

Optimal algorithms for testing monotonicity and Lipschitz over hypergrids

C. Seshadhri

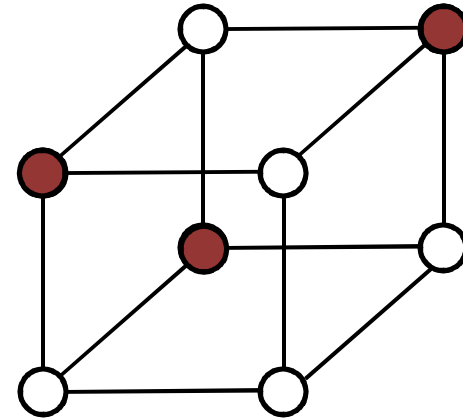
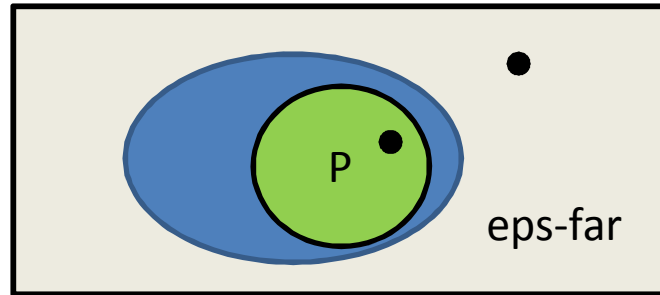
(Sandia National Labs, Livermore)

Joint work with Deeparnab Chakrabarty
(Microsoft Research, Bangalore)

Property Testing in a Slide

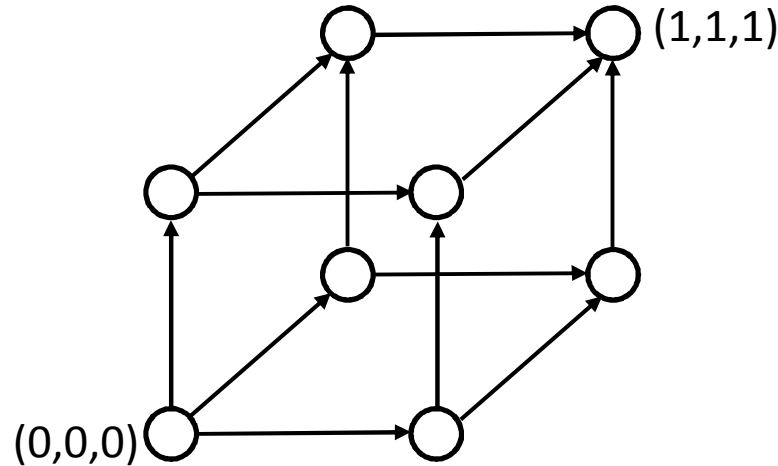
- $f: D \rightarrow R$
 - Amen
- A “property” is some subset of these functions
 - Linear functions, monotone functions, bounded gradient, etc.
- Hamming distance $d(f,g)$ between two functions:
 $(\# x \text{ s.t. } f(x) \neq g(x)) / |D|$
 - $d(f,P) = \min_g d(f,g)$. The minimum fraction of values you need to change so f “has P ”.

Property Testing in (another) Slide



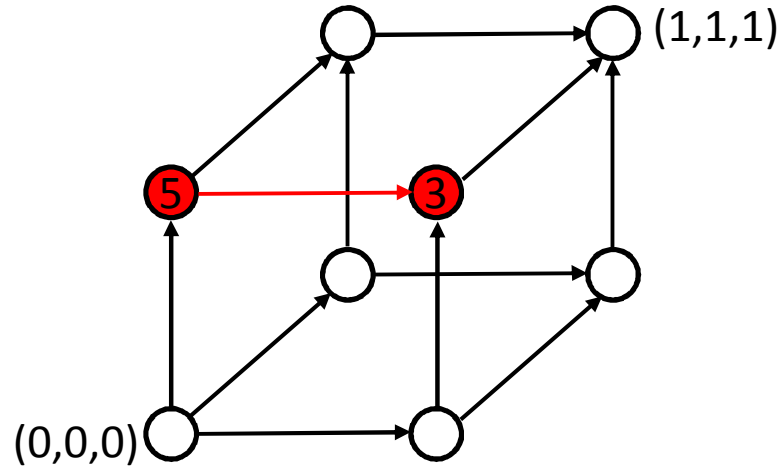
- Suppose you have query access to f
- Standard decision problem: is f in P or $d(f,P) \neq 0$?
- Relaxed version with $\epsilon \in (0,1)$: is f in P or $d(f,P) > \epsilon$
 - Can hopefully decide with much fewer queries than $2n$ (using randomization)
- Property tester: if input f has P , accept with prob $> 2/3$.
If input f is ϵ -far from P , reject with prob $> 2/3$.
 - No. of queries is $t(n,\epsilon) \ll 2n$

Monotonicity



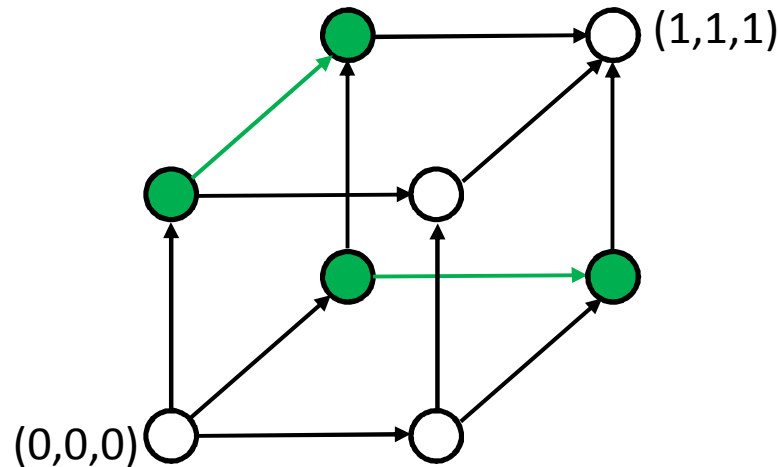
- Take product ordering on $\{0,1\}^n$: $x \leq y$ if $\forall i x_i \leq y_i$
 - Standard partial order given by containment in $\{0,1\}^n$
- f is monotone if $\forall x < y, f(x) \leq f(y)$
 - Come up in many places:
- [Goldreich et al 00] “Testing monotonicity”. Introduced this problem and the edge tester
- Let B be the hypercube graph. Edge (u,v) is violation if $f(u) > f(v)$.

Monotonicity



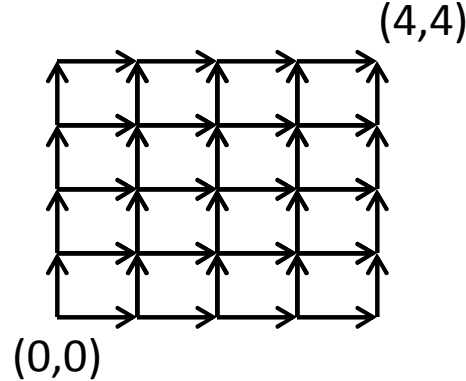
- Take product ordering on $\{0,1\}^n$: $x \leq y$ if $\forall i x_i \leq y_i$
 - Standard partial order given by containment in $\{0,1\}^n$
- f is monotone if $\forall x < y, f(x) \leq f(y)$
- [Goldreich et al 00] “Testing monotonicity”. Introduced this problem and the edge tester
- Let B be the hypercube graph. Edge (u,v) is violation if $f(u) > f(v)$.

The edge tester



- [Goldreich et al 00] Edge tester: sample t uniform random edges. If any violation detected, reject. Otherwise, accept.
- [GGLRS00, DGLRRS99]: For $f: \{0,1\}^n \rightarrow \mathbb{R}$. If f is ϵ -far, edge tester with $O(n(\log R)/\epsilon)$ queries rejects with high prob.
 - Worst case $O(n^2/\epsilon)$
 - Do you need dependence on R ?
 - [Blais Brody Matulef 11] $\Omega(n/\epsilon)$ queries required.
 - [Bertinoro 11] Considered important open problem in property testing
- **Theorem: Edge tester is $O(n/\epsilon)$ -query monotonicity tester.**

Monotonicity for hypergrids

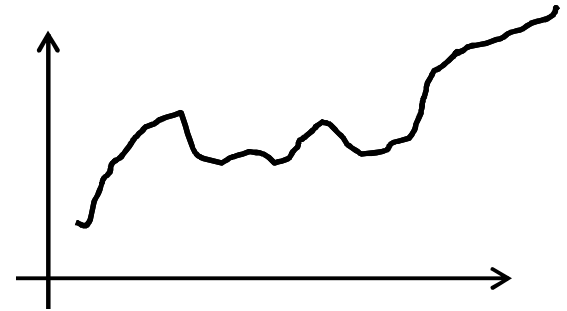


- $f: [k]^n \rightarrow \mathbb{R}$. $x \leq y$ if $\forall i, x_i \leq y_i$
 - $k=2$, hypercube. $n=1$, total order. And everything in between
- [DGLRRS99, EKK+99, AC04, HK04] $O(\epsilon^{-1} k \log n \log R)$ and $O(\epsilon^{-1} 2^n \log k)$ testers
- **Theorem: There is a $O(\epsilon^{-1} n \log k)$ -query tester for monotonicity on hypergrids.**
 - Somewhat analogous to edge tester, but also samples distant pairs
 - [Blais-Raskhodnikova 13, Chakrabarty S 13] Matching lower bounds found

The Lipschitz property

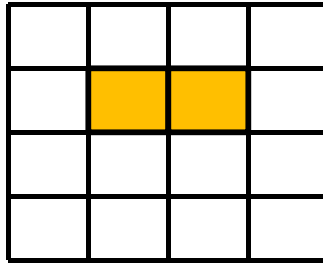
4	5	4	4
5	4	3	3
4	3	2	2
3	2	2	1

4	5	4	4
5	4	4	3
4	3	2	2
3	2	2	1



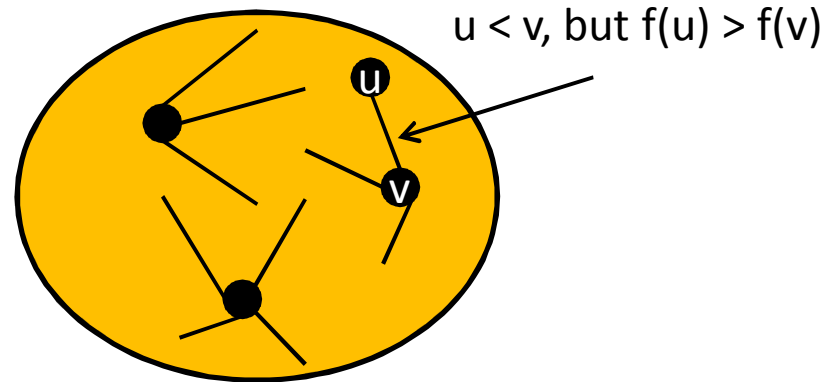
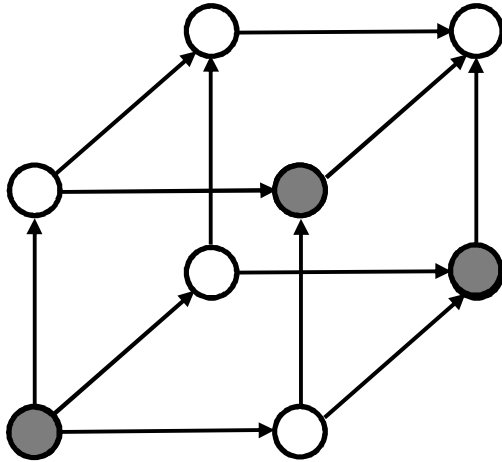
- f is c -Lipschitz if for all nbers (x,y) , $|f(x) - f(y)| \leq c$
- By changing any coordinate by 1, $f(x_1, x_2, \dots, x_k)$ changes by at most c
- For all (x,y) $|f(x) - f(y)| \leq c|x - y|_1$
- Natural property: comes up in statistics, program analysis, differential privacy
- [JR11] “Testing c -Lipschitz”. Applications to differential privacy
- [JR11, AJMR12] $R = \delta Z$. There is $O((\delta \epsilon)^{-1} n^{2k} \log k)$ tester.
- **Theorem: There is a $O(\epsilon^{-1} n \log k)$ -query tester for c -Lipschitz on hypergrids.**

The bounded derivative problem



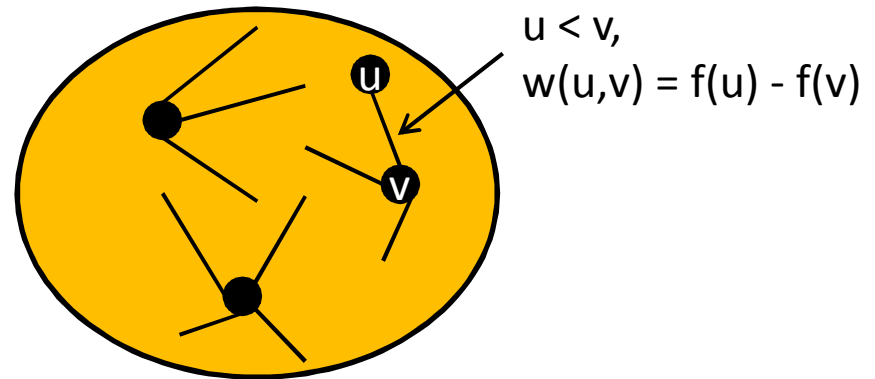
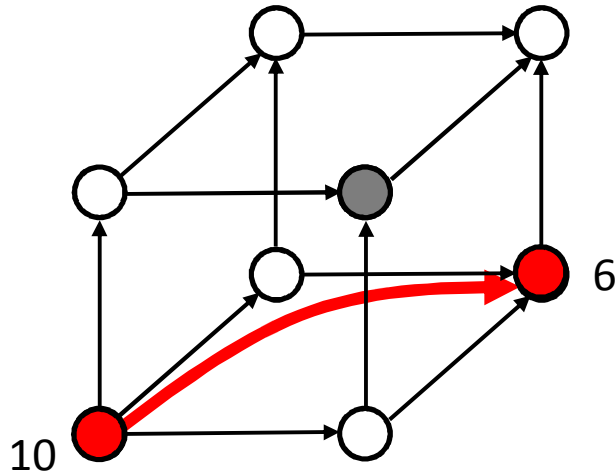
- Fix α, β . Consider the (α, β) -Lipschitz property
- $f: [k] \rightarrow \mathbb{R}$. For all nbrs $x < y$, $\alpha \leq f(x) - f(y) \leq \beta$
 - Different upper and lower bounds for derivative
 - If $\alpha = 0$, $\beta = \infty$, monotonicity
 - If $\alpha = \beta = c$, c -Lipschitz
- **Theorem: There is a $O(\epsilon^{-1} n \log k)$ -query tester for (α, β) -Lipschitz on hypergrids.**
 - No dependence on α, β
 - Same tester. Simply sample from predefined distribution on pairs, and check (α, β) -Lipschitz condition
- Single unified proof matching/improving previous results: [EKK99], [GGRLS00], [DDGLRS99], [AC04], [JR11], [AJMR12]

The violation graph



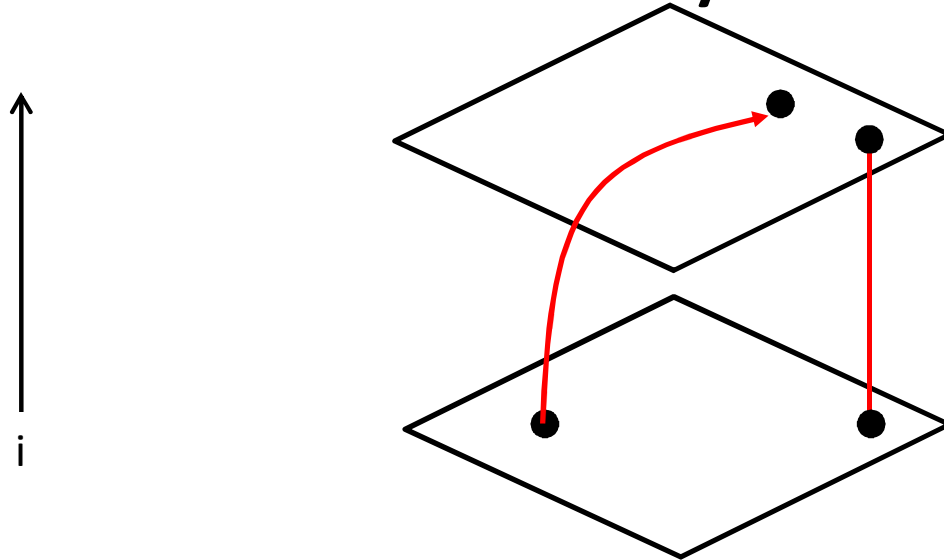
- Undirected edge (u,v) in VG if (u,v) is violation
- Consider any vertex cover S of VG.
 - No edges if we delete S = No violations if $f(x)$ undefined for all x in S
 - Simple induction shows we can fill in values at S to get monotone f'
 - $\text{eps}_f 2^n = \min \text{ vertex cover}$
- [FLNRRS02] Let M be maximal violation matching. $|M| > \epsilon_f 2^{n-1}$

From matchings to edges



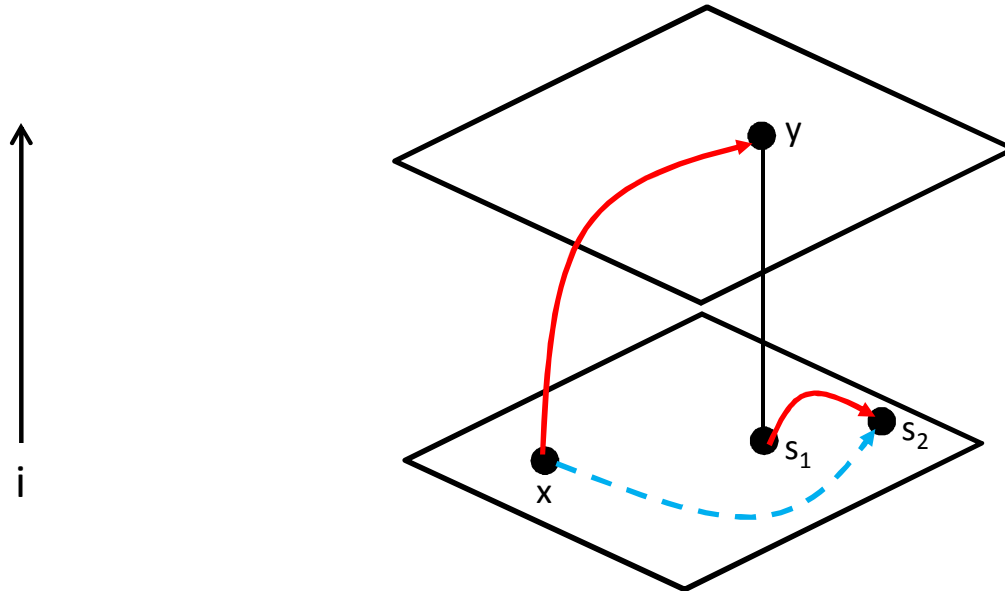
- Let VG be weighted with “magnitude of violation”
- Key idea: let M be matching of maximum weight. (M must be maximal, so $|M| > \epsilon_f 2^{n-1}$.)
- Theorem: # violated edges $> |M|$
 - Immediately implies success of edge tester. Total # edges = $n2^{n-1}$. Probability uniform random edge is violation is n/ϵ_f

Dimension by dimension



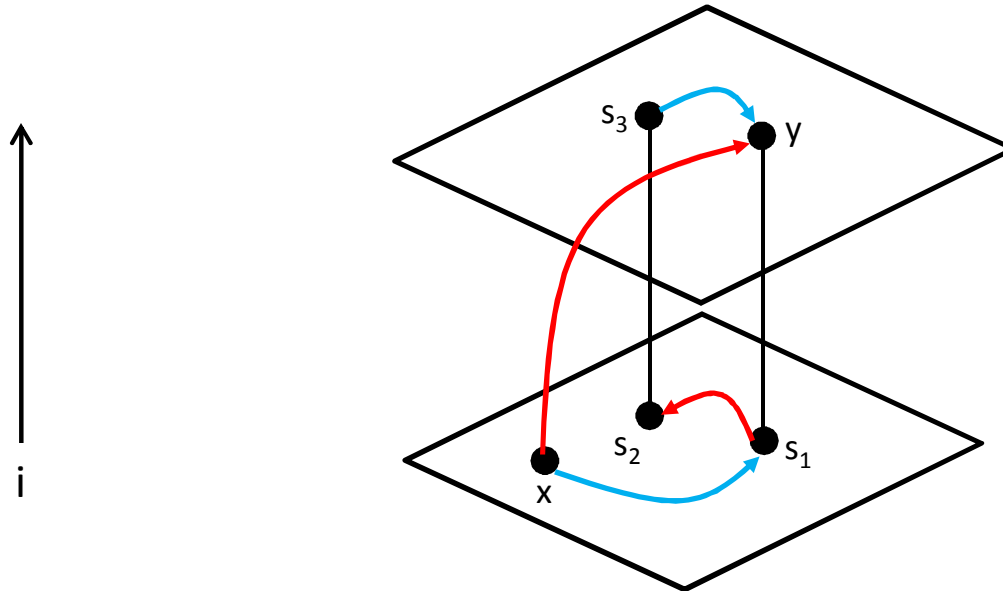
- Let M_i be pairs in M crossing dimension i .
 - Not a partition, but $\cup_i M_i = M$
- **Lemma: #violated dimension i edges $\geq |M_i|$**
 - Main theorem follows immediately

Trying to find a violated edge



- Start with (x, y) in M_i . So $f(x) > f(y)$
- Project down along dimension i to get edge (s_1, y) .
 - If violation, done. Otherwise $f(s_1) < f(y)$.
- $f(x) - f(s_1) > f(x) - f(y)$
 - If s_1 was M -unmatched, then we contradict maximum weight property of M .
- Suppose $s_1 < s_2$, so $f(s_1) > f(s_2)$. So $x < s_2$.
- $f(x) - f(s_2) = [f(x) - f(s_1)] - [f(s_1) - f(s_2)]$
 $> [f(x) - f(y)] - [f(s_1) - f(s_2)]$
- Max weight of M contradicted

A few more steps...



$$f(x) > f(y)$$

$$s_1 > s_2$$

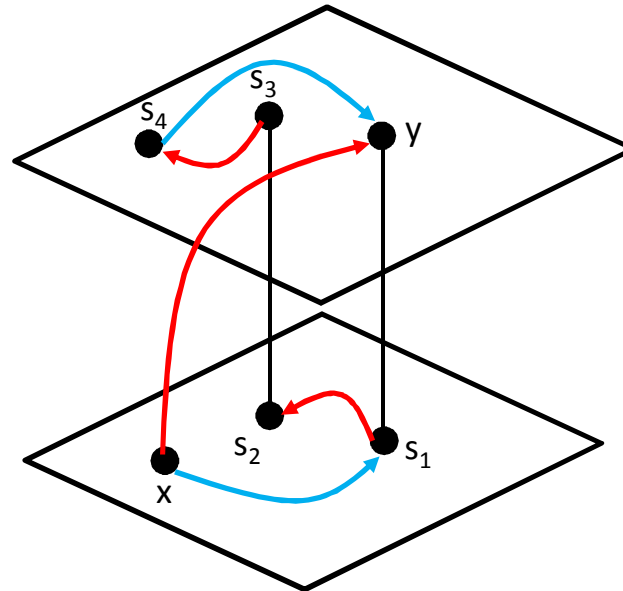
$$f(s_2) > f(s_1)$$

Project up s_2 to get s_3 .

So $s_3 < y$

- If (s_2, s_3) violated, done. So $f(s_2) < f(s_3)$
- Suppose s_3 is M -unmatched.
- $[f(x) - f(s_1)] + [f(s_3) - f(y)] = [f(x) - f(y)] + [f(s_3) - f(s_1)]$
 $> [f(x) - f(y)] + [f(s_2) - f(s_1)]$ (Contradiction!)

A few more steps...



$$f(x) > f(y)$$

$$f(y) > f(s_1)$$

$$s_1 > s_2$$

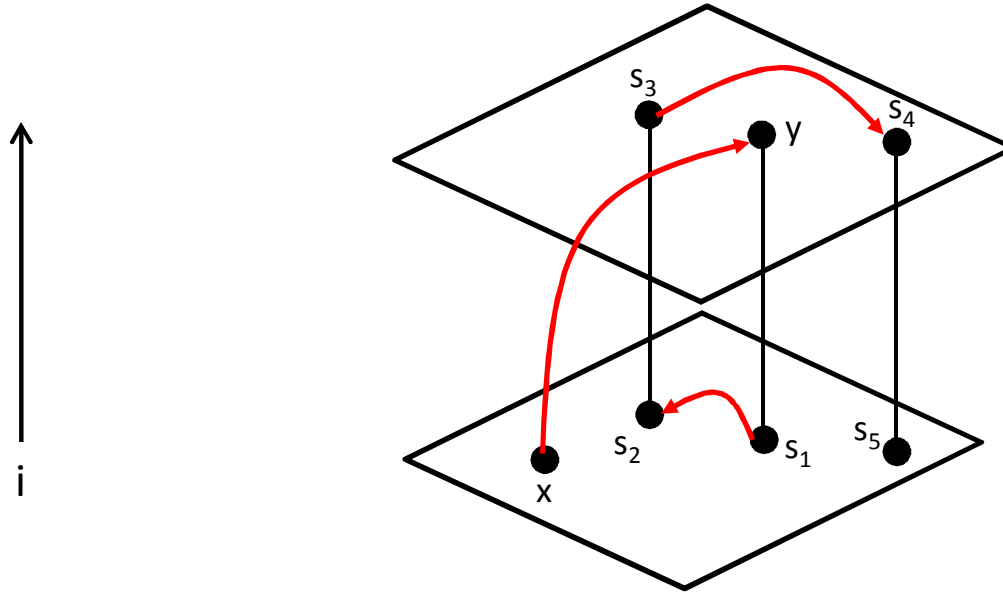
$$f(s_2) > f(s_1)$$

Project up s_2 to get s_3 .

So $s_3 < y$

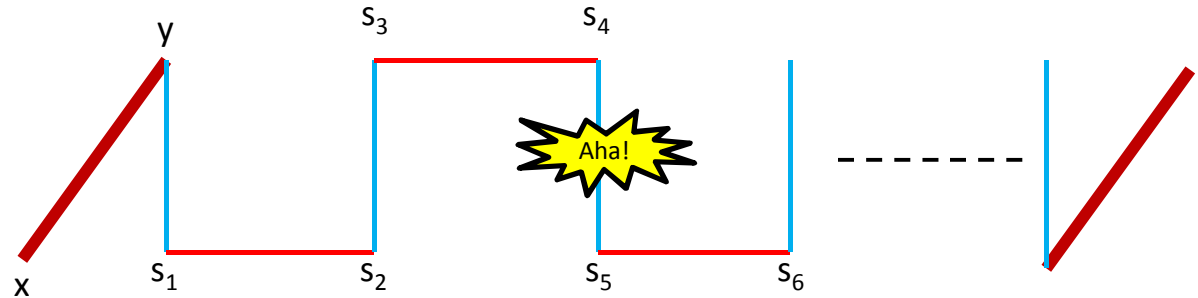
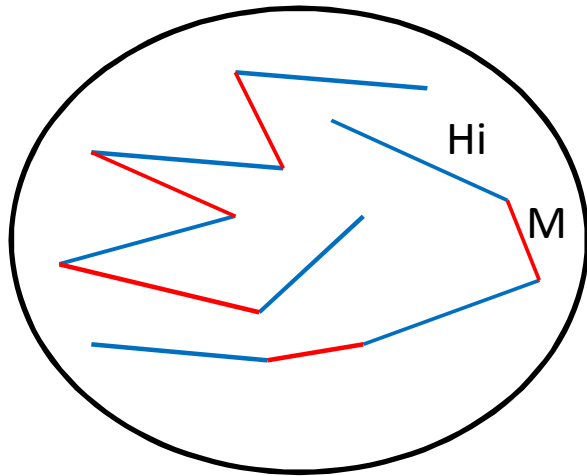
- If (s_2, s_3) violated, done. So $f(s_2) < f(s_3)$
- Suppose s_3 is M -unmatched.
- $[f(x) - f(s_1)] + [f(s_3) - f(y)] = [f(x) - f(y)] + [f(s_3) - f(s_1)]$
 $> [f(x) - f(y)] + [f(s_2) - f(s_1)]$ (Contradiction!)
- Suppose $s_4 < s_3$, so $s_4 < y$.
- $[f(x) - f(s_1)] + [f(s_4) - f(y)] > [f(x) - f(y)] + [f(s_2) - f(s_1)]$
 $+ [f(s_4) - f(s_3)]$ (Contradiction!)

Where we are



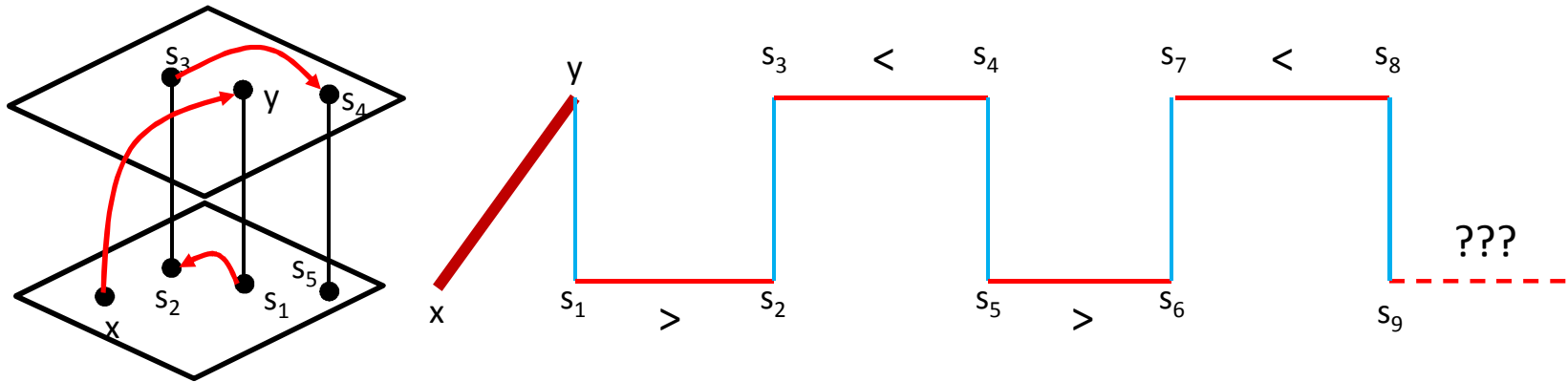
- No violated edge encountered so far. Hence, s_4 exists and $s_3 < s_4$.
- Project down to s_4 , and continue...?

Alternating paths: what's really going on



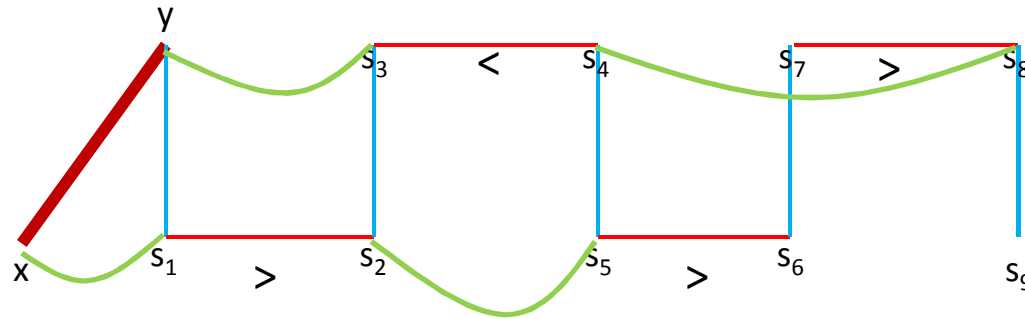
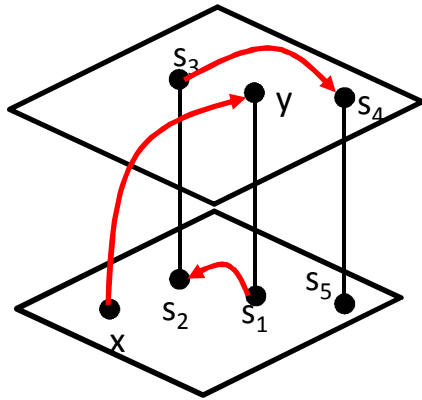
- Let H_i be i th dimension edge. It's a perfect matching.
- Symmetric diff of H_i and M is collection of alternating paths and cycles
- Partition this into segments between M_i -pairs
- Claim: Each segment has a violating edge.

The basic tools



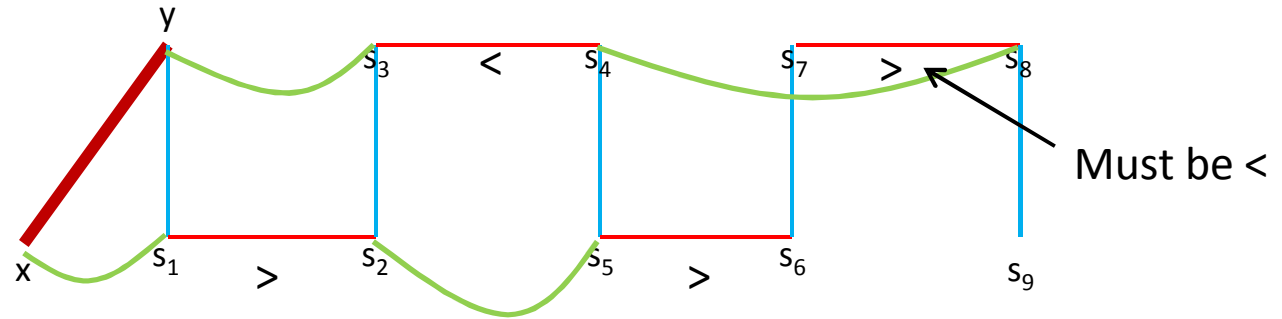
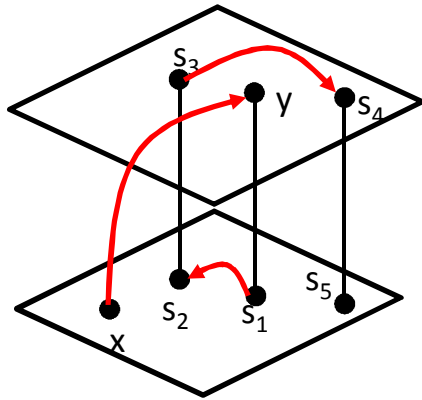
- For even i : if s_i exists, so does s_{i+1} . (H_i is perfect matching.)
- What happens at odd i ?
- Suppose no violated edge seen so far.
- Structure lemma: For $j = 1 \pmod{4}$, $s_j < s_{j+1}$. For $j = 3 \pmod{4}$, $s_j > s_{j+1}$
- Progress lemma: s_i is matched in M , so s_{i+1} also exists. (Structure implies (s_i, s_{i+1}) not in M_i .)
- Hence, if no violated edge ever seen, path goes on forever. Contradiction.

Proving structure



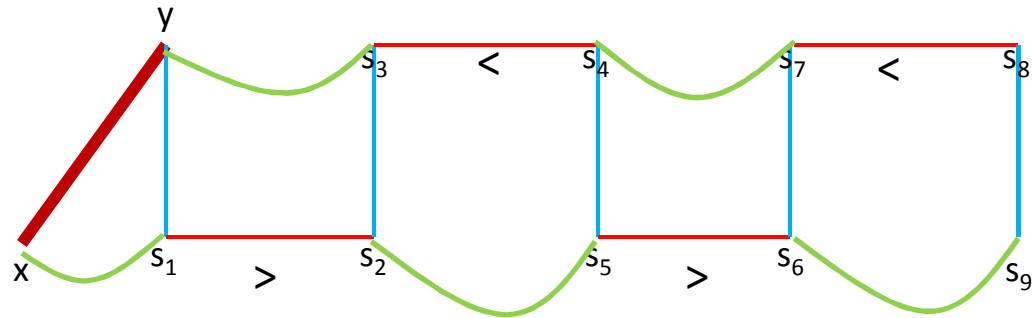
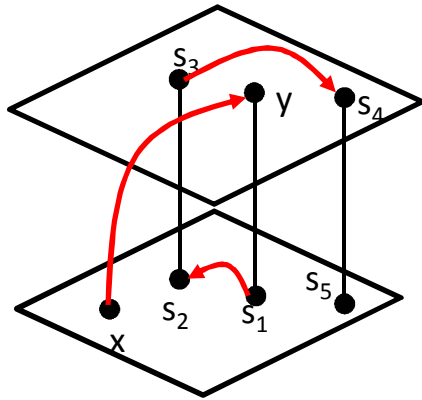
- s_i exists. For $j < i$: if $j = 1 \pmod{4}$, $s_j < s_{j+1}$. If $j = 3 \pmod{4}$, $s_j > s_{j+1}$
- Prove by induction on j . (And a contradiction)
- Suppose true for all $j' < j$. And $s_j < s_{j+1}$.
- Then remove red pairs and add green pairs from M .
- Argue (index chasing) that weight has increased.
 - Induction hypothesis gives handle on red weight
- $[f(x) - f(s_1)] + [f(s_3) - f(y)] + [f(s_2) - f(s_5)] + [f(s_8) - f(s_4)]$
 $= [f(x) - f(y)] + [f(s_2) - f(s_1)] + [f(s_3) - f(s_4)] + [f(s_8) - f(s_5)]$
 $= [f(x) - f(y)] + [f(s_2) - f(s_1)] + [f(s_3) - f(s_4)] + [f(s_6) - f(s_5)] + [f(s_8) - f(s_7)]$
 $\quad\quad\quad + [f(s_7) - f(s_6)]$

Proving structure



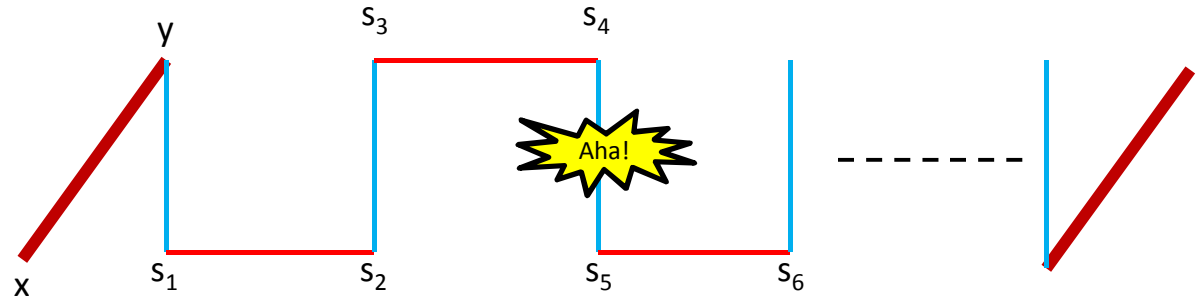
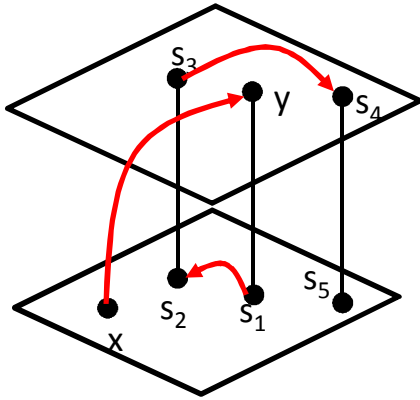
- s_i exists. For $j < i$: if $j = 1 \pmod{4}$, $s_j < s_{j+1}$. If $j = 3 \pmod{4}$, $s_j > s_{j+1}$
- Then remove red pairs and add green pairs from M .
- Argue (index chasing) that weight has increased.
 - Induction hypothesis gives handle on red weight
- $[f(x) - f(s_1)] + [f(s_3) - f(y)] + [f(s_2) - f(s_5)] + [f(s_8) - f(s_4)]$
 $= [f(x) - f(y)] + [f(s_2) - f(s_1)] + [f(s_3) - f(s_4)] + [f(s_8) - f(s_5)]$
 $= [f(x) - f(y)] + [f(s_2) - f(s_1)] + [f(s_3) - f(s_4)] + [f(s_6) - f(s_5)] + [f(s_8) - f(s_7)]$
 Positive because (s_6, s_7) not violated edge! $+ [f(s_7) - f(s_6)]$
- $w(\text{Green}) > w(\text{Red})$. Contradiction.

Proving progress



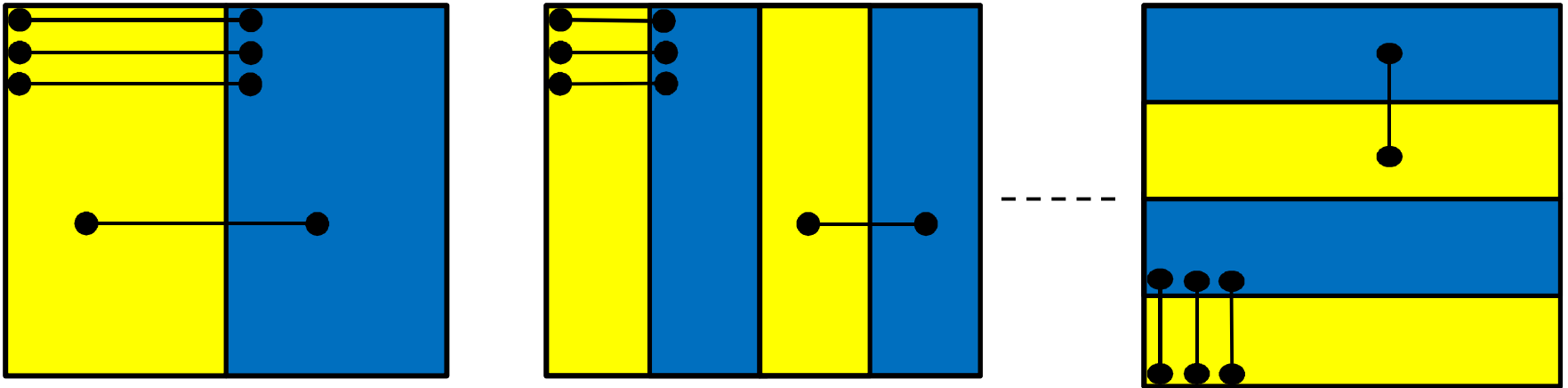
- s_i exists. What about s_{i+1} ?
- Suppose s_i unmatched in M .
- Replace red by green, do index chasing to argue that weight has increases.
- I'll spare you the details...

Recap



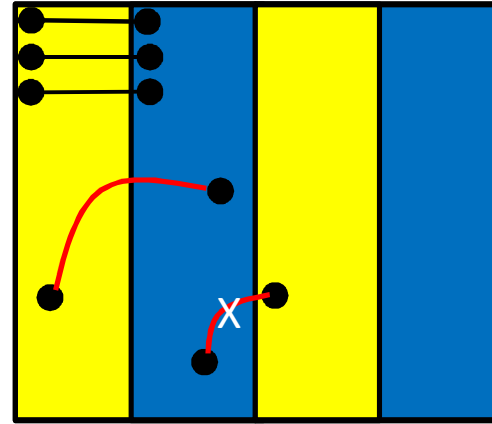
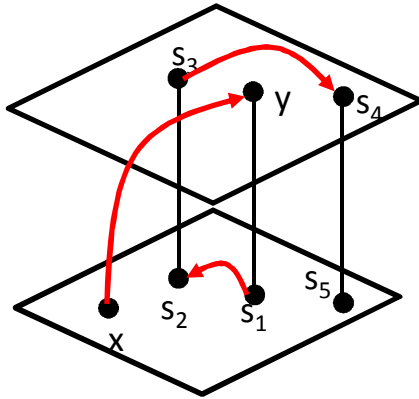
- Segment of alternating path between two M_i pairs has violated edge
 - Hence, # violated H_i edges $> |M_i|$
- Hence, #violated edges $> |M|$
- $|M| > \epsilon_f 2^n$. Hence, fraction of violated edges is ϵ_f/n
- Hence, edge tester with n/ϵ_f samples is valid tester
- The rearrangement idea: alternating paths generated from max weight matchings in VG are highly structured

On to hypergrids



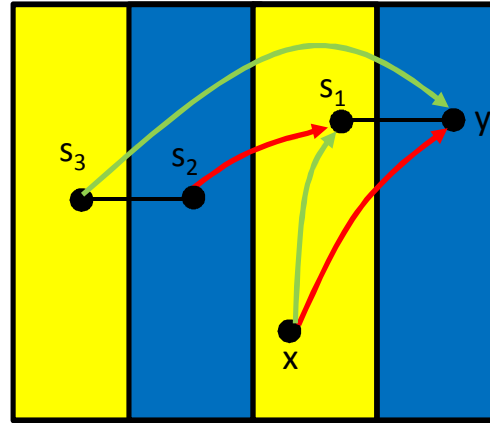
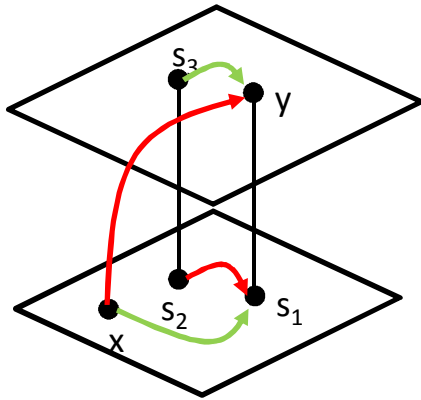
- Hypergrid $[k]^n$
- First, we need the matchings
 - Matching Hir: (x,y) where $|x_i - y_i| = 2^r, x_j = y_j$
- There are $n \log k$ of these.
- Candidate tester: pick u.a.r Hir, then u.a.r. pair from Hir.
 - Pretty much same as previous testers
- Theorem: Total number of violations in $\cup \text{Hir}$ is $\epsilon_f 2^{n-1}$
 - Tester works with $|\cup \text{Hir}| / \epsilon_f 2^{n-1} = O(\epsilon_f^{-1} n \log k)$ samples

Crossing pairs



- Violation graph, maximum weight matching defined as before
- Maximum weight matching M , and $|M| > \epsilon_f 2n-1$
- But when does (x, y) in M “cross” Hir?
 - x is at lower end of Hir, and y is at upper end of Hir
 - Let M_{ir} be pairs crossing Hir
- As before, we could do alternating paths of M_{ir} and Hir.
- Or can we...?

A problem



(s2, s3) not violation

Positive because $s3 > s2$
in hypercube

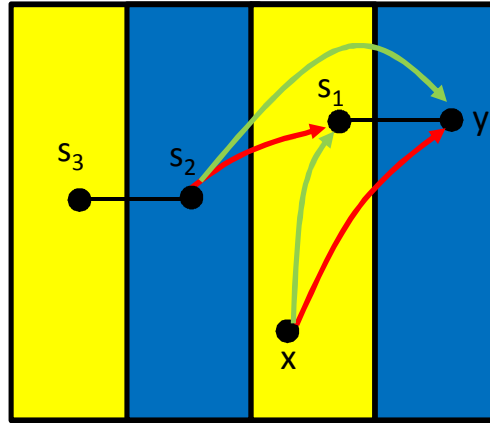
- $s3$ is unmatched.
- $[f(x) - f(s_1)] + [f(s_3) - f(y)] = [f(x) - f(y)] + [f(s_2) - f(s_1)] + [f(s_3) - f(s_2)]$
- Rearrangement does not increase weight for hypergrid
- So alternating path ends, with no violated Hir-pair ☹️

w(Green)

w(Red)

I hate potential arguments

Matching $H_{i,r}: (x,y)$ where
 $|x_i - y_i| = 2^r, x_j = y_j$

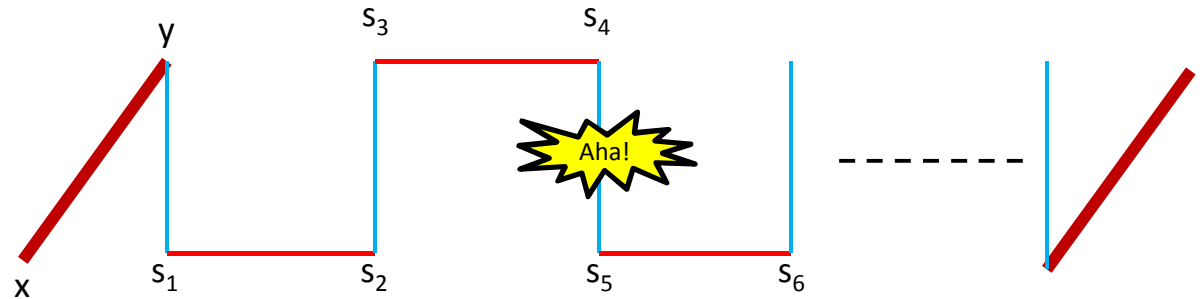
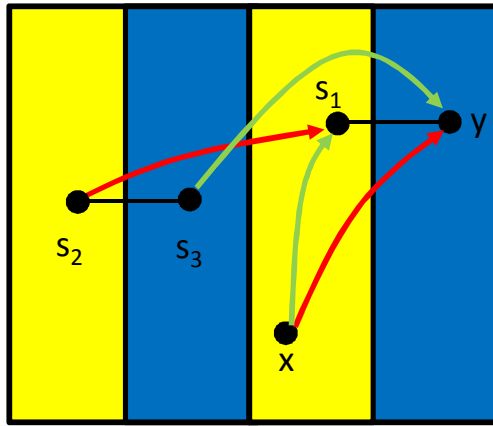


- Not that I dislike situations that may lead to confrontations...
- Let $\text{tz}(a) = p$: $2p$ divides a , but $2p+1$ doesn't

$$\Phi(M) = \sum_{(x,y) \in M} \sum_{i=1}^n \text{tz}(|y_i - x_i|)$$

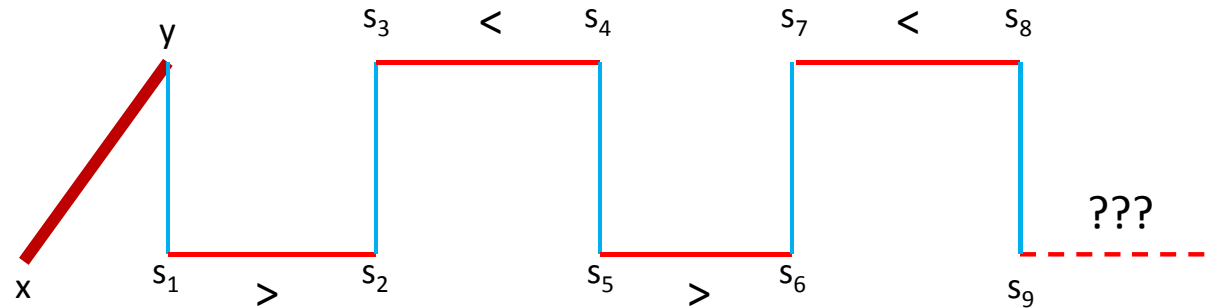
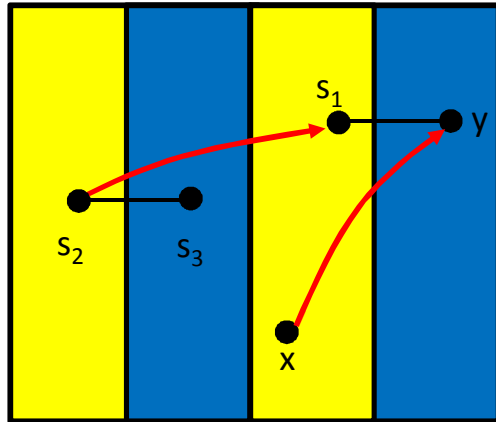
- Choose the maximum weight matching that maximizes potential
- Trying to make differences divisible by powers of 2
 - Bad situations go away

Alternating paths redux



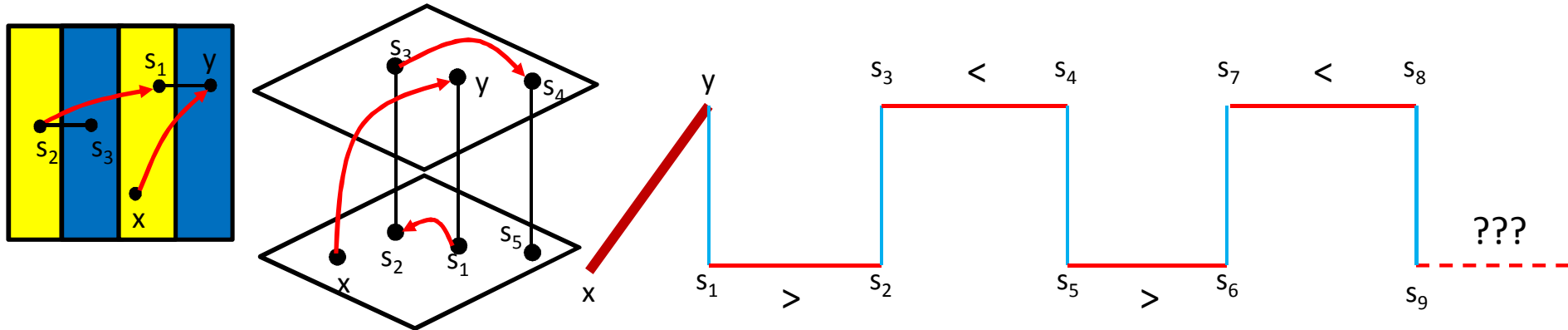
- Takes alternating paths of M and Hir
- Lemma: Every segment between Mir pairs must contain a violated Hir pair

Similar tools



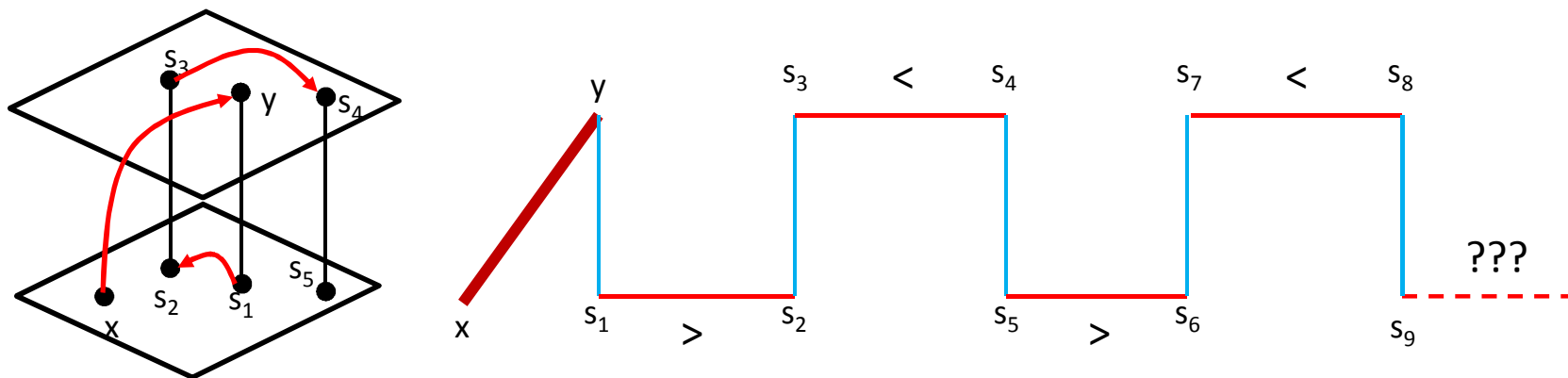
- For even i : if s_i exists, so does s_{i+1} . (H_i is almost perfect matching.)
- What happens at odd i ?
- Suppose no violated edge seen so far.
- Structure lemma: For $j = 1 \pmod{4}$, $s_j < s_{j+1}$. For $j = 3 \pmod{4}$, $s_j > s_{j+1}$
- Progress lemma: s_i is matched in M , so s_{i+1} also exists.
- Disjointness lemma: (s_i, s_{i+1}) not in M_i .
- Proof by contradiction. Rearrange to increase weight or potential Φ .

All in all...



- $O(\epsilon^{-1} k \log n)$ -query tester for monotonicity over $[k]n$.
- Basically a predefined set of pairs (union of H_{ir}). If f is ϵ -far from monotone, ϵ -fraction of pairs are violations
 - And this is the tester for all the properties
- The property of monotonicity, nor the algorithm care for the actual values (purely comparison based)
- But the proof does

The Lipschitz property



- Monotonicity is directed, Lipschitz is not

- (x, y) is edge, $x < y$

$$0 \leq f(y) - f(x)$$

$$-1 \leq f(y) - f(x) \leq 1$$

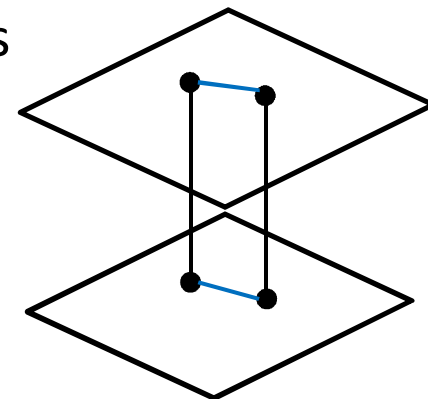
- Basis of argument was the (ordered) structure of alternating path
- So how to generalize ideas to Lipschitz (and larger classes)?

The pseudo-distance $d(x,y)$

- Define a pseudo-distance $d(x,y)$
- $w(x,y) = f(x) - f(y) - d(x,y)$
- (x,y) is violation iff $w(x,y) > 0$

- For monotonicity: $d(x,y) = 0$ if $x < y$, $d(x,y) = \infty$ otherwise
 - So $w(x,y) = f(x) - f(y)$ if $x < y$, $w(x,y) = -\infty$ otherwise
- For c -Lipschitz, $d(x,y) = c|x-y|_1$
 - So $w(x,y) = f(x) - f(y) - |x-y|_1$. Note if $w(x,y) > 0$, then $w(y,x) < 0$.
- For c -Lipschitz, comes from weighted shortest paths

- $d(x,y)$ satisfies triangle inequality
- $d(x,y)$ preserved under “projection”



Thank you