

Shortest reconfiguration paths in the solution space of Boolean formulas

Amer Mouawad

Department of Informatics
University of Bergen

Joint work with Naomi Nishimura, Vinayak Pathak, and Venkatesh Raman

June 7, 2016

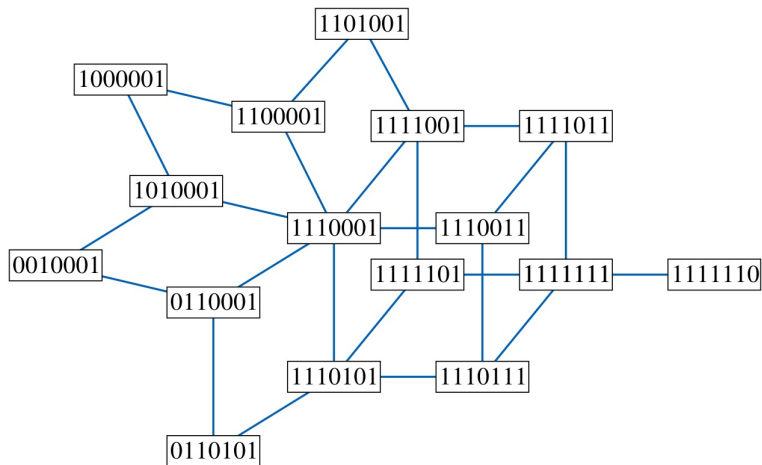
Problem statement

Input: A Boolean formula ϕ (e.g. $x_1 \vee x_2$) and two satisfying assignments \mathbf{s} and \mathbf{t} (e.g. 01 and 10).

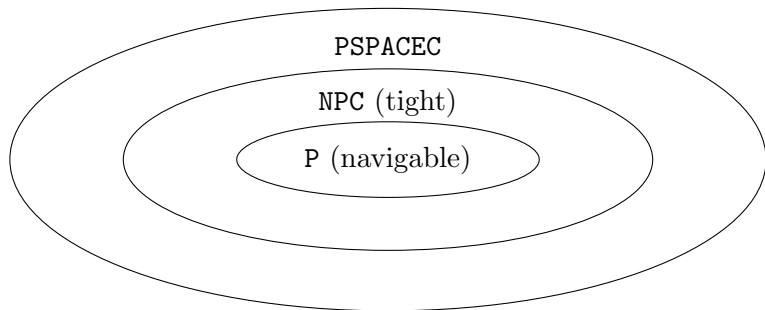
Output: Smallest number of flips to transform \mathbf{s} to \mathbf{t} , where a flip negates a single variable so that the resulting assignment is also satisfying.

Example: $\phi = x_1 \vee x_2$, $\mathbf{s} = 01$, and $\mathbf{t} = 10$.
01 \rightarrow 11 \rightarrow 10 is valid.
01 \rightarrow 00 \rightarrow 10 is **NOT** valid.

Walking in the solution space



The result



Theorem

For any “class” of Boolean formulas computing the shortest reconfiguration path is either in P, NP-complete, or PSPACE-complete.

A bit of history

Schaefer's Theorem (1978)

2-SAT, Horn-SAT, Dual-Horn-SAT, and Affine-SAT are the only (non-trivial) classes in P. Every other class is NP-complete.

Ladner's Theorem (1975)

If $P \neq NP$ then there are an infinite number of non-equivalent SAT classes in NP that are neither in P nor NP-complete.

Catch: Schaefer's definition of "class" is restrictive.

Rest of the talk

- Schaefer's framework
- **st**-connectivity (Gopalan et al., 2009)
- **st**-shortest path

Schaefer's framework

- ▶ Let \mathcal{S} be a finite set of Boolean relations of **constant** arity.
- ▶ A Boolean formula ϕ belongs to the class $\text{CNF}(\mathcal{S})$ if each clause in ϕ is “built” from a relation $R \in \mathcal{S}$.

$$R^0 = \{0, 1\}^3 \setminus \{000\}$$

$$R^1 = \{0, 1\}^3 \setminus \{100\}$$

$$R^2 = \{0, 1\}^3 \setminus \{110\}$$

$$R^3 = \{0, 1\}^3 \setminus \{111\}$$

$$\mathcal{S} = \{R^0, R^1, R^2, R^3\}$$

$$\text{CNF}(\mathcal{S}) \equiv 3\text{CNF}$$

$$(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee \neg x_3) \equiv R^0(x_1, x_2, x_3) \wedge R^2(x_1, x_3, x_2)$$

The reconfiguration graph

- ▶ For any relation R , we let $G_R = (V, E)$ be the graph consisting of one node for each element $\mathbf{v} \in R$ and two nodes \mathbf{u} and \mathbf{v} are connected whenever $Ham(\mathbf{u}, \mathbf{v}) = 1$.
- ▶ For a formula ϕ , we let R_ϕ denote the relation containing all satisfying assignments for ϕ and we let $G_\phi = G_{R_\phi}$.

The reconfiguration graph

- ▶ **st-connectivity**: check if \mathbf{s} and \mathbf{t} belong to the same connected component of G_ϕ .
- ▶ **st-shortest path**: compute the length of a shortest path from \mathbf{s} to \mathbf{t} in G_ϕ .

st-connectivity

Theorem (Folklore)

st-connectivity is in P for 2CNF formulas.

Proof.

We only need to flip variables that are assigned different values in \mathbf{s} and \mathbf{t} . So just keep flipping such variables as long as possible. If we get stuck then there is no path from \mathbf{s} to \mathbf{t} . Otherwise, we will eventually find a (shortest) path.

st-connectivity

Theorem (Gopalan et al., 2009)

st-connectivity is in P for any formula ϕ where each connected component of G_ϕ can be “written” as a 2CNF formula, also called component-wise bijunctive formulas.

Proof.

Generalizes the 2CNF case (also produces shortest paths).

st-connectivity

Definition

A k -ary relation R is called NAND-free (OR-free) if no $k - 2$ of its variables can be assigned values such that the relation induced on the remaining two variables is a NAND (OR), i.e. $\neg x \vee \neg y$ (i.e. $x \vee y$).

001101100	001101100
010010010	010111010
010110010	010110010
010011010	010011010
001101100	001101100

A formula ϕ is NAND-free (OR-free) if it consists entirely of NAND-free (OR-free) relations.

st-connectivity

Theorem (Gopalan et al., 2009)

For a NAND-free formula ϕ , any sequence of flips can be reordered so that all $0 \rightarrow 1$ flips occur before all $1 \rightarrow 0$ flips. The reverse is true for OR-free relations.

Proof. (NAND-free case)

$..0..1.. \rightarrow ..0..0.. \rightarrow ..1..0..$ becomes $..0..1.. \rightarrow ..1..1.. \rightarrow ..1..0..$ If not possible then some relation is not NAND-free.

st-connectivity

Corollary

For a NAND-free (OR-free) formula ϕ , any connected component of G_ϕ contains a unique local maximum (minimum) with respect to hamming weight (polynomial diameter if G_ϕ is connected).

st-connectivity

Definition (Gopalan et al., 2006)

\mathcal{S} is *tight* if at least one of the following conditions holds:

- (1) Every relation in \mathcal{S} is component-wise bijective.
- (2) Every relation in \mathcal{S} is OR-free.
- (3) Every relation in \mathcal{S} is NAND-free.

st-connectivity

Theorem (Gopalan et al., 2006)

For a $\text{CNF}(\mathcal{S})$ formula ϕ , **st**-connectivity is in **P** if \mathcal{S} is tight and **PSPACE**-complete otherwise.

Proof. (NAND-free case)

Try to reach the same assignment by doing $0 \rightarrow 1$ flips from both **s** and **t** (i.e. try to reach the local maximum).

Finding shortest paths

- If not tight \rightarrow PSPACE-complete
- 2CNF and component-wise bijunctive \in P (symmetric difference)

Is there a tight formula for which finding the shortest path is NP-complete?

NP-completeness

- Let $\phi = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4)$ (NAND-free).
- Let $\mathbf{s} = 0000$ and $\mathbf{t} = 1100$.
- If $x_3 = x_4 = 0$, then neither x_1 nor x_2 can be flipped.
- Hence we need at least one of them to be 1.
- Reduction from VERTEX COVER (two clauses per edge).
- Since the formula is tight G_ϕ has polynomial diameter.

A false (but useful) conjecture

- Hardness crucially relied on deciding which common variables should be flipped.
- Conjecture: whenever we have to flip common variables the problem becomes hard.

A false (but useful) conjecture

- Let $P = \{000, 100, 101, 111, 011\}$. So G_P is a path:
 $000 \leftrightarrow 100 \leftrightarrow 101 \leftrightarrow 111 \leftrightarrow 011$.
- There exists formulas in $\text{CNF}(\{P\})$ where any path between two assignments needs to flip common variables.
- However computing the shortest path between any two assignments is in \mathbf{P} for any formula in $\text{CNF}(\{P\})$.
 - Each clause specifies a **unique order** on flips.
 - We just need to check if two clauses are in conflict.

The trichotomy

Definition

\mathcal{S} is *navigable* if at least one of the following conditions holds:

- (1) Every relation in \mathcal{S} is component-wise bijunctive.
- (2) Every relation in \mathcal{S} is OR-free and **Horn-free**.
- (3) Every relation in \mathcal{S} is NAND-free and **dual-Horn-free**.

Theorem

For a $\text{CNF}(\mathcal{S})$ formula ϕ , **st**-shortest path is in **P** if \mathcal{S} is navigable, **NP**-complete if \mathcal{S} is tight but not navigable, and **PSPACE**-complete otherwise.

Proof ideas

- (1) Component-wise bijective: Use the symmetric difference.
- (2) OR-free and **Horn-free**: Unique local minimum + **partial order**.
- (3) NAND-free and **dual-Horn-free**: Unique local maximum + **partial order**.

Open problems

- ▶ Generalize the dichotomy/trichotomy beyond the Boolean domain.
- ▶ How about weighted SAT and/or CSPs?
- ▶ Study the variations in the (parameterized) complexity landscape if we consider simultaneous flips.

Thank you