Truthful Allocation in Graphs and Hypergraphs

George Christodoulou
University of Liverpool, UK

Elias Koutsoupias
University of Oxford, UK

Annamária Kovács
Goethe University, Frankfurt am Main, Germany

Abstract

We study truthful mechanisms for allocation problems in graphs, both for the minimization (i.e., scheduling) and maximization (i.e., auctions) setting. The minimization problem is a special case of the well-studied unrelated machines scheduling problem, in which every given task can be executed only by two pre-specified machines in the case of graphs or a given subset of machines in the case of hypergraphs. This corresponds to a multigraph whose nodes are the machines and its hyperedges are the tasks. This class of problems belongs to multidimensional mechanism design, for which there are no known general mechanisms other than the VCG and its generalization to affine minimizers. We propose a new class of mechanisms that are truthful and have significantly better performance than affine minimizers in many settings. Specifically, we provide upper and lower bounds for truthful mechanisms for general multigraphs, as well as special classes of graphs such as stars, trees, planar graphs, k-degenerate graphs, and graphs of a given treewidth. We also consider the objective of minimizing or maximizing the L^p-norm of the values of the players, a generalization of the makespan minimization that corresponds to $p = \infty$, and extend the results to any $p > 0$.

2012 ACM Subject Classification Theory of computation → Algorithmic mechanism design

Keywords and phrases Algorithmic Game Theory, Scheduling Unrelated Machines, Mechanism Design

Digital Object Identifier 10.4230/LIPIcs.ICALP.2021.56

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2106.03724

Funding *Elias Koutsoupias*: This work was partially supported by ERC Advanced Grant 321171 (ALGAME).

1 Introduction

This work belongs to the area of mechanism design, one of the most researched branches of Game Theory and Microeconomics with numerous applications in environments where a protocol of conduct of selfish participants is required. The goal is to design an algorithm, called mechanism, which is robust under selfish behavior and that produces a social outcome with a certain guaranteed quality. The mechanism solicits the preferences of the participants over the outcomes, in forms of bids, and then selects one of the outcomes. The challenge stems from the fact that the real preferences of the participants are private, and the participants care only about maximizing their private utilities and hence they will lie if a false report is profitable. A *truthful* mechanism provides incentives such that a truthful bid is the best action for each participant.

Despite the importance of the problem the only general positive result for multi-dimensional domains is the celebrated Vickrey-Clarke-Groves (VCG) mechanism [43, 15, 26] and its affine extensions, known as affine maximizers.

In their seminal paper on algorithmic mechanism design, Nisan and Ronen [39] proposed the scheduling problem on unrelated machines as a central problem to understand the algorithmic aspects of mechanism design. The objective is to incentivize n machines to execute m tasks, so that the maximum completion time of the machines, i.e. the makespan, is minimized. Scheduling, a problem that has been extensively studied from the classical algorithmic perspective, proved to be the perfect ground to study the limitations that truthfulness imposes on algorithm design.

Nisan and Ronen applied the VCG mechanism, the most successful generic machinery in mechanism design, which truthfully implements the outcome that maximizes the social welfare. In the case of scheduling, the allocation of the VCG is the greedy allocation in which each task is assigned to the machine with minimum processing time. This mechanism is truthful, but has a poor approximation ratio of n for the makespan. They conjectured that this is the best guarantee that can be achieved by any deterministic (polynomial-time or not) truthful mechanism and this conjecture, known as the Nisan-Ronen conjecture, is widely perceived as the holy grail in algorithmic mechanism design.

An interesting special case of the scheduling problem, which is well-understood, is the single-dimensional mechanism design in which the values of each player are linear expressions of a single parameter. The principal representative is the problem of scheduling *related* machines, where the cost of each machine can be expressed via a single parameter, its *speed*. This was first studied by Archer and Tardos [1], who showed that in contrast to the unrelated machines version, an algorithm that minimizes the makespan can be truthfully implemented – albeit in exponential time. It was subsequently shown that truthfulness has essentially no impact on the computational complexity of the problem. Specifically, a randomized truthful-in-expectation[1] PTAS was given in [18] and a deterministic PTAS was given in [14]; a PTAS is the best possible algorithm even for the pure algorithmic problem (unless $P = NP$).

1.1 Summary of Results

In this work, we show how to combine these two main positive results of VCG and single-dimensional mechanisms into a single mechanism, which we call the *Hybrid Mechanism*. This new mechanism applies to domains in which some players are multidimensional and some players are single-dimensional. A typical example is to schedule m tasks, such that task i can only be executed by player 0 and player i. In this case, player 0 is multidimensional and the other m players are single-dimensional. We call this the *star balancing* problem. This is a multidimensional mechanism design problem for which the VCG mechanism, as well as every other known mechanism, performs very poorly. However, as we show in Section 3.1, the Hybrid Mechanism has approximation ratio 2, optimal among all truthful mechanisms. We generalize the star balancing problem in three directions: *graphs/multigraphs*, *hyperstars* and also to objectives other than makespan minimization. Due to space limitations, omitted proofs are presented in the full version.

(Multi)Graphs. A generalization of the star balancing problem to graphs and multigraphs is the Unrelated Graph Balancing problem (Section 3). This is a special case of unrelated machines scheduling in which there is a (multi)graph whose nodes represent the machines

[1] This is one of the two main definitions of truthfulness for randomized mechanisms, where truth-telling maximizes the expected utility of each player.

and whose edges represent tasks that can be executed only by the incident nodes. For general graphs, all machines are multiparameter, but we can still apply the Hybrid Mechanism, if we first decompose the graph into stars and then apply the Hybrid Mechanism to each one of them. The combined mechanism, which we call the *Star-Cover Mechanism*, has surprisingly good approximation ratio for certain classes of graphs – ratio 4 for trees, 8 for planar graphs, and $2k+2$ for k-degenerate graphs (Corollary 15). These results use as ingredient the analysis of star graphs, in which the Hybrid Mechanism has approximation ratio 2.

Hyperstars. In the hyperstar version, there are k multidimensional players/machines and every task can be executed by any one of these k players or by a task-specific single-dimensional player. Specifically, there are k different *root players* (players $1, 2, \ldots, k$ with bids $(r_{ij})_{k\times m}$) and each of them are allowed to process all tasks. In addition, for each task there is one *leaf player*, which can process only this single task (players $k+1, k+2, \ldots, k+m$ with bids $(\ell_1, \ell_2, \ldots \ell_m)$). Note that the root players without the leaves form a classic input for unrelated scheduling mechanisms with k players and m-tasks. We can now state the Hybrid Mechanism for this case.

▶ **Definition 1** (Hybrid Mechanism). *The Hybrid Mechanism minimizes*

$$\min_T \left\{ \left(\min_{x^T} \sum_{i=1}^{k} \lambda_i r_i \cdot x_i^T \right) + g_T(\ell) \right\},$$

where the λ_i can be arbitrary non-negative real numbers and $(g_T)_{T\subseteq M}$ can be any functions that guarantee that the leaf players are truthful. The output of the mechanism is the subset of tasks T that are allocated to the multidimensional root players together with their allocation matrix x^T. The remaining tasks, $M \setminus T$, are allocated to the leaf players.

VCG fairs poorly, yielding approximation ratio m in this domain, but the Hybrid Mechanism has approximation ratio $k+1$, as stated in the next theorem. Due to space limitations, we provide details and proofs in the full version of the paper.

▶ **Theorem 2.** *For the hyperstar scheduling problem, the Hybrid Mechanism with $g_T(\ell) = \max_{j\notin T} \ell_j$, and with $\lambda_i = 1$, for every i, is $(k+1)$-approximate.*

In Section 4 we provide general definitions as well as necessary and sufficient conditions for truthfulness of the Hybrid Mechanism.

Mechanisms for L^p-norm optimization. In Section 5, we consider the much more general objective of minimizing or maximizing the L^p-norm of the values of the players, for $p > 0$. The scheduling problem is the special case of minimizing the L^∞-norm. We show that the Hybrid Mechanism performs very well for this much more general problem, and in some cases it has the optimal approximation ratio among all truthful mechanisms. This illustrates the applicability and usefulness of the Hybrid Mechanism in applications with various domains and objectives. We emphasize that for all these cases, even for stars, all known mechanisms such as the VCG and affine maximizers have very poor performance.

Relation to the Nisan-Ronen conjecture. Our results on (multi)graphs show that this domain may provide an easier way to attack the Nisan-Ronen conjecture. In a recent work [12], we showed a $\Omega(\sqrt{n})$ lower bound for multistars with edge multiplicity only 2, when the root player has submodular or supermodular valuations. In contrast, our results in this work show that for additive valuations, the Star-Cover Mechanism has approximation ratio 4 on the very same multigraphs. However, the Hybrid and the Star-Cover Mechanisms have high approximation for multistars with high edge-multiplicity or for simple clique graphs. It is natural to ask whether there are other, better mechanisms for these cases. Recently we have proved a $\Omega(\sqrt{n})$ lower bound for the former case, which is the first super-constant lower bound for the Nisan-Ronen problem [11], and we conjecture that the latter case admits similarly a high, perhaps even linear, lower bound.

We remark that all previous lower bound proofs use inherently either (multi)graphs [13, 29, 11] or, recently, hypergraphs with hyperedges of small size [24, 20]. Our work provides new methodological tools to study these objects, that can help to identify certain (hyper)graph structures as good candidates for high lower bounds and to avoid those where low upper bounds exist. For example, the 2.755 lower bound construction of [24] uses a hyperstar with $k = 2$, for which the Hybrid Mechanism achieves an upper bound of 3 (Thm 2).

All our lower bounds are information theoretic and hold independently of the computational time of the mechanisms. Conversely, all upper bounds are polynomial time algorithms when the star decomposition is given. We leave it open whether computing an optimal star decomposition of a graph is in P, although it follows from our results that it can be approximated with an additive term of 1 in polynomial time (actually in linear time).

1.2 Related Work

The Nisan-Ronen conjecture [39] has become one of the central problems in Algorithmic Game Theory, and despite intensive efforts it remains open. The original paper showed that no truthful deterministic mechanism can achieve an approximation ratio better than 2 for two machines, which was later improved to 2.41 [13] for three machines, and finally to 2.618 [29] which was the best known bound for over a decade. Recent progress improved this bound to 2.755 [24], to 3 [20] and finally to the first non-constant lower bound of $1 + \sqrt{n-1}$ [11]. The best known upper bound is n [39].

The purely algorithmic problem of makespan minimization on unrelated machines is one of the most important scheduling problems. The seminal paper of Lenstra, Shmoys and Tardos [32], gave a 2-approximation algorithm, and also showed that it is NP-hard to approximate within a factor of 3/2. Closing this gap has remained open for 30 years, and is considered one of the most important open questions in scheduling.

In this work we consider the design of truthful mechanisms for the *Unrelated Graph Balancing* problem, a special but quite rich case of the unrelated machines problem, which was previously studied by Verschae and Wiese [42], for which each task can only be assigned to two machines. This can be formulated as a graph problem, where given an undirected (multi)-graph $G = (V, E)$, each vertex corresponds to a machine, and each edge corresponds to a task. The goal is to allocate each edge to one of its nodes, in a way that minimizes the maximum (weighted) in-degree.

The special case of this problem where each direction of an edge corresponds to the same processing time $t(e)$ is known as Graph Balancing, and was introduced by Ebenlendr, Krcál, and Sgall [21] who showed an 1.75-approximate algorithm and also demonstrated that the problem retains the hardness of the unrelated machines problem, by showing that it is NP-hard to approximate within a factor better than 3/2.

Graph Balancing. As was already mentioned, for the pure graph balancing problem, the best approximation ratio for classical polynomial time algorithms is 1.75 by [21]. Wang and Sitters [44] showed a different LP-based algorithm with a higher ratio of $11/6 \approx 1.83$, while Huang and Ott [27] designed a purely combinatorial approximation algorithm but with also a higher guarantee of 1.857.

Jansen and Rohwedder [28] studied the so-called *configuration LP* which was introduced by Bansal and Sviridenko [6]. They showed that it has an integrality gap of at most 1.749 breaking the 1.75 barrier of the integrality gaps of the previous LP formulations. This leaves open the possibility of using this LP to produce an approximation algorithm with a ratio better than 1.75.

Verschae and Wiese [42] studied the *unrelated* version of graph balancing (whose strategic variant we consider in this paper) and showed that the integrality gap of the configuration LP is equal to 2, which is much higher comparing to graph balancing. They also showed a 2-approximation algorithm for the problem of maximizing the minimum load, which is the best possible unless P=NP.

The problem has been studied for various special graph classes. For the case of simple graphs (also known as Graph Orientation), Asahiro et al [2] showed that the problem is in P for the case of trees, while Asahiro, Miyano and Ono [3] showed that it becomes strongly NP-hard for planar and bipartite graphs. Finally, Lee, Leung and Pinedo [31] concluded the case of trees in the case of multiple edges, showing an FPTAS which is the best possible, given that the problem in multi-graphs is immediately NP-hard even for the simple case of two vertices (due to reduction from Subset Sum).

Truthful Scheduling. The lack of progress in the original unrelated machine problem led to the study of special cases where progress has been made. Ashlagi et al.[4], resolved a restricted version of the Nisan-Ronen conjecture, for the special but natural class of *anonymous* mechanisms. Lavi and Swamy [30] studied a restricted input domain which however retains the multi-dimensional flavour of the setting. They considered inputs with only two possible values “low” and “high”, that are publicly known to the designer. For this case they showed an elegant deterministic mechanism with an approximation factor of 2. They also showed that even for this setting achieving the optimal makespan is not possible under truthfulness, and provided a lower bound of 11/10. Yu [45] extended the results for a range of values, and Auletta et al. [5] studied multi-dimensional domains where the private information of the machines is a single bit.

Randomization has led to mildly improved guarantees. There are two extensions of truthfulness for randomized mechanisms; *universal truthfulness* if the mechanism is described as a probability distribution over deterministic truthful mechanisms, and *truthfulness-in-expectation*, if in expectation no player can benefit by lying. The former notion was first considered in [39] for two machines, it was later extended to n machines by Mu’alem and Schapira [38] and finally Lu and Yu [35] showed a $0.837n$-approximate mechanism, which is currently the best known. Lu and Yu [36] showed a truthful-in-expectation mechanism with an approximation guarantee of $(m + 5)/2$. Mu’alem and Schapira [38], showed a lower bound of $2 - 1/m$, for both notions of randomization. Christodoulou, Koutsoupias and Kovács [10] extended this lower bound for fractional mechanisms, where each task can be split to multiple machines, and they also showed a fractional mechanism with a guarantee of $(m + 1)/2$. The special case of two machines [34, 36] is still unresolved; currently, the best upper bound is 1.587 due to Chen, Du, and Zuluaga [9].

The case of *related* machines is well understood. It falls into the so-called *single-dimensional* mechanism design in which the valuations of a player are linear expressions of a single parameter. In this case, the cost of each machine is expressed via a single parameter, its *(inverse) speed* multiplied by the workload allocated to the machine, instead of an m-valued vector, as it is the case for the unrelated machines and the Graph Balancing setting. Archer and Tardos [1] showed that, in contrast to the unrelated machines version, the optimal makespan can be achieved by an (exponential-time) truthful algorithm, while [14] gave a deterministic truthful PTAS which is the best possible even for the pure algorithmic problem (unless P=NP).

Truthful implementation of other objectives was considered by Mu'alem and Schapira [38] for multi-dimensional problems and by Epstein, Levin and van Stee [22] for single-dimensional ones. Leucci, Mamageishvili and Penna [33] demonstrated high lower bounds for other min-max objectives on some combinatorial optimization problems on graphs, showing essentially that VCG is the best mechanism for these problems. Minooei and Swamy [37] considered a multi-dimensional vertex cover problem, and approached in by decomposition into single parameter problems.

The Bayesian setting, where the players costs are drawn from a probability distribution has also been studied. Daskalakis and Weinberg [17] showed a mechanism that is at most a factor of 2 from the *optimal truthful mechanism*, but not with respect to the optimal makespan. Chawla et al. [8] provided bounds of prior-independent mechanisms (where the input distribution is unknown to the mechanism), while Giannakopoulos and Kyropoulou [25] showed that the VCG mechanism achieves a factor of $O(\log n/\log\log n)$ under some distributional and symmetry assumptions.

Recently Christodoulou, Koutsoupias, and Kovács [12] showed a lower bound of $\sqrt{n-1}$ for all deterministic truthful mechanisms, when the cost of processing a subset of tasks is given by a submodular (or supermodular) set function, instead of an additive function which is assumed in the standard scheduling setting.

2 Preliminaries

Scheduling. In the classical *unrelated machines scheduling* there is a set N of n machines and a set M of m tasks that need to be scheduled on the machines. The input is given by a nonnegative matrix $t = (t_{ij})_{n\times m}$: machine i needs time $t_{ij} \in \mathbb{R}_{\geq 0}$ to process task j, and her *costs* are additive, i.e., the processing time for machine i for a set of tasks $X_i \subset M$ is $t_i(X_i) := \sum_{j\in X_i} t_{ij}$. The objective is to minimize the makespan (min-max objective). An allocation to all machines $X = (X_1, X_2, \ldots, X_n)$, (which is a partition of M) can also be denoted by the characteristic matrix $x = (x_{ij})$ where $x_{ij} = 1$ if $j \in X_i$, and $x_{ij} = 0$ otherwise.

The current work essentially considers a special case of unrelated scheduling, in which every task can be processed by two designated machines. The tasks can thus be modelled by the edges of a graph, and the associated problem is also known as *Unrelated Graph Balancing*. More formally, in the Unrelated Graph Balancing problem, there is a given undirected graph $G = (V, E)$; the vertices correspond to a set of machines $N = V$ and the edges to a set of tasks $M = E$. For each edge $e \in E$ only its two incident vertices can process the job e, and they have in general different processing times $t_i(e)$, and $t_{i'}(e)$. The goal is to assign a direction to each edge $e = (i, i')$ (allocate the corresponding task) of the graph, to one of the incident vertices (machines). The *completion time* of each vertex i is then the total processing time of the jobs X_i assigned to it $t_i(X_i) = \sum_{e\in X_i} t_i(e)$. The objective is to find an allocation that minimizes the *makespan*, i.e. the maximum completion time over all vertices.

Mechanism design setting. We assume that each machine $i \in N$ is controlled by a selfish agent that is reluctant to process the tasks and the cost function t_i is private information (also called the *type* of agent i). A *mechanism* asks the agents to report *(bid)* their types t_i, and based on the collected bids it allocates the jobs, and gives payments to the agents. A player may report a false cost function $b_i \neq t_i$, if this serves her interests.

Formally, a mechanism (X, P) consists of two parts:

An allocation algorithm: The allocation algorithm X allocates the tasks to the machines depending on the players' bids $b = (b_1, \ldots, b_n)$. We denote by $X_i(b)$ the subset of tasks assigned to machine i in the bid profile b.

A payment scheme: The payment scheme $P = (P_1, \ldots, P_n)$ determines the payments also depending on the bid values b. The functions $P_1, \ldots, P_n$ stand for the payments that the mechanism hands to each agent.

The *utility* u_i of a player i is the payment that she gets minus the *actual* time that she needs to process the set of tasks assigned to her, $u_i(b) = P_i(b) - t_i(X_i(b))$. We are interested in *truthful* mechanisms. A mechanism is truthful, if for every player, reporting his true type is a *dominant strategy*. Formally,

$$u_i(t_i, b_{-i}) \geq u_i(t'_i, b_{-i}), \qquad \forall i \in N, \;\; t_i, t'_i \in \mathbb{R}^m_{\geq 0}, \;\; b_{-i} \in \mathbb{R}^{(n-1)\times m}_{\geq 0},$$

where b_{-i} denotes the reported bidvectors of all players disregarding i.

We are looking for *truthful* mechanisms with *low approximation ratio* of the allocation algorithm for the makespan irrespective of the running time to compute X and P. In other words, our lower bounds are information-theoretic and do not take into account computational issues.

A useful characterization of truthful mechanisms in terms of the following monotonicity condition, helps us to get rid of the payments and focus on the properties of the allocation algorithm.

▶ **Definition 3** (Weak Monotonicity). *An allocation algorithm X is called* weakly monotone (WMON) *if it satisfies the following property: for every two inputs $t = (t_i, t_{-i})$ and $t' = (t'_i, t_{-i})$, the associated allocations X and X' satisfy $t_i(X_i) - t_i(X'_i) \leq t'_i(X_i) - t'_i(X'_i)$.*

It is well known that the allocation function of every truthful mechanism is WMON [7], and also that this is a sufficient condition for truthfulness in convex domains [41].

The following lemma was essentially shown in [39] and has been a useful tool to show lower bounds for truthful mechanisms for several variants (see for example [13, 38]).

▶ **Lemma 4.** *Let t be a bid vector, and let $S = X_i(t)$ be the subset assigned to player i by a weakly monotone allocation X. For any bid vector $t' = (t'_i, t_{-i})$ such that only the bid of machine i has changed and in such a way that for every task in S it has decreased (i.e., $t'_{ij} < t_{ij}, j \in S$) and for every other task it has increased (i.e., $t'_{ij} > t_{ij}, j \in M \setminus S$). Then the mechanism does not change the allocation to machine i, i.e., $X_i(t') = X_i(t) = S$.*

In general, when the values of a machine change, the allocation of the other machines may change, this issue being the pivotal difficulty of truthful unrelated scheduling. Allocation algorithms that "promise" not to change the allocation of other machines as long as changing (only) t_i does not affect the set X_i, are less problematic. These allocation rules are called *local* in [39], where it is shown that local truthful mechanisms cannot have a better than n approximation.

▶ **Definition 5** (Local mechanisms). *A mechanism is* local *if for every $i \in N$, for every t_{-i}, and t_i, t'_i for which $X_i(t_i, t_{-i}) = X_i(t'_i, t_{-i})$ also holds that $X_j(t_i, t_{-i}) = X_j(t'_i, t_{-i}) \quad (\forall j \in N)$.*

There are several special classes of mechanisms that satisfy this property, perhaps the most prominent one is the class of *affine minimizers* (see, e.g., [13]).

3 Graph Balancing

In this section we focus on the (Unrelated) Graph Balancing problem, which is a special case of makespan minimization of scheduling unrelated machines. The Graph Balancing is a multi-parameter mechanism design problem that retains most of the difficulty of the Nisan-Ronen conjecture, yet has certain features that make it more amenable.

One of the difficulties in dealing with truthful mechanisms is that while truthfulness is a local property (i.e., independent truthfulness conditions, one per player), the allocation algorithm is a global function (that involves all players). Local algorithms attempt to reconcile this tension by insisting that the allocation is also "local", but they take this notion too far. The results of this work show that locality in mechanisms is very restrictive in some domains, where the Hybrid Mechanism outperforms every local mechanism.

The Graph Balancing problem is more amenable than the general scheduling problem because it exhibits another kind of locality, *domain locality*: when a machine does not get a task, we know which machines gets it. Yet, this locality is not very restrictive and the problem retains most of its original difficulty.

In this section, we take advantage of domain locality to obtain an optimal mechanism for stars. It turns out that this mechanism, the Hybrid Mechanism, is a special case of a more general mechanism. But since the Hybrid Mechanism does not apply to general graphs, we also propose the Star-Cover mechanism for general graphs: decompose the graph into stars and apply the Hybrid Mechanism independently to each star. In this way, we obtain a 4-approximation algorithm for trees and similar positive results for other types of graphs.

Makespan minimization is the special case, when $p = \infty$, of minimizing the L^p-norm of the values of the players. Other special cases of the L^p-norm optimization is the case $p = 1$, which corresponds to welfare maximization, and the case $p = 0$, which is related to Nash Social Welfare [16]. We deal with this more general problem in another section (Section 5). Most of the results and proofs of this section generalize to any $p \geq 1$. We provide almost all the proofs in this section, because we believe that the Graph Balancing problem is an important problem in its own right and because the treatment is simpler and more intuitive, and we omit most of the (more general) results of Section 5 that deals with the L^p-norm minimization, due to space limitations.

3.1 Stars and the Hybrid Mechanism

In this subsection, we focus on star graphs, where there are $n = m + 1$ players and m tasks. Player 0 is the root of the star, and has processing times given by a vector $r = (r_1, r_2, \dots r_m)$. We also refer to this player as the *root player* or *r-player*. For given bids r of the root player, and task set $T \subseteq M$ we use the short notation $r(T) = \sum_{j \in T} r_j$.

There are also m *leaf-players*, one for each leaf of the star with processing times $\ell = (\ell_1, \dots, \ell_m)$ respectively. Each task j can only be assigned to two players; either to the root, with processing time r_j, or to the leaf with processing time ℓ_j.

As usual, we denote by r_{-i} the vector of bids of the root player except for the bid for task i, and similarly ℓ_{-i} denotes the bids of all leaf-players, except for player i. The vector of all input bids is given by $t = (r, \ell)$.

As we show later in the Lower Bound section (Section 3.3), all previously known mechanisms for the Unrelated Graph Balancing problem, e.g. affine minimizers and task independent mechanisms, have approximation ratio at least $\sqrt{n-1}$ for graphs, even for stars.

In contrast, we now show that the Hybrid Mechanism has constant approximation ratio for stars.

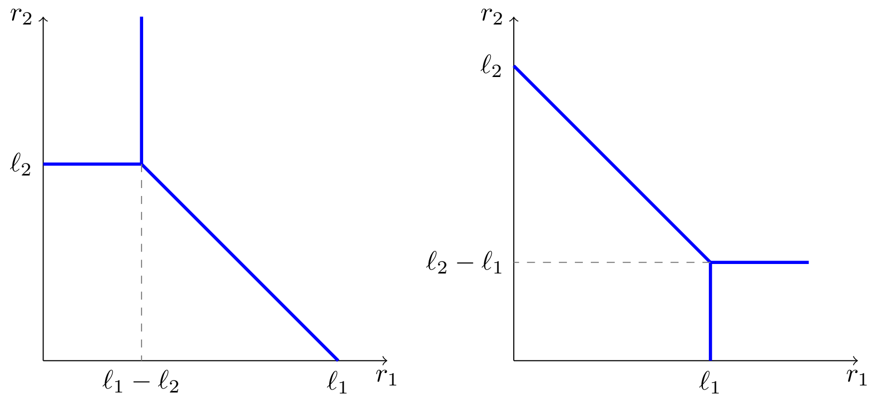

Figure 1 An instance of the Hybrid Mechanism, for the star of $m = 2$ leaves. It shows the partition of bid-space of the root player induced by the allocation of the Hybrid Mechanism when $\ell_1 \geq \ell_2$ (left) and when $\ell_2 \geq \ell_1$ (right). In the left case, the root gets both tasks in the area near $(0, 0)$, it gets only task 1 when $r_1 \leq \ell_1 - \ell_2$ and $r_2 \geq \ell_2$, and it gets neither task otherwise. Note that, in contrast to VCG, for every set of fixed values for the leaves, only three allocations are possible.

▶ **Definition 6** (Hybrid Mechanism for Graph Balancing)**.** *Consider an instance of the Unrelated Graph Balancing problem on a star of n nodes and set of tasks M. Let*

$$S \in \arg\min_{T \subseteq M} \{r(T) + \max_{i \notin T} \ell_i\}. \tag{1}$$

The mechanism assigns a set of tasks S to the root and the remaining tasks to leaves. Ties are broken in a deterministic way (e.g., lexicographically).

Figure 1 shows the partition of the space of the root player induced by the Hybrid Mechanism for a star of two leaves.

The argmin expression that defines the Hybrid Mechanism and a corresponding expression that defines the VCG mechanism are similar: in the definition of VCG, instead of $\max_{i \notin T} \ell_i$, we have $\sum_{i \notin T} \ell_i$. It is a happy coincidence that replacing the operator sum with max preserves the truthfulness of the mechanism, a fact that rarely holds.

▶ **Lemma 7.** *The Hybrid Mechanism for Graph Balancing on stars is truthful and has approximation ratio 2.*

Proof. The root player has no incentive to lie since $-\max_{i \notin T} \ell_i$ can be interpreted as its payments. The reason that leaf players have no incentive to lie comes essentially from the fact that the expression in (1) is monotone in ℓ_i (see Section 4, for a more rigorous and extensive treatment of the truthfulness of the general Hybrid Mechanism).

Let $S^* = \arg\min_{T \subseteq M} \max\{r(T), \max_{i \notin T} \ell_i\}$ be the subset assigned to the root in the optimal allocation, OPT be the optimal makespan, and ALG be the makespan achieved by the Hybrid Mechanism. Then we have

$$ALG \leq \min_{T \subseteq M} \{r(T) + \max_{i \notin T} \ell_i\} \leq r(S^*) + \max_{i \notin S^*} \ell_i \leq 2 \max\{r(S^*), \max_{i \notin S^*} \ell_i\} = 2OPT. \quad ◀$$

3.2 Upper bound for general graphs and multigraphs

We now turn our attention to positive (upper bound) results for general graphs and multigraphs. We will need a few definitions first.

▶ **Definition 8** (Star decomposition). *A* star decomposition *of a (multi)graph $G(V, E)$ is a partition $T = \{T_1, \ldots, T_k\}$ of its edges into stars (see Figure 2 for an example). Let $V(T_i)$ denote the vertex set of the star spanned by T_i. The star contention number of a star decomposition is the maximum number of stars that include a node either as a root or as a leaf: $c(T) = \max_{v \in V} |\{i : v \in V(T_i), i = 1, \ldots, k\}|$. The* star contention number of a (multi)graph *is the minimum star contention number among all its star decompositions.*

In an optimal star decomposition of a graph (but not multigraph), we can assume that every node is the root of at most one star, otherwise we can merge stars with common root without changing the star contention number.

A related notion to star decomposition that has been studied extensively is the notion of edge orientation of a multigraph (or of load balancing when we consider multigraphs).

▶ **Definition 9** (Edge orientation number). *Define the orientation number of a given orientation of the edges of a multigraph G, as its maximum in-degree. The* edge orientation number $o(G)$ *of a multigraph G is the minimum orientation number among all its possible orientations.*

Indeed the two notions are closely related: every star decomposition corresponds to a graph orientation by orienting the edges in all stars from roots to leaves, and vice versa a graph orientation gives rise to a star decomposition in which every node with its outgoing edges defines a star. Given that in an optimal star decomposition of a graph, each node is the root of at most one star, we get that for *every graph G*:

$$o(G) \leq c(G) \leq o(G) + 1.$$

This relation for *multigraphs* is similar only that in the right hand side we add the maximum edge multiplicity w instead of 1, i.e., $o(G) \leq c(G) \leq o(G) + w$.

The following definition utilizes the Hybrid Mechanism on stars to obtain a general mechanism for arbitrary graphs (and multigraphs).

▶ **Definition 10** (Star-Cover Mechanism). *Let $G = (V, E)$ be a multigraph and let $T = \{T_1, \ldots, T_k\}$ be a fixed star decomposition. The Star-Cover mechanism runs the Hybrid Mechanism on every star of T independently. That is, if $S_{i,h}$ is the subset of tasks allocated to a player i by the Hybrid Mechanism when applied to a star T_h, the set of tasks allocated to player i is $S_i = \cup_{h=1}^{k} S_{i,h}$.*

We can now state and prove the general positive theorem of this section.

▶ **Theorem 11.** *The Star-Cover mechanism for a given multigraph G that uses the Hybrid Mechanism on every star of a fixed star decomposition $T = \{T_1, \ldots, T_k\}$ is truthful and has an approximation ratio at most $2c(T)$.*

Proof. Fix some player i and let $S_{i,h}$ be the subset of tasks allocated to player i by the Star Mechanism when applied to a star T_h, $h = 1, \ldots, k$. Truthfulness is an immediate consequence of the following two observations. First, since the fixed star decomposition is independent of player i's processing times, player i cannot affect it by lying. Second, $S_{i,h}$ is independent of player i's processing times $t_i(e)$ for all edges $e \notin T_h$, therefore player i cannot alter the assignment on T_h by changing its values outside T_h.

To see the approximation guarantee, let OPT, $OPT(T_h)$ be the optimal makespan on G and T_h respectively, and let ALG and $ALG(T_h)$ be the makespan achieved by the Star-Cover mechanism on G and T_h.

$$ALG \leq \max_{h=1,\ldots,k} c(T) \cdot ALG(T_h) \leq \max_{h=1,\ldots,k} c(T) \cdot 2OPT(T_h) \leq 2c(T) \cdot OPT. \qquad \blacktriangleleft$$

Due to the connection between star decompositions and edge orientations in graphs, we get

▶ **Corollary 12.** *The approximation ratio for graphs with edge orientation number $o(G)$ is at most $2o(G) + 2$.*

In the sequel, we consider particular bounds for certain classes of graphs. It is known that the edge orientation number of a given graph can be computed in polynomial time [2]. In fact, by an application of the max-flow-min-cut theorem it can be shown that $o(G) \leq \gamma$ iff for every subgraph H of G it holds that $|E(H)| \leq \gamma |V(H)|$. Since this equivalent condition[2] holds for planar graphs with $\gamma = 3$, we immediately obtain:

▶ **Theorem 13.** *For every planar graph, there exists a truthful mechanism with approximation ratio 8.*

A natural class of graphs fulfilling this property (with $\gamma = k$) is k-degenerate graphs. A graph $G(V, E)$ is called *k-degenerate* [23] (or *k-inductive*) if there is an ordering $v_1, \ldots, v_n$ of its nodes such that the number of neighbors of v_i in $\{v_{i+1}, \ldots, v_n\}$ is at most k. Many interesting classes of graphs are k-degenerate for some small k. Besides planar graphs (with $k = 5$), another example is given by k-trees [40]: by definition, a k-tree is a degenerate graph with an ordering such that every v_i (except for the last k nodes of the ordering) has exactly k neighbors in $\{v_{i+1}, \ldots, v_n\}$ and these k neighbors form a clique. Since graphs of treewidth k are subgraphs of k-trees [40], they are also k-degenerate. In particular, trees are 1-degenerate. We give here a direct proof and illustration of a star decomposition for k-degenerate graphs:

▶ **Theorem 14.** *For every k-degenerate graph, there is a truthful mechanism with approximation ratio $2k + 2$.*

Proof. Consider a k-degenerate graph G. It suffices to show that it admits a star decomposition with contention number $k + 1$. Let $v_1, \ldots, v_n$ be an inductive ordering of the nodes of G. We consider the star covering $\{T_2, \ldots, T_n\}$ where T_i is the star with root v_i and leaves all its neighbors in $\{v_1, \ldots, v_{i-1}\}$. Note that stars are created in the opposite direction of the inductive order (see Figure 2). This star decomposition has contention number $k + 1$ since every node belongs to at most one star as a root and to at most k stars as a leaf. ◀

▶ **Corollary 15.** *There exist truthful mechanisms with approximation ratio at most 4 for trees, and generally of ratio at most $2k + 2$ for graphs of treewidth k.*

3.3 Lower Bounds for Graph Balancing

In this subsection, we show corresponding negative results for the positive results of the previous subsection. We first observe that the natural candidate mechanisms for the Graph Balancing problem have very poor performance, in stark contrast to the Hybrid Mechanism.

▶ **Theorem 16.** *All local mechanisms for stars, including VCG, affine minimizers and task-independent mechanisms, have approximation ratio at least $\sqrt{m} = \sqrt{n-1}$.*

---

[2] This characterization of the orientation number $o(G)$ implies that a truthful mechanism with constant approximation ratio exists for any minor-closed class of graphs, because for every class of graphs with forbidden minors, there exists some constant γ that satisfies the property (see Theorems 7.2.3, 7.2.4 and Lemma 12.6.1. in [19]). We are grateful to an anonymous referee for pointing this out.

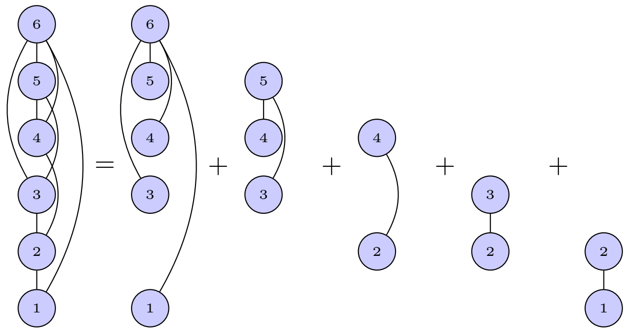

Figure 2 The star decomposition used in Theorem 14 of a 2-degenerate graph. The inductive order is upwards, while the stars are "pointing" downwards.

Proof. Consider the following input

$$t = \begin{pmatrix} \frac{1}{\sqrt{m}} & \frac{1}{\sqrt{m}} & \cdots & \frac{1}{\sqrt{m}} \\ 1 & \infty & \cdots & \infty \\ \infty & 1 & \cdots & \infty \\ \infty & \infty & \cdots & 1 \end{pmatrix}.$$

If, in the allocation of the mechanism, the root player takes all the tasks, then this allocation has approximation $\sqrt{m}$, as the optimal allocation is to assign the tasks to the leaves with makespan equal to 1. Otherwise, assume that (at least) one of the tasks, is given to some other player, say w.l.o.g. task 1 is given to player 1. By a series of applications of Lemma 4, and by exploiting the locality of the mechanism, we set the value of the owner of task j to 0 for every $j \neq 1$.

In particular, let S be the set of tasks assigned to the root player, and $M \setminus S$ be the tasks assigned to their respective leaf-player. Let $t^1 = (r', \ell_1, \dots, \ell_m)$, with r' defined as follows for some arbitrarily small ϵ.

$$r'_j = \begin{cases} 0 & j \in S \\ \frac{1}{\sqrt{m}} + \epsilon & \text{otherwise.} \end{cases}$$

By applying Lemma 4, the root player receives again the set S, and therefore, the set $M \setminus S$ is assigned to the leaves. We proceed by changing the bids of the leaf-players for the tasks in $M \setminus S$ to 0, i.e., defining a sequence t^j for $j \in M \setminus S$, with $t^j = (r', \ell'_j = 0, \ell^{j-1}_{-j})$

Again, by Lemma 4 and by locality, we get that the allocation of the tasks remains the same for the leaf j, *and* for all the other players as well.

We end up with an instance t' where player 1 still takes the first task, while the rest of the tasks are assigned to a player with 0 processing time. For t', the optimal makespan is $1/\sqrt{m}$, while the mechanism achieves makespan equal to 1. We illustrate the case when $S = \emptyset$, that is, the allocation gives all the tasks to the leaves of the star.

$$t = \begin{pmatrix} \frac{1}{\sqrt{m}} & \frac{1}{\sqrt{m}} & \cdots & \frac{1}{\sqrt{m}} \\ 1 & \infty & \cdots & \infty \\ \infty & 1 & \cdots & \infty \\ \infty & \infty & \cdots & 1 \end{pmatrix} \to t' = \begin{pmatrix} \frac{1}{\sqrt{m}} & \frac{1}{\sqrt{m}} & \cdots & \frac{1}{\sqrt{m}} \\ 1 & \infty & \cdots & \infty \\ \infty & 0 & \cdots & \infty \\ \infty & \infty & \cdots & 0 \end{pmatrix}$$ ◀

In the previous subsection, we showed that the Hybrid Mechanism outperforms all known mechanisms and has approximation ratio at most 2. The next theorem shows that this ratio is the best possible among all possible mechanisms for stars.

▶ **Theorem 17.** *There is no deterministic mechanism for stars that can achieve an approximation ratio better than 2.*

This is a special case of a more general lower bound for the L^p-norm objective (Theorem 30), but we give the proof here anyway, since it will be an ingredient of the proof of the following theorem (Theorem 18).

Proof. Let's assume that the mechanism takes an input where the processing time of the root player is $r_j = a^{j-1}$, for each task j, where $a > 1$ is a parameter, and the processing time of the corresponding leaf player for task j is $\ell_j = a^j$, as also shown in the following table.

$$t = \begin{pmatrix} 1 & a & \cdots & a^{m-2} & a^{m-1} \\ a & \infty & \cdots & \infty & \infty \\ \infty & a^2 & \cdots & \infty & \infty \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \infty & \infty & \cdots & a^{m-1} & \infty \\ \infty & \infty & \cdots & \infty & a^m \end{pmatrix}$$

If the mechanism assigns all tasks to the root player, then the makespan for this input is $(a^m-1)/(a-1)$, while the optimal makespan is a^{m-1}, yielding a ratio of $(a^m-1)/((a-1)a^{m-1})$. Otherwise, let X be the nonempty set of tasks assigned to the leaf players. Let k be the task with the maximum index in X. Since it is processed by the leaf player, its processing time is a^k. Now consider the input in which we change the processing times of the root player to

$$r'_j = \begin{cases} 0 & j \notin X \\ r_j + \epsilon & \text{otherwise} \end{cases}$$

for some arbitrarily small $\epsilon > 0$. By weak monotonicity (Lemma 4), the set of tasks assigned to the root player remains the same, and as a result the whole allocation stays the same. Therefore task k is still assigned to the leaf player k and the makespan of the mechanism is at least a^k. Notice that the optimum allocation for this input is $a^{k-1} + \epsilon$ which yields an approximation ratio of a, as ϵ tends to 0.

In conclusion, the approximation ratio is $\min\{(a^m-1)/((a-1)a^{m-1}), a\}$, for every $a > 1$. By choosing $a = 2$, we see that the ratio is $2 - 1/2^{m-1}$, which shows that for the class of stars no mechanism can have approximation ratio better than 2. For fixed m, the lower bound is slightly better than $2 - 1/2^{m-1}$, by selecting a to be the positive root of the equation $(a^m - 1)/((a-1)a^{m-1}) = a$. ◀

We now show how to extend the previous result to get a lower bound of $1+\varphi \approx 2.618$ for trees, and thus for graphs. This matches the best lower bound for the Nisan-Ronen setting [29] that was known until the recent improvements [24, 20, 11], suggesting that studying the special case of scheduling in graphs may be useful in attacking the Nisan-Ronen conjecture.

▶ **Theorem 18.** *No mechanism for trees can achieve approximation ratio* $1 + \varphi \approx 2.618$.

Proof. The proof mimics the proof of Theorem 17 on the tree shown in Figure 3. The tree consists of a star with root 0 and leaves $1, \ldots, k$ in which we add a new node $\overline{v}$ for each node v of the star and connect it to v. These new nodes (players), which we call dummy will not

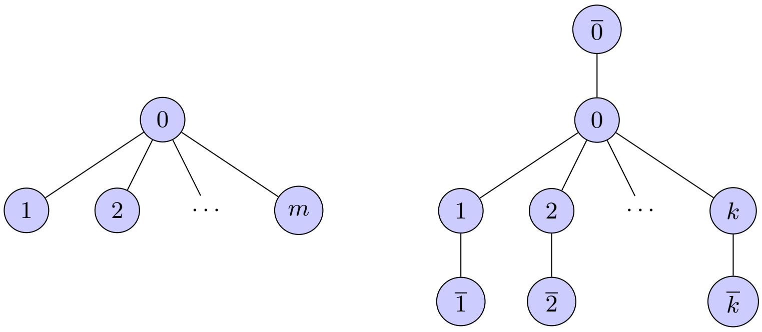

Figure 3 A star with root 0 and leaves $1, \ldots, m$ and its extension to a tree with dummy nodes.

be assigned any task by any efficient mechanism since we set their processing times to an arbitrarily high value H. The processing times of the edges of the star are exactly the same as in the proof of Theorem 17: $r_j = a^{j-1}$ and $\ell_j = a^j$, for some $a > 1$. The processing times for all edges are given below:

$$\begin{aligned} r_j &= a^{j-1} & \ell_j &= a^j & j &= 1, \ldots, k \\ \overline{r} &= 0 & \overline{\ell}_j &= 0 \end{aligned}$$

where $\overline{r}$ and $\overline{\ell}_j$ are the processing times of the star vertices of their respective dummy tasks. The dummy nodes themselves have a very large processing time $H \gg 1$ on these tasks.

We consider two cases. In the first case, all tasks of the star are assigned to the root player 0. We then consider a new instance in which we slightly lower the processing time of the root on the tasks of the star (i.e., $r_j = a^{j-1} - \epsilon$ for some $\epsilon > 0$) and increase the processing time of its dummy task $\overline{r} = a^k$. By weak monotonicity (Lemma 4), the r-player will take this task and all tasks of the star with a total processing time slightly less than $1 + a + \ldots + a^k = (a^{k+1} - 1)/(a - 1)$. It is easy to see that the optimal allocation for this instance is a^k, and the approximation ratio $(a^{k+1} - 1)/((a - 1)a^k)$.

In the second case, at least one task of the star is allocated to a leaf. Let p be the star task allocated to a leaf with the maximum index (that is, task p of the star is allocated to leaf-player p and tasks $p + 1, \ldots, k$ are allocated to the root). We consider the instance in which we change the processing times of the root player as follows: all processing times of the tasks allocated to the root become 0 and all processing times of the root player for the remaining tasks increase slightly. By weak monotonicity (Lemma 4), the r-player will still get the same set of tasks. We now create a new instance by increasing the processing time of the p-th dummy task: $\overline{\ell}_p = a^{p-1}$ and slightly decreasing the processing time of the leaf p for its task in the star: $\ell_p = a^p - \epsilon$, for some $\epsilon > 0$. Then again by weak monotonicity (Lemma 4), player p will get these two tasks. Although the allocation of the other tasks may change, the cost for the mechanism is at least $a^p + a^{p-1} - \epsilon$, while the optimal allocation has cost a^{p-1}. Therefore, in this case the mechanism has approximation ratio $(a^p + a^{p-1})/a^{p-1} = a + 1$, as $\epsilon \to 0$. In any case, the mechanism has approximation ratio $\min\{(a^{k+1} - 1)/((a - 1)a^k), a + 1\}$. By selecting $a = \varphi$, we get a ratio at least $1 + \varphi$ (as $k \to \infty$). ◀

Closing the gap between the above lower bound 2.618 of Theorem 18 and the upper bound 4 (Corollary 15) for mechanisms for trees is a crisp intriguing question.

4 Hybrid Mechanisms

Here we provide the general definitions related to Hybrid Mechanisms, and show necessary and sufficient conditions for truthfulness on stars (and hyper-stars[3]). We emphasize that this is a multi-dimensional mechanism design setting. Each leaf j has a single dimensional valuation, given by the scalar ℓ_j but a root has multi-dimensional preferences, given by the vector of values. For the sake of convenience, we call non-decreasing real functions *increasing*, and non-increasing functions *decreasing*. We say *strictly increasing/decreasing* if we want to emphasize strict monotonicity.

It is known, that an allocation rule can be equipped by a truthful payment scheme iff it is *weakly monotone* [41]. The next two propositions give a characterization of the weak monotonicity property in our case, for the leaf-players, and for the root player, respectively:

▶ **Proposition 19.** *An allocation rule is weakly monotone for a leaf-player i, iff for every r and every ℓ_{-i}, whenever leaf-player i gets task i with bid ℓ_i, then he also gets the task with every smaller bid $\ell_i' < \ell_i$.*

▶ **Proposition 20.** *An allocation rule is weakly monotone for the root player if and only if for every fixed bid vector ℓ of the other players, and every $T \subseteq M$ a constant $g_T(\ell)$ (i.e., independent of r) exists, such that for every r the root player is allocated a set $S \in$* $\arg\min_T \{r(T) + g_T(\ell)\}$.

The canonical choice for truthful payments to the r-player is then $P^0_S(\ell) = g_\emptyset(\ell) - g_S(\ell)$, and all other truthful payments can be obtained by an additive shift by an arbitrary $c(\ell)$.

We assume w.l.o.g. that for every fixed ℓ the payments P^0_S correspond to an increasing set-function of S,[4] because a set of tasks with higher cost and less payments can not be allocated to player 0 by a truthful mechanism.[5] Motivated by Proposition 20 we restrict our search for truthful mechanisms on star graphs as follows:

▶ **Definition 21** (Hybrid Mechanism). *Assume that an m-variate function $g_T : \mathbb{R}^m \to \mathbb{R}$ is given for every $T \subseteq M$, so that for every fixed vector $\ell \geq 0$ the values $\{g_T(\ell)\}_{T\subseteq M}$ correspond to a decreasing setfunction of T. For any input (r, ℓ), a* Hybrid Mechanism *(for the functions $\{g_T\}_{T\subseteq M}$) allocates a set S to the root player such that*

$$S \in \arg\min_T \{r(T) + g_T(\ell)\};$$

if there are more than one such sets S, the mechanism breaks ties according to the lexicographic order over all subsets of M. The items in $M \setminus S$ are assigned to the leaves.

Now for any $i \in M$ fix all bids in the input except for r_i, i.e., fix the vectors r_{-i} and ℓ. The following function $\psi_i[r_{-i}, \ell]$ defines the so called *critical value* for the bid r_i. We omit the argument r_{-i}, ℓ whenever they are obvious from the context.

▶ **Definition 22.**

$$\psi_i = \psi_i[r_{-i}, \ell] = \min_{T: i \notin T} \{r(T) + g_T(\ell)\} - \min_{T: i\in T} \{r(T \setminus \{i\}) + g_T(\ell)\}$$

[3] For simplicity of presentation we give here all definitions and lemmata for the case of stars, and discuss the necessary changes for hyper-stars in the full version.

[4] We call a setfunction P *increasing*, if $P(S') \leq P(S)$ whenever $S' \subset S$; we call it *strictly increasing* if the inequality is strict.

[5] See also the *virtual payments* in [12].

The next lemma states that ψ_i is nonnegative, and is, indeed, a critical value function. The proofs are straightforward, and due to space limitations are deferred to the full version.

▶ **Lemma 23.** *Let $i \in M$, and arbitrary nonnegative bid vectors r_{-i} and ℓ be fixed. Then $\psi_i[r_{-i}, \ell] \geq 0$, furthermore for every $r_i < \psi_i$ the root player receives task i, and for every $r_i > \psi_i$ the leaf player with bid ℓ_i receives task i.*

The following lemma provides various necessary or sufficient conditions for the truthfulness of Hybrid Mechanisms in terms of monotonicity of the critical value function ψ_i as a function of ℓ_i. For the proof of the lemma see the full version. There we also present an example mechanism showing that conditions (b) and (c) are both not necessary for the Hybrid Mechanism to be truthful.

▶ **Lemma 24.** *For the truthfulness of the Hybrid Mechanism with given $\{g_T\}_{T\subseteq M}$ functions (i.e., for a truthful payment scheme to exist),*

(a) *it is necessary that for every $i \in M$ and every fixed (r_{-i}, ℓ_{-i}) the function $\psi_i(\ell_i) = \psi_i[r_{-i}, \ell_{-i}](\ell_i)$ is an increasing function of ℓ_i;*

(b) *it is sufficient that for every $i \in M$ and every fixed (r_{-i}, ℓ_{-i}) the function $\psi_i(\ell_i) = \psi_i[r_{-i}, \ell_{-i}](\ell_i)$ is a* strictly *increasing function of ℓ_i;*

(c) *it is sufficient that for every i and ℓ_{-i} the $g_T(\ell_i, \ell_{-i})$ is an increasing function of ℓ_i whenever $i \notin T$, and decreasing function of ℓ_i whenever $i \in T$.*

▶ **Corollary 25.** *The Hybrid Mechanism for Graph Balancing and the Hybrid L^p Mechanism on stars are truthful.*

Proof. The first statement follows from the fact that the Hybrid Mechanism for Graph Balancing fulfils (c). Clearly, $g_T(\ell) = \max_{i\notin T}\{\ell_i\} = \max_{i\in M\setminus T}\{\ell_i\}$ is an increasing setfunction of the sets $M \setminus T$, and therefore a decreasing setfunction of the sets T, for fixed ℓ. For fixed T, $\max_{i\notin T}\{\ell_i\}$ is an increasing function of ℓ_i for every $i \notin T$, and it is independent of ℓ_i (constant function) if $i \in T$. Finally, it is easy to see that the Hybrid L^p Mechanism (see Section 5) fulfils (b) as well as (c). ◀

5 Mechanisms for L^p-norm optimization

In this section we generalize some of the results of Section 3 to the objective of minimizing the L^p-norm of the values of the agents, i.e., minimizing, over all allocations X the expression

$$\left(\sum_{i=1}^{n} t_i(X)^p\right)^{1/p}. \tag{2}$$

The makespan scheduling problem is the special case of $p = \infty$. We consider all positive values of p, but we deal separately with the case $p \geq 1$, in which L^p is a proper norm, and the case $p \in (0, 1)$, where the L^p function is not subadditive (i.e., the triangle inequality does not hold). Due to space limitations we postpone most of the results and their proofs to the full version of the paper. There we also consider the *maximization* case, which for $p = 1$ corresponds to auctions.

Consider an instance of the Unrelated Graph Balancing problem on a star of n nodes and set of tasks M. Notice that for stars the objective of minimizing the L^p-norm corresponds to minimizing $(r(T)^p + \sum_{i\notin T} \ell_i^p)^{\frac{1}{p}}$ over all task sets $T \subseteq M$ given to the r-player.

▶ **Definition 26** (Hybrid L^p Mechanism for stars)**.** *For a given $0 < p \leq \infty$, and an instance of the Unrelated Graph Balancing problem on a star of n nodes and set M of tasks, let*

$$S \in \arg\min_{T \subseteq M} \left\{ r(T) + \left(\sum_{i \notin T} \ell_i^p \right)^{1/p} \right\}. \tag{3}$$

The mechanism assigns S to the root and the remaining tasks to leaves. Ties are broken in a deterministic way (e.g., lexicographically).

The argmin expression that defines the Hybrid L^p Mechanism coincides with the VCG mechanism for $p = 1$ and with the Hybrid Mechanism of Section 3 for $p \to \infty$. As it is shown in Corollary 25, the Hybrid L^p mechanism is truthful.

Next we show two upper bound results for the approximation ratio (for the L^p-norm objective) separately in case $p \geq 1$, and in case $0 < p \leq 1$, respectively. We summarize here the inequalities that we will use:

▶ **Lemma 27.** *For any $p \geq 1$ it holds*

$$\sum_{i=1}^k x_i^p \leq \left(\sum_{i=1}^k x_i \right)^p \leq k^{p-1} \sum_{i=1}^k x_i^p. \tag{4}$$

Similarly for any $0 < p \leq 1$ it holds

$$\sum_{i=1}^k x_i^{1/p} \leq \left(\sum_{i=1}^k x_i \right)^{1/p} \leq k^{1/p-1} \sum_{i=1}^k x_i^{1/p}. \tag{5}$$

▶ **Theorem 28.** *For the problem of minimizing the L^p-norm, the Hybrid L^p Mechanism for stars has approximation ratio of at most $2^{(p-1)/p}$, when $p \geq 1$, and $2^{(1-p)/p}$, when $0 < p < 1$.*

Proof. Let $S^* = \arg\min_{T \subseteq M} (r(T)^p + \sum_{i \notin T} \ell_i^p)^{\frac{1}{p}}$ be the subset assigned to the root in the optimal allocation, S be the subset assigned to the root by the L^p Mechanism, OPT be the optimal L^p-norm, and ALG be the L^p-norm achieved by the Hybrid L^p Mechanism.

We first consider the case $p \geq 1$. We have

$$\begin{aligned} ALG = \left(r(S)^p + \sum_{i \notin S} \ell_i^p \right)^{1/p} &\leq r(S) + \left(\sum_{i \notin S} \ell_i^p \right)^{1/p} \\ &\leq r(S^*) + \left(\sum_{i \notin S^*} \ell_i^p \right)^{1/p} \\ &\leq 2^{(p-1)/p} \left(r(S^*)^p + \sum_{i \notin S^*} \ell_i^p \right)^{1/p} = 2^{(p-1)/p} OPT, \end{aligned}$$

where the first inequality follows from the triangle inequality, the second from the definition of the Hybrid L^p Mechanism, while the last one from Jensen's inequality (Lemma 27, Equation (4)) for $k = 2$, $x_1 = r(S^*)$, and $x_2 = (\sum_{i \notin S^*} \ell_i^p)^{1/p}$.

The case of $p < 1$, is essentially the same, but the proof is slightly different.

$$\begin{aligned} ALG = \left(r(S)^p + \sum_{i \notin S} \ell_i^p \right)^{1/p} &\leq 2^{\frac{1}{p}-1} \left(r(S) + (\sum_{i \notin S} \ell_i^p)^{1/p} \right) \\ &\leq 2^{\frac{1}{p}-1} \left(r(S^*) + (\sum_{i \notin S^*} \ell_i^p)^{1/p} \right) \\ &\leq 2^{\frac{1}{p}-1} \left(r(S^*)^p + \sum_{i \notin S^*} \ell_i^p \right)^{1/p} = 2^{\frac{1}{p}-1} OPT. \end{aligned}$$

The first inequality follows from Jensen's inequality for $x_1 = (r(S))^p$, and $x_2 = \sum_{i \notin S} \ell_i^p$; the second from the definition of the L^p Mechanism, while the last one from the fact that $(\alpha + \beta)^p \leq \alpha^p + \beta^p$, when $0 < p \leq 1$. ◀

As in the case of makespan, we can use the mechanism to other domains by decomposing them. We can apply the Star-Cover mechanism (Definition 10) to get good approximation ratios for general domains:

▶ **Theorem 29.** *For $p \geq 1$, the Star-Cover mechanism for a given multigraph G that uses the Hybrid L^p Mechanism on every star of a fixed star decomposition $T = \{T_1, \dots, T_k\}$ is truthful and has an approximation ratio at most $(2c(T))^{(p-1)/p}$ of the L^p-norm of the machines' costs, where $c(T)$ is the star contention number of the decomposition.*

We also provide corresponding negative results for mechanisms. For the case, of $p \geq 1$, the next theorem shows that the Hybrid L^p Mechanism has optimal approximation ratio.

▶ **Theorem 30.** *For any $p \geq 1$, there is no deterministic mechanism for stars that can achieve an approximation ratio better than $2^{1-1/p}$ for the L^p-objective.*

We point out that all known (local) mechanisms perform much worse that the Hybrid Mechanism. Observe that for $p = 1$, the VCG is optimal, but for large p the inefficiency of all local mechanisms grows and tends to $\sqrt{m}$:

▶ **Theorem 31.** *For minimizing the L^p-norm on stars, all* local *mechanisms, including affine minimizers and task-independent mechanisms, have approximation ratio of at least $m^{\frac{1}{2}(1-1/p)} = (n-1)^{\frac{1}{2}(1-1/p)}$, when $p \geq 1$.*

The lower bound that we give for the case of $p < 1$ does not match exactly the upper bound, which leaves open the possibility that there exists a mechanism with better approximation ratio than the Hybrid L^p Mechanism. Notice that the following approximation ratio tends to infinity as p tends to 0.

▶ **Theorem 32.** *For any $0 < p \leq 1$ and every $a > 1$, there is no deterministic mechanism for stars that can achieve an approximation ratio better than*

$$\min \left\{ a, \frac{(a+1)^{1/p}}{a^{1/p} + a)} \right\}. \tag{6}$$

By selecting an appropriate a, this is $\Omega(p^{-1}/\ln(p^{-1}))$.

References

1 Aaron Archer and Éva Tardos. Truthful mechanisms for one-parameter agents. In *Proc. of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 482–491, 2001.

2 Yuichi Asahiro, Jesper Jansson, Eiji Miyano, and Hirotaka Ono. Graph orientation to maximize the minimum weighted outdegree. *Int. J. Found. Comput. Sci.*, 22(3):583–601, 2011. doi:10.1142/S0129054111008246.

3 Yuichi Asahiro, Eiji Miyano, and Hirotaka Ono. Graph classes and the complexity of the graph orientation minimizing the maximum weighted outdegree. *Disc. Appl. Math.*, 159(7):498–508, 2011.

4 Itai Ashlagi, Shahar Dobzinski, and Ron Lavi. Optimal lower bounds for anonymous scheduling mechanisms. *Mathematics of Operations Research*, 37(2):244–258, 2012. doi:10.1287/moor.1110.0534.

5 Vincenzo Auletta, George Christodoulou, and Paolo Penna. Mechanisms for scheduling with single-bit private values. *Theory Comput. Syst.*, 57(3):523–548, 2015. doi:10.1007/s00224-015-9625-5.

6 Nikhil Bansal and Maxim Sviridenko. The Santa Claus problem. In *STOC*, pages 31–40. ACM, 2006. doi:10.1145/1132516.1132522.

7 Sushil Bikhchandani, Shurojit Chatterji, Ron Lavi, Ahuva Mu'alem, Noam Nisan, and Arunava Sen. Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica*, 74(4), 2006.

8 Shuchi Chawla, Jason D. Hartline, David L. Malec, and Balasubramanian Sivan. Prior-independent mechanisms for scheduling. In *STOC'13*, pages 51–60. ACM, 2013. doi:10.1145/2488608.2488616.

9 Xujin Chen, Donglei Du, and Luis Fernando Zuluaga. Copula-based randomized mechanisms for truthful scheduling on two unrelated machines. *Theory Comput. Syst.*, 57(3):753–781, 2015.

10 George Christodoulou, Elias Koutsoupias, and Annamária Kovács. Mechanism design for fractional scheduling on unrelated machines. *ACM Transactions on Algorithms*, 6(2), 2010. doi:10.1145/1721837.1721854.

11 George Christodoulou, Elias Koutsoupias, and Annamária Kovács. On the Nisan-Ronen conjecture. *CoRR*, abs/2011.14434, 2020. arXiv:2011.14434.

12 George Christodoulou, Elias Koutsoupias, and Annamária Kovács. On the Nisan-Ronen conjecture for submodular valuations. In *STOC*, pages 1086–1096. ACM, 2020.

13 George Christodoulou, Elias Koutsoupias, and Angelina Vidali. A lower bound for scheduling mechanisms. *Algorithmica*, 55(4):729–740, 2009. doi:10.1007/s00453-008-9165-3.

14 George Christodoulou and Annamária Kovács. A deterministic truthful ptas for scheduling related machines. *SIAM J. Comput.*, 42(4):1572–1595, 2013. doi:10.1137/120866038.

15 Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 8, 1971.

16 Richard Cole and Vasilis Gkatzelis. Approximating the Nash social welfare with indivisible items. *SIAM Journal on Computing*, 47(3):1211–1236, January 2018.

17 Constantinos Daskalakis and S. Matthew Weinberg. Bayesian truthful mechanisms for job scheduling from bi-criterion approximation algorithms. In *SODA*, pages 1934–1952. SIAM, 2015. doi:10.1137/1.9781611973730.130.

18 Peerapong Dhangwatnotai, Shahar Dobzinski, Shaddin Dughmi, and Tim Roughgarden. Truthful approximation schemes for single-parameter agents. *SIAM J. on Computing*, 40(3):915–933, 2011. doi:10.1137/080744992.

19 Reinhard Diestel. *Graph Theory, 4th Edition*. Springer, 2012.

20 Shahar Dobzinski and Ariel Shaulker. Improved lower bounds for truthful scheduling. *CoRR*, abs/2007.04362, 2020. arXiv:2007.04362.

21 Tomás Ebenlendr, Marek Krcál, and Jirí Sgall. Graph balancing: A special case of scheduling unrelated parallel machines. *Algorithmica*, 68(1):62–80, 2014. doi:10.1007/s00453-012-9668-9.

22 Leah Epstein, Asaf Levin, and Rob van Stee. A unified approach to truthful scheduling on related machines. In *Proc. SODA*, pages 1243–1252, 2013.

23 Paul Erdős and András Hajnal. On chromatic number of graphs and set-systems. *Acta Mathematica Hungarica*, 17(1-2):61–99, 1966.

24 Yiannis Giannakopoulos, Alexander Hammerl, and Diogo Poças. A new lower bound for deterministic truthful scheduling. In *SAGT 2020*, volume 12283, pages 226–240. Springer, 2020. doi:10.1007/978-3-030-57980-7_15.

25 Yiannis Giannakopoulos and Maria Kyropoulou. The VCG mechanism for bayesian scheduling. *ACM Trans. Economics and Comput.*, 5(4):19:1–19:16, 2017. doi:10.1145/3105968.

26 Theodore Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

27 Chien-Chung Huang and Sebastian Ott. A combinatorial approximation algorithm for graph balancing with light hyper edges. In *ESA 2016*, volume 57 of *LIPIcs*, pages 49:1–49:15, 2016. doi:10.4230/LIPIcs.ESA.2016.49.

28 Klaus Jansen and Lars Rohwedder. Local search breaks 1.75 for graph balancing. In *ICALP 2019*, volume 132 of *LIPIcs*, pages 74:1–74:14, 2019.

29 Elias Koutsoupias and Angelina Vidali. A lower bound of $1+\phi$ for truthful scheduling mechanisms. *Algorithmica*, pages 1–13, 2012.

30 Ron Lavi and Chaitanya Swamy. Truthful mechanism design for multidimensional scheduling via cycle monotonicity. *Games and Economic Behavior*, 67(1):99–124, 2009. doi:10.1016/j.geb.2008.08.001.

31 Kangbok Lee, Joseph Y.-T. Leung, and Michael L. Pinedo. A note on graph balancing problems with restrictions. *Inf. Process. Lett.*, 110(1):24–29, 2009. doi:10.1016/j.ipl.2009.09.015.

32 Jan Karel Lenstra, David B Shmoys, and Eva Tardos. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming*, 46(1-3):259–271, 1990.

33 Stefano Leucci, Akaki Mamageishvili, and Paolo Penna. No truthful mechanism can be better than n approximate for two natural problems. *Games and Economic Behavior*, 111:64–74, 2018. doi:10.1016/j.geb.2018.05.003.

34 Pinyan Lu. On 2-player randomized mechanisms for scheduling. In *WINE*, volume 5929 of *Lecture Notes in Computer Science*, pages 30–41. Springer, 2009. doi:10.1007/978-3-642-10841-9_5.

35 Pinyan Lu and Changyuan Yu. An improved randomized truthful mechanism for scheduling unrelated machines. In *STACS*, volume 1 of *LIPIcs*, pages 527–538, 2008.

36 Pinyan Lu and Changyuan Yu. Randomized truthful mechanisms for scheduling unrelated machines. In *WINE*, pages 402–413, 2008. doi:10.1007/978-3-540-92185-1_46.

37 Hadi Minooei and Chaitanya Swamy. Truthful mechanism design for multidimensional covering problems. In *WINE 2012*, volume 7695 of *LNCS*, pages 448–461. Springer, 2012. doi:10.1007/978-3-642-35311-6_33.

38 Ahuva Mu'alem and Michael Schapira. Setting lower bounds on truthfulness. *Games and Economic Behavior*, 110:174–193, 2018.

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

40 Donald J Rose. On simple characterizations of k-trees. *Disc. Math.*, 7(3-4):317–322, 1974.

41 Michael E. Saks and Lan Yu. Weak monotonicity suffices for truthfulness on convex domains. In *Proceedings 6th ACM Conference on Electronic Commerce (EC)*, pages 286–293, 2005. doi:10.1145/1064009.1064040.

42 José Verschae and Andreas Wiese. On the configuration-lp for scheduling on unrelated machines. *J. Scheduling*, 17(4):371–383, 2014. doi:10.1007/s10951-013-0359-4.

43 William Vickrey. Counterspeculations, auctions and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

44 Chao Wang and René Sitters. On some special cases of the restricted assignment problem. *Inf. Process. Lett.*, 116(11):723–728, 2016.

45 Changyuan Yu. Truthful mechanisms for two-range-values variant of unrelated scheduling. *Theoretical Computer Science*, 410(21-23):2196–2206, 2009. doi:10.1016/j.tcs.2009.02.001.