Min CSPs
on
Complete Instances

by

Aditya Anand

Euiwoong Lee

Amatya Sharma

Abstract

- This talk is about
  - Minimization Constraint Satisfiability Problems on Complete Instances.
- What is new?
  - $O(1)$-approx for Min-2-SAT on Complete Instances.
  - $n^{O(\log n)}$ time algorithm for k-CSPs, $\forall k \geq 2.$
  - NP-Hardness for 2-CSPs with large alphabet
- Why is this new?
  - Prior best approximation for Min-2-SAT: $O(\sqrt{\log n})$
  - First systematic study on Min-CSPs

CSP example

3-SAT:

- Variables $x_1, x_2, \dots, x_n$
- Clauses:

  $(x_1 \vee \neg x_3 \vee x_4), (\neg x_4 \vee \neg x_{13} \vee x_7), \dots$

  Clause is unsatisfied iff all literals are false

NAE-3-SAT

- Clauses:

  $(x_1, \neg x_3, x_4), (\neg x_4, \neg x_{13}, x_7), \dots$

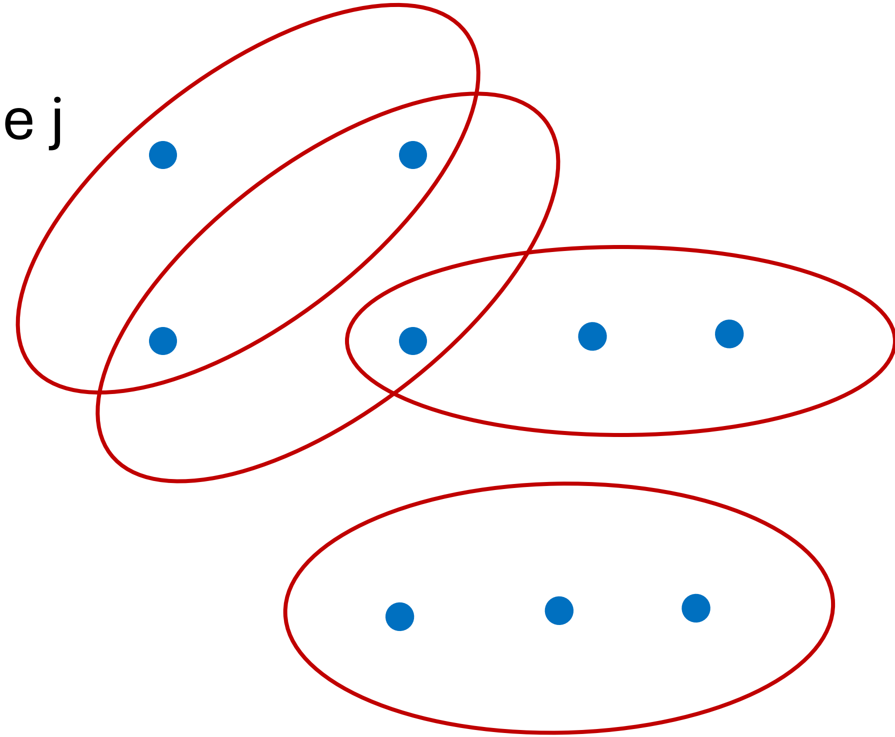
  Clause is unsatisfied iff all literals have the same assignment

Many more examples, 3-LIN, 3-AND, 2-SAT, Horn-SAT, SAT...

Constraint Satisfiability Problems

k-CSP:

- n Boolean Variables
- m Clauses with
  - k-variables each clause
  - Predicate $P_j : \{0,1\}^k \mapsto \{sat, unsat\}$ for each clause j



Instance is a k-uniform hypergraph

Objectives

Find an assignment to the variables that…

…satisfies all the clauses?

- Decision CSP
  - SAT, 3-SAT, NAE-SAT, k-Coloring

… maximizes #satisfied clauses

- Max-CSP
  - e.g. Max-Cut, Max-2-SAT, Max-k-SAT, Unique Games

…minimizes #unsatisfied clauses?

- Min-CSP
  - e.g. Min-Uncut, Min-2-SAT, Min-3-SAT

A Brief History about CSPs

Max-CSPs

General Instances

- Applications
  - NP-Optimization
  - Graph Cuts
  - ETH
- Tools
  - Linear, Semi Definite Programming
  - Probabilistically Checkable Proofs
  - Unique Games Conjecture

Optimal* Approximation Algorithms

- Max-3-SAT, Max-3-LIN, Max-CUT, Unique Games [Has01, Kho02, KKMO07]

Structured Instances

Instance is a hypergraph H.

What are some basic structural assumptions?

- Dense Instance
  - H is ‘’almost’’ complete, $|E| = \Omega(n^k)$
- Expanding Instance
  - H has some expansion properties

Max-CSPs

Structured Instances

- Tools
  - Random Sampling
  - Convex Hierarchies
  - Regularity Lemmas

- Every Max-CSP has a PTAS on dense/expanding instances via **any tool**

Min-CSPs

- General Instances
  - Unlike Max-CSPs*, not all Min-CSPs admit $O(1)$-approx.
  - Structural Characterization identifies optimal approximations as one of
    $1, O(1), \text{poly}\log(n), \text{poly}(n)$ [KSTW01]

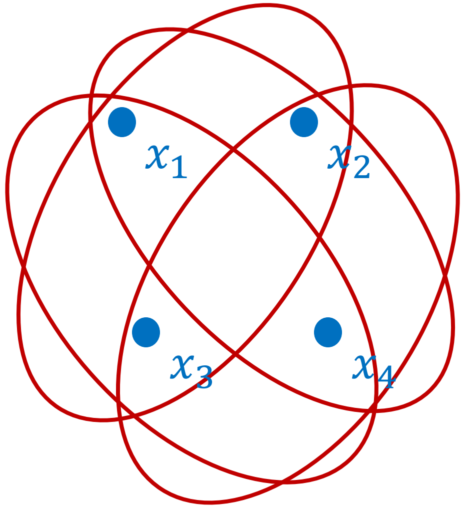
- Structured Instances
  - Min Uncut & Unique Games
    - $O(1)$- approximation for on dense/expanding instances [BFdLVK03, KS09, GS11, MdMMN23]
  - Fragile CSPs
    - PTAS on dense instances [KS09]
  - Min-2-SAT
    - UGC-Hard to get $O(1)$-approximation

Complete Instances

- Min-k-CSP on complete instance hypergraph is complete
- Every k-tuple of variables is a clause

e.g. Min-3-SAT complete instance on 4 variables:

$(x_1 \vee \neg x_2 \vee x_3)$,
$(x_1 \vee \neg x_2 \vee \neg x_4)$,
$(x_1 \vee x_3 \vee x_4)$,
$(x_2 \vee \neg x_3 \vee x_4)$.



Motivation

- Fine Grained Understanding of Instance Structures
- Connections to DS/ML
  - Correlation Clustering
  - Low rank approximation
- New Algorithmic Techniques
  - Combination of tools for Max-CSPs
    - Random Sampling, Convex Hierarchies, Regularity Lemmas

Our Results

- Min-CSPs on complete instances

Thm. poly(n) time $O(1)$-approx for Min-2-SAT on complete instances

What about generalizing to Min-3-SAT, Min-k-SAT?

- Before approximation, comes decision…

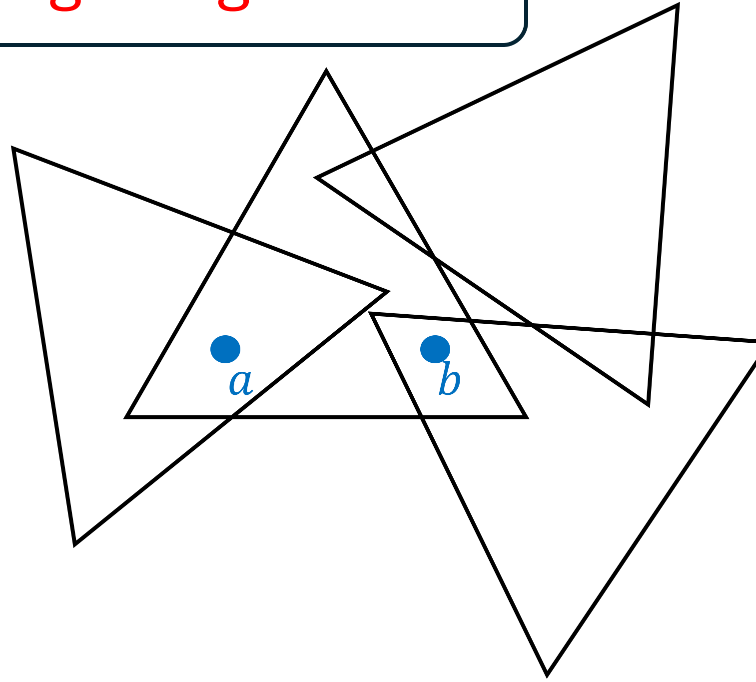
Thm. $n^{O(\log n)}$ time algorithm for decision k-CSP on complete instances

Toy Example1: Decision 2-SAT

Thm. Any 2-SAT complete instance has $O(n)$ satisfying assignments.

Proof.

- VC-dim $\leq k$ => #sets $\leq O(n^{k-1})$ [Sau72, She72]
- Elements: n variables
- Every satisfying assignment forms a set
  - Set of all the true variables



[Fed94] all assignments for 2-SAT can be found in time $O(n * \#satisfying\ assignments + m)$

Corr. All the satisfying assignments can be found in $O(n^2)$ time.

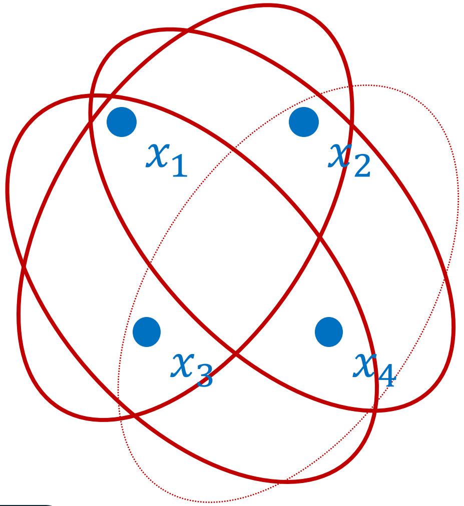
Toy Example1: Decision 2-SAT

Thm. Any k-CSP complete instance has $O(n^{k-1})$ satisfying assignments.

Unlike 2-CSP, no easy way to find these assignments in polynomial time.

For 3-CSP, we do it in time $n^{O(\log n)}$
and for 4-CSP, k-CSP in general.

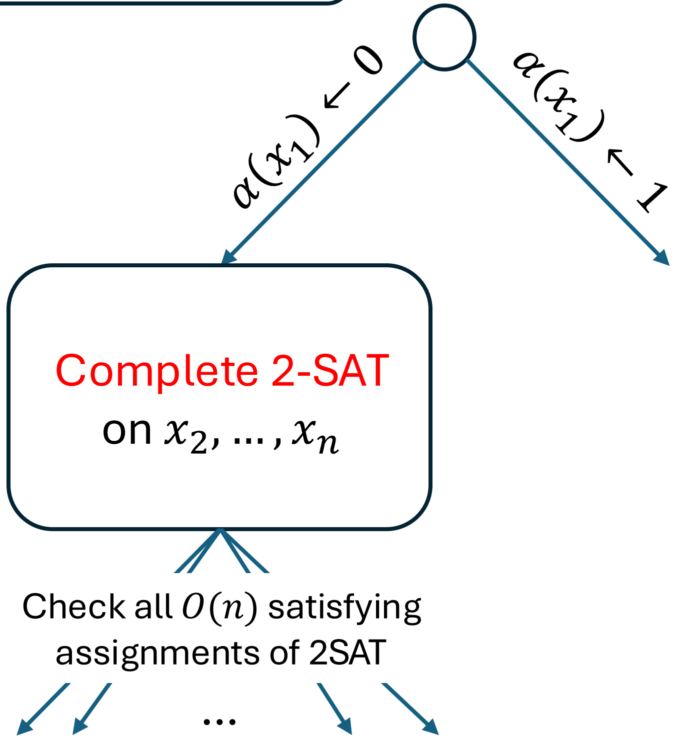
Toy Example2: NAE-3-SAT



- Clause (x, y, z) is unsat iff x,y,z are all assigned the same.

Thm. $O(n^2)$ time algorithm for complete NAE-3-SAT.

- Key Observation:
  - Guess assignment of x, say 1.
    - Both y and z can not be 1
    - 2 SAT constraint $(\neg y \vee \neg z)$
  - Complete instance
    - x has clauses with all $n - 1$ variables
    - 2 SAT constraints for all $n - 1$ variables
    - Complete 2-SAT instance



Idea for 3 SAT

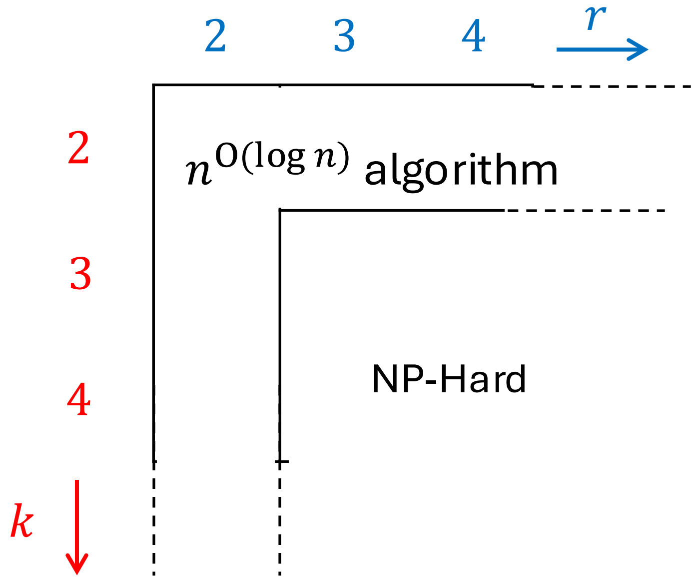
- $(x \vee y \vee z)$ if assignment of x, y is both 0, then z is "**fixed**" to 1.
- If there is a "good" pair (x, y), that fixes $\Omega(n)$ variables:
  - Guess the pair, guess its assignment.
  - $O(\log n)$ rounds fixes all the variables.
- Otherwise,
  - Obtain a complete 2SAT instance
  - Enumerate all $O(n^2)$ 2SAT assignments.

- Runtime $n^{O(\log n)}$

Our Results

Thm. $n^{O(\log n)}$ time algorithm for k-CSP on complete instances

Thm. Complete Classification of (**k**,**r**)-CSP

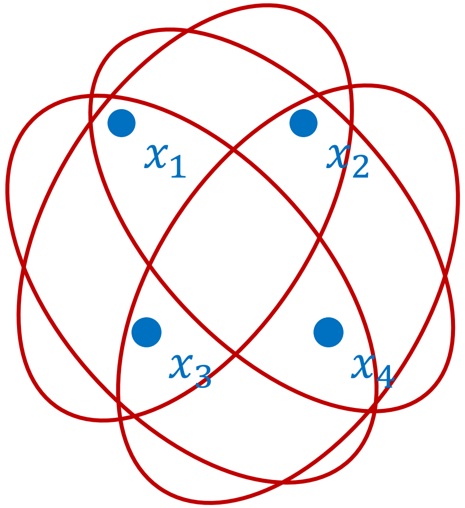


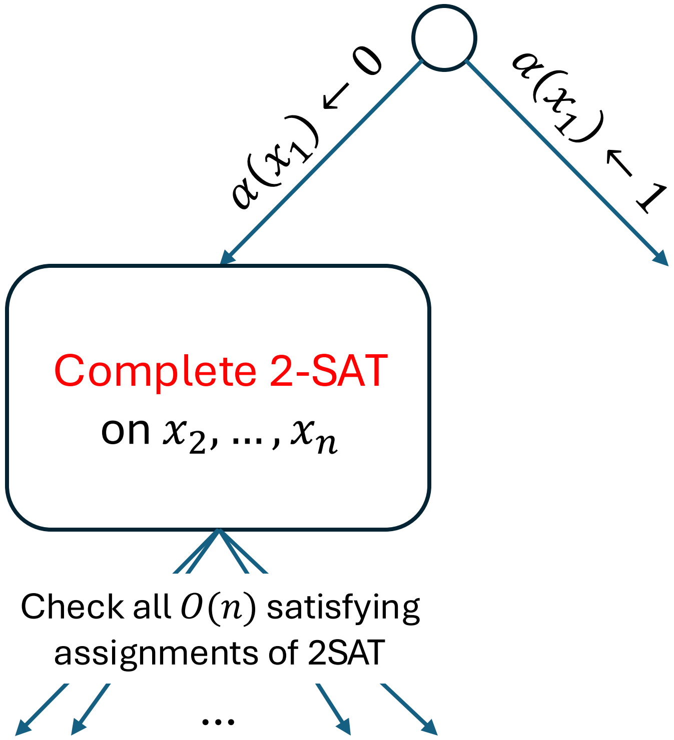
Thm. poly(n) time $O(1)$-approx for Min-2-SAT on complete instances

Open Questions

- Poly time algorithm for complete k-CSP?
  - Even complete 3-SAT?
- Approximating Min-k-CSPs?
  - Exact Approximation is hard unless NP⊆BPP
  - Possibly quasi time?
    - $n^{O(\log n)}$ is optimal for (2,poly(n))-CSPs
- More fine-grained characterizations?
  - k-LIN
  - k-AND

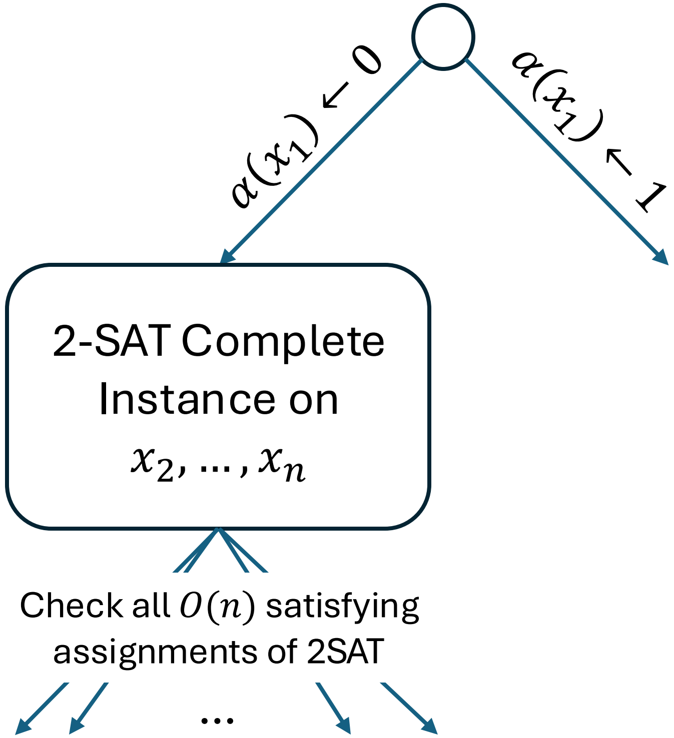
Thank you



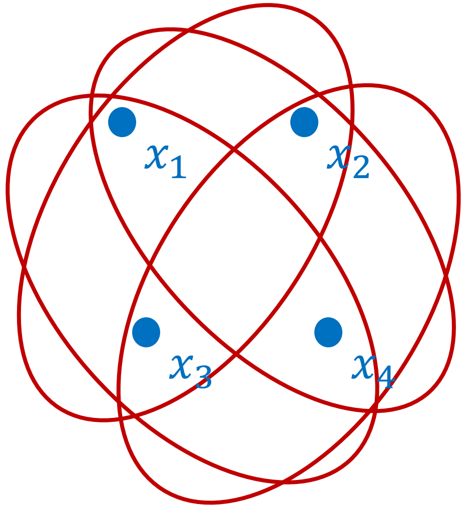


Registration and travel support for this presentation was provided by National Science Foundation.

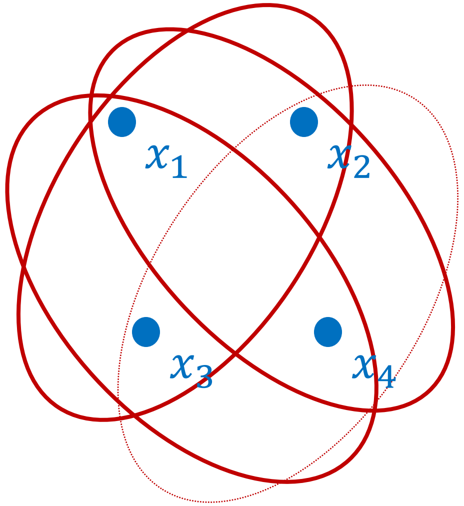
Rough



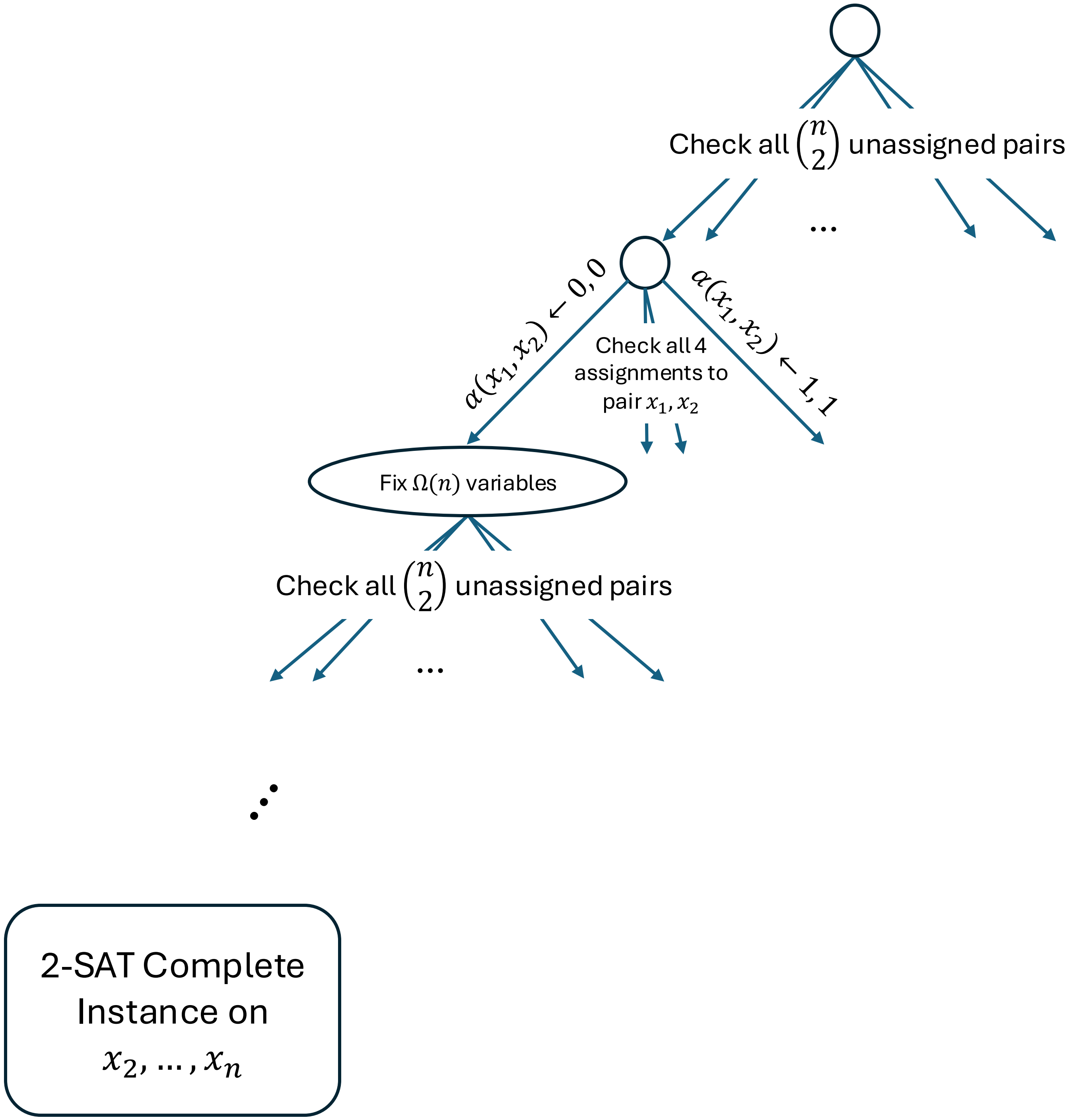
$\alpha(x_1) \leftarrow 0$
$\alpha(x_1) \leftarrow 1$
2-SAT Complete Instance on $x_2, \ldots, x_n$
Check all $O(n)$ satisfying assignments of 2SAT
...



x_1
x_2
x_3
x_4



x_1
x_2
x_3
x_4



Check all $\binom{n}{2}$ unassigned pairs
...
$\alpha(x_1, x_2) \leftarrow 0,0$
Check all 4 assignments to pair x_1, x_2
$\alpha(x_1, x_2) \leftarrow 1,1$
Fix $\Omega(n)$ variables
Check all $\binom{n}{2}$ unassigned pairs
...
2-SAT Complete Instance on $x_2, \ldots, x_n$