the packing problem (e.g., potential deals to notify), then a packing problem arrives probabilistically based on that subset, and finally, during the 2nd-stage, the algorithm solves the packing problem as best it can.

Notations and Conventions.

Throughout this paper, V will denote the ground set of a set system, and k will denote the maximum size of any set in the set system under consideration. We will sometimes find it convenient to represent set systems as hypergraphs (V, E) , where each hyperedge $e \in E$ is a subset of V . Given a vector x with coordinates indexed by a set A , and a set $B \subseteq A$, we will write $x(B)$ as shorthand for $\sum_{b \in B} x(b)$. For any $\mathcal{D}' \subseteq \mathcal{D}$ let $\mathcal{E}_{\mathcal{P}}(\mathcal{D}')$ equal the expected weight of the maximum weight packing in a random set system (V, \mathcal{D}'') , where each $S \in \mathcal{D}''$ is included in \mathcal{D}'' independently with probability $p(S)$. Given a set system (V, \mathcal{D}) and $S \in \mathcal{D}$, we let $\Gamma(S) := \{S' \mid S' \in \mathcal{D}, S' \cap S \neq \emptyset\}$ be the sets of \mathcal{D} intersecting S . For $v \in V$ we let $\mathcal{D}_v := \{S \mid S \in \mathcal{D}, v \in S\}$ be the set of deals that include v . We will recall these definitions as we use them.

2. EXPECTED PACKING WEIGHT

When designing an algorithm for the SPMP problem, one of the first difficulties we encounter is how to derive a good upper bound on the value of the optimal solution. Even setting the issue of compensation costs aside, deriving tight bounds on the expected payoffs the optimal solution receives appears difficult. In the case that $p(S) = 1$ for all S , the problem is conventional k -Set Packing, so we cannot expect do better than $\Omega(k/\log k)$. Even for $k = 3$, computing this quantity is **NP**-hard via the trivial reduction to 3-Dimensional Matching; in fact for $k \geq 3$ it is known to be **NP**-hard to approximate within some constant greater than one [14]. And all this is before the addition of randomness. Consider that if $k = 2$ so that the set system is a graph G , and $p(e) = p$ for all edges e , it is still unclear how to efficiently compute $\mathcal{E}_{\mathcal{P}}(G)$, which equals the expected cost of a maximum weight matching in a random subgraph of G . It may well be **#P**-hard to do so, though we leave that as an open problem.

Difficulties aside, in this section we will be concerned with estimating $\mathcal{E}_{\mathcal{P}}(G)$ for an arbitrary hypergraph G . Whereas the random graph theory literature typically strives for asymptotically tight results on quantities such as $\mathcal{E}_{\mathcal{P}}(G)$ for certain special instances [9], we will have to content ourselves with the derivation of a polynomial time $O(k)$ approximation. At this point it is worth noting that while random sampling and applying an approximation algorithm for conventional set packing to the samples would also yield an $O(k)$ approximate estimate, it does not provide the analytic framework which we need.

Now, given a set system (V, \mathcal{D}) , with probabilities $p(S)$ and payoffs $w(S)$ for all $S \in \mathcal{D}$, we write down the following linear program. Recall $\mathcal{D}_v := \{S \in \mathcal{D} \mid v \in S\}$ and $x(\mathcal{D}_v) := \sum_{S \in \mathcal{D}_v} x(S)$.

$$\begin{aligned} & \text{maximize} && \sum_{S \in \mathcal{D}} w(S)x(S) \\ & \text{subject to} && 0 \leq x(S) \leq p(S) \quad \forall S \in \mathcal{D} \\ & && x(\mathcal{D}_v) \leq 1 \quad \forall v \in V \end{aligned} \quad (2.1)$$

THEOREM 2.1. *Let x be an optimal solution to the above LP and let $Z^* = \sum_{S \in \mathcal{D}} w(S)x(S)$ be the value it obtains. Then $Z^*/4k \leq \mathcal{E}_{\mathcal{P}}(\mathcal{D}) \leq Z^*$*

PROOF. To prove $\mathcal{E}_{\mathcal{P}}(\mathcal{D}) \leq Z^*$ fix any algorithm \mathcal{A} that computes an optimal maximum weight set packing. Let $\mathcal{A}(\mathcal{D}')$ denote the output of \mathcal{A} on input \mathcal{D}' . Now define $y(S) := \Pr[S \in \mathcal{A}(\mathcal{D}')$, where the distribution is over the sampled set packing instance $\mathcal{D}' \subseteq \mathcal{D}$. It is easy to see that y is feasible for the LP. Furthermore,

$$\begin{aligned} \mathcal{E}_{\mathcal{P}}(\mathcal{D}) &= \sum_{\mathcal{D}' \subseteq \mathcal{D}} \Pr[\mathcal{D}'] \cdot \sum_{S \in \mathcal{A}(\mathcal{D}')} w(S) \\ &= \sum_{S \in \mathcal{D}} w(S) \cdot \sum_{\mathcal{D}' \subseteq \mathcal{D}: S \in \mathcal{A}(\mathcal{D}')} \Pr[\mathcal{D}'] \\ &= \sum_{S \in \mathcal{D}} w(S) \cdot \Pr[S \in \mathcal{A}(\mathcal{D}')] \\ &= \sum_{S \in \mathcal{D}} w(S)y(S) \end{aligned}$$

Thus $\mathcal{E}_{\mathcal{P}}(\mathcal{D}) \leq Z^*$. We show that $Z^*/4k \leq \mathcal{E}_{\mathcal{P}}(\mathcal{D})$ by rounding the fractional LP solution x as follows. First, *mark* each set S independently with probability $x(S)/kp(S)$. Each set S then appears with probability $p(S)$. We call a set that appears and is marked an *active* set. Our set packing P will consist of all active sets that do not intersect other active sets. Recall $\Gamma(S) := \{S' \mid S' \in \mathcal{D}, S' \cap S \neq \emptyset\}$. For each $S \in \mathcal{D}$ we have $\Pr[S \text{ active}] = x(S)/k$ and

$$\Pr[S \in P] = (x(S)/k) \cdot \prod_{S' \in \Gamma(S)} (1 - x(S')/k) \quad (2.2)$$

$$\geq (x(S)/k) \cdot 4^{-\sum_{S' \in \Gamma(S)} x(S')/k} \quad (2.3)$$

Here we have used $(1 - x) \geq 4^{-x}$ for all $x \in [0, 1/2]$ and $k \geq 2$ for nontrivial instances. Since x is feasible for the LP, $\sum_{S' \in \Gamma(S)} x(S') \leq \sum_{v \in S} x(\mathcal{D}_v) \leq |S|$, and so $\Pr[S \in P] \geq x(S)/4k$. Thus $\mathbf{E}[w(P)] \geq \sum_S w(S)x(S)/4k = Z^*/4k$, and clearly $\mathcal{E}_{\mathcal{P}}(\mathcal{D}) \geq \mathbf{E}[w(P)]$. \square

3. SPARSIFICATION RESULTS

Sparsification of a graph typically involves removing many (often most) of its edges, while attempting to (approximately) preserve some feature(s) of the original such as connectivity, expansion, etc. In this paper, we are concerned with preserving the value of $\mathcal{E}_{\mathcal{P}}(\cdot)$. Positive sparsification results along these lines would indicate that we can ignore many or most of the potential deals, while approximately preserving the value of the deals that are executed. In other words, sparsification results indicate a way to dramatically reduce the cost of lost deals while only slightly reducing the payoff from executed deals. In light of the above, Theorem 2.1 has some interesting corollaries. For example, suppose we are in the graph case, i.e., $k = 2$, and further suppose the input graph $G = (V, E)$ is bipartite and the probability that each edge appears is uniform, so that $p(e) = p$ for all edges e . We let $w(e)$ be arbitrary. For simplicity, assume $1/p$ is an integer – this assumption is fairly easy to

eliminate at some small cost to the approximation factor. We can scale the x values by $1/p$ and, letting $\partial v := \{e \mid e \text{ incident on } v\}$, rewrite LP (2.1) as

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} p \cdot w(e)x(e) \\ & \text{subject to} && 0 \leq x(e) \leq 1 && \forall e \in E \\ & && x(\partial v) \leq 1/p && \forall v \in V \end{aligned} \quad (3.4)$$

If input graph is bipartite, the constraint matrix for this LP is known to be *totally unimodular* (see e.g., §18.2 of [22]). Thus this LP is integral. (In fact we can find an integral optimal solution combinatorially via a maximum matching algorithm running on an appropriately defined graph built from the input graph.) Now find an optimal integral solution y and let $H := \{e \mid y(e) = 1\}$. Let $Z^*(G)$ be the optimal LP value using the edges in G . Then $Z^*(H) = Z^*(G)$. Thus using Theorem 2.1 we can prove the following corollary.

COROLLARY 3.1. *Given a bipartite graph $G = (V, E)$, probability p such that each edge appears independently with probability p , and edge weights $w : E \rightarrow \mathbb{R}_{\geq 0}$, we can construct a subgraph H of G of maximum degree $\lceil 1/p \rceil$ in polynomial time such that $\mathcal{E}_{\mathcal{P}}(H) \geq \mathcal{E}_{\mathcal{P}}(G)/8$.*

In the extreme case, we can imagine $G = (V, E)$ is the complete bipartite graph $K_{n,n}$, and imagine $p = \epsilon$ is some small constant. Corollary 3.1 then says that we can reduce the number of edges by a factor of about ϵn , while only losing a factor of at most 8 in $\mathcal{E}_{\mathcal{P}}(G)$. Incidentally, the constant 8 is not tight, and a more careful application of our techniques will slightly improve it, however if G is a star of infinite degree and as p approaches zero, we see that this constant cannot be improved beyond $e/(e-1)$. Looking at Corollary 3.1, we can see that there is nothing particularly special about $\lceil 1/p \rceil$, and in fact we can push this line of reasoning a bit further.

COROLLARY 3.2. *Given a bipartite graph $G = (V, E)$, probability p such that each edge appears independently with probability p , and edge weights $w : E \rightarrow \mathbb{R}_{\geq 0}$, then for any $c \in (0, 1]$ we can construct a subgraph H of G of maximum degree $\lceil c/p \rceil$ in polynomial time such that*

$$\mathcal{E}_{\mathcal{P}}(H) \geq \frac{\mathcal{E}_{\mathcal{P}}(G)}{8} \cdot \frac{\lceil c/p \rceil}{\lceil 1/p \rceil}$$

This follows from the fact that we can replace the LP constraints $x(\partial v) \leq \lceil 1/p \rceil$ with $x(\partial v) \leq \lceil c/p \rceil$ while losing only a factor of $\lceil c/p \rceil / \lceil 1/p \rceil$ in the optimal value of the LP. Total unimodularity ensures that this change does not alter the integrality of the LP. In general, these results will hold for any instances of k -Set Packing whose constraint matrices are totally unimodular, however we will lose a factor of k in the results and obtain a subhypergraph H with degree at most $\lceil c/p \rceil$ (i.e., each ground element is contained in at most $\lceil c/p \rceil$ hyperedges) such that

$$\mathcal{E}_{\mathcal{P}}(H) \geq \frac{\mathcal{E}_{\mathcal{P}}(G)}{4k} \cdot \frac{\lceil c/p \rceil}{\lceil 1/p \rceil}$$

For problems with constraint matrices that are not totally unimodular, we note that so long as we can round an optimal fractional LP solution x to an integral solution y while losing a factor of α , if we again define $H := \{e \mid y(e) = 1\}$ then $Z^*(H) \geq Z^*(G)/\alpha$, where G is the hypergraph that represents the Set Packing instance. Thus we obtain the following

COROLLARY 3.3. *Given a k -Set Packing instance represented as hypergraph $G = (V, \mathcal{D})$, weights $w : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ and a probability $p \in [0, 1]$ such that each hyperedge appears independently with probability p , and given an algorithm that efficiently rounds a fractional solution to the LP described in (3.4) defined on G to an integral solution while losing a factor of at most α in the objective, we can efficiently construct a subhypergraph H of G of maximum degree $\lceil 1/p \rceil$ such that*

$$\mathcal{E}_{\mathcal{P}}(H) \geq \frac{\mathcal{E}_{\mathcal{P}}(G)}{4\alpha k}$$

For example, if $k = 2$ and G is an arbitrary, possibly non-bipartite graph, then $\alpha \leq 3/2$ and we can obtain a subgraph H of G of maximum degree $\lceil 1/p \rceil$ such that $\mathcal{E}_{\mathcal{P}}(H) \geq \mathcal{E}_{\mathcal{P}}(G)/12$.

3.1 Easier Instances and the Compression Gap

Some real world instances of k -Set Packing in combinatorial auctions may be significantly easier than the worst case [4]. In this section we will thus consider the problem of finding a sparse subhypergraph H of G that approximates G well with respect to $\mathcal{E}_{\mathcal{P}}(\cdot)$ with particular emphasis on “easier” instances of Set Packing whose linear programming relations have small *integrality gaps* (e.g., totally unimodular instances). The integrality gap of an IP formulation of a maximization problem is the worst case (i.e., maximum) ratio between the optimum value of the LP relaxation of the IP and the optimum value of the IP itself.

Consider a class of k -Set Packing instances with integrality gap α , such that we can efficiently round any fractional solution x to their LP relaxations to an integral solution of value at least $1/\alpha$ times the value of x . We require that this class be closed under the operation of removing sets from the instance.

Now, given an SPM as a hypergraph $G = (V, E)$, $p : E \rightarrow [0, 1]$, and $w : E \rightarrow \mathbb{R}_{\geq 0}$ consider the following LP, which we explain below.

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} w(e)z(e) \\ & \text{subject to} && z(e) = \sum_{G' \subseteq G} \Pr[G'] \cdot y(G', e) && \forall e \in E \\ & && 0 \leq y(G', e) \leq 1 && \forall G' \subseteq G, e \in E \\ & && \sum_{e: e \in G', v \in e} y(G', e) \leq 1 && \forall G' \subseteq G, v \in V \end{aligned} \quad (3.5)$$

In LP (3.5) above, there are variables $y(G', e)$ which indicate the extent to which hyperedge e is selected when G' is exactly the set of hyperedges that appears. The z variables are there only for simplicity of exposition. $\Pr[G']$ is the probability that exactly the hyperedges in G' appear, that is $\Pr[G'] = \prod_{e \in G'} p(e) \cdot \prod_{e \notin G'} (1 - p(e))$. Let IP (3.5) indicate the IP where we have replaced constraints of the form $0 \leq y(G', e) \leq 1$ with constraints of the form $y(G', e) \in \{0, 1\}$. Note that the optimal value of IP (3.5) is exactly $\mathcal{E}_{\mathcal{P}}(G)$. Now consider the following procedure for sparsifying a hypergraph:

1. Solve LP (2.1) optimally and round the resulting (fractional) solution x to a solution x' such that $x'(e) \in \{0, p(e)\}$ while losing a factor of α for as small an α as possible. Let $H := \{e \mid x'(e) = p(e)\}$.
2. Construct a fractional solution (y, z) that is feasible for LP (3.5) using only hyperedges in H (that is $x'(e) = 0$ implies $y(G', e) = 0$ for all G') such that $\sum_e w(e)z(e) \geq \frac{1}{\lambda} \sum_e w(e)x'(e)$ for as small a λ as possible.
3. Round (y, z) to a feasible solution (y', z') for IP (3.5) such that $e \notin H$ implies $y'(G', e) = 0$ for all G' , while losing a factor of α' for as small an α' as possible.

Only step 1 needs to be executed to find a sparse subhypergraph. Steps 2 and 3 are only for purposes of proof, and need not be efficient at all.

FACT 3.4. *If the above procedure is possible for some α, λ , and α' , then $\mathcal{E}_{\mathcal{P}}(H) \geq \mathcal{E}_{\mathcal{P}}(G)/\alpha\lambda\alpha'$.*

PROOF. Consider the following chain of inequalities.

$$\begin{aligned} \mathcal{E}_{\mathcal{P}}(H) &\geq \sum_e w(e)z'(e) \\ &\geq \frac{1}{\alpha'} \sum_e w(e)z(e) \\ &\geq \frac{1}{\lambda\alpha'} \sum_e w(e)x'(e) \\ &\geq \frac{1}{\alpha\lambda\alpha'} \sum_e w(e)x(e) \\ &\geq \frac{\mathcal{E}_{\mathcal{P}}(G)}{\alpha\lambda\alpha'} \end{aligned}$$

The first inequality is true because $\mathcal{E}_{\mathcal{P}}(H)$ is the value of the optimal solution to IP (3.5) under the additional constraints that $e \notin H$ implies $y(G', e) = 0$ for all G' , and (y', z') is feasible for IP (3.5) and respects these additional constraints. The second inequality is true because by definition we lose at most a factor of α' in converting (y, z) to (y', z') . Similarly the third and fourth inequalities are true by definition of λ and α , respectively. Finally, the last inequality is true from Theorem 2.1. \square

Now, for the class of Set Packing problems whose constraint matrix is totally unimodular (which is closed under taking subinstances) and in which $p(e)$ is the same for all hyperedges e , we have that $\alpha = \alpha' = 1$. For these problems the key concern is then the parameter λ , which we call the *compression gap*, based on the idea that LP (2.1) is in some sense a (lossy) compressed representation of LP (3.5). We will prove the following bound on λ

THEOREM 3.5. *For instances of Set Packing with maximum set size k , the compression gap is bounded by $\lambda = O(\log k)$, even when $p : E \rightarrow [0, 1]$ and $w : E \rightarrow \mathbb{R}_{\geq 0}$ are arbitrary.*

PROOF. Let x be an arbitrary feasible solution to LP (2.1). For all G' and $e \in G'$ let $y(G', e) = \frac{x(e)}{d \cdot p(e)}$, where $d \approx \lambda$ is a parameter to be determined later. Otherwise $y(G', e) = 0$. Clearly, $0 \leq y(G', e) \leq 1/d$, and $z(e) = \sum_{G' \subseteq G, e \in G'} \Pr[G'] \cdot \frac{x(e)}{d \cdot p(e)} = x(e)/d$. Now fix any $v \in V$, and note that

$$\mathbf{E}_{G'} \left[\sum_{e: e \in G', v \in e} y(G', e) \right] \leq 1/d$$

That is, in expectation over the random sampling of G , hyperedge e containing v contributes $p(e) \cdot \frac{x(e)}{d \cdot p(e)}$ to $\sum_{e: e \in G', v \in e} y(G', e)$, and by feasibility of x we have $\sum_{e: e \in G, v \in e} x(e) \leq 1$. However, we require that $\sum_{e: e \in G', v \in e} y(G', e) \leq 1$ for every G' , so (y, z) is not necessarily feasible. We respond by decreasing y on some coordinates to obtain a feasible solution y' . Call a hyperedge e_0 *problematic* for G' if $\sum_{e: e \in G', v \in e} y(G', e) > 1$ for some $v \in e_0$. If e is problematic for G' , set $y'(G', e) = 0$. Otherwise, set $y'(G', e) = y(G', e)$. By construction, (y', z') is feasible for LP (3.5) for some z' . The compression gap is bounded by the ratio of $\sum_e w(e)x(e)$ to $\sum_e w(e)z'(e)$. Now, let Y_v be the random variable that maps G' to $\sum_{e: e \in G', v \in e} y(G', e)$. Letting $Y_{v,e}$

be a random variable that maps G' to $y(G', e)$, we see that Y_v is the sum of independent random variables $Y_{v,e}$ which take on values in the range $[0, 1/d]$ and $\mathbf{E}[Y_v] \leq 1/d$. Thus we can use a Chernoff-type inequality (see e.g., [3]) to show that $\Pr[Y_v \geq 1] \leq \exp\{-\Theta(d)\}$ for $d \geq 1$. Thus we can set $d = \Theta(\log k)$ to ensure that $\Pr[Y_v > 1] \leq 1/100k$. Fixing any e and taking a union bound over all $v \in e$ we see that $\Pr[e \text{ problematic}] \leq 1/100$, and so $\sum_e w(e)z'(e) \geq \frac{99}{100} \cdot \sum_e w(e)z(e)$. Since $z(e) = x(e)/d$, we obtain a compression gap of $\lambda \leq \frac{100 \cdot d}{99} = O(\log k)$. \square

COROLLARY 3.6. *Given a k -Set Packing instance represented as hypergraph $G = (V, \mathcal{D})$ whose constraint matrix is totally unimodular, weights $w : \mathcal{D} \rightarrow \mathbb{R}_{\geq 0}$ and a probability $p \in [0, 1]$ such that each hyperedge appears independently with probability p , we can efficiently construct a subhypergraph H of G of maximum degree $\lceil 1/p \rceil$ such that $\mathcal{E}_{\mathcal{P}}(H) = \Omega(\mathcal{E}_{\mathcal{P}}(G)/\log k)$*

4. THE MAIN ALGORITHM

We will now describe the SPMP algorithm, and prove the main result of the paper:

THEOREM 4.1. *Let OPT be the value of an optimal SPMP solution, and let $\mathbf{E}[A]$ be the expected value of the output returned by the algorithm in Figure 1. Define $\rho := \max_S (c(S)/w(S))$ and assume $\rho \leq 1$. Then $\text{OPT}/\mathbf{E}[A] \leq 4k/(1 - \rho/2)$.*

To describe the algorithm, we first define a family of linear programs parameterized by β as follows

$$\begin{aligned} &\text{maximize} && \sum_{S \in \mathcal{D}} w(S)x(S) \\ &\text{subject to} && 0 \leq x(S) \leq p(S) \quad \forall S \in \mathcal{D} \\ &&& x(\mathcal{D}_v) \leq \beta \quad \forall v \in V \end{aligned} \quad (4.6)$$

For a fixed β , let $LP(\beta)$ denote the corresponding LP in the family. The algorithm for selecting which deals to notify is given in Figure 1.

SPMP(V, \mathcal{D}, p, w, c) {
 Let $\hat{\beta} := 1/2k$.
 Optimally solve $LP(\hat{\beta})$. Let x be the solution obtained.
 Round x to an integral solution $y \in \{0, 1\}^{|\mathcal{D}|}$ via simple randomized rounding:
 For each $S \in \mathcal{D}$ independently
 set $y(S) = 1$ with probability $x(S)/p(S)$,
 otherwise set $y(S) = 0$.
 Output $\mathcal{D}' := \{S \mid y(S) = 1\}$.
}

Figure 1: Main algorithm for the Stochastic Packing Market Planning problem.

4.1 Analysis

We first show that the expected payoff of the algorithm's output is reasonably large. Later, we will prove that the expected compensation cost, $\mathbf{E}[\sum \{c(S) \mid S \text{ a lost deal}\}]$, is not too great. Below, the constants were chosen for clarity of exposition, and have not been optimized.

LEMMA 4.2. *Let \mathcal{D} be the set of all potential deals, and let \mathcal{D}' be the output of the algorithm. Then $\mathbf{E}[\mathcal{E}_{\mathcal{P}}(\mathcal{D}')] \geq \frac{\mathcal{E}_{\mathcal{P}}(\mathcal{D})}{4k}$*

PROOF. Let $Z^*(\beta)$ be the value of an optimal solution to $LP(\beta)$. Note for all $\beta \in [0, 1]$, $Z^*(\beta) \geq Z^*(1) \cdot \beta$ since we can map feasible solutions for $LP(1)$ to feasible solution for $LP(\beta)$ via the map $x \mapsto \beta x$. Applying Theorem 2.1 we obtain $\hat{\beta} \mathcal{E}_{\mathcal{P}}(\mathcal{D}) \leq \hat{\beta} Z^*(1) \leq Z^*(\hat{\beta})$, where $\hat{\beta} := 1/2k$. Thus it suffices to show $\mathbf{E}[\mathcal{E}_{\mathcal{P}}(\mathcal{D}')] \geq \frac{Z^*(\hat{\beta})}{4k\hat{\beta}} = \frac{Z^*(\hat{\beta})}{2}$. By analogy with the proof of Theorem 2.1, let x be an optimal solution to $LP(\hat{\beta})$, and mark each set S independently with probability $x(S)/p(S)$. Each set S then appears with probability $p(S)$. We call a set that appears and is marked an *active set*. Here, the marked sets correspond to the algorithm's output. As before, our set packing P will consist of all active sets that do not intersect other active sets. Since the events $\{S \text{ active}\}$ are independent and $\Pr[S \text{ active}] = x(S)$, suitable modifications to equations 2.2 and 2.3 give us that

$$\Pr[S \in P] \geq x(S) \cdot 4^{-\sum_{S' \in \Gamma(S)} x(S')}$$

Since x is feasible for $LP(\hat{\beta})$, we infer that

$$\sum_{S' \in \Gamma(S)} x(S') \leq \sum_{v \in S} x(\mathcal{D}_v) \leq \hat{\beta} |S| \leq 1/2$$

Thus $\Pr[S \in P] \geq x(S)/2$ and

$$\begin{aligned} \mathbf{E}[\mathcal{E}_{\mathcal{P}}(\mathcal{D}')] &\geq \mathbf{E}[w(P)] \\ &\geq \frac{1}{2} \sum_S w(S)x(S) \\ &= Z^*(\hat{\beta})/2 \\ &\geq \hat{\beta} \mathcal{E}_{\mathcal{P}}(\mathcal{D})/2 \\ &= \frac{\mathcal{E}_{\mathcal{P}}(\mathcal{D})}{4k} \end{aligned}$$

□

LEMMA 4.3. *Let \mathcal{D}' be the output of the algorithm, and let $\rho := \max_S (c(S)/w(S))$. Using a greedy second stage algorithm, the expected compensation cost for \mathcal{D}' , $\mathbf{E}[\sum \{c(S) \mid S \text{ a lost deal}\}]$, is at most $\frac{\rho}{2} \mathbf{E}[\mathcal{E}_{\mathcal{P}}(\mathcal{D}')]$.*

PROOF. As before, let x be an optimal solution to $LP(\hat{\beta})$, mark each set S independently with probability $x(S)/p(S)$, and let each set S appear with probability $p(S)$. Call set S active if it is marked and appears. In decreasing order of $w(S)$, greedily construct a set packing P using only active sets. For each active $S' \notin P$, amortize the compensation cost $c(S')$ by charging it to the set $S \in P \cap \Gamma(S')$ maximizing $w(S)$. Now fix any $S \in P$ and let Y_S be a random variable indicating the number of active sets in $\Gamma(S)$, which is an upper bound on the number of sets charged to S . Since $\Pr[S' \text{ active}] = x(S')$

$$\mathbf{E}[Y_S] = \sum_{S' \in \Gamma(S)} x(S') \leq \sum_{v \in S} x(\mathcal{D}_v) \leq |S| \hat{\beta}$$

Note that if we charge $c(S')$ to S then by the construction of P , $w(S') \leq w(S)$. By the definition of ρ , $c(S') \leq \rho w(S)$ for all S . Thus, the expected charge to $S \in P$ is at most $|S| \cdot \hat{\beta} \cdot \rho \cdot w(S) \leq \rho w(S)/2$. Thus for any set packing P that we construct, the expected total compensation cost (conditioned on P being the output of the greedy set packing algorithm) is at most $\frac{\rho}{2} \sum_{S \in P} w(S)$, from which we may immediately infer that the expected total compensation cost is at most $\frac{\rho}{2} \mathbf{E}[\mathcal{E}_{\mathcal{P}}(\mathcal{D}')]$. □

We can now bound the approximation ratio of the optimal SPM value OPT to the expected value of the algorithm's output, which we will denote as $\mathbf{E}[A]$. Note $\text{OPT} \leq \mathcal{E}_{\mathcal{P}}(\mathcal{D})$, since $\mathcal{E}_{\mathcal{P}}(\mathcal{D})$ is the expected payoff received if all deals are notified and no compensation is made for lost deals. Lemmas 4.2 and 4.3 together with linearity of expectation imply that $\mathbf{E}[A] \geq \frac{\mathcal{E}_{\mathcal{P}}(\mathcal{D})}{4k} (1 - \frac{\rho}{2})$, which concludes the proof of Theorem 4.1.

A Word on Second Stage Algorithms.

Note that after the deals in \mathcal{D}' are notified, and $\mathcal{F} \subseteq \mathcal{D}'$ are committed to by all the relevant parties, any algorithm may be used to find a set packing $P \subseteq \mathcal{F}$ that maximizes $\sum_{S \in P} w(S) - \sum_{S \notin P} c(S)$. One possibility is the greedy algorithm for Set Packing. (The greedy algorithm tries to add sets to its output in decreasing order of $w(S)$, and will add S if and only if S does not intersect a set it has already added.) Another algorithm, which we will call the *random mark* algorithm, simply marks each set that appears independently with some probability α and outputs the sets S that are marked and have no marked neighbors (i.e., for all $S' \in \Gamma(S)$, S' is unmarked). If $\alpha = 1$, we call this the *trivial* algorithm. Note that in the proof of Lemma 4.2 we show that the expected payoff which the trivial algorithm receives is at least $Z^*(\hat{\beta})/2$, while in the proof of Lemma 4.3 we show that the expected compensation the greedy algorithm pays is at most $\rho/2$ of the payoff it collects. Because the greedy algorithm returns a strictly better set packing than the trivial algorithm, these observations imply that using the greedy algorithm to select which deals to make results in an expected profit to the system of $\frac{\text{OPT}}{4k} (1 - \frac{\rho}{2})$.

5. TRUTHFULNESS IN EXPECTATION

So far we have left strategic issues aside, instead focusing exclusively on the design of approximation algorithms. In this section we show how to devise a randomized mechanism for the SPM problem that is truthful *in expectation*, assuming the compensation costs are not paid to the agents, but rather to some third party.

Recall that a (direct revelation) mechanism is an algorithm that receives as input n valuation functions $v_i : \mathcal{O} \rightarrow \mathbb{R}_{\geq 0}$, where \mathcal{O} is the set of possible outcomes, and must select an outcome $\sigma \in \mathcal{O}$, as well as prices $\rho_i \in \mathbb{R}$ to charge each of the n agents. The i^{th} input v_i is provided by agent i , who may report any of the allowed valuation functions. We let \bar{v}_i denote agent i 's true valuation function. Given a tuple of valuation functions v , let $\sigma(v) \in \mathcal{O}$ denote the outcome and let $\rho(v) := (\rho_1(v), \dots, \rho_n(v))$ denote the prices selected by the mechanism on input v . We assume the agents' utilities are quasi-linear. That is, the utility that agent i obtains from outcome σ is $\bar{v}_i(\sigma) - \rho_i$. As usual, each agent seeks to maximize its utility. It is often desirable that a mechanism be *truthful*, by which we mean that each agent may maximize its utility by reporting its true valuation function as input, irrespective of the behavior of the other agents. A randomized mechanism is said to be *truthful in expectation* if each agent may maximize its *expected* utility by reporting its true valuation function as input, irrespective of the behavior of the other agents.

In our setup, agents place bids and asks for various goods and services. These bids induce the set of potential deals \mathcal{D} and payoffs. These, together with our estimates for the deal probabilities and the compensation costs generate a stochastic packing market. Note that $w(S)$ is the social surplus of executing deal S . We then notify some of the potential deals, and agents probabilistically appear. The agents may not change their bids. A second stage algorithm then determines which deals are to be executed and what payments are to be made. The valuation functions we consider have

the form $\{(S_i, b_i, t_i) \mid i \in I\}$, where S_i is a set of goods and services, $b_i \in \mathbb{R}$ is its perceived value, $t_i \in \{\text{bid}, \text{ask}\}$ is the type of the bid, and I is a finite set. The value to the agent of executing the deals $\{S_i \mid i \in J\}$ for some $J \subseteq I$ (without making or receiving any payments) is then $\sum_{j \in J: t_j = \text{bid}} b_j - \sum_{j \in J: t_j = \text{ask}} b_j$. For simplicity of exposition, we will imagine that each agent is associated with a single set (either as a supplier or customer), and simulate agents with several bids as a group of virtual agents.

5.1 Our Approach

Lehmann *et al.* [18] have shown that merely using an approximation algorithm instead of an exact algorithm may render a truthful mechanism non-truthful. However, as noted previously, Lavi and Swamy [17] gave a general procedure for converting certain linear programming based approximation algorithms for combinatorial auctions into randomized mechanisms that are truthful in expectation. At a high level, their approach is as follows. First, they encode an auction as a packing IP, relax the integrality constraints, and use a linear programming algorithm in conjunction with the VCG approach to obtain a truthful fractional mechanism (which may assign fractions of an item to various agents). Next, they extend previous work on randomized rounding. They show how a LP-based approximation algorithm that bounds the integrality gap of the LP relaxation by some parameter α can be used to obtain a convex decomposition of any fractional assignment of items into α times some weighted average of polynomially many integral assignments. The weights can then be treated as probabilities to convert the convex decomposition into a probability distribution over integral assignments with polynomially sized support. The randomized mechanism then draws an outcome from this distribution. The prices are set as follows. Let $\sigma^F(v)$ be the outcome of the fractional mechanism and let $\rho^F(v)$ be its prices. Suppose the randomized mechanism selects outcome $\sigma(v)$ and $v_i(\sigma^F(v)) > 0$. Then set

$$\rho_i(v) = \frac{v_i(\sigma(v))}{v_i(\sigma^F(v))} \cdot \frac{\rho_i^F(v)}{\alpha}$$

and otherwise set $\rho_i(v) = 0$.

There are some difficulties with applying this technique in our case. First, the natural LP relaxation of our problem (which we will not work with directly, but still give below for completeness) has both exponentially many variables and constraints. (In LP (5.7) below, the $x(e)$ variable indicates the extent to which hyperedge e is notified. The $y(G', e)$ variables indicate the extent to which hyperedge e is executed when G' is exactly the set of hyperedges that appears. Finally, the $z(e)$ and $f(e)$ variables indicate the probability that we collect payment for e or pay the compensation cost for e , respectively.)

$$\begin{aligned} & \text{maximize} && \sum_{e \in E} (w(e)z(e) - c(e)f(e)) && (5.7) \\ & \text{subject to} && z(e) = \sum_{G' \subseteq G} \mathbf{Pr}[G'] \cdot y(G', e) && \forall e \in E \\ & && f(e) = p(e)x(e) - z(e) && \forall e \in E \\ & && 0 \leq y(G', e) \leq x(e) \leq 1 && \forall G' \subseteq G, e \in E \\ & && \sum_{e: e \in G', v \in e} y(G', e) \leq 1 && \forall G' \subseteq G, v \in V \end{aligned}$$

It is unclear how we could solve this LP exactly to run the corresponding fractional VCG mechanism, as the Lavi and Swamy procedure would dictate. Note that we need not use this particular fractional mechanism. In fact, we can use *any* truthful fractional mechanism satisfying the following properties.

1. The mechanism can be efficiently converted into a truthful

randomized mechanism whose outcomes are integral allocations of sets.

2. The resulting randomized mechanism well approximates the social welfare for the original SPMP problem if the agents report their true valuations.

The fractional mechanism we will use is the VCG mechanism derived from LP (4.6), with $\beta = 1/2k$ as in the main algorithm from section 4. Recall that when the VCG mechanism for a combinatorial exchange is given input v it computes the surplus-maximizing outcome $\sigma^*(v)$, as well as the surplus-maximizing outcome $\sigma_{-i}^*(v)$ in the exchange where we have removed agent i , for each agent i in turn. The VCG mechanism then selects outcome $\sigma^* = \sigma^*(v)$ and sets the payments so that agent i pays the exchange

$$\rho_i(v) = v_i(\sigma^*) - \left(\sum_j v_j(\sigma^*) - \sum_{j: j \neq i} v_j(\sigma_{-i}^*) \right)$$

In our fractional VCG mechanism the outcomes are feasible solutions to LP (4.6) with $\beta = 1/2k$. We extend the valuation functions of the agents to handle fractional allocations in the obvious way: if x is a fractional allocation, $\{y^j \mid j = 1, 2, \dots\}$ are integral allocations, and $x = \sum_j \lambda_j y^j$, then we define $v_i(x) := \sum_j \lambda_j v_i(y^j)$. The high level idea is that we will use the fractional VCG mechanism to determine a fractional outcome x along with prices ρ_i^F . We will then round x exactly as in the main algorithm from section 4 to determine which deals to notify. Next, we will carefully solve the second stage winner-determination problem so that each deal S is executed with probability exactly $\alpha \cdot x(S)$, where $\alpha \in (0, 1)$ is some constant. Finally, we will ensure that each agent i pays exactly $\alpha \cdot \rho_i^F$ in expectation, and that individual rationality is maintained. Agents that only care about maximizing their expected utility will consider the resulting randomized mechanism to be equivalent to the fractional VCG mechanism in which the outcome and prices are scaled by a factor of α . The resulting randomized mechanism will thus be truthful in expectation. One caveat is that we will not be able to handle compensation costs paid directly to the agents. Instead, we will imagine they are paid to some third party. In this way the compensation costs will not affect the strategic behavior of the agents, but will penalize the system for notifying lost deals.

5.2 The Mechanism

The mechanism is given in Figure 2. Recall that $w(S)$ is the social surplus of executing deal S , and is computed from the input vector v of valuation functions. For example, in a combinatorial auction, $w(S)$ would simply be the maximum value any agent bid on S . We will now proceed to prove various properties of the mechanism.

CLAIM 5.1. *The randomized mechanism executes each deal S with probability exactly $\alpha \cdot x(S)$.*

PROOF. Fix a deal S . The deal is executed if S is notified and S materializes and each set $S' \in \Gamma(S)$ fails to materialize and an independent coin toss with bias (α/α_S) comes up heads. The probability of this is $\mathbf{Pr}[S \text{ executed}] = \frac{x(S)}{p(S)} \cdot p(S) \cdot \alpha_S \cdot (\alpha/\alpha_S)$ which equals $\alpha \cdot x(S)$. \square

CLAIM 5.2. *The randomized mechanism ensures each agent i pays $\alpha \cdot \rho_i^F$ in expectation, and maintains individual rationality.*

PROOF. Fix an agent i and let S be the deal which agent i bid on. (Recall that for simplicity we have assumed each agent only

```

First-Stage( $V, \mathcal{D}, p, c, v$ ) {
  Compute  $x$  and  $\rho^F$  using the fractional VCG mechanism.
  Round  $x$  to an integral  $y$  as in the main SPMP algorithm:
  For each  $S \in \mathcal{D}$  independently
    set  $y(S) = 1$  with probability  $x(S)/p(S)$ ,
    otherwise set  $y(S) = 0$ .
  Notify the set of deals  $\mathcal{D}' := \{S \mid y(S) = 1\}$ .
}
Second-Stage( $V, \mathcal{F}, x, \rho^F$ ) {
  For each  $S \in \mathcal{D}$ , let  $\alpha_S := \prod_{S' \in \Gamma(S)} (1 - x(S'))$ .
  Let  $\alpha := \min_{S \in \mathcal{D}} \{\alpha_S\}$ .
  For each  $S \in \mathcal{F}$ 
    If  $\Gamma(S) \cap \mathcal{F} = \emptyset$ 
      Independently with probability  $(\alpha/\alpha_S)$  do
        Execute  $S$ .
        For each agent  $i$  associated with  $S$ ,
          set payments  $\rho_i = \rho_i^F/x(S)$ .
    }
}

```

Figure 2: The Randomized Mechanism. In the second stage, \mathcal{F} is the set of deals that materialized (i.e., the active deals).

bid on one set.) If S is executed, then agent i pays $\rho_i^F/x(S)$. Otherwise agent i pays nothing. Since S is executed with probability $\alpha \cdot x(S)$ by Claim 5.1, the expected payment is $\alpha \cdot \rho_i^F$. Individual rationality is maintained because the fractional VCG mechanism ensures individual rationality, which implies $\rho_i^F \leq v_i(x)$. If we abuse notation slightly and let $v_i(S)$ denote the value to agent i of executing deal S , then by the definition of $v_i(\cdot)$ over fractional domains $v_i(x) = v_i(S)x(S)$. Thus $\rho_i^F/x(S) \leq v_i(S)$ and so agent i never pays more than its bid for S . \square

CLAIM 5.3. *The randomized mechanism $O(k)$ approximates the social surplus in expectation, assuming $\max_S \{ \frac{c(S)}{w(S)} \} \leq 1 - \epsilon$.*

PROOF. By Claim 5.1, each deal S is executed with probability $\alpha \cdot x(S)$ so the expected surplus is $\sum_S w(S) \cdot \alpha \cdot x(S)$. We can lower bound α by noting that $\alpha_S = \prod_{S' \in \Gamma(S)} (1 - x(S')) \geq 4^{-\sum_{S' \in \Gamma(S)} x(S')}$ and $\sum_{S' \in \Gamma(S)} x(S') \leq |S|\beta \leq 1/2$ by feasibility of x . Thus $\alpha_S \geq 4^{-1/2} = 1/2$ for all S , which implies $\alpha := \min_S \{\alpha_S\} \geq 1/2$. To bound the expected compensation cost, note that each deal S is notified and materializes with probability $\Pr[S \text{ active}] = (x(S)/p(S))p(S) = x(S)$. Together with Claim 5.1, this implies the expected compensation cost for S is exactly $c(S) \cdot (1 - \alpha)x(S)$. Assuming $\max_S \{ \frac{c(S)}{w(S)} \} \leq 1 - \epsilon$, the social surplus minus the compensation costs is at least $\frac{\epsilon}{2} \sum_S w(S)x(S)$. However, $\sum_S w(S)x(S) \geq \mathcal{E}_{\mathcal{P}}(\mathcal{D})/2k$, as we showed in the proof of Lemma 4.2. \square

CLAIM 5.4. *The randomized mechanism is truthful in expectation.*

PROOF. By the standard VCG analysis, the fractional VCG mechanism is truthful, because it aligns the computation of the mecha-

nism with each individual agent. That is, the utility of agent i is

$$\begin{aligned}
u_i(\sigma^*) &= \bar{v}_i(\sigma^*) - \rho_i^F \\
&= \bar{v}_i(\sigma^*) - v_i(\sigma^*) + \left(\sum_j v_j(\sigma^*) - \sum_{j:j \neq i} v_j(\sigma_{-i}^*) \right) \\
&= \bar{v}_i(\sigma^*) + \sum_{j:j \neq i} v_j(\sigma^*) - \sum_{j:j \neq i} v_j(\sigma_{-i}^*)
\end{aligned}$$

The last term is independent of i 's bid and can thus be ignored. However σ^* is chosen to maximize $\sum_j v_j(\sigma^*)$, and thus agent i can do no better than revealing its true valuation function, in which case the mechanism maximizes $\bar{v}_i(\sigma^*) + \sum_{j:j \neq i} v_j(\sigma^*)$ and thus $u_i(\sigma^*)$. Clearly, scaling the outcome and the prices by a factor of α does not affect truthfulness. Moreover, replacing the α -scaled fractional VCG mechanism with the randomized mechanism does not change the behavior of agents who only care about their expected utility, since their expected values and payments are the same in both cases by Claims 5.1 and 5.2. In other words, if we assume the each agent i is indifferent between receiving an $\alpha \cdot x(S)$ fraction of S and paying $\alpha \cdot \rho_i^F$ on the one hand, and receiving all of S with probability $\alpha \cdot x(S)$ and paying $\rho_i^F/x(S)$ in that event on the other hand, then the agents will not distinguish between the α -scaled fractional VCG mechanism and the randomized mechanism. Thus the randomized mechanism is truthful in expectation. \square

6. APPLICATIONS

Excess Capacity Markets.

We begin by revisiting the airline seat market example from the introduction. Let us represent the SPM by a hypergraph. There will be a vertex for each available seat on each flight in the market, as well as a vertex for each customer. Customers bid on pairs of flights, corresponding to desired departing and returning flights, or perhaps on sets of flights that correspond to tours of some country or region. These bids are represented as sets in the SPM that include their flights and the customer's vertex. The latter ensures that each customer is awarded only one deal at a time. We can add hotel rooms and rental cars by adding vertices for them as well. If a typical bid consists of the customer, two flights, a hotel and a rental car, then $k = 5$. Thus our model can support a bidding language that can express arbitrary disjunctions of conjunctions (e.g., "I want (A and B at price P_1) OR (A and C at price P_2) OR ..."), where each conjunction has at most $k - 1$ items.

In addition to selling residual airline capacity, we can use the SPMP algorithm to run an exchange market to sell "residual capacity" in other markets where there is a low marginal cost to provide service to an additional customer. Consider the market for movie tickets. Customers will have different substitution effects based on geometry (preferring theaters near their residences), dates and times of shows, and which movie is playing. Naturally the same applies for concert tickets for a symphony or an opera, or any kind of performances that occur on a recurring basis.

Online Dating Services.

In an online dating service, user profiles can be used to create a graph of potentially romantically-compatible people. We can imagine there are payoffs on edges associated with the value of initiating a relationship between the people incident on the edge. Notifying an edge corresponds to suggesting a date. Presumably there is a non-negligible cost associated with suggesting a date only to have

exactly one of the parties agree to it. These lost deals (aka romantic rejections) are likely to drive customers away from the service. However our sparsification and SPMP results give a way to dramatically reduce the number of lost deals while losing only a constant factor in the number of dates that are made.

Coordinated Second-hand Markets.

The SPM framework provides a structured way of looking at everything from used car markets to used book markets. The system can be used in at least two ways. First, it can act primarily as a recommendation system, as in the case of an online dating service. That is, it can notify agents of potential deals and then leave the agents to negotiate the terms themselves. Combined with some algorithm to estimate the deal probabilities, this system provides a nice way to facilitate trade without flooding the agents with spam. Alternately, the system can determine which deals are to be made, and what the payments should be. Of course, many other payment schemes are possible, and there are complex design tradeoffs and strategic issues inherent in combinatorial exchange markets. (Certain tradeoffs are necessary. See [19] for a treatment of impossibility results for combinatorial exchange markets).

7. CONCLUSIONS AND OPEN PROBLEMS

In this paper we show how to approximate a stochastic generalization of k -Set Packing nearly as well as k -Set Packing itself, and show how to apply this result to perform approximate market clearing in a market with probabilistic supply and demand while maintaining truthfulness in expectation. We also show how to $O(k)$ -approximate the expected weight of a maximum weight set packing in a random set system sampled from an arbitrary set system with maximum set size k . Thus, for example, we provide a linear programming based 8-approximation on the expected size of a matching in a random subgraph of an arbitrary graph. Lastly, we have given sparsification results that show how to dramatically reduce the number of potential deals to consider (as well as the number of unwanted messages sent to the users), while approximately preserving the expected value of the maximum weight set packing. Of course, there remain many fascinating open problems, including

1. Estimation of deal probabilities $p(S)$: How can we estimate the deal probabilities over many rounds of market clearing in a way that is unbiased and discourages strategic manipulation of the system?
2. Handling more complex distributions over subhypergraphs: Researchers working on two stage stochastic covering problems with recourse have developed algorithms that give good approximation guarantees even in a *black box* model [23], where the distribution on future scenarios is arbitrary and the algorithm can only learn about the distribution by sampling from it. Following their lead, can we obtain good algorithms in the case that the subhypergraph of the notified hyperedges that appears is drawn from an arbitrary distribution represented by a black box?
3. Online variants: In an online setting, what kind of guarantees on competitive ratio can be obtained? For example, suppose edges in a graph arrive one by one, and upon arrival you must decide to pursue it or not. If you pursue edge e , with probability $p(e)$ you receive $w(e)$ and the vertices incident on e are used up. Otherwise, you pay $c(e)$ and the vertices remain available for future edges.

4. Strategic issues: There is the full range of combinatorial exchange market design (including truthfulness, welfare maximization, budget balancing), with the additional complexities of probabilistic demand and supply and the economic friction of costs to send information to the agents. For example, can we obtain truthfulness with certainty rather than truthfulness in expectation? Can we maintain truthfulness if the compensation costs are paid to the agents rather than some third party?

Acknowledgments

Thanks to Vahab Mirrokni for providing the initial question that served as the impetus for this work, and thanks to Guy Blelloch, Ryan O'Donnell, Alan Frieze, Matt Streeter, and David Brumley for helpful discussions.

8. REFERENCES

- [1] Ravi Boppana and Magnus M. Halldórsson. Approximating maximum independent sets by excluding subgraphs. *BIT*, 32(2):180–196, 1992.
- [2] Barun Chandra and Magnus M. Halldórsson. Greedy local improvement and weighted set packing approximation. *J. Algorithms*, 39(2):223–240, 2001.
- [3] Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities – a survey. *Internet Math.*, to appear.
- [4] Peter Cramton, Yoav Shoham, and Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006.
- [5] Sven de Vries and Rakesh V. Vohra. Combinatorial Auctions: A Survey. *INFORMS Journal on Computing*, 15:284–309, 2003.
- [6] Brian C. Dean, Michel X. Goemans, and Jan Vondrák. Adaptivity and approximation for stochastic packing problems. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 395–404, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
- [7] Shahar Dobzinski, Noam Nisan, and Michael Schapira. Truthful randomized mechanisms for combinatorial auctions. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 644–652, New York, NY, USA, 2006. ACM Press.
- [8] Uriel Feige. On maximizing welfare when utility functions are subadditive. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 41–50, New York, NY, USA, 2006. ACM Press.
- [9] Alan Frieze and Colin McDiarmid. Algorithmic theory of random graphs. *Random Struct. Algorithms*, 10(1-2):5–42, 1997.
- [10] Tobias Galla, Michele Leone, Matteo Marsili, Mauro Sellitto, Martin Weigt, and Riccardo Zecchina. Statistical mechanics of combinatorial auctions. *Physical Review Letters*, 97:128701, 2006.
- [11] M. M. Halldórsson, J. Kratochvíl, and J. A. Telle. Independent sets with domination constraints. *Discrete Appl. Math.*, 99(1-3):39–54, 1999.
- [12] Magnus M. Halldórsson. Approximations of weighted independent set and hereditary subset problems. *J. Graph Algorithms and Applications*, 4(1):1–16, 2000.
- [13] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182(1):105–142, 1999.

- [14] Elad Hazan, Muli Safra, and Oded Schwartz. On the complexity of approximating k -set-packing. *Computational Complexity*, 15(1):20–39, 2006.
- [15] C. A. J. Hurkens and A. Schrijver. On the size of systems of sets every t of which have an SDR, with an application to the worst-case ratio of heuristics for packing problems. *SIAM J. Discret. Math.*, 2(1):68–72, 1989.
- [16] Jayant Kalagnanam and David C. Parkes. Auctions, bidding and exchange design. In David Simchi-Levi, S. David Wu, and Max Shen, editors, *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, Int. Series in Operations Research and Management Science, chapter 5. Kluwer, 2004.
- [17] Ron Lavi and Chaitanya Swamy. Truthful and near-optimal mechanism design via linear programming. In *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 595–604, Washington, DC, USA, 2005. IEEE Computer Society.
- [18] Daniel Lehmann, Liadan Ita O’Callaghan, and Yoav Shoham. Truth revelation in approximately efficient combinatorial auctions. *J. ACM*, 49(5):577–602, 2002.
- [19] Y. Narahari and Pankaj Dayama. Combinatorial auctions for electronic business. *Sadhana, Indian Academy of Sciences Proceedings in Engineering Sciences*, Special Issue on Electronic Commerce and Electronic Business, 2005.
- [20] Noam Nisan and Amir Ronen. Computationally feasible VCG mechanisms. In *EC '00: Proceedings of the 2nd ACM conference on Electronic commerce*, pages 242–252, New York, NY, USA, 2000. ACM Press.
- [21] Aleksandar Pekec and Michael H. Rothkopf. Combinatorial auction design. *Manage. Sci.*, 49(11):1485–1503, 2003.
- [22] Alexander Schrijver. *Combinatorial optimization : polyhedra and efficiency*. Volume A, paths, flows, matchings, chapter 1-38. Springer, 2003. Schrijver.
- [23] C. Swamy and D.B. Shmoys. Approximation algorithms for 2-stage stochastic optimization problems. In *To appear in Proceedings of FSTTCS*, 2006.