On the Nisan-Ronen Conjecture

GEORGE CHRISTODOULOU
Aristotle University of Thessaloniki
and
ELIAS KOUTSOUPIAS
University of Oxford
and
ANNAMARIA KOVACS
Goethe University, Franfurt M.

The Nisan-Ronen conjecture states that no truthful mechanism for makespan-minimization when allocating m tasks to n unrelated machines can have approximation ratio less than n. Over more than two decades since its formulation, little progress has been made in resolving it and the best known lower bound was a small constant. This note gives an overview of our recent paper that gives a lower bound of $1 + \sqrt{n-1}$.

Categories and Subject Descriptors: F.2 [**Theory of computation**]: Mechanism design

General Terms: Algorithms, Economics, Theory

Additional Key Words and Phrases: Algorithmic mechanism design, Nisan-Ronen conjecture

Mechanism design, a main branch of Game Theory and Microeconomics, studies a special class of algorithms, called mechanisms. Unlike traditional algorithms that get their input from a single user, mechanisms solicit the input from different participants (called agents, players, bidders), in the form of preferences over the possible outputs (outcomes). The difficulty of designing such algorithms stems from the fact that the actual preferences of the participants are private information, unknown to the algorithm. The participants are assumed to be utility maximisers who will provide some input that suits their objective and may differ from their true preferences. A *truthful* mechanism provides incentives such that a truthful input is the best action for each participant.

The question is what kind of problems can be solved within this framework. In their seminal paper that launched the field of algorithmic mechanism design, Nisan and Ronen [Nisan and Ronen 2001] proposed the scheduling problem on unrelated machines as a central problem to capture the algorithmic and information-theoretic aspects of mechanism design. In the classical form of the scheduling problem, which has been extensively studied from the algorithmic perspective, there are n machines that process a set of m tasks; each machine i takes time t_{ij} to process task j. The objective of the algorithm is to allocate each task to a machine in order to minimize the makespan, i.e., the maximum completion time over all machines. In the mechanism design setting, each machine provides as input its processing time

Authors' addresses:
gichristo@csd.auth.gr, elias@cs.ox.ac.uk, panni@cs.uni-frankfurt.de

for each task. The selection by Nisan and Ronen of this version of the scheduling problem to study the limitations that truthfulness imposes on algorithm design was a masterstroke, because it turned out to be an extremely rich and challenging setting.

Nisan and Ronen applied the VCG mechanism [Nisan et al. 2007], 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: each task is independently assigned to the machine with minimum processing time. This mechanism is truthful, but has poor approximation ratio, n. They boldly conjectured that this is the best guarantee that can be achieved by any deterministic (polynomial-time or not) truthful mechanism. The Nisan-Ronen conjecture has been a central problem in algorithmic mechanism design in the last two decades. This note is about a recent paper [Christodoulou et al. 2021a] that made progress towards this conjecture:

Theorem 1. *There is no deterministic truthful mechanism with approximation ratio better than* $1 + \sqrt{n-1}$ *for the problem of scheduling* n *unrelated machines.*

This bound is information-theoretic in the sense that it holds for all deterministic mechanisms, regardless of their running time.

It is well known [Saks and Yu 2005; Archer and Kleinberg 2008; Bikhchandani et al. 2006] that a mechanism is truthful if its allocation function is monotone in the values of each machine. Monotonicity in one dimension (i.e., a single task) is the usual notion of monotonicity of the allocation function, and for two or more dimensions, it takes a particular very natural form that is tightly related to the theory of convex functions. Thus one can restate the above theorem as "no monotone algorithm, polynomial-time or not, has approximation ratio less than $1+\sqrt{n-1}$ for the problem of scheduling n unrelated machines." In contrast, the approximation ratio for the usual (non-monotone) class of algorithms is trivially 1, for exponential-time algorithms, and at most 2 for polynomial-time ones [Lenstra et al. 1990].

This is the first non-constant lower bound of the Nisan-Ronen problem. Previous results include a lower bound of 2 [Nisan and Ronen 2001], which was improved to 2.41 [Christodoulou et al. 2009], and later to 2.61 [Koutsoupias and Vidali 2012], as well as recent improvements to 2.755 [Giannakopoulos et al. 2020] and to 3 [Dobzinski and Shaulker 2020]. Some ideas used in the proof of the above theorem first appeared in a recent publication [Christodoulou et al. 2020], which established 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 of the standard scheduling setting.

A crucial role in the proof of the above theorem is played by use of special inputs in which each task can be reasonably allocated to only two machines. This is achieved by setting the values of the other machines sufficiently high so that every algorithm with relatively small approximation ratio will avoid them. With this, we focus on a special case of the scheduling problem, the *multi-graph scheduling problem* [Christodoulou et al. 2021b]: the input is a multi-graph of n nodes (machines) and each edge (task) has two values, one value for each of its nodes; the mechanism must allocate each edge to one of its nodes. Actually the proof of the above theorem employs *multi-stars*. Using graphs instead of general instances allows us to take advantage of a useful characterization of mechanisms for 2 machines; no such good characterization is known for 3 or more machines.

1. OUTLINE OF THE PROOF

We provide an outline for a slightly worse[1] lower bound of $\sqrt{n-1}$.

1.1 The construction

We consider instances, with n players (machines) and m tasks that are partitioned into $n-1$ clusters $C_1, \ldots, C_{n-1}$. Each cluster C_i contains ℓ tasks and is associated with player $i \in [n-1]$; the number ℓ of tasks per cluster needs to be at least exponential in n for the proof to work. The processing time for a task $j \in C_i$, $i \in [n-1]$, is described by two values: t_j of player 0 and s_j of player i; these values are usually in $(0,1]$. The processing time of every other player $k \notin \{0, i\}$ for a task $j \in C_i$ is sufficiently high, so that no mechanism with bounded approximation ratio would ever allocate j to them. Hence, we describe an instance T by only two values $[t_j, s_j]$ per task j.

1.2 Definitions

The proof relies on a characterization of 2×2 mechanisms [Christodoulou et al. 2020] that concerns two players and two tasks. Although we consider a multi-player setting, we are able to use it, by fixing all other values except for the values of two tasks p and p' of the *same* cluster (which we call *siblings*). We refer to this set of instances as a (p, p')-*slice*; the resulting allocation for p, p' corresponds to an allocation of a 2×2 truthful mechanism.

The central part of the argument that shows the lower bound, uses an induction on the number k of clusters. The values of the tasks in the remaining $n-k-1$ clusters, which we call trivial clusters, play a limited role, but it is important that they do not affect substantially the approximation ratio. Intuitively, a cluster is called *trivial* if the optimal allocation for all tasks of the cluster has cost sufficiently small (say at most n^{-2}).

We usually select a single task from each non-trivial cluster, and we call such a selection of tasks *regular*. A set of instances is called *standard* for a set of clusters $\mathcal{C}$ if the value of every task $j \in \cup\mathcal{C}$ is $[t_j = 0,\ s_j = 1]$, and the remaining clusters are trivial. The following definition of a *good set of tasks* is at the heart of the proof.

Definition 2 *Good set of tasks – bad task.* Consider a truthful mechanism and let $\alpha = 1/\sqrt{n-1}$. Fix a standard instance T for a set of k clusters $\mathcal{C}$, and a set of regular tasks $P = \{p_1, \ldots, p_k\}$ from $\mathcal{C}$. The set of tasks P is called *good*,[2] if

---

[1]We expose the main ideas of the proof, and try to provide intuition, hiding intricate details in our definitions and statements which inevitably sacrifices the rigour of many of the statements. We refer the reader to the full version of our paper for complete arguments and for the slightly improved bound of $1 + \sqrt{n-1}$.

[2]Our goal is to communicate the high level idea, as a result this definition is oversimplified and far from the more intricate definition that we have in the full version. For example, what we actually need is that there exists a vector V of open intervals (around the values of the tasks in P) such that the mechanism allocates all tasks in P to player 0 *for every* instance in the set. We call such sets of instances V-*perturbations* – and *witness* of goodness in particular –, which are crucial for correctness and which we consider an important conceptual contribution of our work.

when we replace the value of every task $p_j \in P$ with $[t_j = \alpha,\ s_j = 1]$, the mechanism allocates *all* tasks in P to player 0. If the latter property is not satisfied, we call P a *bad* set. A singleton bad set is simply called *bad task*. (See Fig 1 for an illustration.)

$$T = \begin{bmatrix} 0 & 0 & \alpha^* & \alpha^* & 0 & 0 & \alpha^* & 0 & 0 \\ 1 & 1 & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \end{bmatrix}, \qquad T' = \begin{bmatrix} 0 & \alpha & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1^* & 1 & & & & & & \\ & & & 1 & 1 & 1 & & & \\ & & & & & & 1 & 1 & 1 \end{bmatrix}$$

Fig. 1. T gives an example of a good set $P = \{3, 4, 7\}$ of size 3. Indeed, tasks $3 \in C_1, 4 \in C_2, 7 \in C_3$ are regular, as they belong to distinct clusters, and have values $[t_j = \alpha, s_j = 1]$. The other values are trivial, and the mechanism allocates the tasks in P to the 0 player (indicated with '$*$'). In T', task 2 is a bad task, as it has values $[t_2 = \alpha, s_2 = 1]$, all other tasks are trivial, and the task is allocated to player 1.

LEMMA 3 (MAIN LEMMA). *At least one of the following two properties hold for every truthful mechanism:*

(i) there exists a bad task

(ii) there exists a good set of $n-1$ tasks.

The Main Lemma immediately implies the desired lower bound on the approximation ratio. The existence of a bad task (Property *(i)*) means that there is only one non-trivial task j with values $[t_j = \alpha,\ s_j = 1]$ and the mechanism does not give it to player 0, which would be the optimal decision. In this case, the approximation ratio is approximately $1/\alpha = \sqrt{n-1}$. Finally, a good set of $n-1$ tasks (Property *(ii)*) has approximation ratio $(n-1)\alpha = \sqrt{n-1}$, giving the desired result.

To obtain a proof of the Main Lemma, we show that there exists a good set of k tasks for every $k \in [n-1]$, by induction on k. We start with some regular set of k tasks, which we call *potentially-good* set, such that *all its subsets of $k-1$ tasks are good*. To satisfy all the requirements in the proof, the precise structure of a potentially-good set of tasks is complicated and it is detailed in the full version of our paper.

1.3 Outline of the proof of the Main Lemma

We now give a rough outline of the argument that establishes the Main Lemma (Lemma 3). We consider regular instances T for sets of k clusters $\mathcal{C}$. If there is a bad task, then we are done. Hence, we show that otherwise, for $k = n-1$ there is a good set of tasks. By induction on k, we show the stronger claim that there are *many sets of tasks that are good*, the base case ($k = 1$) being true due to the fact that there are no bad tasks.

We use a probabilistic argument to show that the probability b_k of a random regular set of tasks $P = (p_1, \dots, p_k)$ being a bad set is small. In particular we conclude that $b_{n-1} < 1$, which establishes the existence of a good set of $n-1$ tasks.

Showing that b_k is small. We show that b_k is small by establishing the following two facts:

Fact a. With high probability a randomly selected regular set $P = (p_1, \ldots, p_k)$ is potentially-good.

Fact b. If $P = (p_1, \ldots, p_k)$ is a potentially-good set of tasks, either P is good itself or (P_{-k}, p'_k) is good with sufficiently high probability, where p'_k is a random sibling of p_k. Roughly speaking, either P is good or *almost all other sets are good* (with exponentially small probability of the negative event).

Taking into account that $b_1 = 0$, since there are no bad tasks, we can combine these two facts to get that b_k is bounded above by a decreasing function on the number ℓ of tasks per cluster. By selecting ℓ to be sufficiently large, this establishes that $b_{n-1} < 1$.

Showing Fact b. The difficult part is to establish the second of the above two facts (Fact b). Let's assume that P is potentially-good but not good. We show that (P_{-k}, p'_k) is good for *many* p'_k's, as follows:

—First, we observe that none of the tasks in $P = (p_1, \ldots, p_k)$ is given to player 0. This essentially follows from the definition of potentially-good and weak monotonicity.

—Let p'_k be a sibling of p_k and consider the (p_k, p'_k)-slice mechanism. This is exactly the point where we exploit the 2×2 characterization. The proof proceeds by treating carefully all possible cases, as they appear in the characterization [Christodoulou et al. 2020]:

(1) **Affine minimizers:** we show that the mechanism is not an affine minimizer almost surely

(2) **Relaxed affine minimizers:** we show that the probability that the mechanism is a relaxed affine minimizer is at most $2n^2/\ell$

(3) **1-dimensional and constant mechanisms:** we show that 1-dimensional and constant mechanisms do not occur, or the approximation ratio is high

(4) **Task independent or relaxed task independent mechanisms:** we show that if the mechanism is task independent or relaxed task independent for each of k appropriately selected random instances from the witness, then (P_{-k}, p'_k) is good

—We conclude that for a random sibling p'_k, the mechanism must be either task independent or relaxed task independent for all these k instances with probability at least $1 - k\,2n^2/\ell$; therefore (P_{-k}, p'_k) is good with probability at least $1 - 2n^3/\ell$.

The first item, i.e., to show that affine minimizers are sparse, exploits an interesting use of (potential-)goodness and linearity, the latter being an important implication of affine maximization. The proof of relaxed affine minimizers uses the same machinery, but it has an extra layer of difficulty, as such mechanisms may have non-linear parts which are hard to handle. In fact, we might have a positive probability (at most $2n^3/\ell$) to pick a wrong sibling p'_k due to this deficiency. The proof of the last item about task independent and relaxed task independent mechanisms has very similar flavor. It is essentially this part that takes away the complications that arise from having to deal with an additive domain.

2. CONCLUSION

The major problem left open is to settle the Nisan-Ronen conjecture. We expect the techniques of this work to be helpful in this direction. The case of randomized or fractional mechanisms appears also very challenging; the best known lower bound of the approximation ratio is 2 [Mu'alem and Schapira 2018; Christodoulou et al. 2010], embarrassingly lower than the best known upper bound $(n + 1)/2$. The bottleneck of applying the techniques of the current work to these variants appears to be the lack of a good characterization of 2×2 fractional mechanisms. Finally, although the result of this work indicates that mechanisms constitute a limited subclass of allocation algorithms, a more direct demonstration would be to find a useful characterization of mechanisms for the domain of scheduling and its generalizations.

REFERENCES

Archer, A. and Kleinberg, R. 2008. Truthful germs are contagious: a local to global characterization of truthfulness. In *EC*. ACM, 21–30.

Bikhchandani, S., Chatterji, S., Lavi, R., Mu'alem, A., Nisan, N., and Sen., A. 2006. Weak monotonicity characterizes deterministic dominant strategy implementation. *Econometrica 74,* 4, 1109–1132.

Christodoulou, G., Koutsoupias, E., and Kovács, A. 2010. Mechanism design for fractional scheduling on unrelated machines. *ACM Trans. Algorithms 6,* 2, 38:1–38:18.

Christodoulou, G., Koutsoupias, E., and Kovács, A. 2020. On the Nisan-Ronen conjecture for submodular valuations. In *STOC*. ACM, 1086–1096. (Full version arXiv:1907.12733).

Christodoulou, G., Koutsoupias, E., and Kovács, A. 2021a. On the Nisan-Ronen conjecture. In *FOCS*. IEEE, 839–850.

Christodoulou, G., Koutsoupias, E., and Kovács, A. 2021b. Truthful allocation in graphs and hypergraphs. In *ICALP*. Vol. 198. 56:1–56:20.

Christodoulou, G., Koutsoupias, E., and Vidali, A. 2009. A lower bound for scheduling mechanisms. *Algorithmica 55,* 4, 729–740.

Dobzinski, S. and Shaulker, A. 2020. Improved Lower Bounds for Truthful Scheduling.

Giannakopoulos, Y., Hammerl, A., and Poças, D. 2020. A new lower bound for deterministic truthful scheduling. In *SAGT*. Vol. 12283. 226–240.

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

Lenstra, J. K., Shmoys, D. B., and Tardos, E. 1990. Approximation algorithms for scheduling unrelated parallel machines. *Mathematical programming 46,* 1-3, 259–271.

Mu'alem, A. and Schapira, M. 2018. Setting lower bounds on truthfulness. *Games and Economic Behavior 110*, 174–193.

Nisan, N. and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior 35*, 166–196.

Nisan, N., Roughgarden, T., Tardos, E., and Vazirani, V. V. 2007. *Algorithmic Game Theory.* Cambridge University Press.

Saks, M. E. and Yu, L. 2005. Weak monotonicity suffices for truthfulness on convex domains. In *EC*. ACM, 286–293.