Frugal and Truthful Auctions for Vertex Covers, Flows, and Cuts

David Kempe*, Mahyar Salek* and Cristopher Moore‡
*University of Southern California
‡University of New Mexico and The Santa Fe Institute

***Abstract*—We study truthful mechanisms for hiring a team of agents in three classes of set systems: Vertex Cover auctions, k-flow auctions, and cut auctions. For Vertex Cover auctions, the vertices are owned by selfish and rational agents, and the auctioneer wants to purchase a vertex cover from them. For k-flow auctions, the edges are owned by the agents, and the auctioneer wants to purchase k edge-disjoint s-t paths, for given s and t. In the same setting, for cut auctions, the auctioneer wants to purchase an s-t cut. Only the agents know their costs, and the auctioneer needs to select a feasible set and payments based on bids made by the agents.**

We present constant-competitive truthful mechanisms for all three set systems. That is, the maximum overpayment of the mechanism is within a constant factor of the maximum overpayment of any truthful mechanism, for *every* set system in the class. The mechanism for Vertex Cover is based on scaling each bid by a multiplier derived from the dominant eigenvector of a certain matrix. The mechanism for k-flows prunes the graph to be minimally $(k+1)$-connected, and then applies the Vertex Cover mechanism. Similarly, the mechanism for cuts contracts the graph until all s-t paths have length exactly 2, and then applies the Vertex Cover mechanism.

***Keywords*-mechanism design; frugality; spectral; Vertex Covers; Flows; Cuts;**

I. Introduction

Many tasks require the joint allocation of multiple resources belonging to different bidders. For instance, consider the task of routing a packet through a network whose edges are owned by different agents. In this setting, it is necessary to obtain usage rights for multiple edges simultaneously from the agents. Similarly, if the agents own the vertices of a graph, and we want to monitor all edges, we need the right to install monitoring devices on nodes, and again obtain these rights from distinct agents.

Providing access to edges or nodes in such settings makes the agents incur a cost c_e, which the agents should be paid for. A convenient way to determine "appropriate" prices to pay the agents is by way of *auctions*, wherein the agents e submit bids b_e to an *auctioneer*, who selects a *feasible subset* S of agents to use, and determines prices p_e to pay the agents. The most basic case is a single-item auction. The auctioneer requires the service of any one of the agents, and their services are interchangeable. Single-item auctions have a long history of study, and are fairly well understood [16], [17]. Motivated by applications in computer networks and electronic commerce, several recent papers have considered the extension to a setup termed *hiring a team of agents* [3], [9], [10], [15], [22]. In this setting, there is a collection of *feasible sets*, each consisting of one or more agent. The auctioneer, based on the agents' bids b_e, selects one feasible set S, and pays each agent $e \in S$ a price p_e.

Some of the well-studied special cases of set systems are *path auctions* [3], [10], [15], [20], [24], in which the feasible sets are paths from a given source s to a given sink t, and *spanning tree auctions* [4], [12], [15], [22], in which the feasible sets are spanning trees of a connected graph. In both cases, the agents are the edges of the graph. In this paper, we will extend the study to more complex examples of set systems, namely:

1) Vertex Covers: The agents own the *vertices* of the graph G, and the auctioneer needs to select a vertex cover [5], [9], [22]. Not only are vertex covers of interest in their own right, but they give a key primitive for many other set systems as well, as we will explore in depth in this paper.
2) Flows: The agents are the edges of G, and the auctioneer wants to select k edge-disjoint paths from s to t. Thus, this scenario generalizes path auctions; the generalization turns out to require significant new techniques in the design and analysis of mechanisms.
3) Cuts: In the same setting as for flows, the auctioneer wants to purchase an s-t cut.

In choosing an auction mechanism for a set system, the auctioneer needs to take into account that the agents are selfish. Ideally, the auctioneer would like to know the agents' true costs c_e. However, the costs are private information, and a rational and selfish agent will submit a bid $b_e \neq c_e$ if doing so leads to a higher profit. The area of *mechanism design* [19], [20], [21] studies the design of auctions for selfish and rational agents.

We are interested in designing *truthful* (or *incentive-compatible*) auction mechanisms: auctions under which it is always optimal for selfish agents to reveal their private costs c_e to the auctioneer. Such mechanisms are societally desirable, because they make the computation of strategies a trivial task for the agents, and obviate the need for gathering information about the costs or strategies of competitors. They are also desirable from the point of view of analysis, as they allow us to identify bids with costs, and let us dispense with any kinds of assumptions about the distribution of agents' costs. In that sense, truthful mechanisms have inherently much more stable outcomes than games only possessing Nash Equilibria, and may give bidders more confidence that the right outcome will be reached. For this reason, truthful mechanism design has been a mainstay of game theory for a long time.

It is well known that any truthful mechanism will have to pay agents more than their costs at times; in this paper, we study mechanisms approximately minimizing the "overpayment". The ratio between the payments of the "best" truthful mechanism and natural lower bounds has been termed the "Price of Truth" by Talwar [22], and studied in a number of recent papers [3], [4], [9], [10], [12], [15], [22], [24]. In particular, [15] and [9] define and analyze different natural measures of lower bounds on payments, and define the notions of frugality ratio and competitiveness. The *frugality ratio* of a mechanism is the worst-case ratio of payments to a natural lower bound (formally defined in Section II), over all cost vectors of the agents. A mechanism is *competitive* for a class of set systems if its frugality ratio is within a constant factor of the frugality ratio of the best truthful mechanism, for *all* set systems in the class.

A. Our Contributions

In this paper, we present novel frugal mechanisms for three general classes of set systems: Vertex Covers, k-Flows, and Cuts. Vertex Cover auctions can be considered a very natural primitive for more complicated set systems. Under the natural assumption that there are no isolated vertices, they capture set systems with "minimal competition": if the auction mechanism decides to exclude an agent v from the selected set, this immediately forces the mechanism to include all of v's neighbors, thus giving these neighbors a monopoly. Thus, a different interpretation of Vertex Cover auctions is that they capture any set system where feasible sets can be characterized by positive 2SAT formulas: each edge (i, j) corresponds to a clause $(x_i \vee x_j)$, stating that any feasible set must include at least one of agents i and j.

Our mechanism for Vertex Cover works as follows: based solely on the structure of the graph G, we define an appropriate matrix K and compute its top eigenvector $\mathbf{q}$. After agents submit their bids b_v, the mechanism first scales each bid to $c'_v = b_v/q_v$, and then simply runs the VCG mechanism [23], [8], [13] with these modified bids. We prove that this mechanism has a frugality ratio of α (the largest eigenvalue of K), and that this is within a factor of at most 2 of the frugality ratio of any mechanism. The lower bound is based on pairwise "competition" between adjacent bidders for *any* truthful mechanism, and in a sense can be considered the natural culmination to which the past techniques of [10], [15] can be taken in deriving lower bounds. The upper bound is based on carefully balancing all possible worst cases of a single non-zero cost against each other, and showing that the worst case is indeed one of these cost vectors. We stress here that the mechanism does not in general run in polynomial time: the entries of K are derived from fractional clique sizes in G, which are known to be hard to compute, even approximately. We discuss the issue of polynomial time briefly in Section VI.

Based on our Vertex Cover mechanism, we present a general methodology for designing frugal truthful mechanisms. The idea is to take the original set system, and prune agents from it until it has "minimal competition" in the above sense; subsequently, the Vertex Cover auction can be invoked. So long as the pruning is "composable" in the sense of [1] (see Section III), the resulting auction is truthful. The crux is then to prove that the pruning step (which removes a significant amount of competition) does not increase the lower bound on payments too much. We illustrate the power of this approach with two examples.

1) For the k-flow problem, we show that pruning the graph to a minimum cost $(k+1)$ s-t-connected graph H is composable, and increases the lower bound at most by a factor of $k+1$. Hence, we obtain a $2(k+1)$-competitive mechanism. Establishing the bound of $k+1$ requires significant technical effort.
2) For the cut problem, we show that pruning the graph to a minimum-cost set of edges such that each s-t path is cut at least twice gives a composable selection rule. Furthermore, it increases the lower bound by at most a factor of 2, leading to a 4-competitive mechanism. For the pruning step, we develop a primal-dual algorithm generalizing the Ford-Fulkerson Minimum-Cut algorithm.

We note that while the Vertex Cover mechanism is in general not polynomial, for both special cases derived here, the running time will in fact be polynomial.

B. Relationship to Past and Parallel Work

As discussed above, a line of recent papers [3], [4], [9], [10], [12], [15], [22], [24] analyze frugality of auctions in the "hiring a team" setting, where the auctioneer wants to obtain a feasible set of agents, while paying not much more than necessary. In this context, the papers by Karlin et al. [15] and Elkind et al. [9] are particularly related to our work.

Karlin et al. [15] introduce the definitions of frugality and competitiveness which we use here. They also give competitive mechanisms for path auctions, and for so-called r-out-of-k systems, in which the auctioneer can select any r out of k disjoint sets of agents. At the heart of both mechanisms is a mechanism for r-out-of-$(r+1)$ systems. Our mechanism for Vertex Covers can be considered a natural generalization of this mechanism. Furthermore, both r-out-of-k systems and path auctions are special cases of k-flows. (Notice that choosing an r-flow when the graph consists of k vertex-disjoint s-t paths is equivalent to r-out-of-k systems.) Our approach of pruning the graph is similar in spirit to the approach in [15], where graphs were also first pruned to be minimally 2-connected, and set systems were reduced to an r-out-of-$(r+1)$ system. However, the combinatorial structure of k-flows makes this pruning (and its analysis) much more involved in our case.

Elkind et al. [9] study truthful mechanisms for Vertex Cover. They present a polynomial-time mechanism with frugality ratio bounded by 2Δ (the maximum degree of the graph), and also show that there exist graphs where the best truthful mechanism must have frugality ratio at least $\Delta/2$. Notice, however, that this does not guarantee the mechanism to be competitive: indeed, there are graphs where the best truthful mechanism has frugality ratio significantly smaller than $\Delta/2$, and our goal is to have a mechanism which is within a constant factor of best possible *for every graph*.

Results very similar to ours have been derived independently by [6] using somewhat similar techniques. Both papers first derive mechanisms for Vertex Cover auctions. Our mechanism is based on scaling bids by eigenvector entries of a scaled adjacency matrix. It has constant competitive ratio for all inputs, but may not run in polynomial time. Their mechanism, on the other hand, uses eigenvectors of the unscaled adjacency matrix. It may not be constant competitive on some inputs, but always runs in polynomial time. On the inputs derived from flow and cut problems, both mechanisms coincide, and are thus competitive and run in polynomial time.

II. PRELIMINARIES

A set system $(E, \mathcal{F})$ has n *agents* (or *elements*), and a collection $\mathcal{F} \subseteq 2^E$ of *feasible sets*. We call a set system *monopoly-free* if no element is in all feasible sets, i.e., if $\bigcap_{S\in\mathcal{F}} S = \emptyset$. The three classes of set systems studied in this paper are:

1) Vertex Covers: here, the agents are the *vertices* of a graph G, and $\mathcal{F}$ is the collection of all vertex covers of G. To avoid confusion, we will denote the agents by u, v instead of e in this case. Notice that every Vertex Cover set system is monopoly-free.
2) k-flows: here, we are given a graph G with source s and sink t. The agents are the *edges* of G. A set of edges is feasible if it contains (at least) k edge-disjoint s-t paths. A k-flow set system is monopoly-free if and only if the minimum s-t cut cuts at least $k+1$ edges.
3) Cuts: With the same setup as for k-flows, a set of edges is feasible if it contains an s-t cut. Thus, the set system is monopoly-free if and only if G contains no edge from s to t.

$(E, \mathcal{F})$ is common knowledge to the auctioneer and all agents. Each agent $e \in E$ has a *cost* c_e, which is private, i.e., known only to e. If e is selected by the mechanism, it incurs cost c_e. We write $c(S) = \sum_{e\in S} c_e$ for the total cost of a set S of agent, and also extend this notation to other quantities (such as bids or payments). A *mechanism* for a set system proceeds as follows:

1) Each agent submits a sealed bid b_e.
2) Based on the bids b_e, the auctioneer selects a feasible set $S \in \mathcal{F}$ as the winner, and computes a payment $p_e \geq b_e$ for each agent $e \in S$. The agents $e \in S$ are said to *win*, while all other agents *lose*.

Each agent, knowing the algorithm for computing the winning set and the payments, will choose a bid b_e maximizing her own *profit*, which is $p_e - c_e$ if the agent wins, and 0 otherwise. We are interested in mechanisms where self-interested agents will bid $b_e = c_e$. More precisely, a mechanism is *truthful* if, for any fixed vector $\mathbf{b}_{-e}$ of bids by all other agents, e maximizes her profit by bidding $b_e = c_e$. If a mechanism is known to be truthful, we can use b_e and c_e interchangeably. It is will-known [3], [17] that a mechanism is truthful only if its selection rule is *monotone* in the following sense: If all other agents' bids stay the same, then a losing agent cannot become a winner by raising her bid. Once the selection rule is fixed, there is a unique payment scheme to make the mechanisms truthful: each agent is paid her *threshold bid*: the supremum of all winning bids she could have made given the bids of all other agents.

A. Nash Equilibria and Frugality Ratios

To measure how much a truthful mechanism "overpays", we need a natural bound to compare the payments to. Karlin et al. [15] proposed using as a bound the solution of a natural minimization problem. However, as pointed out by Elkind et al. [9] and Chen and Karlin [7], the proposed bound has several undesirable non-monotonicity properties: introducing more competition can lead to an increase in the bound, as can lowering costs of losing agents or increasing costs of winning agents.

Instead, we use the value of the following natural *maximization LP* proposed by Elkind et al. [9]. Let S be the cheapest feasible set with respect to the true costs c_e; ties are broken lexicographically.

$$\begin{array}{lll} \text{Maximize} & \nu(\mathbf{c}) := \sum_{e\in S} x_e & \\ \text{subject to} & \text{(i)}\ x_e \geq c_e & \forall e \\ & \text{(ii)}\ x_e = c_e & \forall e \notin S \\ & \text{(iii)} \sum_{e\in S} x_e \leq \sum_{e\in T} x_e & \forall T \in \mathcal{F} \end{array} \tag{1}$$

(Elkind et al. [9] refer to $\nu(\mathbf{c})$ as $\mathrm{NTU}_{\max}(\mathbf{c})$). The intuition for this LP is that it captures the bids of agents in the most expensive "Nash Equilibrium" of a first-price auction with full information, under the assumption that the actual cheapest set S wins, and the losing agents all bid their costs. That is, the mechanism selects the cheapest set with respect to the bids x_e, and pays each winning agent her bid x_e. The first constraint captures individual rationality, and the third constraint states that the bids x_e are such that S still wins. We say that a vector $\mathbf{x}$ is *feasible* if it satisfies the LP (1).[1]

In a slight abuse of terminology, we will refer to x_e as the *Nash Equilibrium bid* of agent e, or simply the *Nash bid* of e, despite the fact that these bids technically may not constitute a Nash Equilibrium.

Notice that $\nu(\mathbf{c})$ is defined for all monopoly-free set systems. We now formally define the frugality ratio of a mechanism $\mathcal{M}$ for a set system $(E, \mathcal{F})$, and the notion of a competitive mechanism.

Definition 2.1 (Frugality Ratio, Competitive Mechanism): Let $\mathcal{M}$ be a truthful mechanism for the set system $(E, \mathcal{F})$ and let $P_{\mathcal{M}}(\mathbf{c})$ denote the total payments of $\mathcal{M}$ when the actual costs are $\mathbf{c}$.

1) The *frugality ratio* of $\mathcal{M}$ is

$$\phi_{\mathcal{M}} = \sup_{\mathbf{c}} \frac{P_{\mathcal{M}}(\mathbf{c})}{\nu(\mathbf{c})}.$$

2) The frugality ratio of the set system $(E, \mathcal{F})$ is

$$\Phi_{(E,\mathcal{F})} = \inf_{\mathcal{M}} \phi_{\mathcal{M}},$$

where the infimum is taken over all truthful mechanisms $\mathcal{M}$ for $(E, \mathcal{F})$.

3) A mechanism $\mathcal{M}$ is κ*-competitive* for a class of set systems $\{(E_1, \mathcal{F}_1), (E_2, \mathcal{F}_2), \ldots\}$ if $\phi_{\mathcal{M}}$ is within a factor κ of $\Phi_{(E_i,\mathcal{F}_i)}$ for all i.

Remark 2.2: The frugality ratio of a mechanism is defined as instance-based. The frugality ratio of a set system captures the inherent structural complexity of that instance, which can be "exploited" with careful worst-case choices of costs.

Competitiveness, on the other hand, is defined over a class of set systems. If a single mechanism, such as the ones defined in this paper, is competitive, it does as well on each set system in the class as the best mechanism, which could possibly be tailored to this specific instance. The nomenclature "competitive" is motivated by the analogy with online algorithms.

The instance-based definition [15], [9] allows us a more fine-grained distinction between mechanisms than earlier work (e.g., [3], [20]), where a lower bound in terms of a worst case over all instances was used.

III. Vertex Cover Auctions

In this section, we describe and analyze a constant-competitive mechanism for Vertex Cover auctions. We then show how to use it as the basis for a methodology for designing frugal mechanisms for other set systems. The graph is denoted by $G = (V, E)$, with n vertices. We write $u \sim v$ to denote that $(u, v) \in E$.

The mechanism consists of running VCG with different multipliers for different agents. The multipliers capture "how important" an agent is for the solution, in the sense of how many other agents can be omitted by including this agent. They are computed as eigenvector entries of a certain matrix K. As we will see, the computation of K is NP-hard itself, so the mechanism will in general not run in polynomial time unless P=NP.

As a first step, the mechanism prunes all isolated vertices, since they will never be part of any vertex cover. From now on, we assume that each vertex is incident to at least one edge. Let $\mathbf{1}_v$ (for any vertex v) be the vector with 1 in coordinate v, and 0 in all other coordinates. We define $\nu_v = \nu(\mathbf{1}_v) \geq 1$ to be the total "Nash Equilibrium" payment of the first-price auction (in the sense of [9]) if agent v has cost 1 (and thus loses), and all other agents have cost 0. ν_v is exactly the fractional clique number of the graph induced by the neighbors of v, without v itself. This fact is proved in Section III-A; it implies that unless ZPP=NP, ν_v cannot be approximated to within a factor $O(n^{1-\epsilon})$ in polynomial time, for any $\epsilon > 0$. Our inability to compute ν_v is the chief obstacle to a constant-competitive polynomial-time mechanism.

Let A be the adjacency matrix of G (with diagonal 0). Define $D = \mathrm{diag}(1/\nu_1, 1/\nu_2, \ldots, 1/\nu_n)$, and $K = DA$. Thus, if $u \sim v$, then $k_{u,v} = 1/\nu_u$; otherwise, $k_{u,v} = 0$. K has the same eigenvalues as $K' = D^{-1/2} K D^{1/2} = D^{1/2} A D^{1/2}$, and the eigenvectors of K are of the form $D^{1/2} \cdot \mathbf{e}$, where $\mathbf{e}$ is an eigenvector of K'. Because K' is symmetric and has non-negative entries, by the Perron-Frobenius Theorem, its eigenvalues are real, and its top eigenvector has positive entries. Hence, the same holds for K.

Let α be the largest eigenvalue of K, and $\mathbf{q}$ a corresponding eigenvector (with positive entries). Notice that given K, α and $\mathbf{q}$ can be computed efficiently and without knowledge of the agents' bids or costs.

The mechanism $\mathcal{EV}$ is now as follows: after all nodes v submit their bids b_v, the algorithm sets $c'_v = b_v / q_v$, and computes a minimum cost vertex cover S with respect to the costs c'_v (ties broken lexicographically). S is chosen as the winning set, and each agent in S is paid her threshold bid. (Notice that the second step of the mechanism again requires the solution to an NP-hard problem.)

$\mathcal{EV}$ is truthful since the selection rule is clearly monotone, and the payments are the threshold bids. Thus, we can assume without loss of generality that bids and costs coincide. In the following, we analyze the frugality ratio of $\mathcal{EV}$, and show that $\mathcal{EV}$ is competitive.

Lemma 3.1: $\mathcal{EV}$ has frugality ratio at most α.

Proof: We start by considering only cost vectors with only one non-zero entry, i.e., of the form $\mathbf{c} = c_v \cdot \mathbf{1}_v$. For such a cost vector, consider any agent $u \neq v$. If u bids more than $\frac{q_u}{q_v} \cdot c_v$ (and all agents besides u, v bid 0), then the set $V \setminus \{u\}$ is cheaper with respect to costs $\mathbf{c}'$ (the new costs after u raising its bid) than $\{u\}$, and u cannot be part of the winning vertex cover. Thus, the payment to u is at

[1] While the LP is inspired by the analogy of Nash Equilibria, it should be noted that first-price auctions do not in general have Nash Equilibria due to tie-breaking issues (see a more detailed discussion in [14], [15]).

most $\frac{q_u}{q_v} \cdot c_v$. Hence, the total payment of $\mathcal{EV}$ is at most $P(\mathbf{c}) = \frac{1}{q_v} \cdot c_v \cdot \sum_{u \sim v} q_u$. On the other hand, by definition of ν_v and linearity of ν, we have that $\nu(\mathbf{c}) = c_v \nu_v$, so the frugality ratio is

$$\begin{aligned} \frac{\frac{1}{q_v} \cdot c_v \cdot \sum_{u \sim v} q_u}{c_v \nu_v} &= \frac{1}{q_v} \cdot \sum_{u \sim v} \frac{1}{\nu_v} \cdot q_u \\ &= \frac{1}{q_v} \cdot \alpha \cdot q_v \quad = \quad \alpha, \end{aligned}$$

where the second equality followed because the vector $\mathbf{q}$ is an eigenvector of K with eigenvalue α. Thus, for any cost vector with only one non-zero entry, the frugality ratio is at most α.

Now consider an arbitrary cost vector $\mathbf{c}$, and write it as $\mathbf{c} = \sum_v c_v \mathbf{1}_v$. We claim that $P(\mathbf{c}) \leq \sum_v c_v P(\mathbf{1}_v)$. For consider any vertex $u \in S$ winning with cost vector $\mathbf{c}$. When the cost vector is $c_v \mathbf{1}_v$ instead, u's payment is $q_u/q_v \cdot c_v$. On the other hand, when the cost vector is $\mathbf{c}$, if u bids strictly more than $\sum_{v \sim u} q_u/q_v \cdot c_v$, then u cannot be in the winning set, as replacing u with all its neighbors would give a cheaper solution with respect to the costs $\mathbf{c}'$. Thus, each node u gets paid at most $\sum_{v \sim u} q_u/q_v \cdot c_v$ with cost vector $\mathbf{c}$, and the total payment is at most

$$\sum_u \sum_{v \sim u} \frac{q_u}{q_v} \cdot c_v \quad = \quad \sum_v c_v \cdot \sum_{u \sim v} \frac{q_u}{q_v} \quad = \quad \sum_v c_v P(\mathbf{1}_v).$$

On the other hand, we have that

$$\nu(\mathbf{c}) \quad \geq \quad \sum_v c_v \nu(\mathbf{1}_v) \quad = \quad \sum_v c_v \nu_v,$$

because of the following argument: For each v, let $\mathbf{x}^{(v)}$ be a an optimal solution for the LP (1) with cost vector $\mathbf{1}_v$. Then, simply by linearity, the vector $\mathbf{x} = \sum_v c_v \mathbf{x}^{(v)}$ is feasible for the LP (1) with cost vector $\mathbf{c}$, and achieves the sum of the payments. Thus, the optimal solution to the LP (1) with cost vector $\mathbf{c}$ can have no smaller total payments.

Combining the results of the previous two paragraphs, we have the following bound on the frugality ratio:

$$\max_{\mathbf{c}} \frac{P(\mathbf{c})}{\nu(\mathbf{c})} \quad \leq \quad \max_{\mathbf{c}} \frac{\sum_v c_v P(\mathbf{1}_v)}{\sum_v c_v \nu_v} \quad \leq \quad \max_v \frac{P(\mathbf{1}_v)}{\nu_v} \leq \alpha.$$

■

Next, we prove that no other mechanism can do asymptotically better.

Lemma 3.2: Let $\mathcal{M}$ be any truthful vertex cover mechanism on G. Then, $\mathcal{M}$ has frugality ratio at least $\frac{\alpha}{2}$.

Proof: We construct a directed graph G' from G by directing each edge e of G in at least one direction. Consider any edge $e = (u, v)$ of G. Let $\mathbf{c}$ be the cost vector in which $c_u = q_u, c_v = q_v$, and $c_i = 0$ for all $i \neq u, v$. When $\mathcal{M}$ is run on the cost/bid vector $\mathbf{c}$, at least one of u and v must be in the winning set S; otherwise, it would not be a vertex cover. If $u \in S$, then add the directed edge (v, u) to G'. Similarly, if $v \in S$, then add (u, v) to G'. (If both $u, v \in S$, then both edges are added.) By doing this for all edges $e \in G$, we eventually obtain a graph G'.

Now give each node v a weight $w_v = q_v$. Each node-weighted directed graph contains at least one node v such that $\sum_{u:(v,u) \in E} w_u \geq \sum_{u:(u,v) \in E} w_u$ (see, e.g., the proof of Lemma 11 in [15]). Fix any such node v in G' with respect to the weights q_v.

Now consider the cost vector $\mathbf{c}$ with $c_v = q_v$ and $c_i = 0$ for all $i \neq v$. By monotonicity of the selection rule of $\mathcal{M}$ (which follows from the truthfulness of $\mathcal{M}$), at least all nodes u such that $(v, u) \in G'$ must be part of the selected set S of $\mathcal{M}$, and must be paid at least q_u. Therefore, the total payment of $\mathcal{M}$ is at least

$$\begin{aligned} \sum_{u:(v,u) \in G'} q_u &\geq \frac{1}{2} \sum_{u \sim v} q_u \\ &= \frac{1}{2} \nu_v \sum_{u \sim v} \frac{1}{\nu_v} q_u \quad = \quad \frac{1}{2} \nu_v \cdot \alpha q_v, \end{aligned}$$

where the last equality followed from the fact that $\mathbf{q}$ is an eigenvector of the matrix K.

On the other hand, as in the proof of Lemma 3.1, $\nu(\mathbf{c}) = \nu_v q_v$ for our cost vector $\mathbf{c}$, so the frugality ratio under $\mathbf{c}$ is at least $\frac{1}{2}\alpha$. ■

Combining Lemma 3.1 and Lemma 3.2, we have proved the following theorem:

Theorem 3.3: $\mathcal{EV}$ is 2-competitive for Vertex Cover auctions.

Remark 3.4:
1) The lower bound of $\frac{1}{2}\alpha$ on the frugality ratio of any mechanism can potentially be large. For instance, for a complete bipartite graph $K_{n,n}$, we have $\alpha = \Theta(n)$. Thus, such large overpayments are inherent in truthful mechanisms in general. However, truthful mechanisms may be much more frugal on specific classes of graphs.
2) $\mathcal{EV}$ in general does not run in polynomial time. For the final step, computing a minimum-cost vertex cover with respect to the scaled costs, we could use a monotone 2-approximation, as suggested by Elkind et al. [9]. The hardness of computing K is more severe. However, notice that for specific classes of graphs, such as degree-bounded or triangle-free graphs, K can be computed efficiently, giving us non-trivial polynomial-time mechanisms for Vertex Cover on those classes. This issue is discussed more in Section VI.

A. Relationship between Nash Equilibria and the Fractional Clique Problem

Proposition 3.5: Let G_v be the subgraph induced by the neighborhood of v (but without v itself). Then, ν_v is exactly the fractional clique number of G_v (and thus equal to the fractional chromatic number of G_v).

Proof: Recall that the fractional clique number is the solution to the linear program

$$\begin{array}{lll} \text{Maximize} & \sum_u x_u & \\ \text{subject to} & \sum_{u \in I} x_u \leq 1 & \text{for all indep. sets } I \\ & x_u \geq 0 & \text{for all } u \end{array} \quad (2)$$

Let $\mathbf{x}$ be any bid vector feasible for the LP (1). First, for all vertices u that do not share an edge with v, we must have $x_u = 0$, because $V \setminus \{u, v\}$ is a feasible set. So we can restrict our attention to G_v. If I is any independent set in G_v, then $x(I) \leq 1$. The reason is that the set $V \setminus I$ is also feasible, and would cost less than $V \setminus \{v\}$ if the sum of bids exceeded 1. Thus, any feasible bid vector $\mathbf{x}$ induces a feasible solution to the LP (2), of the same total cost.

Conversely, if we have a feasible solution to the LP (2), we can extend it to a bid vector for all agents by setting $x_v = 1$, and $x_u = 0$ for all vertices not neighboring u. We need to show that each feasible set T (vertex cover) has total bid at least as large as the set $V \setminus \{v\}$. If T does not contain v, it must contain all of v's neighbors, and thus has the same bid as $V \setminus \{v\}$ by definition. Otherwise, because $V \setminus T$ is an independent set, the feasibility for the LP (2) implies that $x(V \setminus T) \leq 1$. Thus, $x(T) \geq x(V) - 1 = x(V \setminus \{v\})$. Thus, the two LPs (1) and (2) have the same value. ■

The dual of the Fractional Clique Problem is the Fractional Coloring problem. Since the fractional chromatic number of a graph and the (integer) chromatic number are within a factor $O(\log n)$ of each other (see, e.g., [18]; a simple proof follows from standard randomized rounding arguments), any approximation hardness results for Graph Coloring also apply to the Fractional Clique Problem with at most a loss of logarithmic factors. In particular, the result of Feige and Kilian [11] implies that unless ZPP=NP, ν_v cannot be approximated to within a factor $O(n^{1-\epsilon})$ in polynomial time, for any $\epsilon > 0$.

B. Composability and a General Design Approach

Vertex Cover auctions can be used naturally as a way to deal with other types of set systems: first pre-process the set system by removing a subset of agents, turning the remaining set system into a Vertex Cover instance; then, run $\mathcal{EV}$ on that instance.

The important part is then to choose the pre-processing rule to ensure that the overall mechanism is both truthful and competitive. A condition termed *composability* in [1, Definition 5.2] is sufficient to ensure truthfulness; we show that a comparison between lower bounds is sufficient to show competitiveness.

Definition 3.6 (Composability [1]): Let σ be a selection rule mapping bid vectors to the set of (remaining) agents. σ is *composable* if $\sigma(\mathbf{b}) = T$ implies that $\sigma(b'_e, \mathbf{b}_{-e}) = T$ for any $e \in T$ and $b'_e \leq b_e$. In other words, not only can a winning agent not become a loser by bidding lower; she cannot even change *which* set containing her wins.

Formally, when we talk about "removing" a set of agents from a set system, we are replacing $(E, \mathcal{F})$ with $(T, \mathcal{F}|_T)$, where $T = \sigma(\mathbf{b})$, and $\mathcal{F}|_T := \{S \in \mathcal{F} \mid S \subseteq T\}$.

Theorem 3.7: Let σ be a composable selection rule with the following property: For all monopoly-free set systems $(E, \mathcal{F})$ in the class, and all cost vectors $\mathbf{c}$, writing $(E', \mathcal{F}') := (\sigma(\mathbf{c}), \mathcal{F}|_{\sigma(\mathbf{c})})$: (1) $(E', \mathcal{F}')$ is a Vertex Cover instance, and (2) $\nu_{(E',\mathcal{F}')}(\mathbf{c}) \leq \kappa \cdot \nu_{(E,\mathcal{F})}(\mathbf{c})$.

Let the *Remove-Cover Mechanism* $\mathcal{RCM}$ consist of running $\mathcal{EV}$ on $(E', \mathcal{F}')$. Then, $\mathcal{RCM}$ is a truthful 2κ-competitive mechanism.

Proof: Truthfulness is proved in [1, Lemma 5.3]. The upper bound on the frugality ratio of $\mathcal{RCM}$ follows simply from Lemma 3.1 and the assumption of the theorem:

$$\begin{aligned} P_{\mathcal{RCM}}(\mathbf{c}) &\leq \alpha((E', \mathcal{F}')) \cdot \nu_{(E',\mathcal{F}')}(\mathbf{c}) \\ &\leq \alpha((E', \mathcal{F}')) \cdot \kappa \cdot \nu_{(E,\mathcal{F})}(\mathbf{c}). \end{aligned}$$

To prove the lower bound, let $\mathcal{M}$ be any truthful mechanism for $(E, \mathcal{F})$, and let $(E', \mathcal{F}')$ be the Vertex Cover set system maximizing $\alpha((E', \mathcal{F}'))$. We consider cost vectors $\mathbf{c}$ with $c_e = \infty$ (or some very large finite values) for $e \notin E'$. For such cost vectors, we can safely disregard all elements $e \notin E'$ altogether, as they will not affect the solutions to the LP (1), nor be part of any solution selected by $\mathcal{M}$.

But then, $\mathcal{M}$ is exactly a mechanism selecting a feasible solution to the Vertex Cover instance $(E', \mathcal{F}')$. By Lemma 3.2, $\mathcal{M}$ thus has frugality ratio at least $\alpha((E', \mathcal{F}'))/2$, completing the proof. ■

A simple general way to obtain a composable rule is to minimize the sum of costs:

Lemma 3.8: Let σ be any rule selecting a set S minimizing $b(S)$ over all sets S with a certain property P. Then, σ is composable.

Proof: Consider any agent e who is part of the winning set S with respect to $\mathbf{b}$. If e's bid decreases by ϵ, the cost of S decreases by ϵ, while the costs of all other sets decrease by at most ϵ. Thus, so long as ties are broken consistently, S will still be selected. ■

IV. A Mechanism for Flows

We apply the methodology of Theorem 3.7 to design a mechanism $\mathcal{FM}$ for purchasing k edge-disjoint s-t paths. We are given a (directed) graph $G = (V, E)$, source s, sink t, and target number k. As discussed earlier, the agents are edges of G. We assume that G is monopoly-free, which is equivalent to saying that the minimum s-t cut contains at least $k+1$ edges. For convenience, we will refer to k edge-disjoint s-t paths simply as a k-flow, and omit s and t.

To specify $\mathcal{FM}$, all we need to do is describe a composable pre-processing rule σ. Our rule is simple: Choose $(k+1)$ edge-disjoint s-t paths, of minimum total bid with respect to $\mathbf{b}$; ties are broken lexicographically. We call such a subgraph a $(k+1)$*-flow*, where it is implicit that we are only interested in integer flows, and identify the flow with its edge set. Call the minimum-cost $(k+1)$-flow H. (In Section III-B, we generically referred to this set system as $(E', \mathcal{F}')$.)

Theorem 4.1: The mechanism $\mathcal{FM}$ is truthful and $2(k+1)$-competitive and runs in polynomial time.

We show this theorem in three parts: First, we establish that the k-flow problem on H indeed forms a Vertex Cover instance (Lemma 4.2). By far the most difficult step is showing that the lower bound satisfies $\nu_H(\mathbf{c}) \leq (k+1) \cdot \nu_G(\mathbf{c})$ for all cost vectors $\mathbf{c}$ (Lemma 4.4). The composability of σ follows from Lemma 3.8. Together, these three facts allow us to apply Theorem 3.7, and conclude that $\mathcal{FM}$ is a truthful $2(k+1)$-competitive mechanism. Finally, we verify that $\mathcal{FM}$ runs in polynomial time (Lemma 4.7).

Lemma 4.2: The instance $(E', \mathcal{F}')$ whose feasible sets are all k-flows on H is a Vertex Cover set system.

Proof: The edges of H are the vertices in the Vertex Cover instance. For clarity, consider explicitly the graph R, which contains a vertex u_e for each edge $e \in H$, and an edge between $u_e, u_{e'}$ if and only if removing e would create a monopoly for e'. This is the case iff there exists at least one minimum s-t cut in H containing both e and e'. For any set of edges E' in H, let $N(E')$ be the corresponding set of nodes in R. Thus, for any minimum s-t cut E', the set $N(E')$ forms a clique in R.

If E' is a k-flow, then for any pair of edges e, e' that lie on a minimum s-t cut, E' must contain at least one of e, e'. Thus, $N(E')$ is a vertex cover of R.

Conversely, let E' be a set of edges in H such that $N(E')$ is a vertex cover of R. We will show that for every s-t cut $F \subseteq E$, at least k edges of E' cross F, i.e., $|E' \cap F| \geq k$. This will imply that E' is a k-flow. Assume for contradiction that $|E' \cap F| < k$. Because $N(E')$ is a vertex cover of R, there can be no edge between any pair from $N(F \setminus E')$ in R. By definition, this means that for any pair $e, e' \in F \setminus E'$, there is no minimum s-t cut containing both e and e'. By Proposition 4.3 below, this is equivalent to saying that for each pair $e, e' \in F \setminus E'$, the graph H contains a path from e to e' or a path from e' to e.

Consider a directed graph whose vertices are the edges $F \setminus E'$, with an edge from e to e' whenever H contains a path from e to e'. By the above argument, this graph is a tournament graph, and thus contains a Hamiltonian path. That is, there is an ordering $e_1, \ldots, e_\ell$ of the edges in $F \setminus E'$ such that each e_{i+1} is reachable from e_i in H. By adding a path from s to e_1 and from e_ℓ to t, we thus obtain an s-t path P containing all edges in $F \setminus E'$. The graph $H \setminus P$ is a k-flow, so the set $E' \cap F$, having size less than k, cannot be an s-t cut in $H \setminus P$. Let P' be an s-t path in $H \setminus P$ disjoint from $E' \cap F$. By construction, P' is also disjoint from $F \setminus E'$. Thus, we have found an s-t path P' in H disjoint from F, contradicting the assumption that F is an s-t cut. ■

Next, we show an alternative characterization of which pairs of edges can be in a minimum cut together in H.

Proposition 4.3: Let H be a graph consisting of $k+1$ edge-disjoint s-t paths, and $e = (u, v), e' = (u', v')$ two edges of H. Then, there is a minimum s-t cut containing both e and e' if and only if there is no path from v to u' and no path from v' to u.

Proof: Assume that there is a path from v to u'. Let P be a concatenation of an s-v path using e, the path from v to u', and a path from u' to t using e. Then, $H \setminus P$ is a k-flow, and therefore has k edge-disjoint paths. Any s-t cut in H must thus contain at least k edges from $H \setminus P$, and no s-t cut with fewer than $k+2$ edges can contain both e and e'.

Conversely, if there is no minimum cut containing both e, e', then every minimum cut in $H \setminus \{e, e'\}$ must contain k edges. Thus, $H \setminus \{e, e'\}$ contains k edge-disjoint s-t paths. Removing these paths from H leaves us with a 1-flow, i.e., one s-t path. By construction, this path must contain e and e'; thus, at least one is reachable from the other. ■

Lemma 4.4: $\nu_H(\mathbf{c}) \leq (k+1) \cdot \nu_G(\mathbf{c})$ for all $\mathbf{c}$.

Proof: Let S be the cheapest k-flow in G with respect to the costs $\mathbf{c}$. Because H is a $(k+1)$-flow, Corollary 4.6 below implies that $\nu_H(\mathbf{c}) = k \cdot \pi_H(\mathbf{c})$, where $\pi_H(\mathbf{c})$ is the cost of the most expensive s-t path in H.

Let $\mathbf{x}$ be a solution to the LP (1) with cost vector $\mathbf{c}$ on the graph G. Define a graph G' consisting of all edges in S, as well as all edges that are in at least one tight feasible set T (i.e., sets T for which the constraint (iii) is tight, meaning that $x(T) = x(S)$).

By definition, G' contains (at least) $k+1$ edge-disjoint s-t paths. Lemma 4.5 (the key step) implies that all s-t paths in G' have the same total bid with respect to $\mathbf{x}$. Let P be an s-t path in G' of maximum total cost $c(P)$. By individual rationality (Constraint (i) in the LP (1)), we have that $x(P) \geq c(P)$, and hence $x(P') \geq c(P)$ for all s-t paths P'. In particular, $\nu_G(\mathbf{c}) = x(S) \geq k \cdot c(P)$. By definition of H (and because G' contains at least $k+1$ edge-disjoint s-t paths), we have that $c(P) \geq \frac{\pi_H(\mathbf{c})}{k+1}$, and thus $\nu_G(\mathbf{c}) \geq k \cdot c(P) \geq \frac{k}{k+1} \cdot \pi_H(\mathbf{c}) = \frac{1}{k+1} \cdot \nu_H(\mathbf{c})$, which completes the proof. ■

Lemma 4.5: Let $\mathbf{x}$ be a solution to LP (1), and G' as in the proof of Lemma 4.4. Let v be an arbitrary node in G, and P_1, P_2 two paths from v to t. Then, $x(P_1) = x(P_2)$.

Proof. Let $\mathcal{F}$ be the collection of all tight k-flows from s to t except S, i.e., the set of all F such that $F \neq S$, F consists of exactly k edge-disjoint s-t paths, and $x(F) = x(S)$. We define a directed multi-graph $\tilde{G}$ as follows: for each $F \in \mathcal{F}$, we add to $\tilde{G}$ a copy of each edge $e \in F$ (creating duplicate copies of edges e which are in multiple flows F). We call these edges *forward edges*. In addition, for each edge $e = (u, v) \in S$, we add $|\mathcal{F}|$ copies of the *backward edge* (v, u) to $\tilde{G}$, i.e., we direct e the other way. We define a mapping $\gamma(e)$, which assigns to each edge $e \in \tilde{G}$ its "original" edge in G. As usual, we extend notation and write $\gamma(R) = \{\gamma(e) \mid e \in R\}$ for any set R of edges.

We will be particularly interested in analyzing collections of cycles in $\tilde{G}$. We say that two cycles C_1, C_2 are *image-disjoint* if $\gamma(C_1) \cap \gamma(C_2) = \emptyset$. A *cycle set* is any set of zero or more image-disjoint cycles in $\tilde{G}$ (which we identify with its edge set), and Γ denotes the collection of all cycle sets. For a cycle set $\mathcal{C} \in \Gamma$, let $\mathcal{C}^{\rightarrow}$ and $\mathcal{C}^{\leftarrow}$ denote the set of forward and backward edges in $\mathcal{C}$, respectively. Then, we define $\phi(\mathcal{C}) = S \cup \gamma(\mathcal{C}^{\rightarrow}) \setminus \gamma(\mathcal{C}^{\leftarrow})$. It is easy to see that for each cycle set $\mathcal{C}$, $\phi(\mathcal{C})$ is a k-flow in G'. Conversely, for every k-flow F in G', there is a cycle set $\mathcal{C} \in \Gamma$ with $\phi(\mathcal{C}) = F$.

We assign each edge $e \in \tilde{G}$ a weight w_e. For forward edges e, we set $w_e = x_{\gamma(e)}$, while for backward edges $e = (v, u)$, we set $w_e = -x_{\gamma(e)}$. Notice that because each copy of S contributes weight $-x(S)$, and each set $F \in \mathcal{F}$ contributes $x(F) = x(S)$, the sum of all weights in $\tilde{G}$ is 0.

Now, let $\mathcal{C}$ be any cycle set, and $F = \phi(\mathcal{C})$ its corresponding feasible set. We claim that F is tight if and only if $\sum_{e \in \mathcal{C}} w_e = 0$. To prove this claim, notice that we can write

$$\sum_{e \in \mathcal{C}} w_e \quad = \quad x(F \setminus S) - x(S \setminus F) \quad = \quad x(F) - x(S),$$

which is 0 if and only if $x(F) = x(S)$, which is exactly the definition of a tight set.

We next show that for any cycle C in $\tilde{G}$, the feasible set $\phi(C)$ is tight. Assume for contradiction that this is not the case, and let C be a cycle with $\sum_{e \in C} w_e \neq 0$. Let $F = \phi(C)$ be the corresponding feasible set. Because we showed above that $\sum_{e \in C} w_e = x(F) - x(S)$, we can rule out that $\sum_{e \in C} w_e < 0$; otherwise, $x(F) < x(S)$, which would violate Constraint (iii) of the LP (1).

If $\sum_{e \in C} w_e > 0$, consider the multigraph obtained by removing C from $\tilde{G}$. Its total weight is $\sum_{e \notin C} w_e < 0$, because the sum of all weights in $\tilde{G}$ is 0 (as shown above). $\tilde{G}$ is Eulerian, i.e., the indegree equals the outdegree for all nodes v. For $v \neq s, t$, this follows trivially because each edge set we added constitutes a flow. For $v = s, t$, it follows because each $F \in \mathcal{F}$ adds k edges out of s (and into t), while the $|\mathcal{F}|$ copies of S add $k|\mathcal{F}|$ edges into s and out of t. Because $\tilde{G}$ is Eulerian and we remove a cycle C, the remaining graph is still Eulerian, and its edges can be partitioned into a collection of edge-disjoint cycles $\{C_1, \ldots, C_\ell\}$. By the Pigeon Hole Principle, at least one of the C_i must have negative total weight; for the feasible set $F_i = \phi(C_i)$ corresponding to C_i, we then derive that $x(F_i) < x(S)$, which again violates the constraint (iii) of the LP (1). Thus, we have shown that for each cycle C, the corresponding feasible set is tight.

Finally, we prove the statement of the lemma. We will prove it by induction on a reverse topological sorting of the vertices v. That is, the index of v is at least as large as the index of any u such that $(v, u) \in G'$. Because G' is acyclic, such a topological sorting exists. The base case of vertex t is of course trivial. For $v \neq t$, let P_1, P_2 be two v-t paths. We distinguish three cases, based on the first edges $e_1 = (v, u_1), e_2 = (v, u_2)$ of the paths P_1, P_2.

1) If $\tilde{G}$ contains a forward edge (v, u_1) and a backward edge (u_2, v) (or vice versa), then because each feasible set is a flow, $\tilde{G}$ must contain a v-t path P'_1 entirely consisting of forward edges and starting with e_1, and a t-v path P'_2 entirely consisting of backward edges and ending with e_2 (backward). By induction hypothesis, applied to u_1 and u_2, and because they share their respective first edges, $x(\gamma(P'_1)) = x(P_1)$, and $x(\gamma(P'_2)) = x(P_2)$. Because $P'_1 \cup P'_2$ forms a cycle, and thus has total weight 0, we get that $x(\gamma(P'_2)) = -w_{P'_2} = w_{P'_1} = x(\gamma(P'_1))$, which proves that $x(P_1) = x(P_2)$.
2) If $\tilde{G}$ contains forward edges (v, u_1) and (v, u_2), then it contains v-t paths P'_1, P'_2 starting with (v, u_1) resp. (v, u_2) and consisting entirely of forward edges. By induction hypothesis, $x(\gamma(P'_1)) = x(P_1)$ and $x(\gamma(P'_2)) = x(P_2)$. Because all feasible sets were flows, $\tilde{G}$ must contain an s-v path P consisting entirely of forward edges. And because S is a feasible set, $\tilde{G}$ must also contain a t-s path P' consisting entirely of backward edges. Because $P \cup P' \cup P'_i$ forms a cycle for each i, we obtain that $x(P_i) = x(\gamma(P'_i)) = -w_{P \cup P'}$ for each i; in particular, $x(P_1) = x(P_2)$.
3) Finally, if $\tilde{G}$ contains backward edges (u_1, v) and (u_2, v), we apply a similar argument. We now have that $\tilde{G}$ contains t-v paths P'_1, P'_2 with respective last edges (u_1, v) and (u_2, v), and by induction hypothesis, $x(\gamma(P'_1)) = x(P_1)$, and $x(\gamma(P'_2)) = x(P_2)$. Because S is a flow, $\tilde{G}$ also contains a v-s path P consisting entirely of backward edges, and an s-v path P' consisting entirely of forward edges. Now, the same argument about $P \cup P' \cup P'_i$ as in the previous case shows that $x(P_1) = x(P_2)$. ■

As a corollary, we can derive a characterization of Nash Equilibria in $(k+1)$-flows.

Corollary 4.6: If G is a $(k+1)$-flow, then bids $\mathbf{x}$ are at Nash Equilibrium if and only if $x(P) = \pi_G(\mathbf{c})$ for all s-t paths P. In particular, all Nash Equilibria have the same total cost $x(S) = k \cdot \pi_G(\mathbf{c})$, where S is the winning set.

Proof: First, because G is a $(k+1)$-flow, the graph G' constructed above actually equals G (since it must contain $k+1$ edge-disjoint s-t paths). If $\mathbf{x}$ is Nash Equilibrium bid vector, then by Lemma 4.5, all s-t paths P have the same total bid $x(P)$. Let $\hat{P}$ be an s-t path maximizing $c(P)$, i.e., $c(\hat{P}) = \pi_G(\mathbf{c})$. $G \setminus P$ is a k-flow, and clearly the cheapest k-flow by definition of $\hat{P}$. Therefore, all agents in $\hat{P}$ lose, and $x(\hat{P}) = c(\hat{P})$ by the Constraint (ii) of the LP (1). ■

Finally, we show that the mechanism $\mathcal{EV}$ runs in polynomial time for the special case of graphs derived from k-flows.

Lemma 4.7: For the Vertex Cover instance derived from computing a k-flow on a $(k+1)$-flow, the mechanism $\mathcal{EV}$ runs in polynomial time.

Proof: There are two steps which are of concern: computing the values ν_v, and finding the cheapest vertex cover with respect to the scaled bids. The latter is exactly a

Minimum Cost Flow problem by Proposition 4.2, and thus solvable in polynomial time with standard algorithms [2]. For the former, we claim that $\nu_{u_e} = k$ for all $u_e \in R$. By Proposition 3.5 and LP duality, ν_{u_e} is upper bounded by the chromatic number of u_e's neighborhood, and lower bounded by the clique number. Since each edge $e \in H$ is part of a minimum cut of size $k+1$, and the edges of the minimum cut form a clique in R, the clique number is (at least) k. On the other hand, we can decompose H into $k+1$ edge-disjoint paths, and color (the vertices corresponding to) each path with its own color in R. By Proposition 4.3, this is a valid coloring, and induces a coloring with k colors in the neighborhood of each vertex u_e. ∎

V. A Mechanism for Cuts

As a second application of our methodology, we give a competitive mechanism $\mathcal{CM}$ for purchasing an s-t cut, given a (directed) graph $G = (V, E)$, source s, and sink t. Again, the agents are edges. Here, the necessary monopoly-freeness is equivalent to G not containing the edge (s, t).

Again, it suffices to specify and analyze a composable pre-processing rule σ. Our pre-processing rule is to compute a minimum-cost set E' of edges (with respect to the submitted bids $\mathbf{b}$), such that E' contains at least two edges from each s-t path. We call such an edge set a *double cut*. Since no edge in $E \setminus E'$ can ever win (i.e., be part of a cut), removing $E \setminus E'$ from the set system is equivalent to contracting all edges in $E \setminus E'$. Thus, our pre-processing step can be equivalently characterized as contracting all edges other than E', producing a new graph H. We begin with a simple structural lemma about H.

Lemma 5.1: In H, all s-t paths have length exactly 2.

Proof: If there were an s-t path of length 1 in H, i.e., an edge (s, t), then consider the edge (u, v) in the original graph corresponding to (s, t). Because u was contracted with s, and v with t, there must be an s-u path and a v-t path in G using only edges from $E \setminus E'$. But then, E' cannot have been a double cut. Similarly, if there were an s-t path P of length at least 3, then at least one edge (u, v) of P has neither s nor t as an endpoint. This edge could be safely contracted, i.e., removed from E'. ∎

Theorem 5.2: The double cut selection rule is composable and produces a Vertex Cover instance with $\nu_H(\mathbf{c}) \le 2\nu_G(\mathbf{c})$. Furthermore, both the selection rule and the subsequent Vertex Cover mechanism can be computed in polynomial time. Thus, $\mathcal{CM}$ is a polynomial-time 4-competitive mechanism.

Again, the final conclusion follows from Theorem 3.7, after establishing composability via Lemma 3.8.

To see that we obtain a Vertex Cover instance, notice that Lemma 5.1 implies that H is of the following form: in addition to s and t, there are vertices $v_1, \ldots, v_\ell$, and for each $i = 1, \ldots, \ell$, a set of parallel edges E_i from s to v_i, and a set of parallel edges E'_i from v_i to t. Any s-t cut has to include, for each i, all of E_i or all of E'_i. Thus, we obtain an equivalent 1-flow instance in a minimally 2-connected graph by having two vertex-disjoint paths of length $|E_i|$ and $|E'_i|$ between u_i and u_{i+1} for each i, and setting $s = u_1$ and $u_{\ell+1} := t$. We can then apply Lemma 4.2. Notice that this equivalence also establishes that $\mathcal{EV}$ runs in polynomial time on the instances produced by this selection rule.

As before, the key part is to analyze the increase in the lower bound.

Lemma 5.3: For all cost vectors $\mathbf{c}$, $\nu_H(\mathbf{c}) \le 2\nu_G(\mathbf{c})$.

Proof: Let $(S, \overline{S})$ the cheapest s-t cut in G with respect to the costs $\mathbf{c}$, and $\mathbf{x}$ a solution to the LP (1) with cost vector $\mathbf{c}$ on the graph G. Let $\mathcal{C}$ be the set of all minimum s-t cuts $(T, \overline{T})$ with respect to the costs $\mathbf{x}$; thus, each of these cuts has cost $x(E(S, \overline{S}))$. Define $T^- = \bigcap_{(T,\overline{T}) \in \mathcal{C}} T$, and $T^+ = \bigcup_{(T,\overline{T}) \in \mathcal{C}} T$. Then, both $(T^-, \overline{T^-})$ and $(T^+, \overline{T^+})$ are minimum s-t cuts as well (see, e.g., [2, Exercise 6.39]).

Furthermore, the edge sets $E(T^-, \overline{T^-})$ and $E(T^+, \overline{T^+})$ are disjoint. For assume that there were an edge $e = (u, v)$ in common between these sets. Then, $u \in \bigcap_{(T,\overline{T}) \in \mathcal{C}} T$ and $v \in \bigcap_{(T,\overline{T}) \in \mathcal{C}} \overline{T}$. In particular, this implies that $u \in S$ and $v \in \overline{S}$. By maximality of the solution $\mathbf{x}$ for the LP (1), there must be a constraint (iii) tight for e; otherwise, x_e could be increased. Let $(T, \overline{T})$ be the cut corresponding to the tight constraint. Thus, $x(E(T, \overline{T})) = x(E(S, \overline{S}))$, and e does not cross $(T, \overline{T})$. Thus, either both u and v are in T, or both are in $\overline{T}$. Because $(T, \overline{T}) \in \mathcal{C}$, this gives a contradiction.

Now define $G' := E(T^-, \overline{T^-}) \cup E(T^+, \overline{T^+})$. Because G' consists of two disjoint s-t cuts, the cost-minimality of H implies that $c(G') \ge c(H)$. By the "individual rationality" LP constraint (i), $x(G') \ge c(G')$, and hence

$$\nu_G(\mathbf{c}) = \frac{x(G')}{2} \ge \frac{c(G')}{2} \ge \frac{c(H)}{2} \ge \frac{\nu_H(\mathbf{c})}{2}.$$

For the last inequality, notice that in the "Nash Equilibrium" on H, for each i, the cheaper of E_i and E'_i will collectively raise their bids to the cost of the more expensive one, so the total bid of the winning set will be at most $\sum_i \max(c(E_i), c(E'_i)) \le c(H)$. ∎

The final claim required to prove Theorem 5.2 is that a minimum-cost double cut can be computed in polynomial time. To this end, we can write the natural linear program for a minimum-cost double cut. One can then prove that it is totally unimodular, and hence has an integral optimum. A more efficient algorithm uses the primal-dual approach, resulting in a natural generalization of the Ford-Fulkerson algorithm. Details of the analysis for both algorithms can be found in the full version of this paper.

VI. Directions for Future Work

In general, the Vertex Cover mechanism does not run in polynomial time, due to two obstacles: first, computing the matrix K requires computing the largest fractional clique size in the neighborhood of v, for each node v. Subsequently, computing the solution with respect to scaled costs requires finding a cheapest vertex cover. For the second obstacle, it seems quite likely that monotone algorithms (such as the one in [9]) could be adapted to our setting, and yield constant-factor approximations. However, the difficulty of computing the entries of K seems more severe. In fact, we conjecture that no polynomial-time truthful mechanism for Vertex Cover can be constant competitive. This result would be quite interesting, in that it would show that the requirements of incentive-compatibility and computational tractability together can lead to significantly worse guarantees than either requirement alone.

While our methodology of designing composable preprocessing algorithms will likely be useful for other problems as well, it does not apply to all set systems. It is fairly easy to construct set systems for which no such pruning algorithm is possible. Even when pruning is in principle possible, it may come with a large blowup in costs.

Thus, the following bigger question still stands: which classes of set systems admit constant-competitive mechanisms? The main obstacle is our inability to prove strong lower bounds on frugality ratios. To date, all lower bounds (here, as well as [10], [15]) are based on pairwise comparisons between agents, which can then be used to show that certain agents, by virtue of losing, will cause large payments. This technique was exactly the motivation for the Vertex Cover approach. In order to move beyond Vertex Cover based mechanisms, it will be necessary to explore lower bound techniques beyond the one used in this paper.

ACKNOWLEDGMENT

We would like to thank Edith Elkind, Uriel Feige, Anna Karlin, Tami Tamir, and Mihalis Yannakakis for useful discussions and pointers, and anonymous reviewers for useful feedback.

REFERENCES

[1] Gagan Aggarwal and Jason D. Hartline. Knapsack auctions. In *Proc. 17th ACM Symp. on Discrete Algorithms*, pages 1083–1092, 2006.

[2] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows*. Prentice Hall, 1993.

[3] Aaron Archer and Eva Tardos. Frugal path mechanisms. In *Proc. 13th ACM Symp. on Discrete Algorithms*, pages 991–999, 2002.

[4] Sushil Bikhchandani, Sven de Vries, James Schummer, and Rakesh Vohra. Linear programming and Vickrey auctions. *IMA Volume in Mathematics and its Applications, Mathematics of the Internet: E-auction and Markets*, 127:75–116, 2001.

[5] Gruia Calinescu. Bounding the payment of approximate truthful mechanisms. In *Proc. 15th Intl. Symp. on Algorithms and Computation*, pages 221–233, 2004.

[6] Ning Chen, Edith Elkind, Nick Gravin, and Fedor Petrov. Frugal mechanism design via spectral techniques. In *Proc. 51st IEEE Symp. on Foundations of Computer Science*, 2010.

[7] Ning Chen and Anna Karlin. Cheap labor can be expensive. In *Proc. 18th ACM Symp. on Discrete Algorithms*, pages 707–715, 2007.

[8] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971.

[9] Edith Elkind, Leslie Goldberg, and Paul Goldberg. Frugality ratios and improved truthful mechanisms for vertex cover. In *Proc. 9th ACM Conf. on Electronic Commerce*, pages 336–345, 2007.

[10] Edith Elkind, Amit Sahai, and Kenneth Steiglitz. Frugality in path auctions. In *Proc. 15th ACM Symp. on Discrete Algorithms*, pages 701–709, 2004.

[11] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998.

[12] Rahul Garg, Vijay Kumar, Atri Rudra, and Akshat Verma. Coalitional games on graphs: core structures, substitutes and frugality. In *Proc. 5th ACM Conf. on Electronic Commerce*, pages 248–249, 2003.

[13] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.

[14] Nicole Immorlica, David R. Karger, Evdokia Nikolova, and Rahul Sami. First-price path auctions. In *Proc. 7th ACM Conf. on Electronic Commerce*, pages 203–212, 2005.

[15] Anna Karlin, David Kempe, and Tami Tamir. Beyond VCG: Frugality of truthful mechanisms. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, pages 615–624, 2005.

[16] Paul Klemperer. Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3):227–286, 1999.

[17] Vijay Krishna. *Auction Theory*. Academic Press, 2002.

[18] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *Journal of the ACM*, 41(5):960–981, 1994.

[19] Andreu Mas-Collel, Michael D. Whinston, and Jerry R. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[20] Noam Nisan and Amir Ronen. Algorithmic mechanism design. In *Proc. 31st ACM Symp. on Theory of Computing*, pages 129–140, 1999.

[21] Christos Papadimitriou. Algorithms, games and the Internet. In *Proc. 33rd ACM Symp. on Theory of Computing*, pages 749–752, 2001.

[22] Kunal Talwar. The price of truth: Frugality in truthful mechanisms. In *Proc. 21st Annual Symp. on Theoretical Aspects of Computer Science*, pages 608–619, 2003.

[23] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *J. of Finance*, 16:8–37, 1961.

[24] Qiqi Yan. On the price of truthfulness in path auctions. In *Proc. 3rd Workshop on Internet and Network Economics (WINE)*, pages 584–589, 2007.