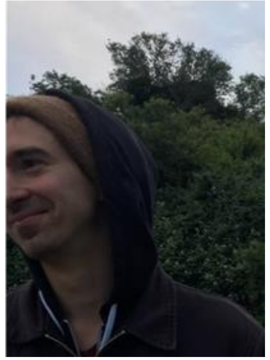
Isoperimetric Inequalities for the Hypercube with Applications to Monotonicity Testing

Sofya Raskhodnikova
Boston University

Joint work with Hadley Black (*UCLA, summer 2020 @ Boston University*)
Iden Kalemaj (*Boston University*)



Based on Iden Kalemaj's slides

BOSTON UNIVERSITY

Isoperimetric Inequalities on the Hypercube

- A tool in the analysis of Boolean functions $f : \{0,1\}^d \to \{0,1\}$
- Study the size of the "boundary" between the points x on which $f(x) = 0$ and on which $f(x) = 1$

Undirected		**Directed**
[Margulis 74]	→	[Chakrabarty, Seshadhri 13]
[Talagrand 93]		[Khot, Minzer, Safra 15]

We generalize these inequalities to **real-valued** functions: $f: \{0,1\}^d \to \mathbb{R}$.

Motivation:

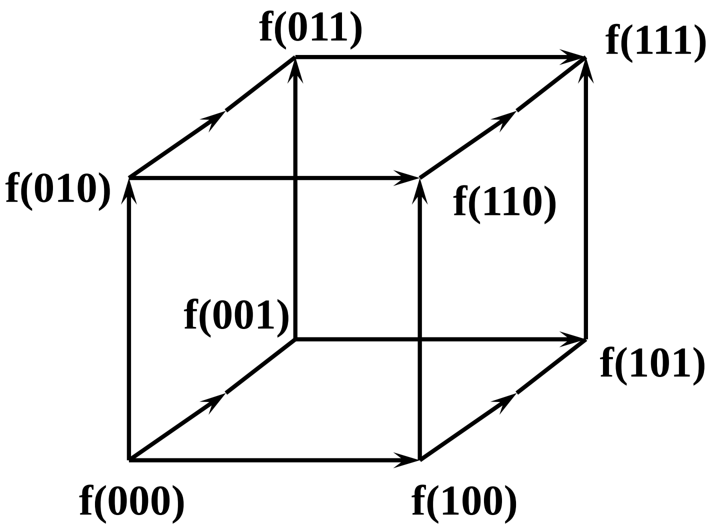
- To understand the structure of real-valued functions.
- To improve sublinear algorithms for monotonicity.

Plan

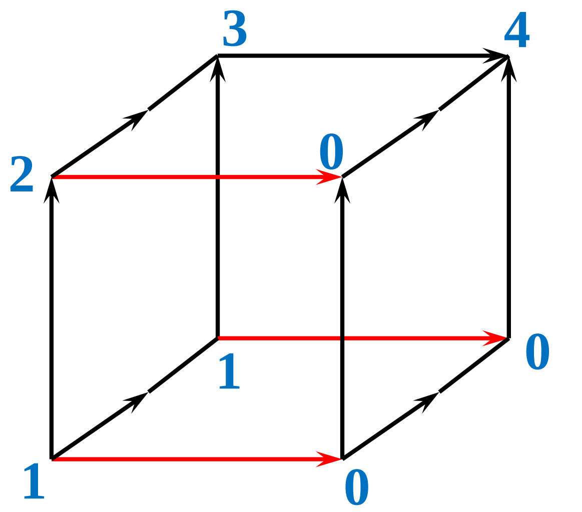
1. Explain our results on sublinear algorithms for monotonicity.
2. Give some background on the isoperimetric inequalities.
3. Prove our generalized inequalities.

The d-Dimensional Hypercube

- Hypercube has 2^d vertices, the points in $\{0,1\}^d$, and $d2^{d-1}$ edges.



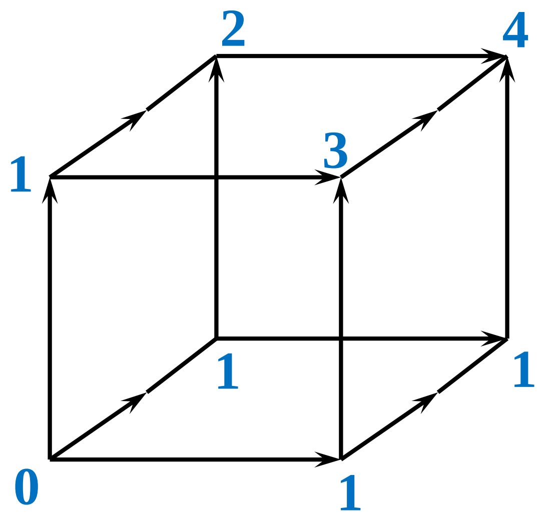
- $x \rightarrow y$ is an edge if:
  $x_i = 0,\, y_i = 1$ for some coordinate $i \in [d]$
  $x_j = y_j$ for all $j \in [n] \backslash \{i\}$



  - Edge $x \rightarrow y$ is influential if $f(x) \neq f(y)$.
  - Edge $x \rightarrow y$ is decreasing if $f(x) > f(y)$.

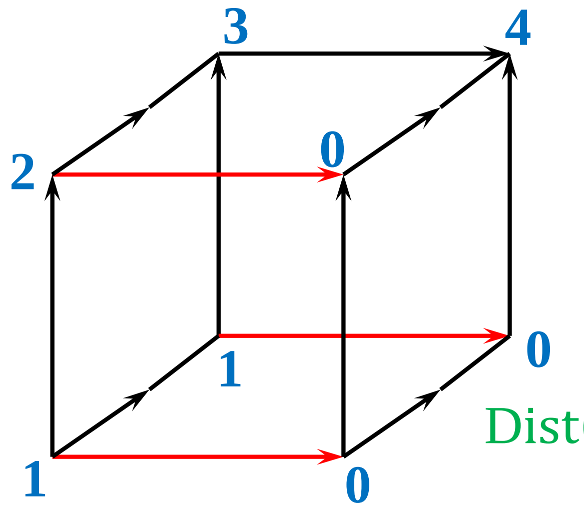
Monotonicity of Functions

- A function $f : \{0,1\}^d \rightarrow \mathbb{R}$ is **monotone** if increasing a bit of x does not decrease $f(x)$, that is, there are no decreasing edges.



f is monotone

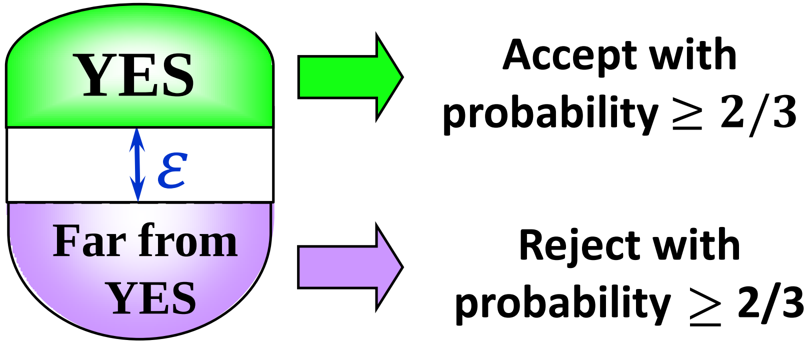
- Distance to monotonicity of f, $\mathbf{Dist}(\boldsymbol{f}, \mathbf{MONO})$, is smallest number of f-values that need to be changed to make it monotone.

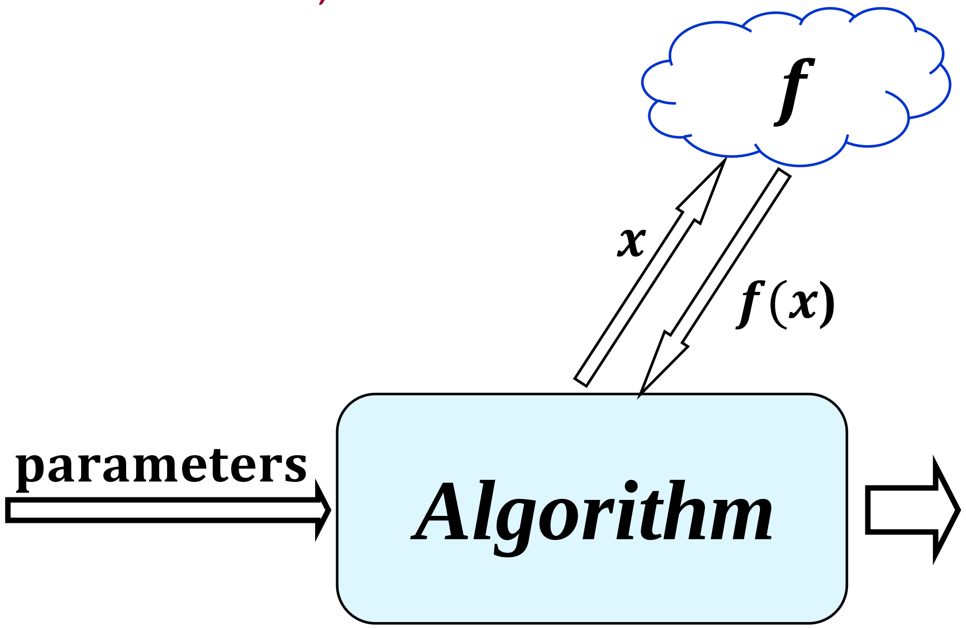


$\mathrm{Dist}(f, \mathrm{MONO}) = 3$

Algorithmic Tasks for Sublinear-Time Algorithms

- Monotonicity testing [Rubinfeld Sudan 96, Goldreich Goldwasser Ron 98, Goldreich Goldwasser Lehman Ron Samorodnitsky 00]





- Approximating distance to monotonicity [Parnas Ron Rubinfeld 06, Fattal Ron 10]
  - with multiplicative and additive error with **probability** $\geq 2/3$

Algorithm is nonadaptive if it makes all queries before receiving any answers.

Results on Monotonicity Testing

- Extensively studied problem [Ergun Kannan Kumar Rubinfeld Viswanathan 00, Lehman Ron 01, Fischer 04, Batu Rubinfeld White 05, Ailon Chazelle 06, Halevy Kushilevitz 08, Bhattacharyya Grigorescu Jung Raskhodnikova Woodruff 12, Briet Chakraborty Soriano Matsliah 12, Berman Raskhodnikova Yaroslavtsev 14, Chakrabarty Seshadhri '13'14'16'19, Chen Servedio Tan 14, Belovs Blais 16, Pallavoor Raskhodnikova Varma 18, Black Chakrabarty Seshadhri '18'20]
- Functions on the hypercube $\{0,1\}^d$, r = number of distinct values of f.

	Boolean	Real-Valued (Previous)	Real-Valued (Our results)
Upper bounds	$\tilde{O}\left(\min\left(\frac{\sqrt{d}}{\varepsilon^2},\frac{d}{\varepsilon}\right)\right)$ [Dodis Goldreich Lehman Raskhodnikova Ron Samorodnitsky 99, Goldreich Goldwassar Ron Lehman Samorodnitsky 00, Khot Minzer Safra 15]	$O\left(\frac{d}{\varepsilon}\right)$ [Chakrabarty Seshadhri 13]	$\tilde{O}\left(\min\left(\frac{r\sqrt{d}}{\varepsilon^2},\frac{d}{\varepsilon}\right)\right)$
Lower Bounds	Nonadaptive: $\tilde{\Omega}(\sqrt{d})$ [Fischer Lehman Newman Raskhodnikova Rubinfeld 02, Chen De Servedio Tan 15, Chen Waingarten Xie 17] Adaptive : $\tilde{\Omega}(d^{1/3})$ [Chen Waingarten Xie 17]	$\Omega(\min(r^2, d))$ [Blais Brody Matulef 12]	$\Omega(\min(r\sqrt{d}, d))$ Nonadaptive, 1-sided error

Results on Approximating the Distance to Monotonicity

- Functions on the hypercube $\{0,1\}^d$, $\boldsymbol{r}$ = number of distinct values of f.
- All algorithms have query complexity polynomial in d and additive error parameter.

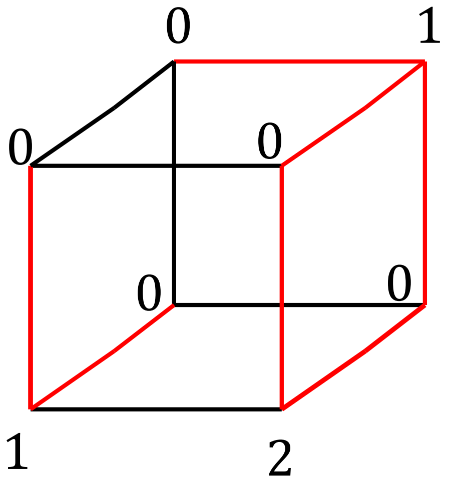
	Boolean [Pallavoor Raskhodnikova Waingarten 20]	**Real-Valued** [Fattal Ron 10]	**Real-Valued (Our results)**
Upper bounds	$O\left(\sqrt{d\log d}\right)$ -factor	$O(d\log \boldsymbol{r})$ -factor	$O\left(\sqrt{d\log d}\right)$-factor
Lower bounds	$\tilde{\Omega}(\sqrt{d})$ -factor nonadaptive		

No dependence on $\boldsymbol{r}$

Isoperimetric Inequalities for Boolean Functions

- Undirected [Talagrand 93]

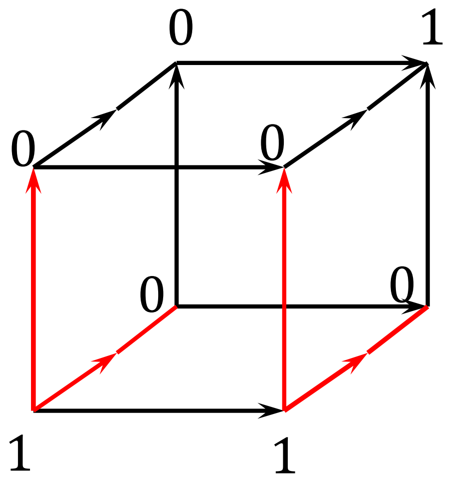
  - Edge (x, y) is influential if $f(x) \neq f(y)$
  - $I_f(x)$ = # influential (x, y) s.t. $f(x) > f(y)$
  - $p_0 =$ fraction of f-values that are 0
    $\mathrm{var}(f) = p_0(1 - p_0)$



$$\sum_{x \in \{0,1\}^d} \left[\sqrt{I_f(x)}\right] = \Omega\left(\mathrm{var}(f) \cdot 2^d\right)$$

- Directed [Khot Minzer Safra 15, Pallavoor Raskhodnikova Waingarten 20]

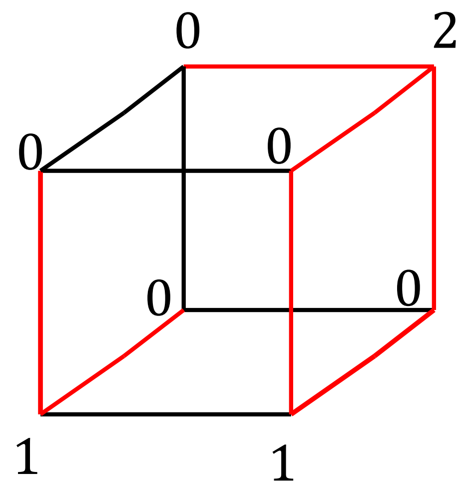
  - Edge $x \rightarrow y$ is decreasing if $f(x) > f(y)$
  - $I_f^-(x)$ = # decreasing edges leaving x



$$\sum_{x \in \{0,1\}^d} \left[\sqrt{I_f^-(x)}\right] = \Omega\left(\mathrm{Dist}(f, \mathrm{MONO})\right)$$

Our Isoperimetric Inequalities for Functions $\boldsymbol{f}: \{\mathbf{0}, \mathbf{1}\}^{\boldsymbol{d}} \to \mathbb{R}$

- Undirected



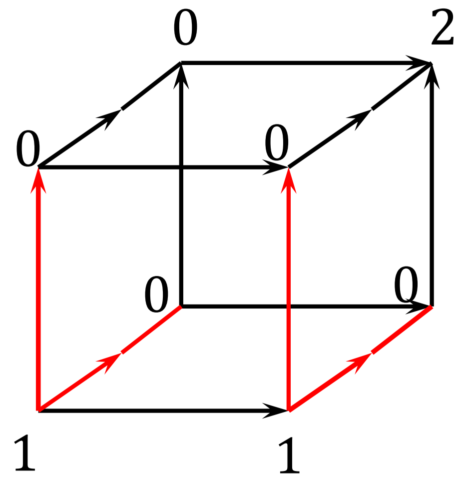
> - Edge (x, y) is influential if $f(x) \neq f(y)$
> - $I_f(x) = \#$ influential (x, y) s.t. $f(x) > f(y)$
> - $p_0 =$ fraction of f-values that are 0
>
> $\mathrm{var}(f) = p_0(1 - p_0)$

$$\sum_{x \in \{0,1\}^d} \left[\sqrt{I_f(x)}\right] = \Omega\big(\mathrm{Dist}(f, \mathrm{CONSTANT})\big)$$

Number of values that need to be changed to make f constant

For a Boolean function f, $\mathrm{var}(f) \cdot 2^d$ and $\mathrm{Dist}(f, \mathrm{CONSTANT})$ are within a factor of 2

- Directed



> - Edge $x \to y$ is decreasing if $f(x) > f(y)$
> - $I_f^-(x) = \#$ decreasing edges leaving x

$$\sum_{x \in \{0,1\}^d} \left[\sqrt{I_f^-(x)}\right] = \Omega\big(\mathrm{Dist}(f, \mathrm{MONO})\big)$$

No dependence on $\boldsymbol{r}$

Main Isoperimetric Inequality (Directed)

Main Inequality

For all functions $f\colon \{0,1\}^d \to \mathbb{R}$,

$$\sum_{x\in\{0,1\}^d}\left[\sqrt{I_f^-(x)}\right] = \Omega\big(\mathrm{Dist}(f,\mathrm{MONO})\big)$$

$I_f^-(x)$ = # decreasing edges leaving x

- The inequality we use in our applications.
- Implies all other inequalities mentioned in this talk.
- We show how to prove it.

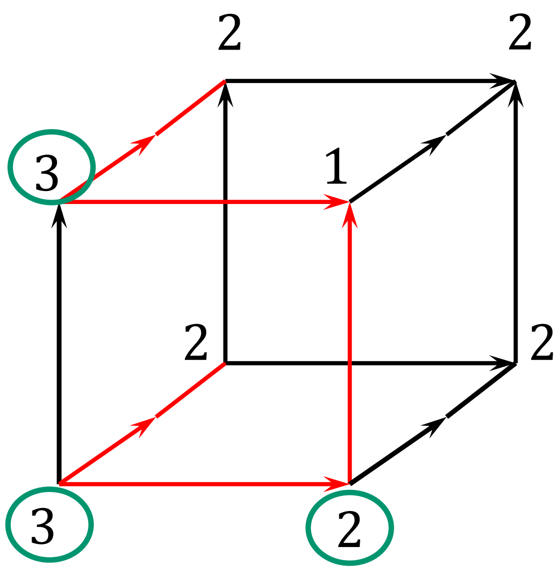
Main Isoperimetric Inequality

Main Inequality

For all functions $f\colon \{0,1\}^d \to \mathbb{R}$,

$$\sum_{x\in\{0,1\}^d}\left[\sqrt{I_f^-(x)}\right] = \Omega\big(\mathrm{Dist}(f,\mathrm{MONO})\big)$$

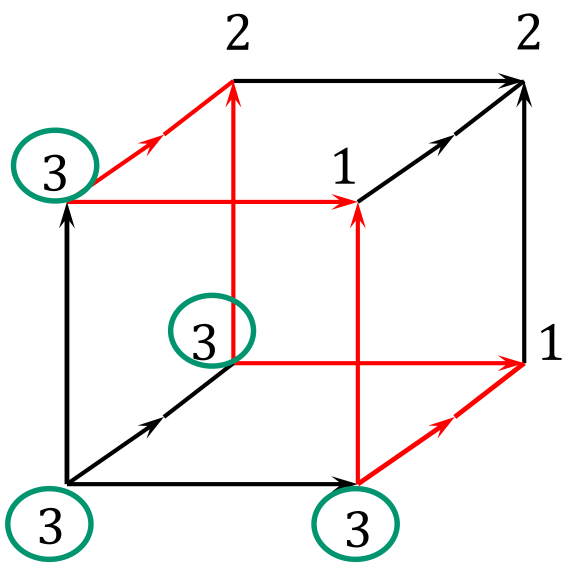
$I_f^-(x)$ = # decreasing edges leaving x



$\mathrm{Dist}(f,\mathrm{MONO}) = 3$

$$\sum_{x\sim\{0,1\}^d}\left[\sqrt{I_f^-(x)}\right]$$

$$= \sqrt{2} + \sqrt{2} + \sqrt{1}$$



$\mathrm{Dist}(f,\mathrm{MONO}) = 4$

$$\sum_{x\sim\{0,1\}^d}\left[\sqrt{I_f^-(x)}\right]$$

$$= \sqrt{2} + \sqrt{2} + \sqrt{2}$$

Main Isoperimetric Inequality

Main Inequality

For all functions $f\colon \{0,1\}^d \to \mathbb{R}$,

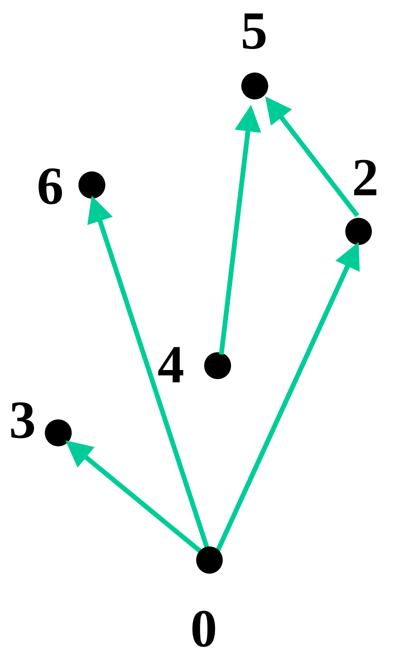
$$\sum_{x\in\{0,1\}^d} \left[\sqrt{I_f^-(x)}\right] = \Omega\big(\mathrm{Dist}(f, \mathrm{MONO})\big)$$

$I_f^-(x)$ = # decreasing edges leaving x

- We prove it by reducing to the Boolean case, via Boolean Decomposition Theorem.

Boolean Decomposition Theorem

- It holds for every partially ordered domain, which we represent as a DAG G.
- Monotonicity testing on POsets first studied by [Fischer Lehman Newman Raskhodnikova Rubinfeld 02]
  - Is equivalent to several other property testing problems
- Vertices $V(G)$, edges $E(G)$.
- $x \preccurlyeq y$ iff there is directed path from x to y.
- Edge $x \rightarrow y$ is decreasing if $f(x) > f(y)$.



$f : V(G) \rightarrow \mathbb{R}$

Boolean Decomposition Theorem

- $\mathrm{DE}(f)$ = set of decreasing edges w.r.t. f

Boolean Decomposition Theorem

Let G be a DAG, and $f: V(G) \to \mathbb{R}$ a nonmonotone function.
There exist $k \geq 1$, Boolean functions $f_1, f_2, \ldots, f_k: V(G) \to \{0,1\}$ and disjoint subgraphs $H_1, H_2, \ldots, H_k$ of G such that:

1) $\mathrm{DE}(f_i) \subseteq E(H_i) \cap \mathrm{DE}(f)$

2) $\sum_{i \in [k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$

Edges decreasing w.r.t. f_i are in H_i and are also decreasing w.r.t. f

Boolean functions f_i capture $\mathrm{Dist}(f, \mathrm{MONO})$

Boolean Decomposition Theorem ⇒ Main Inequality

$I_f^-(x)$ = # decreasing edges leaving x

Main Inequality

For all functions $f\colon \{0,1\}^d \to \mathbb{R}$,

$$\sum_{x\in\{0,1\}^d} \left[\sqrt{I_f^-(x)}\right] = \Omega\big(\mathrm{Dist}(f, \mathrm{MONO})\big)$$

$$\sum_{\boldsymbol{x\in\{0,1\}^d}} \left[\sqrt{I_f^-(x)}\right] \geq \sum_{\boldsymbol{i\in[k]}} \sum_{\boldsymbol{x\in V(H_i)}} \left[\sqrt{I_f^-(x)}\right]$$

Subgraphs H_i are disjoint

$$\geq \sum_{i\in[k]} \sum_{x\in V(H_i)} \left[\sqrt{I_{\boldsymbol{f_i}}^-(x)}\right]$$

Edges decreasing w.r.t. f_i are in H_i and are also decreasing w.r.t. f

$$\geq \sum_{i\in[k]} C \cdot \mathrm{Dist}\big(\boldsymbol{f_i}, \mathrm{MONO}\big)$$

By the Boolean case

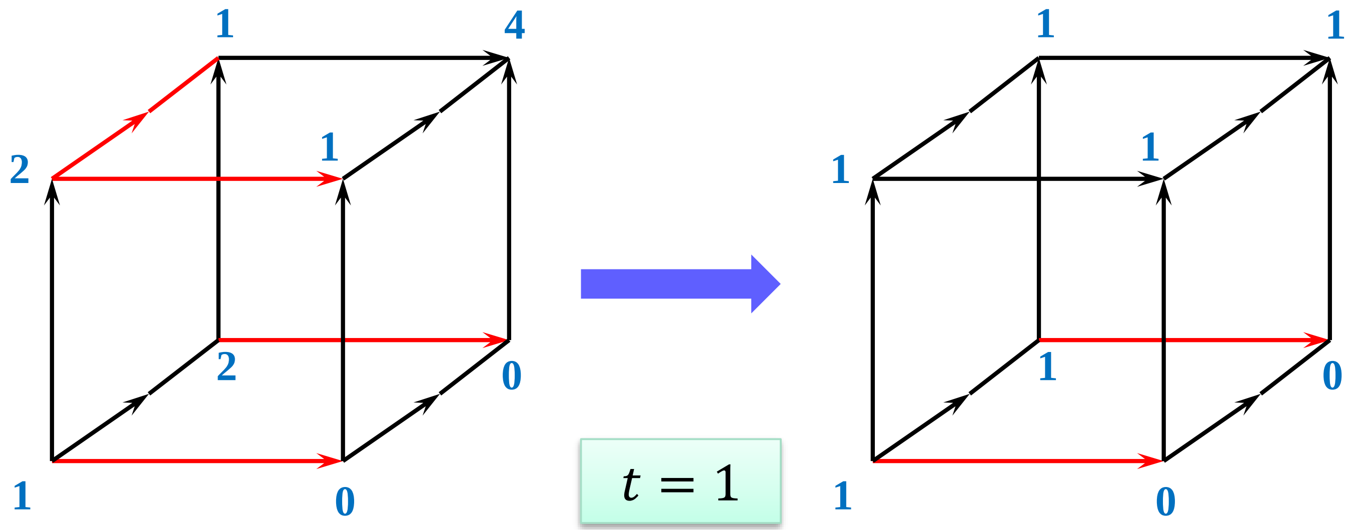
$$\geq \frac{C}{2} \cdot \mathrm{Dist}(f, \mathrm{MONO})$$

Boolean functions f_i capture $\mathrm{Dist}(f, \mathrm{MONO})$

Boolean Decomposition: Thresholding Intuition

- To reduce from real-valued to Boolean functions, we can consider thresholding:

$$h_t(x) = \begin{cases} 1 & \text{if } f(x) \geq t \\ 0 & \text{otherwise} \end{cases}$$

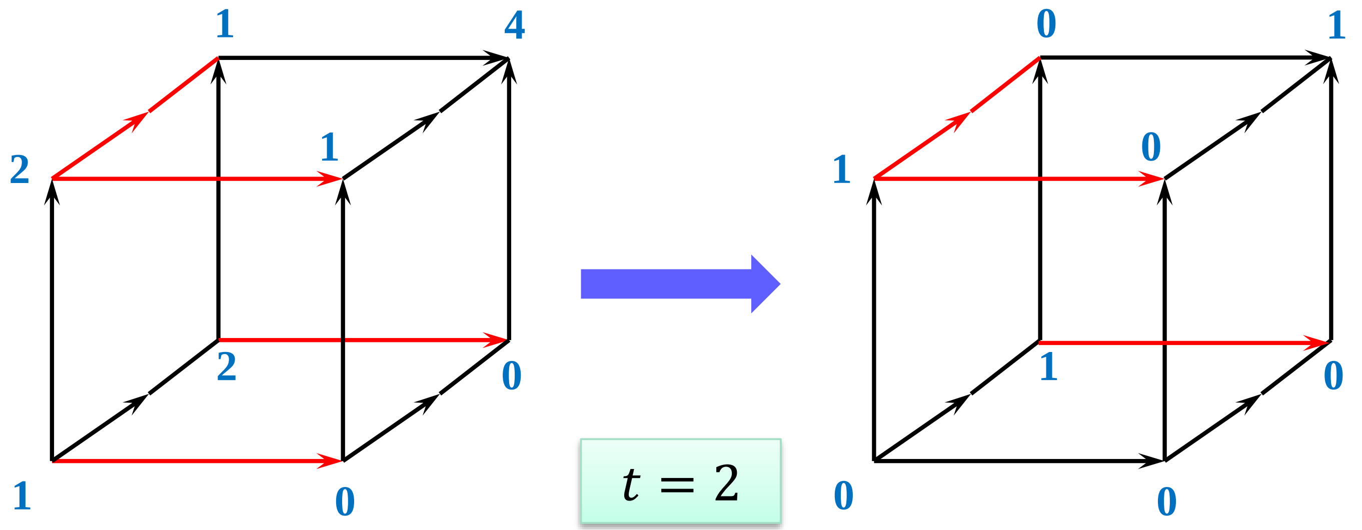


Decreasing edges w.r.t. h_t are also decreasing w.r.t. f

Boolean Decomposition: Thresholding Intuition

- To reduce from real-valued to Boolean functions, we can consider thresholding:

$$h_t(x) = \begin{cases} 1 & \text{if } f(x) \geq t \\ 0 & \text{otherwise} \end{cases}$$



Decreasing edges w.r.t. h_t are also decreasing w.r.t. f

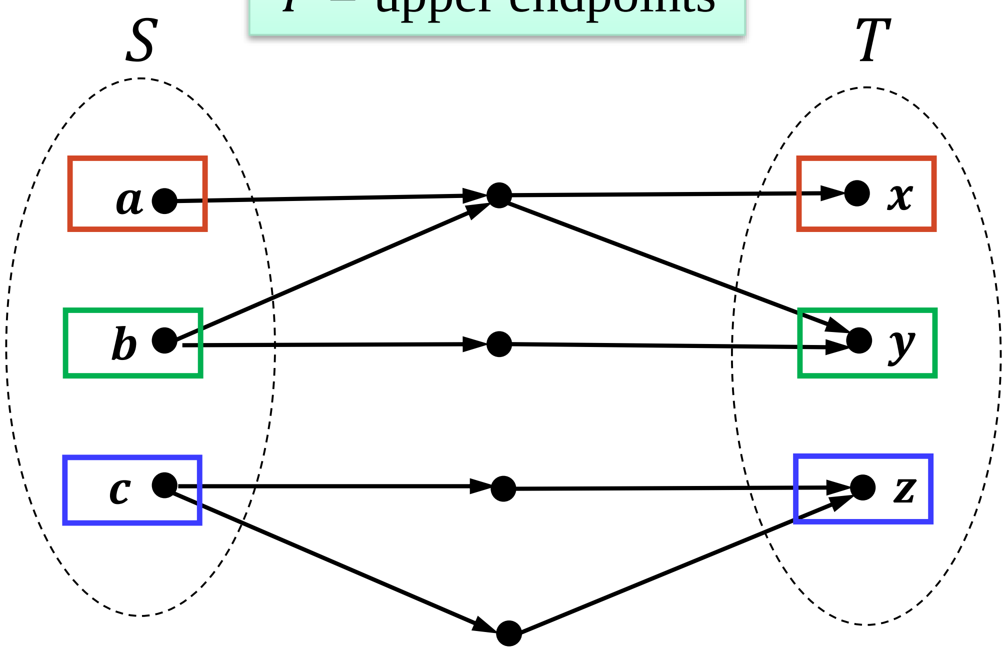
- The undirected version of the Main Inequality can be proved by considering the right t
- Considering all t gives a reduction to the Boolean case for L_1 distance [Berman Raskhodnikova Yaroslavtsev 16]
- But it fails for proving Boolean Decomposition: the same edge could be decreasing w.r.t. different h_t
- To prove Boolean Decomposition, we apply different thresholds in disjoint locations of hypercube.

Boolean Decomposition: Importance of Matchings

A vertex pair (x, y) is decreasing if $x \preccurlyeq y$ and $f(x) > f(y)$.

We consider matchings $M: S \to T$ of decreasing pairs.

S = lower endpoints,
T = upper endpoints



Fact [Fischer Lehman Newman Raskhodnikova Rubinfeld 02]

For every function f and maximal matching M of pairs decreasing w.r.t. f,

$$|M| \leq \text{Dist}(f, \text{MONO}) \leq 2|M|$$

Main Idea: Construct Boolean functions f_i such that every decreasing pair in M is decreasing w.r.t. exactly one f_i

Proof of Boolean Decomposition: Plan

Boolean Decomposition Theorem

Let G be a DAG, and $f: V(G) \to \mathbb{R}$ a nonmonotone function. There exist $k \geq 1$, Boolean functions $f_1, f_2, \dots, f_k: V(G) \to \{0,1\}$ and disjoint subgraphs $H_1, H_2, \dots, H_k$ of G such that:

(1) $\mathrm{DE}(f_i) \subseteq E(H_i) \cap \mathrm{DE}(f)$; (2) $\sum_{i\in[k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$

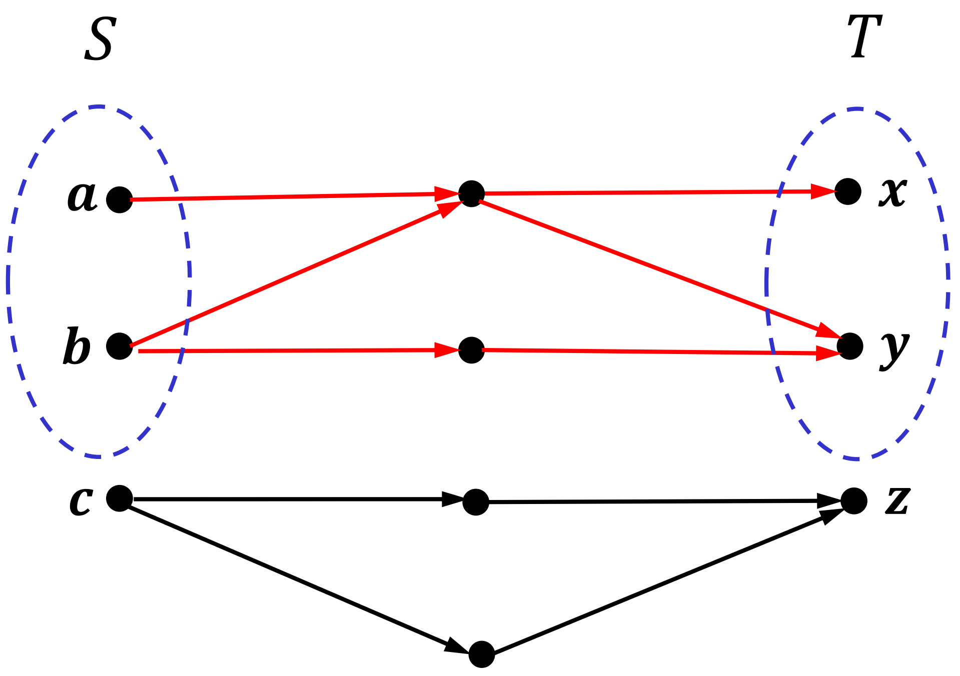
1. Explain how to obtain disjoint subgraphs H_i from a matching of vertices.
2. Specify a special matching M.
3. Define Boolean functions f_i based on subgraphs H_i obtained from M.

Step 1: Obtaining Disjoint Subgraphs H_i

Definition (**Sweeping Graphs**) For two disjoint sets $S, T \subseteq V(G)$,

subgraph $\mathrm{Sweep}(S,T)$ = union of all directed paths from vertices in S to vertices in T

Call (S,T) a **set-pair**



Useful properties:

- $\mathrm{Sweep}(S,T)$ is an induced subgraph
- A vertex outside $\mathrm{Sweep}(S,T)$ cannot be both "above" and "below" $\mathrm{Sweep}(S,T)$

It has a path **from** a vertex in $\mathrm{Sweep}(S,T)$

It has a path **to** a vertex in $\mathrm{Sweep}(S,T)$

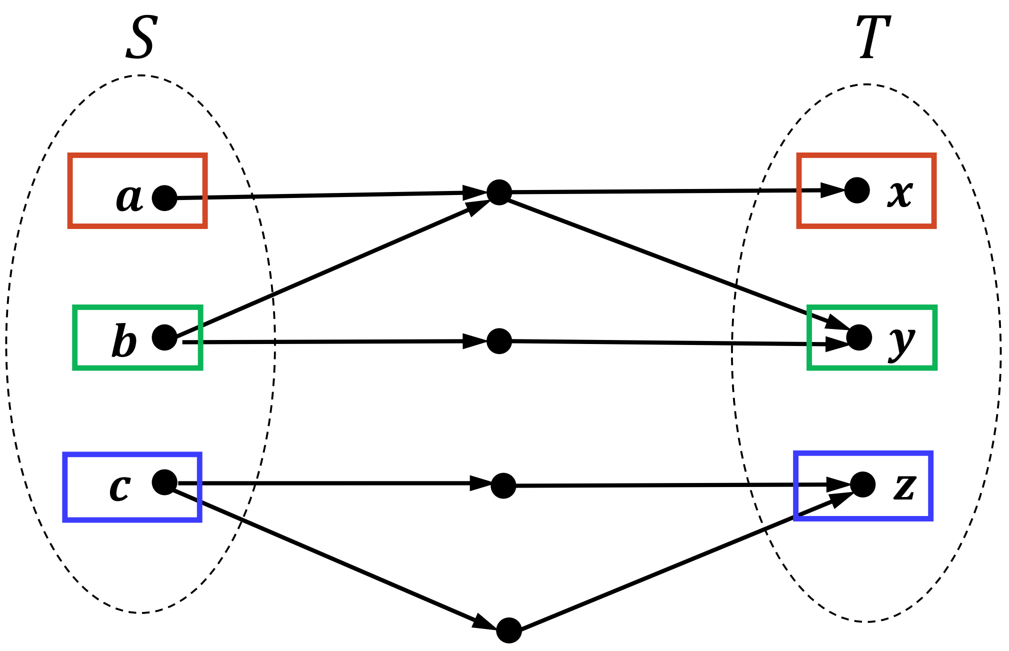
Step 1: Obtaining Disjoint Subgraphs H_i

- $\text{Sweep}(X, Y) =$ subgraph of paths from vertices in X to vertices in Y

- Two set-pairs of vertices (X, Y) and (X', Y') conflict if $\text{Sweep}(X, Y)$ intersects $\text{Sweep}(X', Y')$.

Merge-Conflicts (**Input:** matching $M: S \to T$)

1. Initialize collection of set-pairs to contain $(\{s\}, \{t\})$ for all $(s, t) \in M$
2. Repeat until there are no conflicts:
   If two set-pairs (X, Y) and (X', Y') conflict
   then merge them
   \\\\ *replace them with* $(X \cup X', Y \cup Y')$
3. **Return** collection of set-pairs



$M = \{(a, x), (b, y), (c, z)\}$

Collection $=$ $(\{a\}, \{x\}), (\{b\}, \{y\}), (\{c\}, \{z\})$

Step 1: Obtaining Disjoint Subgraphs H_i

- $\text{Sweep}(X, Y) =$ subgraph of paths from vertices in X to vertices in Y

- Two set-pairs of vertices (X, Y) and (X', Y') conflict if $\text{Sweep}(X, Y)$ intersects $\text{Sweep}(X', Y')$.

Merge-Conflicts (Input: matching $M: S \to T$**)**

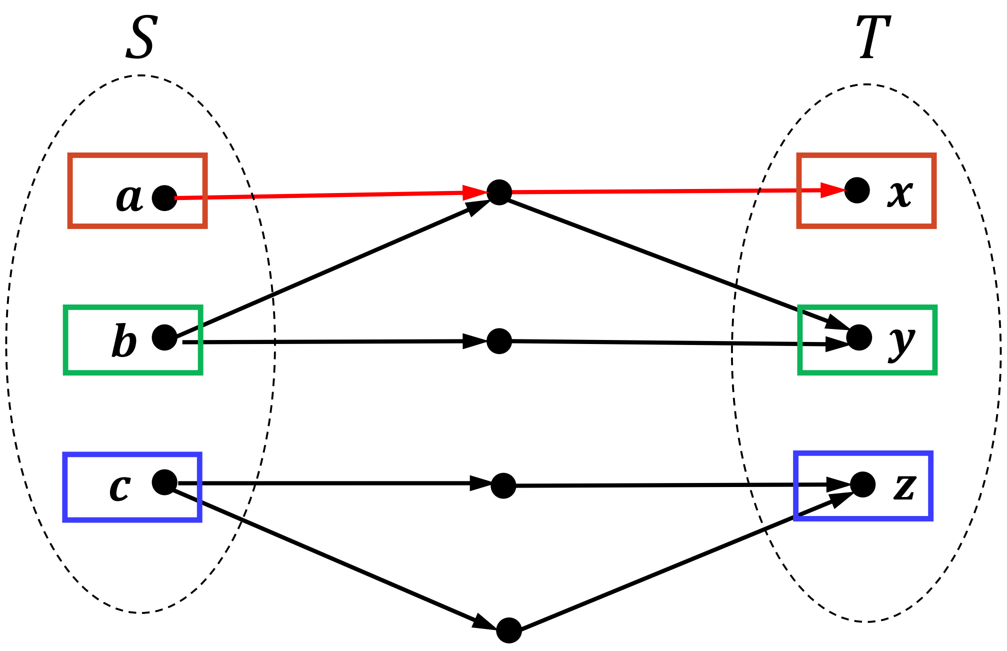
1. Initialize collection of set-pairs to contain $(\{s\}, \{t\})$ for all $(s, t) \in M$
2. Repeat until there are no conflicts:

   If two set-pairs (X, Y) and (X', Y') conflict

   then merge them

   \\\\ *replace them with* $(X \cup X', Y \cup Y')$
3. **Return** collection of set-pairs

Sweep$(\{a\}, \{x\})$
union of paths from $\{a\}$ *to* $\{x\}$



$M = \{(a, x), (b, y), (c, z)\}$

Collection $= (\{a\}, \{x\}), (\{b\}, \{y\}), (\{c\}, \{z\})$

Step 1: Obtaining Disjoint Subgraphs H_i

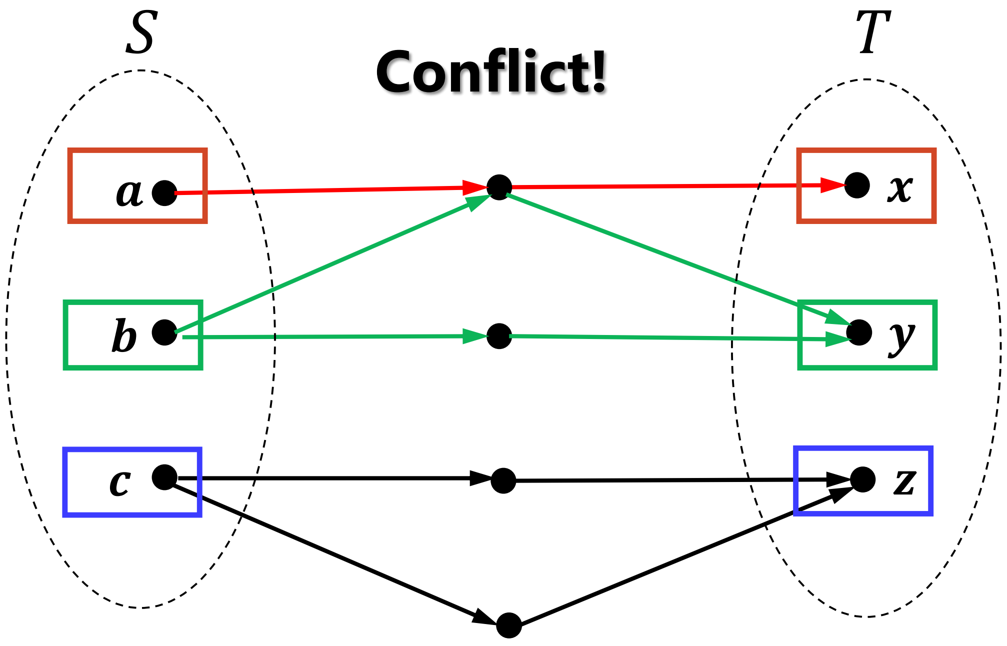
- $\text{Sweep}(X, Y) =$ subgraph of paths from vertices in X to vertices in Y

- Two set-pairs of vertices (X, Y) and (X', Y') conflict if $\text{Sweep}(X, Y)$ intersects $\text{Sweep}(X', Y')$.

Merge-Conflicts (Input: matching $M: S \to T$)

1. Initialize collection of set-pairs to contain $(\{s\}, \{t\})$ for all $(s, t) \in M$
2. Repeat until there are no conflicts:
   If two set-pairs (X, Y) and (X', Y') conflict
   then merge them
   \\ *replace them with* $(X \cup X', Y \cup Y')$
3. **Return** collection of set-pairs

$\text{Sweep}(\{b\}, \{y\})$
union of paths from $\{b\}$ *to* $\{y\}$



$M = \{(a, x), (b, y), (c, z)\}$

Collection $=$ $(\{a\}, \{x\}), (\{b\}, \{y\}), (\{c\}, \{z\})$

Step 1: Obtaining Disjoint Subgraphs H_i

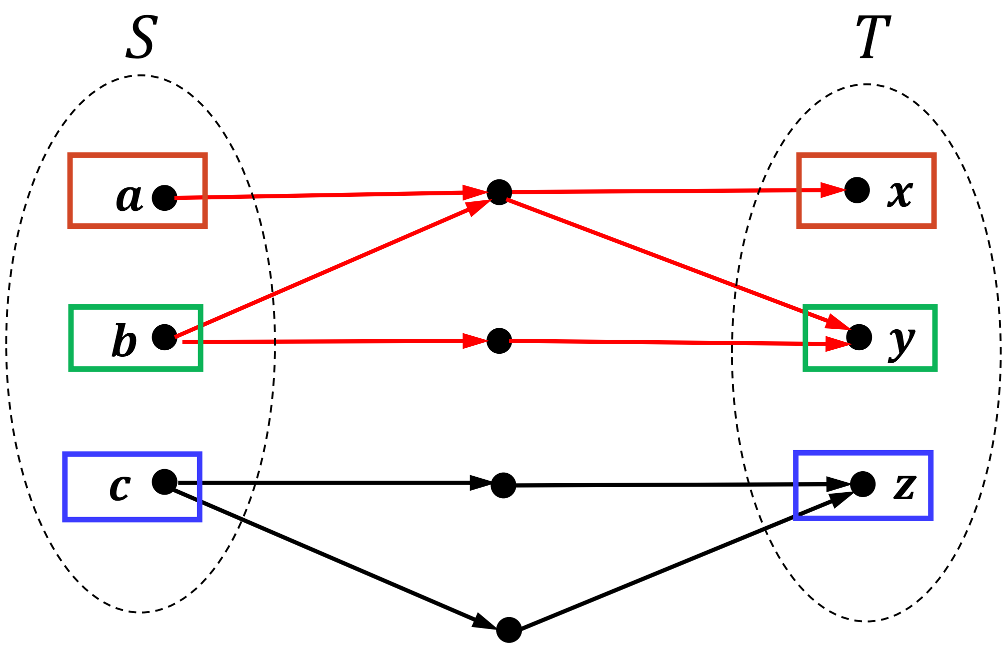
- $\text{Sweep}(X, Y) =$ subgraph of paths from vertices in X to vertices in Y

- Two set-pairs of vertices (X, Y) and (X', Y') conflict if $\text{Sweep}(X, Y)$ intersects $\text{Sweep}(X', Y')$.

Merge-Conflicts (Input: matching $M: S \to T$)

1. Initialize collection of set-pairs to contain $(\{s\}, \{t\})$ for all $(s, t) \in M$
2. Repeat until there are no conflicts:
   If two set-pairs (X, Y) and (X', Y') conflict
   then merge them
   \\ *replace them with* $(X \cup X', Y \cup Y')$
3. **Return** collection of set-pairs

$\text{Sweep}(\{a, b\}, \{x, y\})$
union of paths from $\{a, b\}$ *to* $\{x, y\}$



$M = \{(a, x), (b, y), (c, z)\}$

Collection $= (\{a, b\}, \{x, y\}), (\{c\}, \{z\})$

Step 1: Obtaining Disjoint Subgraphs H_i

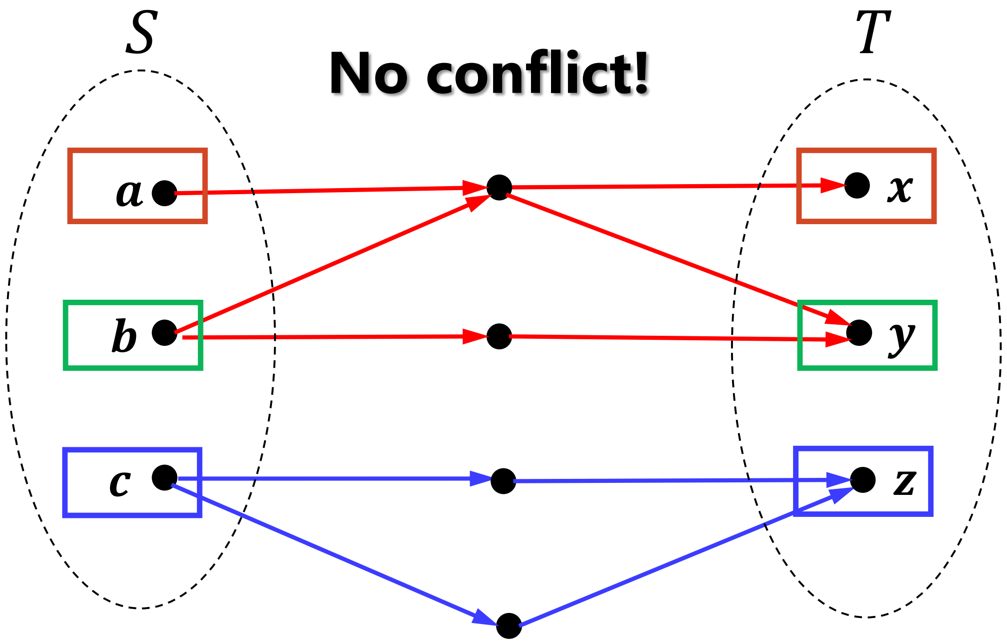
- $\text{Sweep}(X, Y) =$ subgraph of paths from vertices in X to vertices in Y

- Two set-pairs of vertices (X, Y) and (X', Y') conflict if $\text{Sweep}(X, Y)$ intersects $\text{Sweep}(X', Y')$.

Merge-Conflicts (**Input:** matching $M: S \to T$)

1. Initialize collection of set-pairs to contain $(\{s\}, \{t\})$ for all $(s, t) \in M$
2. Repeat until there are no conflicts:
   If two set-pairs (X, Y) and (X', Y') conflict
   then merge them
   \\ *replace them with* $(X \cup X', Y \cup Y')$
3. **Return** collection of set-pairs

$\text{Sweep}(\{c\}, \{z\})$
union of paths from $\{c\}$ *to* $\{z\}$



$M = \{(a, x), (b, y), (c, z)\}$

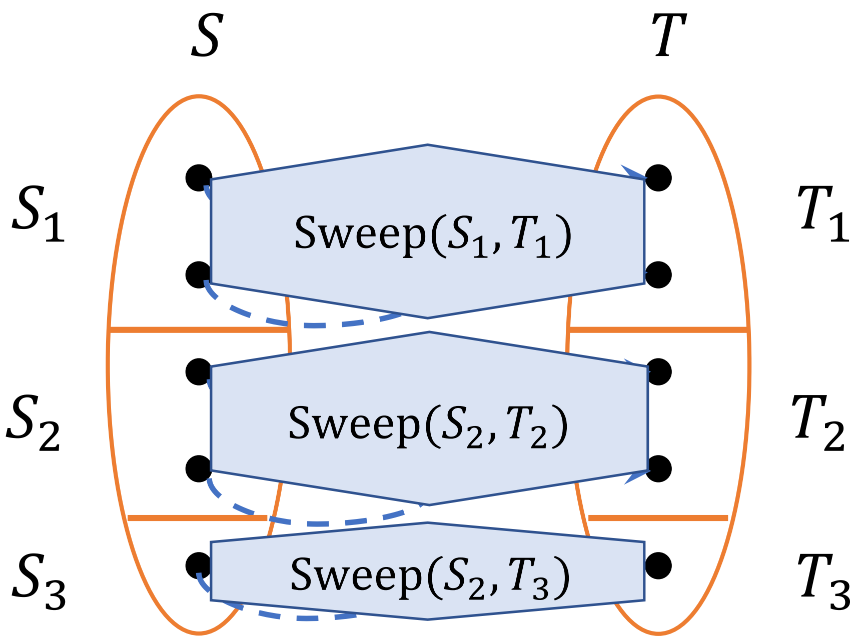
Collection $= (\{a, b\}, \{x, y\}), (\{c\}, \{z\})$

Final collection

Step 1: Obtaining Disjoint Subgraphs H_i

Algorithm **Merge-Conflicts**, given a matching M: $S \to T$, returns set-pairs $(S_1, T_1), \dots, (S_k, T_k)$ such that:

- The sets S_i partition S, the sets T_i partition T.
- The subgraphs $\text{Sweep}(S_i, T_i)$ are vertex-disjoint.
- **(Rematching property)** For all $s \in S_i$, $t \in T_i$ such that $s \preccurlyeq t$:
  there exists another matching M': $S \to T$ that matches (s, t).

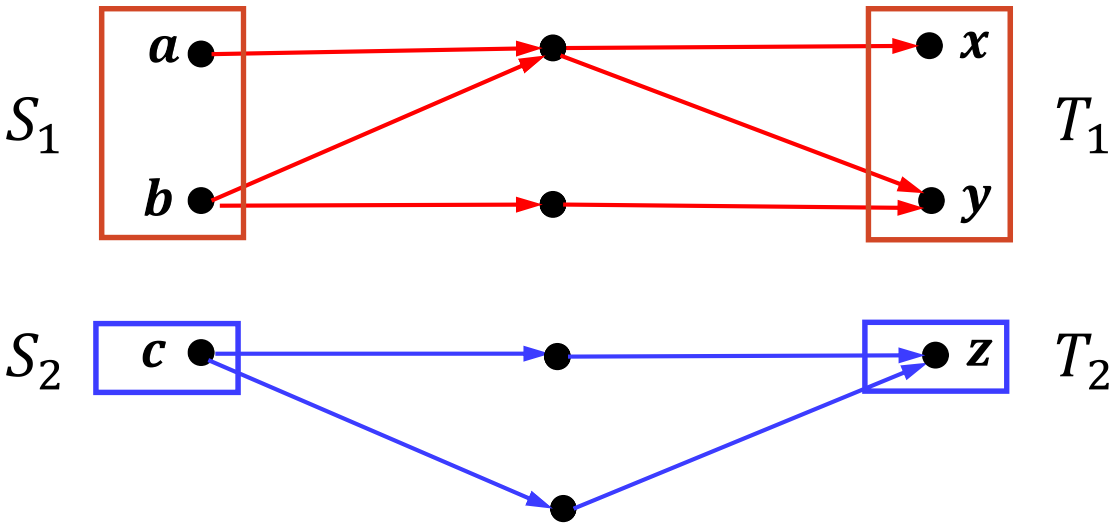


Step 1: Obtaining Disjoint Subgraphs H_i

Algorithm **Merge-Conflicts**, given a matching $M\colon\ S \to\ T$, returns set-pairs $(S_1, T_1), \dots, (S_k, T_k)$ such that:

- The sets S_i partition S, the sets T_i partition T.
- The subgraphs $\mathrm{Sweep}(S_i, T_i)$ are vertex-disjoint.
- **(Rematching property)** For all $s \in S_i,\ t \in T_i$ such that $s \preccurlyeq t$:

  there exists another matching $M'\colon S \to T$ that matches (s, t).



$M = \{(a,x),(b,y),(c,z)\}$ ➡ $M' = \{(a,y),(b,x),(c,z)\}$

Proof of Boolean Decomposition: Plan

Boolean Decomposition Theorem

Let G be a DAG, and $f: V(G) \to \mathbb{R}$ a nonmonotone function. There exist $k \geq 1$, Boolean functions $f_1, f_2, \dots, f_k: V(G) \to \{0,1\}$ and disjoint subgraphs $H_1, H_2, \dots, H_k$ of G such that:

(1) $\mathrm{DE}(f_i) \subseteq E(H_i) \cap \mathrm{DE}(f)$; (2) $\sum_{i \in [k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$

✓ Explain how to obtain disjoint subgraphs H_i from a matching of vertices.

2. Specify a special matching M.

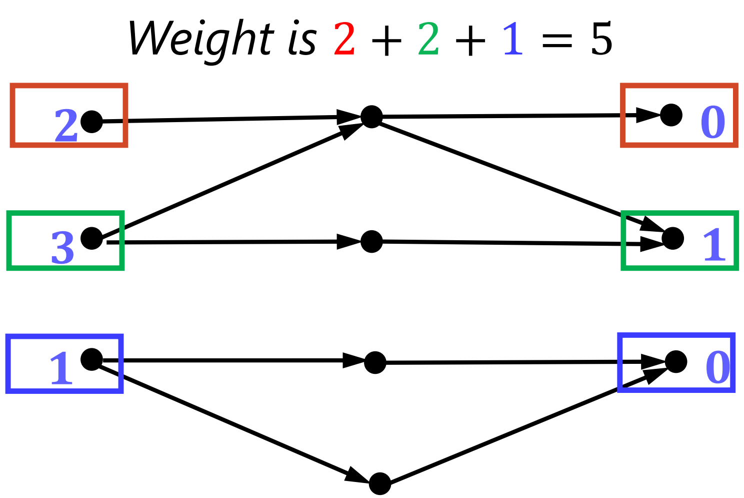
3. Define Boolean functions f_i based on subgraphs H_i obtained from M.

Step 2: Special Matching

Max-weight, min-cardinality matching M of pairs $x \preccurlyeq y$

- maximizes **weight(M)** $= \sum_{(x,y)\in M}(f(x) - f(y))$,
- and amongst such matchings has the fewest pairs.

M is a maximal matching of decreasing pairs



Run algorithm **Merge-Conflicts** with special matching M.

Sweep(S_i, T_i) are the subgraphs H_i

Violation Lemma

The set-pairs $(S_1, T_1), \ldots, (S_k, T_k)$ obtained from special matching satisfy:

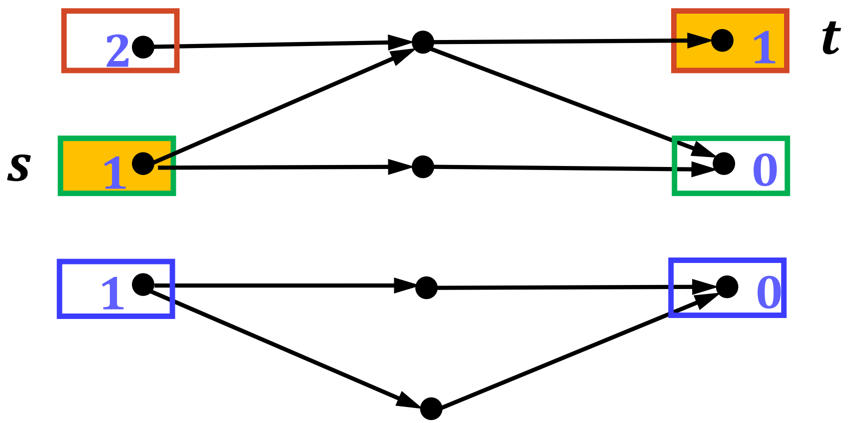
- If $s \preccurlyeq t$ and $s \in S_i,\ t \in T_i$ then $f(s) > f(t)$.

With careful thresholding, we will preserve violations of monotonicity

Step 2: Special Matching

Recall: The set-pairs $(S_1, T_1), \dots, (S_k, T_k)$ returned by **Merge-Conflicts** satisfy:

- **(Rematching property)** For all $s \in S_i,\ t \in T_i$ such that $s \preccurlyeq t$: there exists another matching $M': S \to T$ that matches (s, t).



Violation Lemma

The set-pairs $(S_1, T_1), \dots, (S_k, T_k)$ obtained from special matching satisfy:

- If $s \preccurlyeq t$ and $s \in S_i,\ t \in T_i$ then $f(s) > f(t)$.

Proof (by contradiction):

- Suppose that $f(s) \le f(t)$ for some $s \in S_i, t \in T_i$ with $s \preccurlyeq t$.
- Use **Rematching property** to get a matching $M':\ S \to T$ that matches (s, t).
- $\boldsymbol{weight}(M') = \boldsymbol{weight}(M)$, since the endpoints have not changed.
- $\boldsymbol{weight}(M' \setminus \{(s,t)\}) \ge \boldsymbol{weight}(M)$, because $f(s) - f(t) \ \le 0$.
- But $M' \setminus \{(s,t)\}$ has fewer pairs than M. Contradiction.

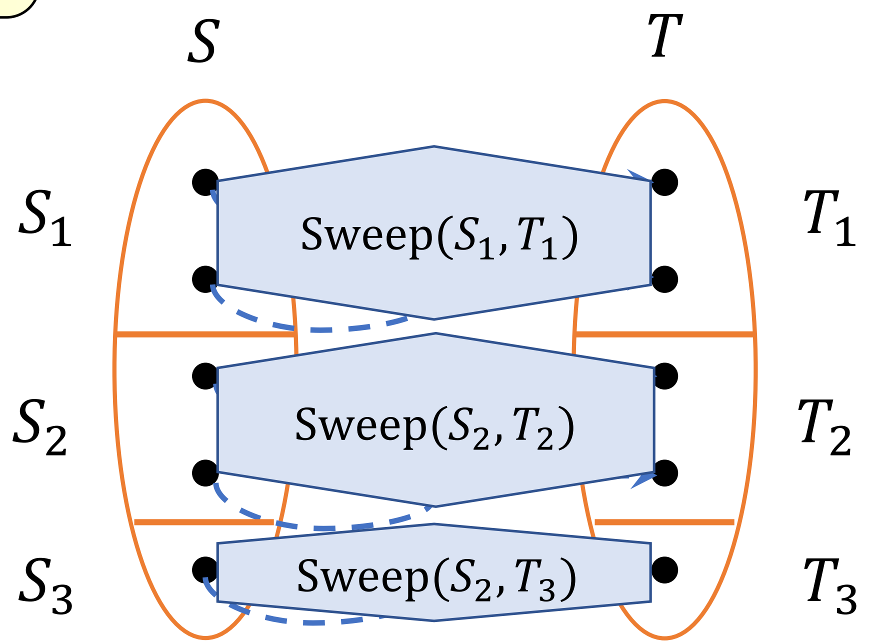
$$\boldsymbol{weight}(\boldsymbol{M}) = \sum_{(x,y) \in M} (f(x) - f(y))$$

Summary of Steps 1-2

- Start with special matching $M: S \to T$ (max weight, min-cardinality).
- M is a maximal matching of decreasing pairs: $|M| \leq \mathrm{Dist}(f, \mathrm{MONO}) \leq 2|M|$.
- Run algorithm **Merge-Conflicts** to obtain set-pairs $(S_1, T_1), (S_2, T_2), \ldots, (S_k, T_k)$.
- The subgraphs $\mathrm{Sweep}(S_i, T_i)$ are vertex-disjoint.

Violation Lemma

If $s \preccurlyeq t$ and $s \in S_i,\ t \in T_i$ then $f(s) > f(t)$.



Proof of Boolean Decomposition: Plan

Boolean Decomposition Theorem

Let G be a DAG, and $f: V(G) \to \mathbb{R}$ a nonmonotone function. There exist $k \geq 1$, Boolean functions $f_1, f_2, \dots, f_k: V(G) \to \{0,1\}$ and disjoint subgraphs $H_1, H_2, \dots, H_k$ of G such that:

(1) $\mathrm{DE}(f_i) \subseteq E(H_i) \cap \mathrm{DE}(f)$; (2) $\sum_{i \in [k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$

✓ Explain how to obtain disjoint subgraphs H_i from a matching of vertices.

✓ Specify a special matching M.

3. Define Boolean functions f_i based on subgraphs H_i obtained from M.

Step 3: Define Boolean Functions

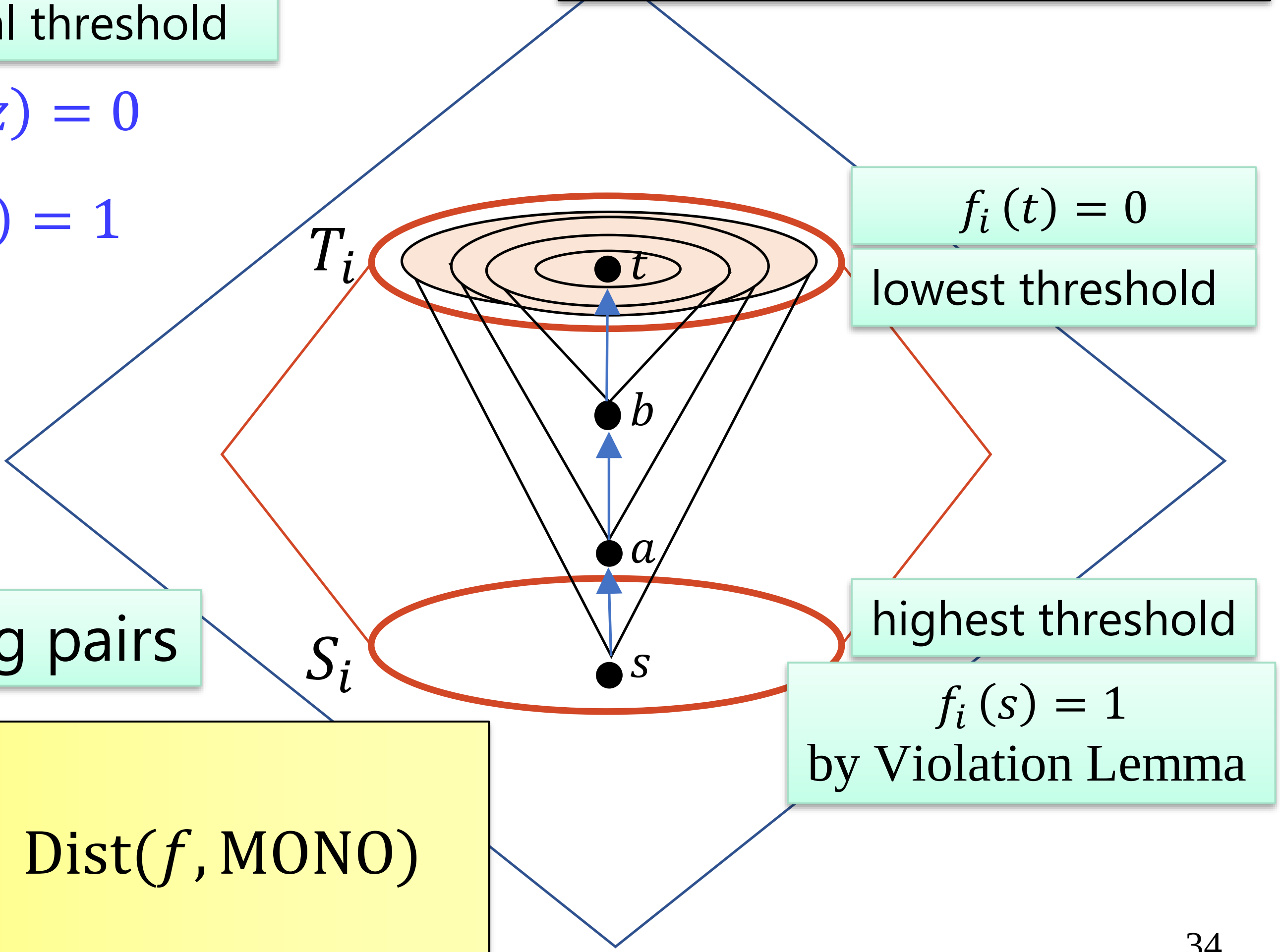
For all set-pairs (S_i, T_i), define $f_i \colon V(G) \to \{0,1\}$

max f-value achieved by points in T_i above z

individual threshold

z
- in Sweep(S_i, T_i)
  - $f(z) \leq \max_{x \in T_i \colon z \preccurlyeq x} f(x)$, then $f_i(z) = 0$
  - else $f_i(z) = 1$
- not in Sweep(S_i, T_i)

An edge inside Sweep(S_i, T_i) is decreasing w.r.t. f_i only if it was decreasing w.r.t. f



f_i has matching $M_i \colon S_i \to T_i$ of decreasing pairs

$$\sum_{i \in [k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \sum_{i \in [k]} |M_i| = |M| \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$$

Step 3: Define Boolean Functions

For all set-pairs (S_i, T_i), define $f_i\colon V(G) \to \{0,1\}$

z

in Sweep(S_i, T_i)

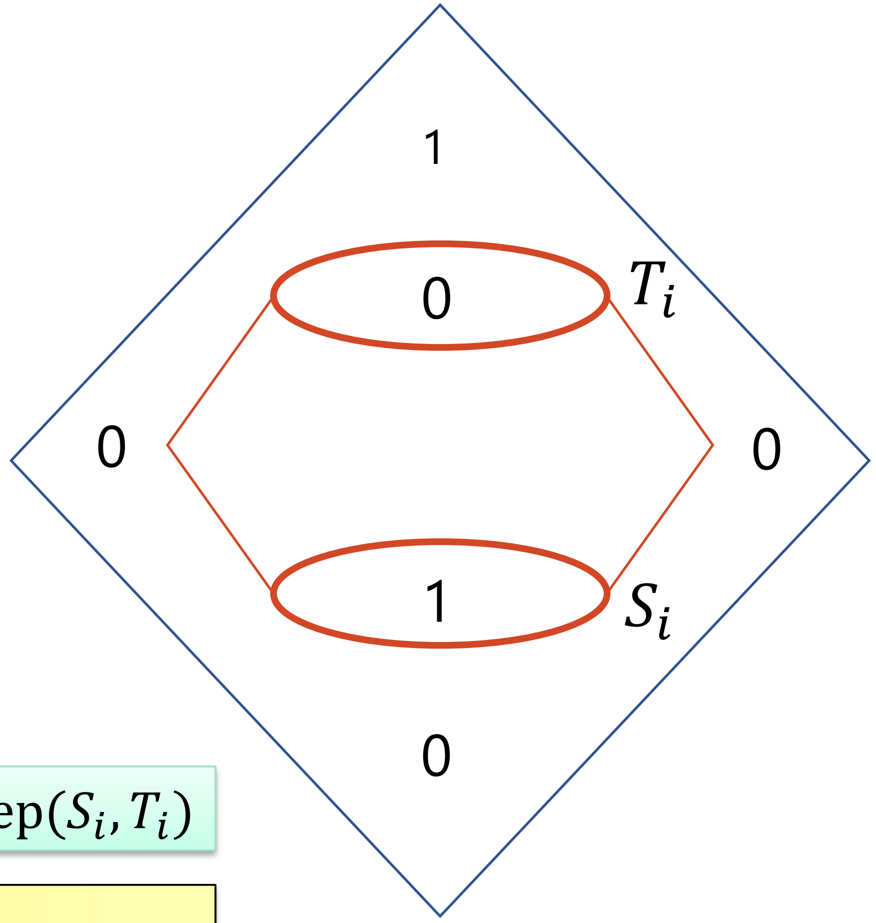
- $f(z) \le \max_{x \in T_i, z \preccurlyeq x} f(x)$, then $f_i(z) = 0$
- else $f_i(z) = 1$

not in Sweep(S_i, T_i)

- above, then $f_i(z) = 1$
- else $f_i(z) = 0$

A vertex cannot be both above and below Sweep(S_i, T_i)

All decreasing edges are inside Sweep(S_i, T_i)



Proof of Boolean Decomposition: Plan

Boolean Decomposition Theorem

Let G be a DAG, and $f: V(G) \to \mathbb{R}$ a nonmonotone function. There exist $k \geq 1$, Boolean functions $f_1, f_2, \ldots, f_k$: $V(G) \to \{0,1\}$ and disjoint subgraphs $H_1, H_2, \ldots, H_k$ of G such that:

(1) $\mathrm{DE}(f_i) \subseteq E(H_i) \cap \mathrm{DE}(f)$; (2) $\sum_{i \in [k]} \mathrm{Dist}(f_i, \mathrm{MONO}) \geq \frac{1}{2} \mathrm{Dist}(f, \mathrm{MONO})$

- ✓ Explain how to obtain disjoint subgraphs H_i from a matching of vertices.
- ✓ Specify a special matching M.
- ✓ Define Boolean functions f_i based on subgraphs H_i obtained from M.

$\Rightarrow$

Main Isoperimetric Inequality

For all functions f: $\{0,1\}^d \to \mathbb{R}$,

$$\sum_{x \in \{0,1\}^d} \left[\sqrt{I_f^-(x)} \right] = \Omega\big(\mathrm{Dist}(f, \mathrm{MONO})\big)$$

Summary

- Improved sublinear algorithms for monotonicity
  - Proved tight bounds for nonadaptive algorithms
- Generalized isoperimetric inequalities
- Proved Boolean Decomposition Theorem

Open Questions

- Role of adaptivity?
  - for property testing and distance approximation
- Does Talagrand inequality generalize to other domains?
  - Specifically, the hypergrid domain $[n]^d$?
  - Weaker inequalities (Margulis) generalize [Black Chakrabarty Seshadhri 18]
  - It would suffice to show such inequality for the Boolean case and then use our Boolean Decomposition Theorem to generalize to real-valued functions.
  - Would improve algorithms for monotonicity testing on hypergrid.