The Computational Complexity of Avoiding Strict Saddle Points in Constrained Optimization

Ioannis Panageas
University of California, Irvine

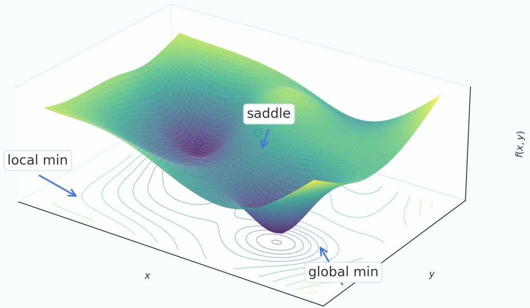
Joint work with: Andreas Kontogiannis (Archimedes AI) and Vasilis Pollatos (Archimedes AI)

June 2026

Non-convex optimization

The basic picture

- We want to solve $\min_{x \in X} f(x)$.
  1. Unconstrained: $X = \mathbb{R}^n$
  2. Constrained: X is convex compact set.
- In general, **global optimization is NP-hard**.
- So algorithmically we often settle for **local certificates**.



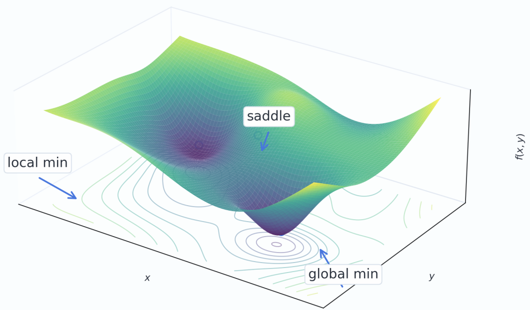
Key message
From **find global optimum** to **find a point where local improvement is impossible.**

Non-convex optimization

The basic picture

- We want to solve $\min_{x \in X} f(x)$.
  1. Unconstrained: $X = \mathbb{R}^n$
  2. Constrained: X is convex compact set.
- In general, **global optimization is NP-hard**.
- So algorithmically we often settle for **local certificates**.

Key message
From **find global optimum** to **find a point where local improvement is impossible**.



Why stationary points?

Why we care

- They are **local certificates**.
- They are what **iterative methods naturally return**.
- They lead to **total search problems**: a solution always exists.

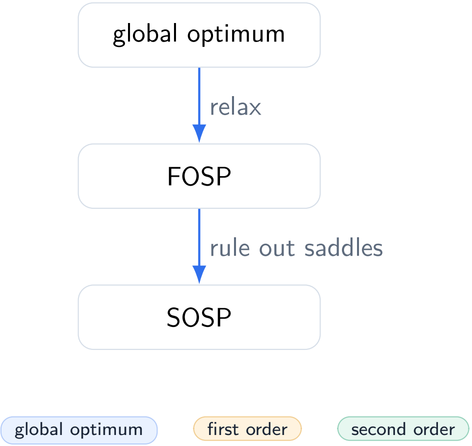
Complexity question
How hard is it to find these certificates in the **worst case**?

Why stationary points?

Why we care

- They are **local certificates**.
- They are what **iterative methods naturally return**.
- They lead to **total search problems**: a solution always exists.

Complexity question
How hard is it to find these certificates in the **worst case**?



Unconstrained FOSP

Definition: ε-FOSP

$$\|\nabla f(x)\| \leq \varepsilon.$$

A point where the gradient is **small**.

- Captures where **gradient descent may stop**.
- Includes local minima, saddles, and maxima.

Known complexity
Approximate unconstrained FOSP is **PLS-complete.** [JPY88, HZ23]

Interpretation
PLS captures **local improvement** with an explicit potential function.

Unconstrained FOSP

Definition: ε-FOSP

$$\|\nabla f(x)\| \le \varepsilon.$$

A point where the gradient is **small**.

- Captures where **gradient descent may stop**.
- Includes local minima, saddles, and maxima.

Known complexity
Approximate unconstrained FOSP is **PLS-complete**. [JPY88, HZ23]

Interpretation
PLS captures **local improvement** with an explicit potential function.

Unconstrained FOSP

Definition: ε-FOSP

$$\|\nabla f(x)\| \leq \varepsilon.$$

A point where the gradient is **small**.

- Captures where **gradient descent may stop**.
- Includes local minima, saddles, and maxima.

Known complexity

Approximate unconstrained FOSP is **PLS-complete**. [JPY88, HZ23]

Interpretation

PLS captures **local improvement** with an explicit potential function.

Why FOSP is not enough

Definition: strict saddle

$$\|\nabla f(x)\| \approx 0, \qquad \lambda_{\min}(\nabla^2 f(x)) < 0.$$

There is a **negative-curvature direction**.

Main issue
A **FOSP can have bad quality compared to local minima.**

- Many dynamics **avoid strict saddles** under assumptions.
- This motivates a **second-order** notion.

[DPG+14, CHM+15, LSJR16, PP17, LPP+19]

Why FOSP is not enough

Definition: strict saddle

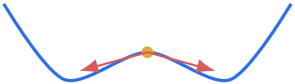
$$\|\nabla f(x)\| \approx 0, \qquad \lambda_{\min}(\nabla^2 f(x)) < 0.$$

There is a **negative-curvature direction**.

Main issue

A **FOSP can have bad quality compared to local minima**.

saddle



escape along negative curvature

- Many dynamics **avoid strict saddles** under assumptions.
- This motivates a **second-order** notion.

[DPG+14, CHM+15, LSJR16, PP17, LPP+19]

Unconstrained SOSP

Definition: $(\varepsilon_G, \varepsilon_H)$-SOSP

$$\|\nabla f(x)\| \le \varepsilon_G, \;\; \lambda_{\min}(\nabla^2 f(x)) \ge -\varepsilon_H.$$

So there is **no large gradient** and **no significant negative curvature**.

Algorithmic side
Cubic regularization, perturbation, and negative-curvature methods all target SOSPs.

Complexity side
Approximate unconstrained SOSP is **PLS-complete**.

[NP06, CDHS18, CR19, KPK+24]

Unconstrained SOSP

Definition: $(\varepsilon_G, \varepsilon_H)$-SOSP

$$\|\nabla f(x)\| \le \varepsilon_G, \;\; \lambda_{\min}(\nabla^2 f(x)) \ge -\varepsilon_H.$$

So there is **no large gradient** and **no significant negative curvature**.

Algorithmic side
Cubic regularization, perturbation, and negative-curvature methods all target SOSPs.

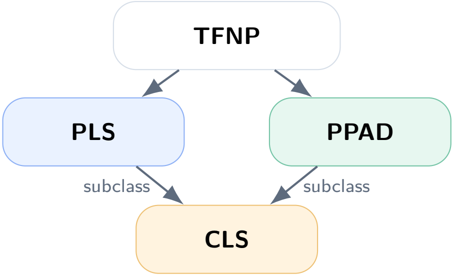
Complexity side
Approximate unconstrained SOSP is **PLS-complete**.

[NP06, CDHS18, CR19, KPK+24]

Complexity language in one slide (Papadimitriou)

Total search classes

- **TFNP**: solutions always exist and are polynomially checkable.
- **PLS**: local search driven by a potential function.
- **CLS**: continuous local search; the class behind many gradient-type problems.



Think of **CLS** as the continuous local-search region where our constrained FOSP problem lives.

For this talk
PLS-complete means: as hard as generic local search.

[MP91, P94, JPY88, DP11]

Constrained FOSP

Now let X be a bounded polytope

$$\min_{x \in X} f(x), \qquad X = \{x \in \mathbb{R}^d : Ax \leq b\}.$$

Definition: projected first-order stationarity

Use the proximal/projected gradient

$$g_\pi(x) = \Big(\Pi_X(x - \nabla f(x)) - x\Big).$$

Then x is approximately stationary when $\|g_\pi(x)\|$ **is small**.

Theorem [FGHS22]
Approximate FOSP aka KKT over bounded polytopes is **CLS-complete**.

Transition
So constraints **lower the complexity** at first order. What about SOSP?

Constrained FOSP

Now let X be a bounded polytope

$$\min_{x \in X} f(x), \qquad X = \{x \in \mathbb{R}^d : Ax \leq b\}.$$

Definition: projected first-order stationarity

Use the proximal/projected gradient

$$g_\pi(x) = \Big(\Pi_X(x - \nabla f(x)) - x\Big).$$

Then x is approximately stationary when $\|g_\pi(x)\|$ **is small**.

Theorem [FGHS22]
Approximate FOSP aka KKT over bounded polytopes is **CLS-complete**.

Transition
So constraints **lower the complexity** at first order. What about SOSP?

Constrained SOSP: the definition issue

Natural definition

- Check second-order behavior over all feasible directions.
- This is mathematically natural.
- But **verification can be NP-hard**.

[MK87, NLR18, MOJ18]

Definition used here

- Restrict to tangent directions of the **active constraints**.
- This version is **polynomial-time checkable**.
- Therefore it fits naturally inside **TFNP**.

[MP91, LRY+20]

Constrained SOSP: the definition issue

Natural definition

- Check second-order behavior over all feasible directions.
- This is mathematically natural.
- But **verification can be NP-hard**.

[MK87, NLR18, MOJ18]

Definition used here

- Restrict to tangent directions of the **active constraints**.
- This version is **polynomial-time checkable**.
- Therefore it fits naturally inside **TFNP**.

[MP91, LRY+20]

Active constraints determine the tangent space

$$A'(x) = \text{rows of } A \text{ active at } x, \qquad A'(x)y = 0.$$

Definition: constrained SOSP (checkable version)

$$\boxed{\|g_\pi(x)\| \le \varepsilon_G} \qquad \text{and} \qquad \boxed{y^\top \nabla^2 f(x) y \ge -\varepsilon_H \ \ \forall y: \ A'(x)y = 0, \ \|y\| = 1.}$$

Equivalently, $\lambda_{\min}(P(x)\nabla^2 f(x)P(x)) \ge -\varepsilon_H$.

check active set | compute tangent space | eigenvalue test

[LRY+20]

The TFNP definition

Active constraints determine the tangent space

$$A'(x) = \text{rows of } A \text{ active at } x, \qquad A'(x)y = 0.$$

Definition: constrained SOSP (checkable version)

$$\boxed{\|g_\pi(x)\| \le \varepsilon_G} \quad \text{and} \quad \boxed{y^\top \nabla^2 f(x) y \ge -\varepsilon_H \ \ \forall y : \ A'(x)y = 0, \ \|y\| = 1.}$$

Equivalently, $\lambda_{\min}(P(x)\nabla^2 f(x) P(x)) \ge -\varepsilon_H$.

check active set | compute tangent space | eigenvalue test

[LRY+20]

Main theorem

Constrained-SOSP is PLS-complete.

Hardness already holds on the **unit square** $[0,1]^2$.

It also holds for the **promise version**: no violation certificates.

It still holds when stationary points are **far from the boundary**.

tractable definition | compact domain | white-box model

[KPP26]

Main theorem

Constrained-SOSP is PLS-complete.

Hardness already holds on the **unit square** $[0,1]^2$.
It also holds for the **promise version**: no violation certificates.
It still holds when stationary points are **far from the boundary**.

tractable definition | compact domain | white-box model

[KPP26]

Why this theorem is surprising

Before our work

	FOSP	SOSP
unconstrained	PLS	PLS
constrained	CLS	?

After our work

	FOSP	SOSP
unconstrained	PLS	PLS
constrained	CLS	**PLS**

Constraints reduce **first-order** complexity,
but not **second-order** complexity.

[FGHS22, HZ23, KPK+24, KPP26]

Reduction overview

The reduction encodes a **discrete local-search path** inside a smooth **two-dimensional landscape**.

Hardness

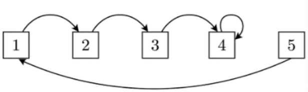
Every approximate SOSP reveals a **solution of the ITER instance**.

Membership

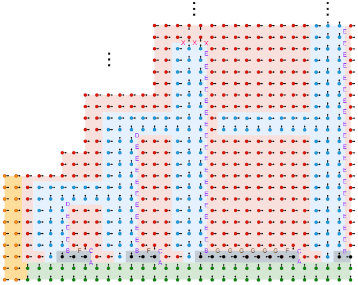
If a point is **not** a constrained SOSP, there is a **local improving neighbor**.

[Mor01, JPY88, FGHS22, HZ23, KPK+24, KPP26]

Hardness picture: embedding ITER



Example ITER instance: $1 \mapsto 2 \mapsto 3 \mapsto 4 \mapsto 4$, and $5 \mapsto 1$.



blue columns = ITER nodes; approximate SOSPs appear only near solutions

Definition 2.1. ITER (informal):
Input: $C : [2^n] \to [2^n]$ with $C(1) > 1$.
Goal: Find $v \in [2^n]$ such that either

- $C(v) < v$, or
- $C(v) > v$ and $C(C(v)) = C(v)$.

High-level idea

- The colors prescribe values and gradients on a grid.
- **Blue columns and corridors** encode the ITER graph.
- Biquintic interpolation makes the function **smooth**.

Membership in PLS

Neighbor map

SNAP update rule

$$h(x) = \begin{cases} \text{PGD step}, & \|g_\pi(x)\| > \varepsilon_G, \\ \text{negative-curvature step}, & \lambda_{\min} < -\varepsilon_H, \\ x, & \text{already SOSP}. \end{cases}$$

Then discretize and set $g(x) = \mathrm{Round}(h(x))$.

Potential function

$$p(x) = f(x) + \eta \dim \mathrm{Null}(A'(x)).$$

- A PGD or curvature step **decreases** f.
- Hitting the boundary decreases the **tangent dimension**.
- Rounding preserves enough decrease to define a **PLS neighborhood**.

[LRY+20, KPP26]

Implications

Suppose a continuous iterative algorithm existed

$$x_{t+1} = \mathrm{Alg}(f, \varepsilon, x_t)$$

- deterministic,
- continuous in x,
- polynomial-time evaluable,
- approximate fixed points are constrained SOSPs.

Consequence
Interior-SOSP would reduce to **Brouwer fixed point search**.

Barrier
This would place a **PLS-hard** problem inside **PPAD**.

No such algorithm unless $\mathrm{PLS} \subseteq \mathrm{PPAD}$

[DGP09, DP11, KPP26]

Implications

Suppose a continuous iterative algorithm existed

$$x_{t+1} = \mathrm{Alg}(f, \varepsilon, x_t)$$

- deterministic,
- continuous in x,
- polynomial-time evaluable,
- approximate fixed points are constrained SOSPs.

Consequence
Interior-SOSP would reduce to **Brouwer fixed point search**.

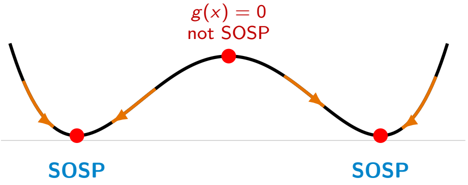
Barrier
This would place a **PLS-hard** problem inside **PPAD**.

No such algorithm unless $\mathrm{PLS} \subseteq \mathrm{PPAD}$

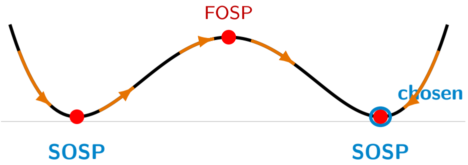
[DGP09, DP11, KPP26]

Implications in a picture

1. Split basins: continuity forces a zero



2. One target: continuous but inefficient



All arrows point to a specific SOSP, but this is not efficient.

Message

Split basins. If a continuous iterative rule sends nearby points to different SOSPs, then at the basin boundary the update/flow must vanish. This zero can be first-order stationary but *not* second-order.

One target. Selecting one SOSP globally avoids the intermediate zero, but it may force all initializations to travel to the same solution, which is inefficient.

Takeaways

1. The **checkable** constrained SOSP notion is still **PLS-complete**.
2. Hardness already appears in **two dimensions** on $[0,1]^2$.
3. Continuous deterministic update rules face a **PLS-vs-PPAD barrier**.

Open: black-box complexity

Open: stronger approximation guarantees