# The Complexity of Gradient Descent

**Rahul Savani**
Department of Computer Science, University of Liverpool, UK

**Abstract**

PPAD and PLS are successful classes that capture the complexity of important game-theoretic problems. For example, finding a mixed Nash equilibrium in a bimatrix game is PPAD-complete, and finding a pure Nash equilibrium in a congestion game is PLS-complete. Many important problems, such as solving a Simple Stochastic Game or finding a mixed Nash equilibrium of a congestion game, lie in both classes. It was strongly believed that their intersection, PPAD ∩ PLS, does not have natural complete problems. We show that it does: any problem that lies in both classes can be reduced in polynomial time to the problem of finding a stationary point of a continuously differentiable function on the domain $[0, 1]^2$. Thus, as PPAD captures problems that can be solved by Lemke-Howson type complementary pivoting algorithms, and PLS captures problems that can be solved by local search, we show that PPAD ∩ PLS exactly captures problems that can be solved by Gradient Descent.

This is joint work with John Fearnley, Paul Goldberg, and Alexandros Hollender. It appeared at STOC'21, where it was given a Best Paper Award [4].

**2012 ACM Subject Classification** Theory of computation → Problems, reductions and completeness; Theory of computation → Mathematical optimization; Mathematics of computing → Continuous functions

**Keywords and phrases** Computational Complexity, Continuous Optimization, TFNP, PPAD, PLS, CLS, UEOPL

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2021.5

**Category** Invited Talk

**Related Version** *Full Version*: `https://doi.org/10.1145/3406325.3451052`

## 1 Talk summary

This talk is about the computational complexity of Gradient Descent, one of the oldest and most widely-used algorithmic approaches to doing optimisation. The approach dates all the way back to an 1847 paper of Cauchy.

When Gradient Descent is constrained to a bounded domain, there are not one but two reasons why it must terminate at an approximate stationary point or boundary point where the gradient is trying to take it outside the domain:

- We are always going downhill, altitude must "bottom out". This puts the search for a solution in the complexity class PLS (polynomial local search).
- Gradient Descent maps any point to a nearby point in the direction of the negative gradient. Brouwer's Fixed Point Theorem guarantees that such a mapping has a point mapped to itself. This puts the search for a solution in the complexity class PPAD.

PPAD and PLS correspond to existence-of-solution proof principles that guarantee solutions, but in a computationally-inefficient way. Both classes have become successful through the fact that they have been shown to exactly characterise the complexity of important problems. Our main result shows that the Gradient Descent solution-existence principle tastefully combines the PLS principle with the PPAD principle:


41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).
Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 5; pp. 5:1–5:2
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

> We show how to efficiently reduce any problem that is in both PPAD and PLS to the problem of finding a stationary point of a continuously differentiable function from $[0,1]^2$ to $[0,1]$.

This is the first natural problem to be shown complete for PPAD $\cap$ PLS. Our results also imply that the class CLS (Continuous Local Search) [2] – which was defined by Daskalakis and Papadimitriou as a more "natural" counterpart to PPAD $\cap$ PLS and contains many interesting problems – is itself equal to PPAD $\cap$ PLS.

Our result has been used to show that computing a mixed equilibrium of a congestion game is also complete for PPAD $\cap$ PLS [1], and, as we discuss in [4], it opens up the possibility of PPAD $\cap$ PLS hardness for other important problems, such as finding Tarski fixed points [3, 6] or finding solutions that are guaranteed to exist by the Colorful Carathéodory theorem [7]. Several of the other problems in the original CLS paper [2], such as the P-matrix Linear Complementarity Problem and finding the fixed point of (piecewise linear) Contraction map, have unique solutions. For these problems, we believe that another class, called UEOPL, for Unique End of Potential Line, is more likely than PPAD $\cap$ PLS to be the correct class to capture their complexity [5].

## References


1. Yakov Babichenko and Aviad Rubinstein. Settling the Complexity of Nash Equilibrium in Congestion Games. In *Proc. of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 1426–1437, 2021.
2. Constantinos Daskalakis and Christos H. Papadimitriou. Continuous Local Search. In *Proc. of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 790–804, 2011.
3. Kousha Etessami, Christos H. Papadimitriou, Aviad Rubinstein, and Mihalis Yannakakis. Tarski's Theorem, Supermodular Games, and the Complexity of Equilibria. In *Proc. of the 11th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 18:1–18:19, 2020.
4. John Fearnley, Paul W. Goldberg, Alexandros Hollender, and Rahul Savani. The Complexity of Gradient Descent: CLS = PPAD $\cap$ PLS. In *Proc. of the 53rd Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 46–59, 2021.
5. John Fearnley, Spencer Gordon, Ruta Mehta, and Rahul Savani. Unique End of Potential Line. *Journal of Computer and System Sciences*, 114:1–35, 2020.
6. John Fearnley and Rahul Savani. A Faster Algorithm for Finding Tarski Fixed Points. In *Proc. of the 38th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 29:1–29:16, 2021.
7. Frédéric Meunier, Wolfgang Mulzer, Pauline Sarrabezolles, and Yannik Stein. The Rainbow at the End of the Line – A PPAD Formulation of the Colorful Carathéodory Theorem with applications. In *Proc. of the 28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1342–1351, 2017.