Improved Upper Bounds for Finding Tarski Fixed Points

Xi Chen, Yuhao Li

Columbia University

{xichen,yuhaoli}@cs.columbia.edu

Introduction

In 1955, Tarski [5] proved that every monotone[1] function $f : L \to L$ over a complete lattice $(L, \preceq)$ has a fixed point, i.e. $x \in L$ with $f(x) = x$. In this paper we study the query complexity of finding a Tarski fixed point in the complete lattice $([n]^k, \preceq)$ over the k-dimensional grid $[n]^k = \{1, \ldots, n\}^k$ and equipped with the natural partial order over $\mathbb{Z}^k$, where $a \preceq b$ if $a_i \le b_i$ for all $i \in [k]$. An algorithm under this model is given n and k and has query access to an unknown monotone function f over $[n]^k$. Each round the algorithm can send a query $x \in [n]^k$ to reveal $f(x)$ and the goal is to find a fixed point of f using as few queries as possible. We will refer to this problem as Tarski(n, k).

Binary Search Algorithm

Theorem [Dang, Qi, and Ye 2011]. There is an algorithm for Tarski(n, k) with query complexity $O(\log^k n)$.

Let's take 2-d cases as an example to describe this algorithm: Consider all points $\mathcal{L}_1 = (x_1, x_2 = \lceil n/2 \rceil)$. Then it's a 1-d lattice and we can find a point x^* such that $f(x^*)_1 = x^*_1$ (a fixed point for the first dimension). No matter the second dimension of x^* goes up or goes down, we can cut the search space by half.

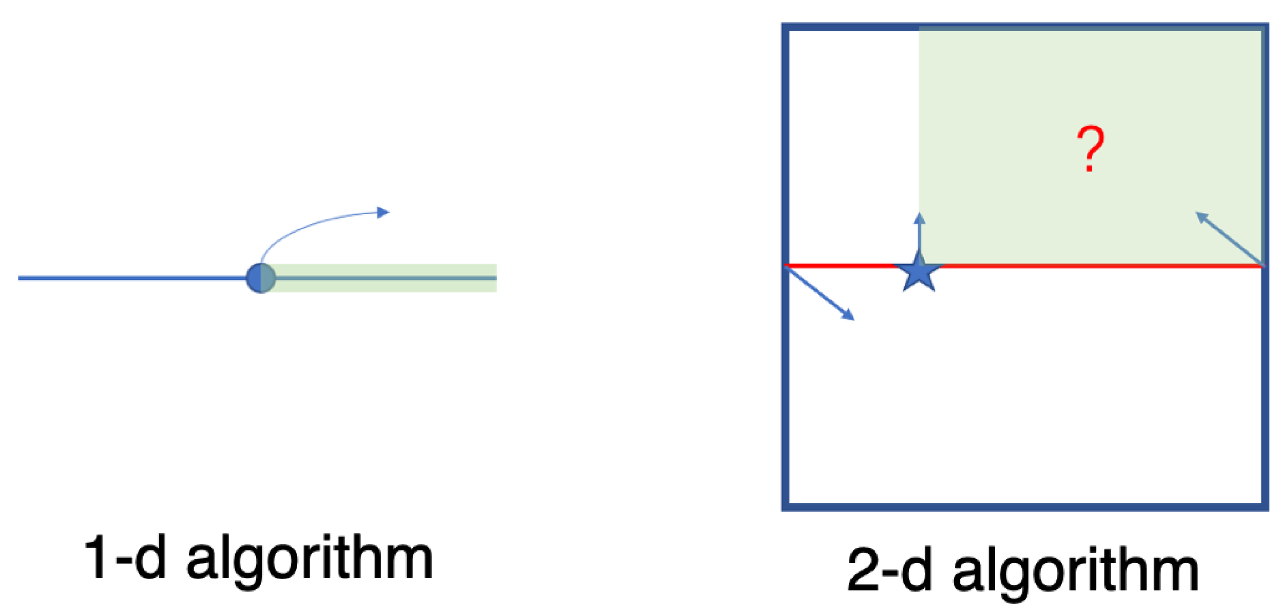

Fig. 1: An illustration of the binary search algorithm.

Scan the QR code to get the full paper!

Lower Bounds

Theorem [Etessami, Papadimitriou, Rubinstein, and Yannakakis 2020]. Any (deterministic or randomized) algorithm must make $\Omega(\log^2 n)$ queries for Tarski$(n, 2)$.

The most interesting part behind the proof is that it shows the best algorithm should solve $\Omega(\log n)$ many independent Tarski$(n, 1)$, which means each of them must take $\Omega(\log n)$ queries.

A Faster Algorithm

Theorem [Fearnley, Pálvölgyi, and Savani 2021]. There is an algorithm for Tarski(n, k) with query complexity $O(\log^{\lceil 2k/3 \rceil} n)$.
This result needs the following two technical theorems.

Faster algorithm for Tarski$(n, 3)$. Recall that the complexity for Tarski$(n, 2)$ has been settled. The main technical contribution of [FPS21] is that when recursively working on 2-d slice, although finding a fixed point should be $\Omega(\log^2 n)$, but it suffices to find a weaker point x^* such that either $f(x^*)_i \le x^*_i$ for all $i \in [3]$ or $f(x^*)_i \ge x^*_i$ for all $i \in [3]$. Then there is an algorithm with query complexity $O(\log n)$ to find such a weaker point, which accelerates the whole algorithm by an $O(\log n)$ factor.

A decomposition theorem for Tarski$(n, a + b)$. If Tarski(n, a) can be solved in $q(n, a)$ queries and Tarski(n, b) can be solved in $q(n, b)$ queries, then Tarski$(n, a + b)$ can be solved in $O(q(n, a) \cdot q(n, b))$ queries.

Tarski$^*(n, k)$

Our key idea is to develop a new decomposition theorem directly on the problem that finds the **weaker** point x^*.
More formally we refer to the following problem as Tarski$^*(n, k)$:

- Given a monotone function $f : [n]^{k+1} \to [n]^{k+1}$, find a point x with $x_{k+1} = \lceil n/2 \rceil$ such that either $f(x)_i \le x_i$ for all $i \in [k + 1]$ or $f(x)_i \ge x_i$ for all $i \in [k + 1]$.

Our Contribution

Theorem. There is an algorithm for Tarski(n, k) with query complexity $O(\log^{\lceil (k+1)/2 \rceil} n)$.

Technical Theorem. If Tarski$^*(n, a)$ can be solved in $q(n, a)$ queries and Tarski$^*(n, b)$ can be solved in $q(n, b)$ queries, then Tarski$^*(n, a + b)$ can be solved in $O(q(n, a) \cdot q(n, b))$ queries.

Now despite sharing the same statement/recursion, the proof of our decomposition theorem requires a number of new technical ingredients compared to that of [2]. This is mainly due to the extra coordinate (i.e., coordinate $k + 1$) that appears in Tarski* but not in the original Tarski.

Discussion and Open Problems

While progress has been made on improving the upper bounds for finding Tarski fixed points, the techniques for lower bounds remain limited.
For the black-box (query complexity) model studied in this paper, the key question left open is to close the gap between $\Omega(\log^2 n)$ and $O(\log^{\lceil (k+1)/2 \rceil} n)$. The first gap is from Tarski$(n, 4)$, where the lower bound is $\Omega(\log^2 n)$ and the upper bound is $O(\log^3 n)$.
With regards to the white-box model, it is known that Tarski is in the intersection of PPAD and PLS [1], and so is in CLS [3] and EOPL [4]. It would also be very interesting to see if Tarski is complete for some computational complexity classes.

References

[1] Kousha Etessami et al. "Tarski's Theorem, Supermodular Games, and the Complexity of Equilibria". In: *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik. 2020.

[2] John Fearnley, Dömötör Pálvölgyi, and Rahul Savani. "A faster algorithm for finding Tarski fixed points". In: *ACM Transactions on Algorithms (TALG)* (2020).

[3] John Fearnley et al. "The complexity of gradient descent: CLS= PPAD ∩ PLS". In: *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. 2021, pp. 46–59.

[4] Mika Göös et al. "Further Collapses in TFNP". In: *arXiv preprint arXiv:2202.07761* (2022).

[5] Alfred Tarski. "A lattice-theoretical fixpoint theorem and its applications." In: *Pacific journal of Mathematics* 5.2 (1955), pp. 285–309.