Competitive Evaluation of Threshold Functions and Game Trees in the Priced Information Model

Ferdinando Cicalese

University of Salerno

Martin Milanič

Bielefeld University

DIMACS-RUTCOR Workshop on Boolean and Pseudo-Boolean Functions
In Memory of Peter L. Hammer
January 19, 2009

The function evaluation problem

Input:

- a function f over the variables $x_1, \ldots, x_n$
- each variable has a positive cost of reading its value
- an unknown assignment $x_1 = a_1, \ldots, x_n = a_n$

Goal:

- Determine $f(a_1, \ldots a_n)$
  - adaptively reading the values of the variables
  - incurring little cost

The function evaluation problem

$$f = (x \text{ and } y) \text{ or } (x \text{ and } z)$$

- x, y, z: binary variables
- for some inputs it is possible to evaluate f without reading all variables

Example:

- $(x, y, z) = (0, 1, 1)$

  It is enough to know the value of x

The function evaluation problem

$$f = (x \text{ and } y) \text{ or } (x \text{ and } z)$$

- x, y, z: binary variables
- for some inputs it is possible to evaluate f without reading all variables

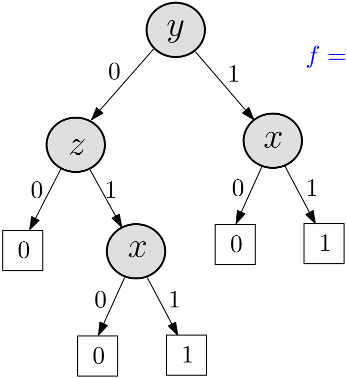
Example:

- $(x, y, z) = (0, 1, 1)$

  It is enough to know the value of x

Algorithms for evaluating f

- Dynamically select the next variable based on the values of the variables read so far
- Stop when the value of f is determined



$f = (x \ and \ y) \ or \ (x \ and \ z)$

Evaluation measure

Evasive Functions

- For any possible algorithm, all the variables must be read in the worst case.
- $f = (x \textit{ and } y) \textit{ or } (x \textit{ and } z)$
- Some important functions are evasive (e.g. game trees, AND/OR trees and threshold trees).

- Worst case analysis cannot distinguish among the performance of different algorithms.
- Instead, we use **competitive analysis** (Charikar et al. 2002)

Evaluation measure

Evasive Functions

- For any possible algorithm, all the variables must be read in the worst case.
- $f = (x \textit{ and } y) \textit{ or } (x \textit{ and } z)$
- Some important functions are evasive (e.g. game trees, AND/OR trees and threshold trees).

- Worst case analysis cannot distinguish among the performance of different algorithms.
- Instead, we use **competitive analysis** (Charikar et al. 2002)

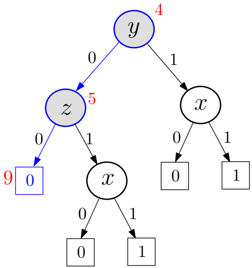
Cost of evaluation

$f = (x \textit{ and } y) \textit{ or } (x \textit{ and } z)$
$cost(x) = 3$, $cost(y) = 4$, $cost(z) = 5$

Asignment (x, y, z)	Value of f	Cheapest Proof	Cost
(0,0,0)	0	$\{$x$\}$	3
(1,1,0)	1	$\{$x,y$\}$	3+4=7

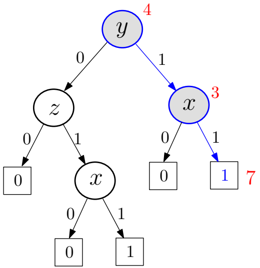
The competitive ratio

(x, y, z)	$f(x, y, z)$	Cost of Cheapest Proof	Algorithm Cost	Ratio
(0,0,0)	0	3	9	3



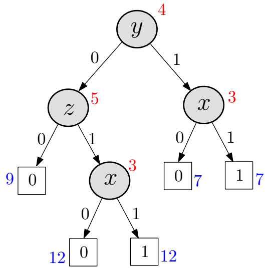
The competitive ratio

(x, y, z)	$f(x, y, z)$	Cost of Cheapest Proof	Algorithm Cost	Ratio
(0,0,0)	0	3	9	3
(1,1,0)	1	7	7	1



The competitive ratio

(x,y,z)	$f(x,y,z)$	Cost of Cheapest Proof	Algorithm Cost	Ratio
(0,0,0)	0	3	9	3
(1,0,0)	0	9	9	1
(0,1,0)	0	3	7	7/3
(0,0,1)	**0**	**3**	**12**	**4**
(1,1,0)	1	7	7	1
(1,0,1)	1	8	12	3/2
(0,1,1)	0	3	7	7/3
(1,1,1)	1	7	7	1



Measures of algorithm's performance

Competitive ratio of algorithm A for (f, c):

$$\max_{\text{all assignments } \sigma} \frac{\text{cost of } A \text{ to evaluate } f \text{ on } \sigma}{\text{cost of cheapest proof of } f \text{ on } \sigma}$$

In this talk:

Extremal *competitive ratio of A for f*:

$$\max_{\substack{\text{all assignments } \sigma,\\ \text{all cost vectors } c}} \frac{\text{cost of } A \text{ to evaluate } f \text{ on } (\sigma, c)}{\text{cost of cheapest proof of } f \text{ on } (\sigma, c)}$$

Measures of algorithm's performance

Competitive ratio of algorithm A for (f, c):

$$\max_{\text{all assignments } \sigma} \frac{\text{cost of } A \text{ to evaluate } f \text{ on } \sigma}{\text{cost of cheapest proof of } f \text{ on } \sigma}$$

In this talk:

Extremal *competitive ratio of A for f*:

$$\max_{\substack{\text{all assignments } \sigma, \\ \text{all cost vectors } c}} \frac{\text{cost of } A \text{ to evaluate } f \text{ on } (\sigma, c)}{\text{cost of cheapest proof of } f \text{ on } (\sigma, c)}$$

Measure of function's complexity

Extremal competitive ratio of f :

$$\min_{\substack{\textit{all deterministic algorithms } A\\ \textit{that evaluate } f}} \left\{ \textit{extremal competitive ratio of } A \textit{ for } f \right\}$$

The function evaluation problem

Given:
a function f over the variables $x_1, x_2, \ldots, x_n$

Combinatorial Goal:

- Determine the extremal competitive ratio of f

Algorithmic Goal:

- Devise an algorithm for evaluating f that:
  1. achieves the optimal (or close to optimal) extremal competitive ratio
  2. is efficient *(runs in time polynomial in the size of f)*

The algorithm knows the reading costs.

The function evaluation problem

Given:
a function f over the variables $x_1, x_2, \ldots, x_n$

Combinatorial Goal:

- Determine the extremal competitive ratio of f

Algorithmic Goal:

- Devise an algorithm for evaluating f that:
  1. achieves the optimal (or close to optimal) extremal competitive ratio
  2. is efficient *(runs in time polynomial in the size of f)*

The algorithm knows the reading costs.

Applications

Applications of the function evaluation problem:

- Reliability testing / diagnosis

  Telecommunications: testing connectivity of networks

  Manufacturing: testing machines before shipping

- Databases

  Query optimization

- Artificial Intelligence

  Finding optimal derivation strategies in knowledge-based systems

- Decision-making strategies (AND-OR trees)

- Computer-aided game playing *for two-player zero-sum games with perfect information, e.g.* **chess** (game trees)

Related work - other models/measures

- **Non-uniform costs & competitive analysis**

  Charikar et al. [STOC 2000, JCSS 2002]
- **Unknown costs**

  Cicalese and Laber [SODA 2006]
- **Restricted costs (selection and sorting)**

  Gupta and Kumar [FOCS 2001], Kannan and Khanna [SODA 2003]
- **Randomized algorithms**

  Snir [TCS 1985], Saks and Wigderson [FOCS 1986], Laber [STACS 2004]
- **Stochastic models**

  Random input, uniform probabilities

  Tarsi [JACM 1983], Boros and Ünlüyurt [AMAI 1999]

  Charikar et al. [STOC 2000, JCSS 2002], Greiner et al. [AI 2005]

  Random input, arbitrary probabilities

  Kaplan et al. [STOC 2005]

  Random costs

  Angelov et al. [LATIN 2008]

How to evaluate general functions?

Good algorithms are expected to test. . .

- cheap variables
- important variables ???

We use

a linear program that captures the impact of the variables

(C.-Laber 2008)

How to evaluate general functions?

Good algorithms are expected to test. . .

- cheap variables
- important variables ???

We use
a linear program that captures the impact of the variables
(C.-Laber 2008)

The minimal proofs

f - a function over $V = \{x_1, \ldots, x_n\}$

R - range of f

Definition

Let $r \in R$. A **minimal proof for** $f(x) = r$ is a minimal set of variables $P \subseteq V$ such that *there is an assignment σ of values to the variables in P such that f_σ is constantly equal to r*.

Example: $f(x_1, x_2, x_3) = (x_1 \text{ and } x_2) \text{ or } (x_1 \text{ and } x_3)$, $R = \{0, 1\}$

minimal proofs for $f(x) = 1$: $\{\{x_1, x_2\}, \{x_1, x_3\}\}$

minimal proofs for $f(x) = 0$: $\{\{x_1\}, \{x_2, x_3\}\}$

The minimal proofs

f - a function over $V = \{x_1, \ldots, x_n\}$

R - range of f

Definition

Let $r \in R$. A **minimal proof for** $f(x) = r$ is a minimal set of variables $P \subseteq V$ such that *there is an assignment σ of values to the variables in P such that f_σ is constantly equal to r*.

Example: $f(x_1, x_2, x_3) = (x_1 \text{ and } x_2) \text{ or } (x_1 \text{ and } x_3)$, $R = \{0, 1\}$

minimal proofs for $f(x) = 1$: $\{\{x_1, x_2\}, \{x_1, x_3\}\}$

minimal proofs for $f(x) = 0$: $\{\{x_1\}, \{x_2, x_3\}\}$

The Linear Program (C.-Laber 2008)

$$\mathbf{LP}(f) \begin{cases} \textit{Minimize} \sum_{x \in V} s(x) \\ \textit{s.t.} \\ \sum_{x \in P} s(x) \geq 1 \quad \text{for every minimal proof } P \text{ of } f \\ \qquad\quad s(x) \geq 0 \quad \text{for every } x \in V \end{cases}$$

Intuitively, $s(x)$ measures the **impact** of variable x.

The feasible solutions to the **LP**(f) are precisely the fractional hitting sets of the set of minimal proofs of f.

The Linear Program (C.-Laber 2008)

$$\textbf{LP}(f) \left\{ \begin{array}{llll} \textit{Minimize } \sum_{x \in V} s(x) & & & \\ \textit{s.t.} & & & \\ \sum_{x \in P} s(x) & \geq & 1 & \text{for every minimal proof } P \text{ of } f \\ s(x) & \geq & 0 & \text{for every } x \in V \end{array} \right.$$

Intuitively, $s(x)$ measures the **impact** of variable x.

The feasible solutions to the $\textbf{LP}(f)$ are precisely the fractional hitting sets of the set of minimal proofs of f.

LP(f): example

$f(x_1, x_2, x_3) = (x_1 \textit{ and } x_2) \textit{ or } (x_1 \textit{ and } x_3)$

minimal proofs for $f = 1$: $\{\{x_1, x_2\}, \{x_1, x_3\}\}$

minimal proofs for $f = 0$: $\{\{x_1\}, \{x_2, x_3\}\}$

$$\textbf{LP}(f) \left\{ \begin{array}{r} \textit{Minimize } s_1 + s_2 + s_3 \\ \textit{s.t.} \qquad\qquad\quad \\ s_1 + s_2 \geq 1 \\ s_1 + s_3 \geq 1 \\ s_1 \geq 1 \\ s_2 + s_3 \geq 1 \\ s_1, s_2, s_3 \geq 0 \end{array} \right.$$

Optimal solution: $s = (1, 1/2, 1/2)$

The Linear Programming Approach

LPA(*f* : **function)**

While the value of f is unknown

Select a feasible solution $s()$ of **LP**(f)

Read the variable u which minimizes $c(x)/s(x)$
(cost/impact)

$c(x) = c(x) - s(x)c(u)/s(u)$

$f \leftarrow$ restriction of f after reading u

End While

The LPA bounds the extremal competitive ratio

The selection of solution s determines both the **computational efficiency** and the **performance** (extremal competitive ratio) of the algorithm.

Key Lemma (C.-Laber 2008)

Let $\mathbf{K}$ *be a positive number. If*

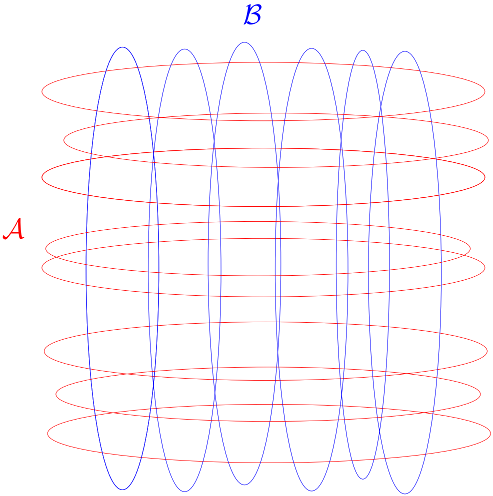
$$\mathit{ObjectiveFunctionValue}(s) \leq \mathbf{K}\,,$$

for every selected solution s

then

$$\mathit{ExtremalCompetitiveRatio}(f) \leq \mathbf{K}\,.$$

Cross-intersecting families



- cross-intersecting: $A \in \mathcal{A}, B \in \mathcal{B} \Rightarrow A \cap B \neq \emptyset$

Minimal proofs and cross-intersecting families

$f : S_1 \times \cdots \times S_n \to S$, a function over $V = \{x_1, \ldots, x_n\}$

R - range of f

For $r \in R$, let $\mathcal{P}(r)$ denote the set of **minimal proofs for** $f(x) = r$.

Then:

- for every $r \neq r'$, the families $\mathcal{P}(r)$ and $\mathcal{P}(r')$ are cross-intersecting

The Linear Program and cross-intersection

$$\mathbf{LP}(f) \begin{cases} \textit{Minimize} \sum_{x \in V} s(x) & \\ \textit{s.t.} & \\ \sum_{x \in P} s(x) \geq 1 & \text{for every } P \in \mathcal{P} \\ \quad\quad s(x) \geq 0 & \text{for every } x \in V \end{cases}$$

$$\mathcal{P} = \cup_{r \in R} \mathcal{P}(r)$$

union of pairwise cross-intersecting families

For every function $f : S_1 \times \cdots \times S_n \to S$, the $\mathbf{LP}(f)$ seeks a *minimal fractional hitting set of a union of pairwise cross-intersecting families*.

Cross-intersecting lemma

Cross-Intersecting Lemma (C.-Laber 2008)

Let $\mathcal{A}$ and $\mathcal{B}$ be two non-empty **cross-intersecting** *families of subsets of V.*
Then, there is a fractional hitting set s of $\mathcal{A} \cup \mathcal{B}$ such that

$$\|s\|_1 = \sum_{x \in V} s(x) \le \max\{|P| : P \in \mathcal{A} \cup \mathcal{B}\}\,.$$

- geometric proof
- generalizes to any number of pairwise cross-intersecting families

Applications of the cross-intersecting lemma

1. $f : S_1 \times \cdots \times S_n \to S$, nonconstant:
   $ExtremalCompetitiveRatio(f) \leq PROOF(f)$
   ($PROOF(f) =$ size of the largest minimal proof of f)

2. Monotone Boolean functions:
   $ExtremalCompetitiveRatio(f) = PROOF(f)$

3. Game trees: $ExtremalCompetitiveRatio(f) \leq TBA$

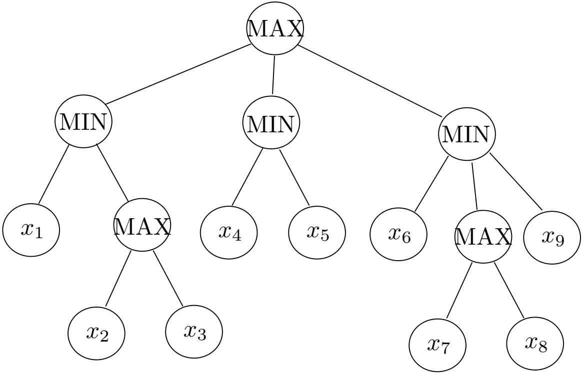
Applications of the cross-intersecting lemma

1. $f : S_1 \times \cdots \times S_n \to S$, nonconstant:
   $\mathit{ExtremalCompetitiveRatio}(f) \leq \mathit{PROOF}(f)$
   ($\mathit{PROOF}(f) =$ size of the largest minimal proof of f)
2. Monotone Boolean functions:
   $\mathit{ExtremalCompetitiveRatio}(f) = \mathit{PROOF}(f)$
3. Game trees: $\mathit{ExtremalCompetitiveRatio}(f) \leq \mathit{TBA}$

Applications of the cross-intersecting lemma

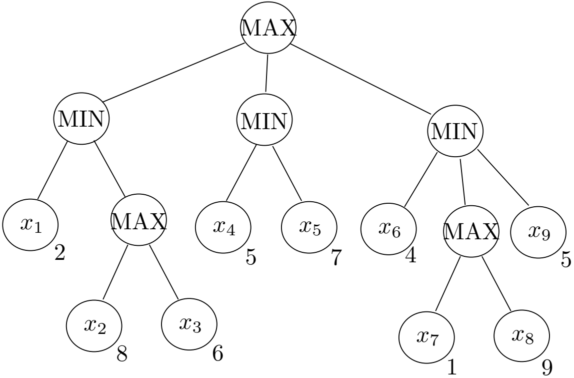
1. $f : S_1 \times \cdots \times S_n \to S$, nonconstant:
   $\mathit{ExtremalCompetitiveRatio}(f) \leq \mathit{PROOF}(f)$
   ($\mathit{PROOF}(f) =$ size of the largest minimal proof of f)
2. Monotone Boolean functions:
   $\mathit{ExtremalCompetitiveRatio}(f) = \mathit{PROOF}(f)$
3. Game trees: $\mathit{ExtremalCompetitiveRatio}(f) \leq \mathit{TBA}$

Game trees



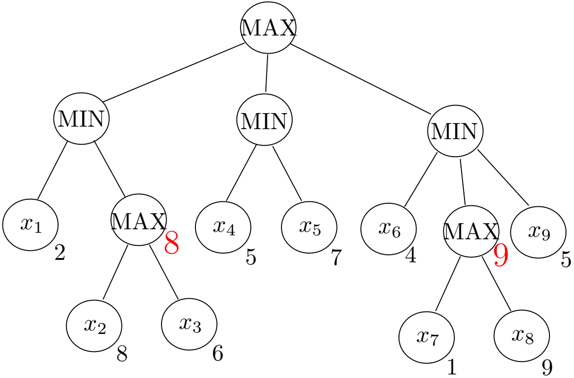
game tree

Game trees



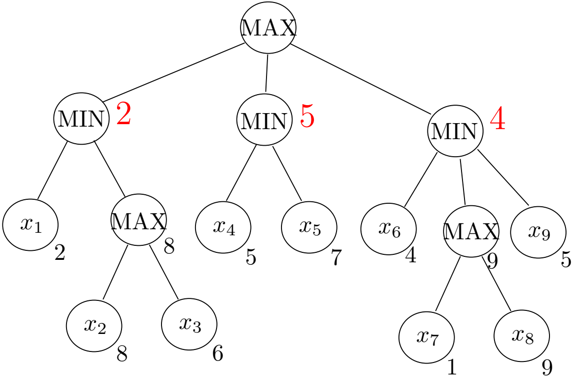
game tree

Game trees



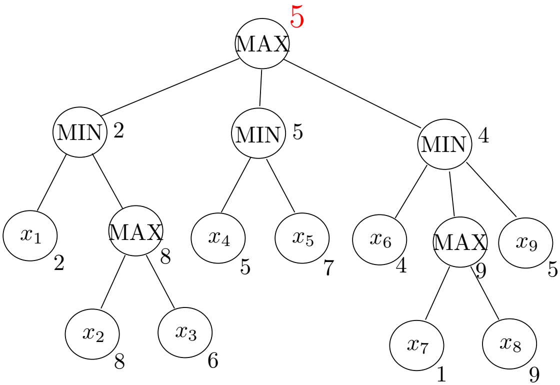
game tree

Game trees



game tree

Game trees



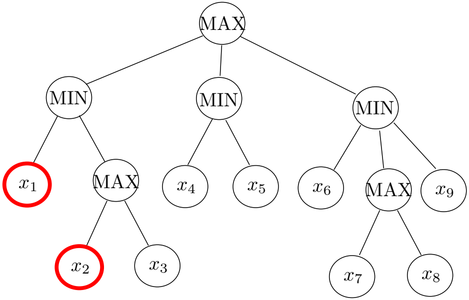
game tree

Minterms of game trees

Minterm: minimal set $A \subseteq V$ of variables such that

$$\textit{value of } f \geq \textit{value of } A := \textit{min value of variables in } A$$

Minterms can prove a lower bound for the value of f.

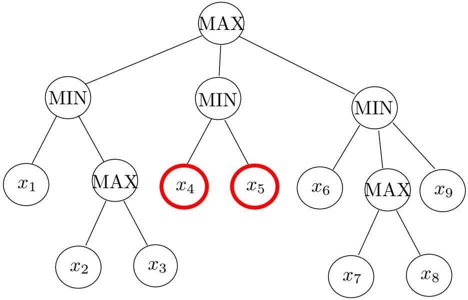


Minterms of game trees

Minterm: minimal set $A \subseteq V$ of variables such that

value of f $\geq$ *value of A* $:=$ *min value of variables in A*

Minterms can prove a lower bound for the value of f.

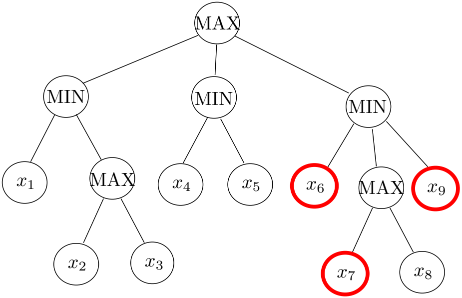


Minterms of game trees

Minterm: minimal set $A \subseteq V$ of variables such that

value of f $\geq$ *value of A* $:=$ *min value of variables in A*

Minterms can prove a lower bound for the value of f.

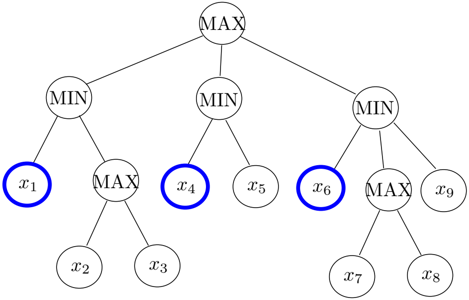


Maxterms of game trees

Maxterm: minimal set $B \subseteq V$ of variables such that

value of f $\leq$ *value of B* $:=$ *max value of variables in B*

Maxterms can prove an upper bound for the value of f.

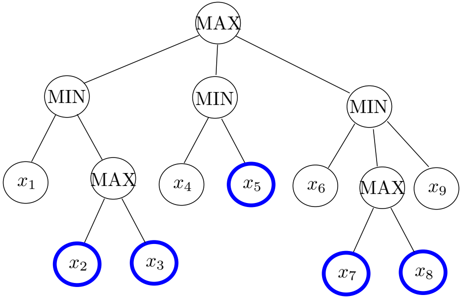


Maxterms of game trees

Maxterm: minimal set $B \subseteq V$ of variables such that

$$\textit{value of } f \leq \textit{value of } B := \textit{max value of variables in } B$$

Maxterms can prove an upper bound for the value of f.



Lower bound for the extremal competitive ratio

- $k(f) = \max\{|A| : A \text{ minterm of } f\}$
- $l(f) = \max\{|B| : B \text{ maxterm of } f\}$

Theorem (Cicalese-Laber 2005)

Let f be a game tree with no minterms or maxterms of size 1. Then,

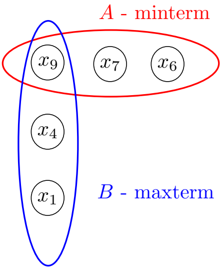
$$ExtremalCompetitiveRatio(f) \geq \max\{k(f), l(f)\}\,.$$

Minimal proofs of game trees

To prove that the value of f is **b**, we need:

- a minterm of value **b** [*proves* $f \geq$ **b**]
- a maxterm of value **b** [*proves* $f \leq$ **b**]

Every minimal proof = union of a minterm and a maxterm



A first upper bound

It follows that

$PROOF(f)$ = size of the largest minimal proof of f
$= k(f) + l(f) - 1.$

Theorem (Cicalese-Laber 2008)

$f : S_1 \times \cdots \times S_n \to S$, nonconstant:
$ExtremalCompetitiveRatio(f) \leq PROOF(f)$

For a game tree f,
$ExtremalCompetitiveRatio(f) \leq k(f) + l(f) - 1.$

Lower bound: $\max\{k(f), l(f)\}$

A first upper bound

It follows that

$PROOF(f) =$ size of the largest minimal proof of f
$= k(f) + l(f) - 1.$

Theorem (Cicalese-Laber 2008)

$f : S_1 \times \cdots \times S_n \to S$, nonconstant:
$ExtremalCompetitiveRatio(f) \leq PROOF(f)$

For a game tree f,
$ExtremalCompetitiveRatio(f) \leq k(f) + l(f) - 1$.

Lower bound: $\max\{k(f), l(f)\}$

A first upper bound

It follows that

$PROOF(f) =$ size of the largest minimal proof of f
$= k(f) + l(f) - 1.$

Theorem (Cicalese-Laber 2008)

$f : S_1 \times \cdots \times S_n \to S$, nonconstant:
$ExtremalCompetitiveRatio(f) \leq PROOF(f)$

For a game tree f,
$ExtremalCompetitiveRatio(f) \leq k(f) + l(f) - 1$.

Lower bound: $\max\{k(f), l(f)\}$

Can we close the gap?

Yes:

Claim

For every restriction f' of f, there is a fractional hitting set s of the set of minimal proofs of f' such that

$$\|s\|_1 \leq \max\{k(f), l(f)\}\,.$$

By the **Key Lemma**,

$$ExtremalCompetitiveRatio(f) \leq \max\{k(f), l(f)\}$$

Can we close the gap?

Yes:

Claim

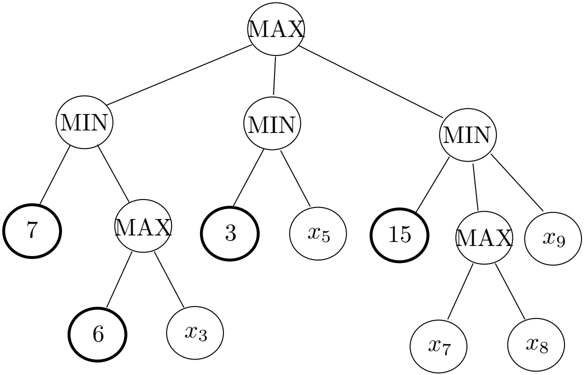
For every restriction f' of f, there is a fractional hitting set s of the set of minimal proofs of f' such that

$$\|s\|_1 \leq \max\{k(f), l(f)\}\,.$$

By the **Key Lemma**,

$$\mathit{ExtremalCompetitiveRatio}(f) \leq \max\{k(f), l(f)\}$$

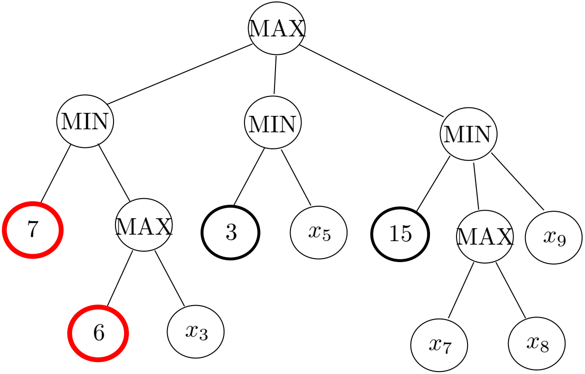
Restrictions of game trees



restriction of f

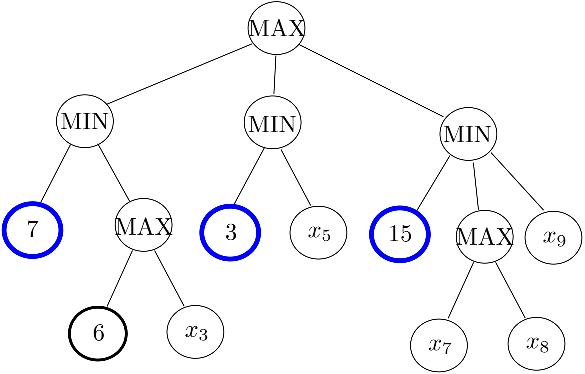
$$f'(x_3, x_5, x_7, x_8, x_9)$$

Restrictions of game trees



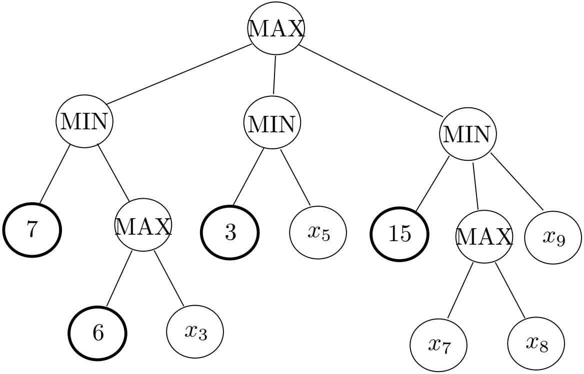
$$\tilde{f}(x_3, x_5, x_7, x_8, x_9) \geq 6$$

Restrictions of game trees



$$\tilde{f}(x_3, x_5, x_7, x_8, x_9) \leq 15$$

Restrictions of game trees

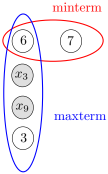


$$\tilde{f}(x_3, x_5, x_7, x_8, x_9) \in [6, 15] = [LB, UB]$$

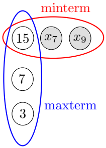
Minimal proofs of a restriction of a game tree

maxterm proofs

$b = LB$

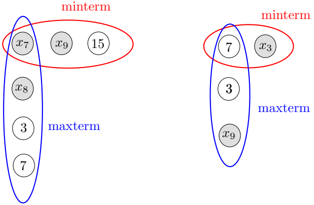


minterm proofs



$b = UB$

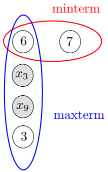
combined proofs

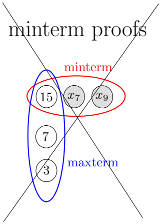


$$LB \leq b \leq UB$$

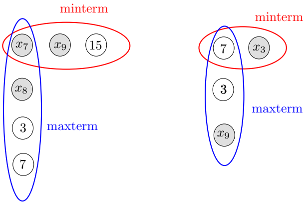
Case 1: No maxterm has been fully evaluated yet

maxterm proofs

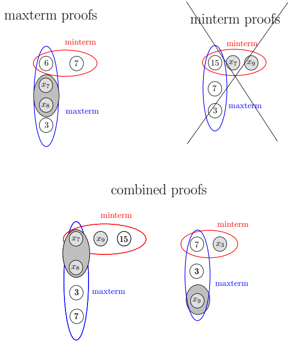




combined proofs



Case 1: No maxterm has been fully evaluated yet

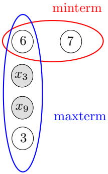


Let s be (the characteristic vector of) **a minimal hitting set of the shaded sets**.

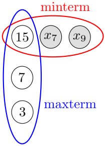
$$\|s\|_1 \leq k(f) \leq \max\{k(f), l(f)\}$$

Case 2: There is a fully evaluated maxterm

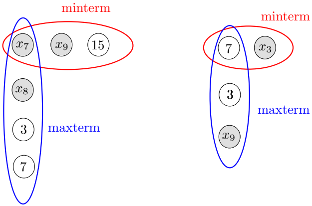
maxterm proofs



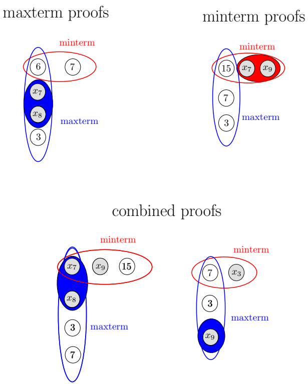
minterm proofs



combined proofs



Case 2: There is a fully evaluated maxterm



R: the family of the minterm proofs

B: the family of the maxterm parts of the non-minterm proofs

Case 2: There is a fully evaluated maxterm

- **R** and **B** are non-empty sets
- **R** and **B** are cross-intersecting
- every minimal proof contains a member of $\mathbf{R} \cup \mathbf{B}$

By the **Cross-intersecting lemma**, there exists a feasible solution s to the $\mathbf{LP}(\mathbf{f}')$ such that

$$\|s\|_1 \leq \max\{|P| : P \in \mathbf{R} \cup \mathbf{B}\} \leq \max\{k(f), l(f)\}\,.$$

Case 2: There is a fully evaluated maxterm

- **R** and **B** are non-empty sets
- **R** and **B** are cross-intersecting
- every minimal proof contains a member of $\mathbf{R} \cup \mathbf{B}$

By the **Cross-intersecting lemma**, there exists a feasible solution s to the $\mathbf{LP}(\mathbf{f}')$ such that

$$\|s\|_1 \leq \max\{|P| : P \in \mathbf{R} \cup \mathbf{B}\} \leq \max\{k(f), l(f)\}\,.$$

The extremal competitive ratio for game trees

Theorem

Let f be a game tree with no minterms or maxterms of size 1. Then,

$$\mathit{ExtremalCompetitiveRatio}(f) = \max\{k(f), l(f)\} .$$

Value dependent costs

Suppose that **the cost of reading a variable can depend on the variable's value:**

$$c(x) = \begin{cases} 50, & \text{if } x = 0; \\ 1000, & \text{if } x = 1. \end{cases}$$

Theorem

Let f be a monotone Boolean function or a game tree. Then,

$$ExtremalCompetitiveRatio(f, r) = r \cdot ECR(f) - r + 1\,,$$

where

$$r = \max_{x \in V} \frac{c_{max}(x)}{c_{min}(x)}\,.$$

Value dependent costs

Suppose that **the cost of reading a variable can depend on the variable's value:**

$$c(x) = \begin{cases} 50, & \text{if } x = 0; \\ 1000, & \text{if } x = 1. \end{cases}$$

Theorem

Let f be a monotone Boolean function or a game tree. Then,

$$\mathit{ExtremalCompetitiveRatio}(f, r) = r \cdot \mathit{ECR}(f) - r + 1\,,$$

where

$$r = \max_{x \in V} \frac{c_{max}(x)}{c_{min}(x)}\,.$$

LPA has a very broad applicability

LPA does not depend on the structure of f

It can be used to derive upper bounds on the extremal competitive ratios of **very different functions**:

- $f =$ minimum of a list: $ExtremalCompetitiveRatio(f) \leq n - 1$ [Cicalese-Laber 2005]
- $f =$ the sorting function: $ExtremalCompetitiveRatio(f) \leq n - 1$ [Cicalese-Laber 2008]
- $f : S_1 \times \cdots \times S_n \to S$, nonconstant: $ExtremalCompetitiveRatio(f) \leq PROOF(f)$ [Cicalese-Laber 2008]
- $f =$ monotone Boolean function: $ExtremalCompetitiveRatio(f) = PROOF(f)$ [Cicalese-Laber 2008]
- $f =$ game tree: $ExtremalCompetitiveRatio(f) \leq max\{k(f), l(f)\}$

Summary

We have seen:

- the Linear Programming Approach for the development of competitive algorithms for the function evaluation problem,
- the **extremal competitive ratio** for **game trees**,
- the more general model of value dependent costs.

Part II

Threshold functions
and
Extended threshold tree functions

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

  This talk:
  - threshold functions (**a quadratic algorithm**)
  - extended threshold tree functions (**a pseudo-polynomial algorithm**)

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

  This talk:
  - threshold functions (**a quadratic algorithm**)
  - extended threshold tree functions (**a pseudo-polynomial algorithm**)

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

  This talk:
  - threshold functions (**a quadratic algorithm**)
  - extended threshold tree functions (**a pseudo-polynomial algorithm**)

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

  This talk:
  - threshold functions (**a quadratic algorithm**)
  - extended threshold tree functions (**a pseudo-polynomial algorithm**)

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

This talk:

- threshold functions (**a quadratic algorithm**)
- extended threshold tree functions (**a pseudo-polynomial algorithm**)

Algorithmic issues: the state of the art

Are there **efficient** algorithms with optimal competitiveness?

- game trees: **there is a polynomial-time algorithm**
- monotone Boolean functions ??? OPEN QUESTION
- subclasses of monotone Boolean functions:
  - AND/OR trees = game trees with 0-1 values [Charikar et al. 2002]
  - threshold tree functions [Cicalese-Laber 2005]

  This talk:
  - threshold functions (**a quadratic algorithm**)
  - extended threshold tree functions (**a pseudo-polynomial algorithm**)

Threshold Functions

$f : \{0,1\}^n \to \{0,1\}$ is a threshold function if $\exists w_1, \ldots, w_n, t$ integers s.t.

$$f(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if} \quad \sum_i x_i w_i \geq t \\ 0 & \text{otherwise} \end{cases}$$

Separating structure

$(w_1, \ldots, w_n; t)$ is a separating structure of f

We assume that $1 \leq w_i \leq t$ for all i.

Threshold Functions

$f : \{0,1\}^n \rightarrow \{0,1\}$ is a threshold function if $\exists w_1, \ldots, w_n, t$ integers s.t.

$$f(x_1, \ldots, x_n) = \begin{cases} 1 & \textit{if} \quad \sum_i x_i w_i \geq t \\ 0 & \textit{otherwise} \end{cases}$$

Separating structure

$(w_1, \ldots, w_n; t)$ is a separating structure of f

We assume that $1 \leq w_i \leq t$ for all i.

Applications

- Switching networks
- Automatic diagnosis
- Mutually exclusive mechanisms
- Decision-making strategies
- Neural networks
- Weighted majority games

Outline

- Separating structures for f and feasible solutions for **LP**(f)
  - Quadratic $\gamma(f)$-competitive algo for threshold functions

$\gamma(f)$ = extremal competitive ratio of f

- The range of separating structures of f
- Extended threshold tree functions
  - Pseudo-polytime $\gamma(f)$-competitive algo for extended threshold tree functions
  - The HLP_f for studying function evaluation

Outline

- Separating structures for f and feasible solutions for **LP**(f)
  - Quadratic $\gamma(f)$-competitive algo for threshold functions

$\gamma(f)$ = extremal competitive ratio of f

- The range of separating structures of f
- Extended threshold tree functions
  - Pseudo-polytime $\gamma(f)$-competitive algo for extended threshold tree functions
  - The HLP_f for studying function evaluation

Outline

- Separating structures for f and feasible solutions for **LP**(f)
  - Quadratic $\gamma(f)$-competitive algo for threshold functions

$\gamma(f)$ = extremal competitive ratio of f

- The range of separating structures of f
- Extended threshold tree functions
  - Pseudo-polytime $\gamma(f)$-competitive algo for extended threshold tree functions
  - The HLP_f for studying function evaluation

Outline

- Separating structures for f and feasible solutions for **LP**(f)
  - Quadratic $\gamma(f)$-competitive algo for threshold functions

$\gamma(f)$ = extremal competitive ratio of f

- The range of separating structures of f
- Extended threshold tree functions
  - Pseudo-polytime $\gamma(f)$-competitive algo for extended threshold tree functions
  - The HLP_f for studying function evaluation

Certificates of a threshold function

f threshold function with separating structure $(w_1, \ldots, w_n; t)$.

X is a minterm (prime implicant) of f

$$\sum_X w_i \geq t \quad \text{and} \quad \sum_{X \setminus \{j\}} w_i < t \text{ for every } j \in X$$

X is a maxterm (prime implicate) of f

$$\sum_{V \setminus X} w_i < t \quad \text{and} \quad \sum_{V \setminus X \cup \{j\}} w_i \geq t \text{ for every } j \in X$$

Technical assumption

$t \leq \frac{1}{2}(\sum_i w_i + 1)$, i.e., f is dual major

then every maxterm contains a minterm

Certificates of a threshold function

f threshold function with separating structure $(w_1, \ldots, w_n; t)$.

X is a minterm (prime implicant) of f

$$\sum_X w_i \geq t \quad \text{and} \quad \sum_{X\setminus\{j\}} w_i < t \text{ for every } j \in X$$

X is a maxterm (prime implicate) of f

$$\sum_{V\setminus X} w_i < t \quad \text{and} \quad \sum_{V\setminus X\cup\{j\}} w_i \geq t \text{ for every } j \in X$$

Technical assumption

$t \leq \frac{1}{2}(\sum_i w_i + 1)$, i.e., f is dual major
then every maxterm contains a minterm

Certificates of a threshold function

f threshold function with separating structure $(w_1, \ldots, w_n; t)$.

X is a minterm (prime implicant) of f

$$\sum_X w_i \geq t \quad \text{and} \quad \sum_{X \setminus \{j\}} w_i < t \text{ for every } j \in X$$

X is a maxterm (prime implicate) of f

$$\sum_{V \setminus X} w_i < t \quad \text{and} \quad \sum_{V \setminus X \cup \{j\}} w_i \geq t \text{ for every } j \in X$$

Technical assumption

$t \leq \frac{1}{2}(\sum_i w_i + 1)$, i.e., f is dual major
then every maxterm contains a minterm

Certificates of a threshold function

f threshold function with separating structure $(w_1, \ldots, w_n; t)$.

X is a minterm (prime implicant) of f

$$\sum_X w_i \geq t \quad \text{and} \quad \sum_{X \setminus \{j\}} w_i < t \text{ for every } j \in X$$

X is a maxterm (prime implicate) of f

$$\sum_{V \setminus X} w_i < t \quad \text{and} \quad \sum_{V \setminus X \cup \{j\}} w_i \geq t \text{ for every } j \in X$$

Technical assumption

$t \leq \frac{1}{2}(\sum_i w_i + 1)$, i.e., f is dual major

then every maxterm contains a minterm

Certificates of a threshold function

f threshold function with separating structure $(w_1, \ldots, w_n; t)$.

X is a minterm (prime implicant) of f

$$\sum_{X} w_i \geq t \quad \text{and} \quad \sum_{X \setminus \{j\}} w_i < t \text{ for every } j \in X$$

X is a maxterm (prime implicate) of f

$$\sum_{V \setminus X} w_i < t \quad \text{and} \quad \sum_{V \setminus X \cup \{j\}} w_i \geq t \text{ for every } j \in X$$

Technical assumption

$t \leq \frac{1}{2}(\sum_i w_i + 1)$, i.e., f is dual major
then every maxterm contains a minterm

Lemma

Every separating structure $(w_1, \ldots, w_n; t)$ for f induces a feasible solution $\mathbf{s} = (s(x_1), \ldots, s(x_n))$ for **LP**(f).

- $s(x_i) = w_i/t$
- X is a minterm of f

$$\sum_{i \in X} s(x_i) = \sum_{i \in X} \frac{w_i}{t} \geq 1.$$

- Y is a maxterm of f

$$\exists X \subseteq Y \text{ s.t. } X \text{ is a minterm } \Rightarrow \sum_{i \in Y} s(x_i) \geq \sum_{i \in X} s(x_i) \geq 1.$$

Separating Structure and LP(f)

Lemma

Every separating structure $(w_1, \ldots, w_n; t)$ for f induces a feasible solution $\mathbf{s} = (s(x_1), \ldots, s(x_n))$ for **LP**(f).

- $s(x_i) = w_i/t$
- X is a minterm of f

$$\sum_{i \in X} s(x_i) = \sum_{i \in X} \frac{w_i}{t} \geq 1.$$

- Y is a maxterm of f

$$\exists X \subseteq Y \text{ s.t. } X \text{ is a minterm } \Rightarrow \sum_{i \in Y} s(x_i) \geq \sum_{i \in X} s(x_i) \geq 1.$$

Separating Structure and LP(f)

Lemma

Every separating structure $(w_1, \ldots, w_n; t)$ for f induces a feasible solution $\mathbf{s} = (s(x_1), \ldots, s(x_n))$ for **LP**(f).

- $s(x_i) = w_i/t$
- X is a minterm of f

$$\sum_{i \in X} s(x_i) = \sum_{i \in X} \frac{w_i}{t} \geq 1.$$

- Y is a maxterm of f

$$\exists X \subseteq Y \text{ s.t. } X \text{ is a minterm } \Rightarrow \sum_{i \in Y} s(x_i) \geq \sum_{i \in X} s(x_i) \geq 1.$$

Separating Structure and LP(f)

Lemma

Every separating structure $(w_1, \ldots, w_n; t)$ *for* f *induces a feasible solution* $\mathbf{s}$ *for* $\mathbf{LP}(f)$ *s.t.*

$$\|\mathbf{s}\|_1 = val(\mathbf{w}; t) = 1/t \sum w_i.$$

How bad can the best separating structure be for the purpose of the $\mathcal{LPA}$?

Lemma

Every separating structure $(w_1, \ldots, w_n; t)$ for f induces a feasible solution $\mathbf{s}$ *for* $\mathbf{LP}(f)$ *s.t.*

$$\|\mathbf{s}\|_1 = val(\mathbf{w}; t) = 1/t \sum w_i.$$

How bad can the best separating structure be for the purpose of the $\mathcal{LPA}$?

Separating Structure and LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t.
$val(\mathbf{w}; t) = \sum w_i/t \leq \max\{k(f), l(f)\}$.

induces an optimal implementation of the $\mathcal{LPA}$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ satisfies

$$\max\left\{\frac{\sum w_i/t}{\sum w_i'/t'}, \frac{\sum w_i'/t'}{\sum w_i/t}\right\} \leq 2$$

Separating Structure and LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $val(\mathbf{w}; t) = \sum w_i/t \le \max\{k(f), l(f)\}$.

induces an optimal implementation of the $\mathcal{LPA}$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ satisfies

$$\max\left\{\frac{\sum w_i/t}{\sum w_i'/t'}, \frac{\sum w_i'/t'}{\sum w_i/t}\right\} \le 2$$

Separating Structure and LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t.
$val(\mathbf{w}; t) = \sum w_i / t \leq \max\{k(f), l(f)\}$.

induces an optimal implementation of the $\mathcal{LPA}$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ satisfies

$$\max\left\{\frac{\sum w_i/t}{\sum w_i'/t'}, \frac{\sum w_i'/t'}{\sum w_i/t}\right\} \leq 2$$

Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i / t \leq \max\{k(f), l(f)\}$.

Let f : $(\mathbf{w}; t)$ and suppose that $\sum w_i / t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
  if not $\sum_V w_i = \sum_P w_i + \sum_{V \setminus P} w_i \leq (l(f) - 1)t + t \leq l(f)t$
  (contradiction)
- Every pair of maxterms P_1, P_2 satisfies $|P_1 \cap P_2| = l(f) - 1$
  if not $\sum_V w_i \leq \sum_{P_1 \cap P_2} w_i + \sum_{V \setminus P_1} w_i + \sum_{V \setminus P_2} w_i \leq (l(f) - 2)t + 2t = l(f)t$
  (contradiction)
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$ for some $x \in P, y \notin P$.

Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let f : $(\mathbf{w}; t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
  if not $\sum_V w_i = \sum_P w_i + \sum_{V \setminus P} w_i \leq (l(f) - 1)t + t \leq l(f)t$ (contradiction)
- Every pair of maxterms P_1, P_2 satisfies $|P_1 \cap P_2| = l(f) - 1$
  if not $\sum_V w_i \leq \sum_{P_1 \cap P_2} w_i + \sum_{V \setminus P_1} w_i + \sum_{V \setminus P_2} w_i \leq (l(f) - 2)t + 2t = l(f)t$ (contradiction)
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$ for some $x \in P, y \notin P$.

Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let f : $(\mathbf{w}; t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
  if not $\sum_V w_i = \sum_P w_i + \sum_{V \setminus P} w_i \leq (l(f) - 1)t + t \leq l(f)t$
  (contradiction)
- Every pair of maxterms P_1, P_2 satisfies $|P_1 \cap P_2| = l(f) - 1$
  if not $\sum_V w_i \leq \sum_{P_1 \cap P_2} w_i + \sum_{V \setminus P_1} w_i + \sum_{V \setminus P_2} w_i \leq (l(f) - 2)t + 2t = l(f)t$
  (contradiction)
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$ for some $x \in P, y \notin P$.

Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let $f : (\mathbf{w}; t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
  if not $\sum_V w_i = \sum_P w_i + \sum_{V \setminus P} w_i \leq (l(f) - 1)t + t \leq l(f)t$
  (contradiction)
- Every pair of maxterms P_1, P_2 satisfies $|P_1 \cap P_2| = l(f) - 1$
  if not $\sum_V w_i \leq \sum_{P_1 \cap P_2} w_i + \sum_{V \setminus P_1} w_i + \sum_{V \setminus P_2} w_i \leq (l(f) - 2)t + 2t = l(f)t$
  (contradiction)
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$ for some $x \in P, y \notin P$.

Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let f : $(\mathbf{w}; t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
  if not $\sum_V w_i = \sum_P w_i + \sum_{V\setminus P} w_i \leq (l(f)-1)t + t \leq l(f)t$
  (contradiction)
- Every pair of maxterms P_1, P_2 satisfies $|P_1 \cap P_2| = l(f) - 1$
  if not $\sum_V w_i \leq \sum_{P_1\cap P_2} w_i + \sum_{V\setminus P_1} w_i + \sum_{V\setminus P_2} w_i \leq (l(f)-2)t + 2t = l(f)t$
  (contradiction)
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$ for some $x \in P, y \notin P$.

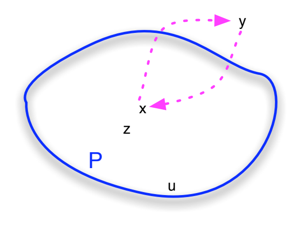
Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let $f : (\mathbf{w}; t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
- Every two maxterms P_1, P_2 satisfy $|P_1 \cap P_2| = l(f) - 1$
- Fix a maxterm P. Then $\forall P'$ (maxterm) $P' = P \setminus \{x\} \cup \{y\}$

for some $x \in P, y \notin P$.



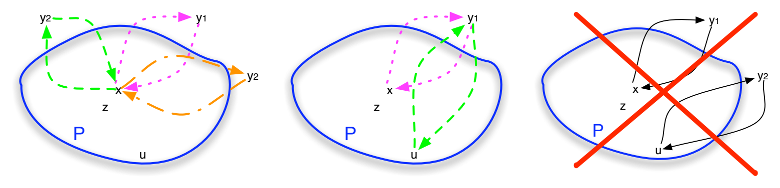
Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, *there exists sep. str.* $(\mathbf{w};t)$ *s.t.* $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Let f : $(\mathbf{w};t)$ and suppose that $\sum w_i/t > \max\{k(f), l(f)\}$

- Every maxterm has size $l(f) \geq k(f)$
- Every maxterms P_1, P_2 satisfy $|P_1 \cap P_2| = l(f) - 1$
- Fix a maxterm P. There are two possible cases:



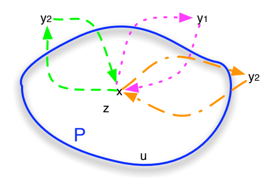
Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t.
$\sum w_i/t \leq \max\{k(f), l(f)\}$.

Fix a maxterm P, we have two cases

- 1. One variable of P is substitutable



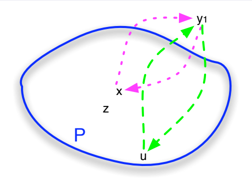
Separating Structure optimal for the LP(f)

Theorem

For any thr. func. f, there exists sep. str. $(\mathbf{w}; t)$ s.t. $\sum w_i/t \leq \max\{k(f), l(f)\}$.

Fix a maxterm P, we have two cases

- Case 2: Only one variable outside P



Linear algorithm for the opt. Separating Structure

f: given by $(\mathbf{w}; t)$

- Verify if f belongs to one of the two above cases (can be done in linear time).
- If so: construct an explicit sep. str. $(\mathbf{w}'; t')$
- If not: the given sep. str. $(\mathbf{w}; t)$ is optimal for **LP**(f).

Combined with the LPA framework:

quadratic $\gamma(f)$-competitive algorithm for threshold functions

Linear algorithm for the opt. Separating Structure

f: given by $(\mathbf{w}; t)$

- Verify if f belongs to one of the two above cases (can be done in linear time).
- If so: construct an explicit sep. str. $(\mathbf{w}'; t')$
- If not: the given sep. str. $(\mathbf{w}; t)$ is optimal for **LP**(f).

Combined with the LPA framework:

quadratic $\gamma(f)$-competitive algorithm for threshold functions

Linear algorithm for the opt. Separating Structure

f: given by $(\mathbf{w}; t)$

- Verify if f belongs to one of the two above cases (can be done in linear time).
- If so: construct an explicit sep. str. $(\mathbf{w}'; t')$
- If not: the given sep. str. $(\mathbf{w}; t)$ is optimal for **LP**(f).

Combined with the LPA framework:

quadratic $\gamma(f)$-competitive algorithm for threshold functions

Linear algorithm for the opt. Separating Structure

f: given by $(\mathbf{w}; t)$

- Verify if f belongs to one of the two above cases (can be done in linear time).
- If so: construct an explicit sep. str. $(\mathbf{w}'; t')$
- If not: the given sep. str. $(\mathbf{w}; t)$ is optimal for $\mathbf{LP}(f)$.

Combined with the LPA framework:

quadratic $\gamma(f)$-competitive algorithm for threshold functions

Any separating structure guarantees $2\gamma(f)$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ *satisfies*

$$\max\left\{\frac{val(\mathbf{w}; t)}{val(\mathbf{w}'; t')}, \frac{val(\mathbf{w}'; t')}{val(\mathbf{w}; t)}\right\} \leq 2$$

$$\tau^*(\mathcal{H}) \leq val(\mathbf{w}; t) \leq \chi^*(\mathcal{H}) \leq \chi(\mathcal{H}) \leq 2\tau^*(\mathcal{H})$$

This bound is sharp:

- $(t, t; t+1)$ for $t \geq 1$ all define the same f
- for $t = 1$, we get $val(\mathbf{w}; t) = 1$
- $val(\mathbf{w}; t) \to 2$ as $t \to \infty$

Any separating structure guarantees $2\gamma(f)$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ *satisfies*

$$\max\left\{\frac{val(\mathbf{w}; t)}{val(\mathbf{w}'; t')}, \frac{val(\mathbf{w}'; t')}{val(\mathbf{w}; t)}\right\} \le 2$$

$$\tau^*(\mathcal{H}) \le val(\mathbf{w}; t) \le \chi^*(\mathcal{H}) \le \chi(\mathcal{H}) \le 2\tau^*(\mathcal{H})$$

This bound is sharp:

- $(t, t; t+1)$ for $t \ge 1$ all define the same f
- for $t = 1$, we get $val(\mathbf{w}; t) = 1$
- $val(\mathbf{w}; t) \to 2$ as $t \to \infty$

Theorem

Every pair of separating structures $(\mathbf{w};t)$, $(\mathbf{w}';t')$ *satisfies*

$$\max\left\{\frac{val(\mathbf{w};t)}{val(\mathbf{w}';t')},\frac{val(\mathbf{w}';t')}{val(\mathbf{w};t)}\right\}\le 2$$

$$\tau^*(\mathcal{H})\le val(\mathbf{w};t)\le\chi^*(\mathcal{H})\le\chi(\mathcal{H})\le 2\tau^*(\mathcal{H})$$

This bound is sharp:

- $(t,t;t+1)$ for $t\ge 1$ all define the same f
- for $t=1$, we get $val(\mathbf{w};t)=1$
- $val(\mathbf{w};t)\to 2$ as $t\to\infty$

Any separating structure guarantees $2\gamma(f)$

Theorem

Every pair of separating structures $(\mathbf{w}; t)$, $(\mathbf{w}'; t')$ *satisfies*

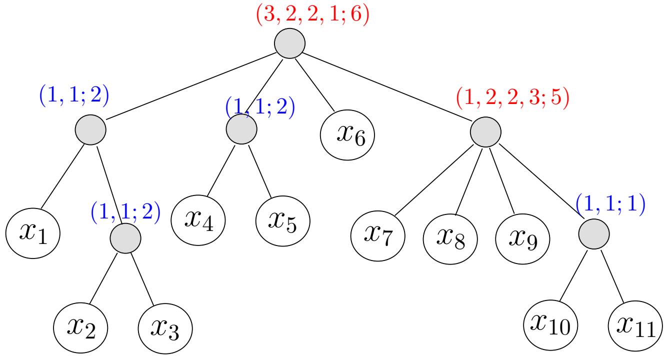
$$\max\left\{\frac{val(\mathbf{w}; t)}{val(\mathbf{w}'; t')}, \frac{val(\mathbf{w}'; t')}{val(\mathbf{w}; t)}\right\} \leq 2$$

$$\tau^*(\mathcal{H}) \leq val(\mathbf{w}; t) \leq \chi^*(\mathcal{H}) \leq \chi(\mathcal{H}) \leq 2\tau^*(\mathcal{H})$$

This bound is sharp:

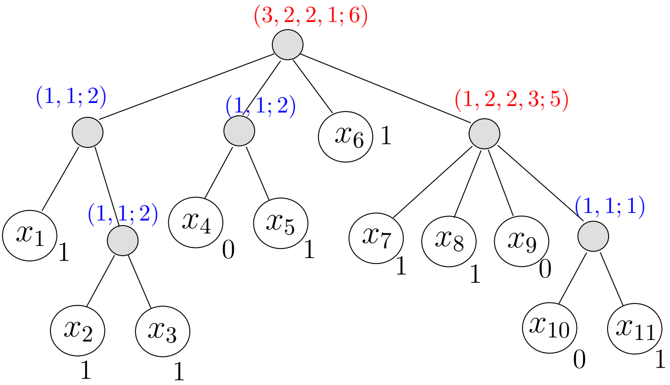
- $(t, t; t+1)$ for $t \geq 1$ all define the same f
- for $t = 1$, we get $val(\mathbf{w}; t) = 1$
- $val(\mathbf{w}; t) \to 2$ as $t \to \infty$

Extended threshold tree functions



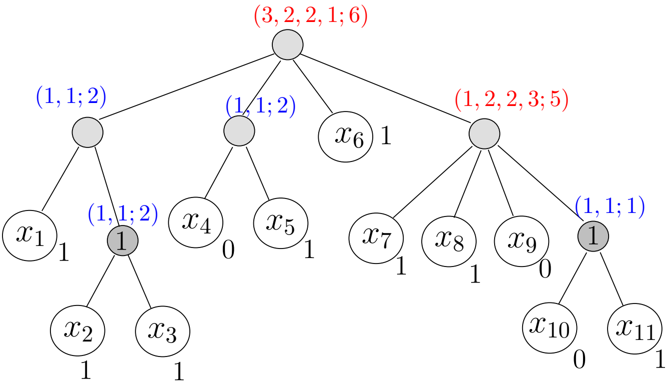
Threshold tree functions: $\mathbf{w} = (1, \ldots, 1)$ at every node
Extended threshold tree functions: no such restriction

Extended threshold tree functions



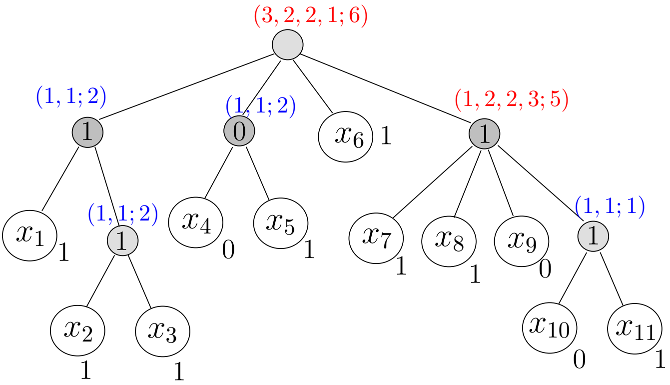
Threshold tree functions: $\mathbf{w} = (1, \ldots, 1)$ at every node
Extended threshold tree functions: no such restriction

Extended threshold tree functions



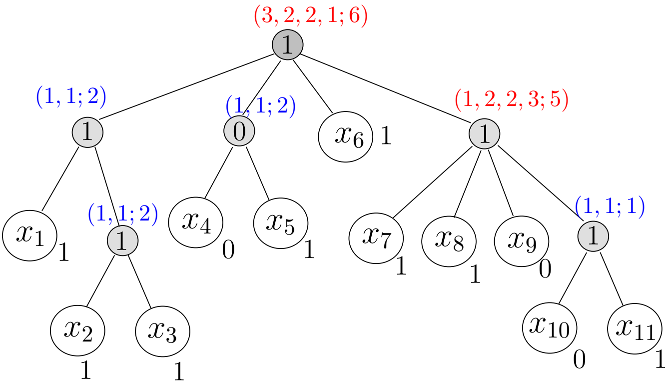
Threshold tree functions: $\mathbf{w} = (1, \ldots, 1)$ at every node
Extended threshold tree functions: no such restriction

Extended threshold tree functions



Threshold tree functions: $\mathbf{w} = (1, \ldots, 1)$ at every node
Extended threshold tree functions: no such restriction

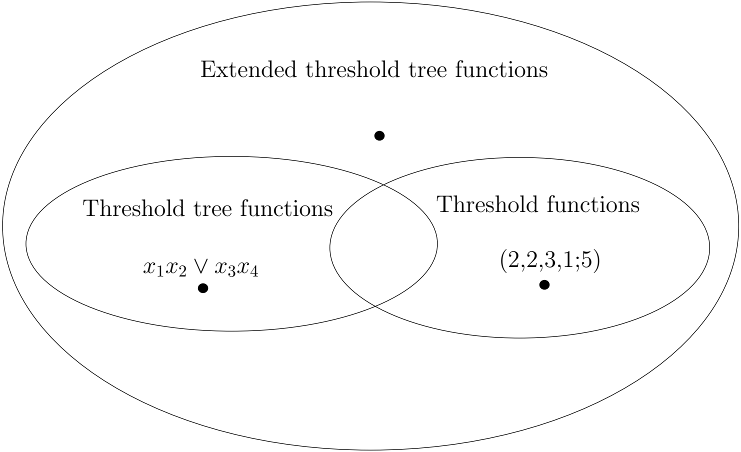
Extended threshold tree functions



Threshold tree functions: **w** $= (1, \ldots, 1)$ at every node
Extended threshold tree functions: no such restriction

Relation to threshold functions and threshold tree functions

This class properly contains the classes of threshold functions and threshold tree functions:



Pseudo-polynomial algorithm based on LPA

LPA(*f*: **function)**

While the value of f is unknown

Select a feasible solution $s()$ of **LP**(f)

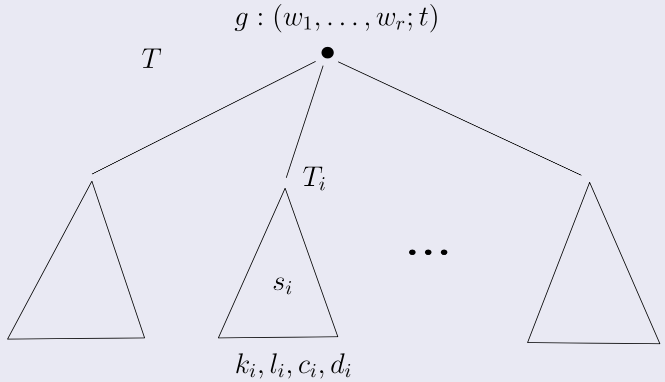
Read the variable u which minimizes $c(x)/s(x)$
(cost/impact)

$c(x) = c(x) - s(x)c(u)/s(u)$

$f \leftarrow$ restriction of f after reading u

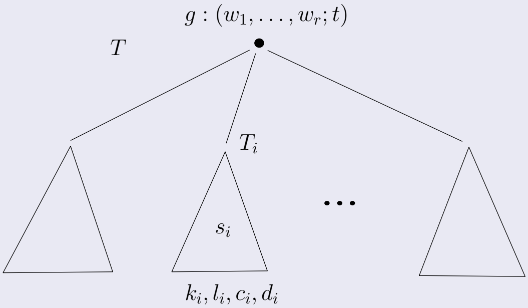
End While

Constructing s



- s_i: the solution to the LP_{T_i}
- k_i, l_i: maximum size of a minterm (maxterm) in T_i
- $c_i, d_i \geq 1$

Constructing s



$$s(x) = \frac{z_i \cdot s_i(x)}{\delta}$$

$\delta > 0$: scaling factor
z: a nontrivial solution of HLP_g

Constructing s (main steps)

1. compute $k(T)$ and $l(T)$: dynamic programming, using $(\mathbf{w}; t)$, the k_i's and the the l_i's

2. compute a solution $\mathbf{z} \neq 0$ to the following system HLP_g:

HLP_g

$$\begin{array}{rcll} \sum c_i k_i z_i & \leq & k(T) \cdot \sum_P c_i z_i & \forall \text{maxterm } P \text{ of } g, \\ \sum d_i l_i z_i & \leq & l(T) \cdot \sum_P d_i z_i & \forall \text{minterm } P \text{ of } g, \\ z_i & \geq & 0 & \forall\, i = 1, \dots, r \end{array}$$

- dynamic prog. algorithm for the separation problem (linearly many knapsack problems)

3. compute $\mathbf{s}$:

$$s(x) = \frac{z_i \cdot s_i(x)}{\delta}$$

Constructing s (main steps)

1. compute $k(T)$ and $l(T)$: dynamic programming, using $(\mathbf{w}; t)$, the k_i's and the the l_i's
2. compute a solution $\mathbf{z} \neq 0$ to the following system HLP_g:

HLP_g

$$\begin{array}{rcll} \sum c_i k_i z_i & \leq & k(T) \cdot \sum_P c_i z_i & \forall \text{maxterm } P \text{ of } g, \\ \sum d_i l_i z_i & \leq & l(T) \cdot \sum_P d_i z_i & \forall \text{minterm } P \text{ of } g, \\ z_i & \geq & 0 & \forall\, i = 1, \ldots, r \end{array}$$

- dynamic prog. algorithm for the separation problem (linearly many knapsack problems)

3. compute **s**:

$$s(x) = \frac{z_i \cdot s_i(x)}{\delta}$$

Constructing s (main steps)

1. compute $k(T)$ and $l(T)$: dynamic programming, using $(\mathbf{w}; t)$, the k_i's and the the l_i's
2. compute a solution $\mathbf{z} \neq 0$ to the following system HLP_g:

HLP_g

$$
\begin{array}{rcll}
\sum c_i k_i z_i & \leq & k(T) \cdot \sum_P c_i z_i & \forall \text{maxterm } P \text{ of } g, \\
\sum d_i l_i z_i & \leq & l(T) \cdot \sum_P d_i z_i & \forall \text{minterm } P \text{ of } g, \\
z_i & \geq & 0 & \forall\, i = 1, \ldots, r
\end{array}
$$

- dynamic prog. algorithm for the separation problem (linearly many knapsack problems)

3. compute $\mathbf{s}$:

$$s(x) = \frac{z_i \cdot s_i(x)}{\delta}$$

Generalization

The algorithm can be generalized to work for any tree function whose node connectives are monotone Boolean functions.

$\mathcal{G}$: a class of functions closed under restrictions

Suppose that all functions at the nodes belong to $\mathcal{G}$.

The complexity of the algo depends on the complexity of:

- Finding a cheapest minterm (maxterm) of a given $g \in \mathcal{G}$.
- Computing a restriction of a given $g \in \mathcal{G}$.
- Testing whether $g \equiv 0$ ($g \equiv 1$) for a given $g \in \mathcal{G}$.

Generalization

The algorithm can be generalized to work for any tree function whose node connectives are monotone Boolean functions.

$\mathcal{G}$: a class of functions closed under restrictions

Suppose that all functions at the nodes belong to $\mathcal{G}$.

The complexity of the algo depends on the complexity of:

- Finding a cheapest minterm (maxterm) of a given $g \in \mathcal{G}$.
- Computing a restriction of a given $g \in \mathcal{G}$.
- Testing whether $g \equiv 0$ ($g \equiv 1$) for a given $g \in \mathcal{G}$.

Generalization

The algorithm can be generalized to work for any tree function whose node connectives are monotone Boolean functions.

$\mathcal{G}$: a class of functions closed under restrictions

Suppose that all functions at the nodes belong to $\mathcal{G}$.

The complexity of the algo depends on the complexity of:

- Finding a cheapest minterm (maxterm) of a given $g \in \mathcal{G}$.
- Computing a restriction of a given $g \in \mathcal{G}$.
- Testing whether $g \equiv 0$ ($g \equiv 1$) for a given $g \in \mathcal{G}$.

The HLP_f for studying function evaluation

f: a monotone Boolean function over $\{x_1, \ldots, x_n\}$
Consider the homogeneous linear system HLP_f:

HLP_f

$$\begin{array}{rcll} \sum z_i & \leq & k(f) \cdot \sum_P z_i & \forall \text{maxterm } P \text{ of } f, \\ \sum z_i & \leq & l(f) \cdot \sum_P z_i & \forall \text{minterm } P \text{ of } f, \\ z_i & \geq & 0 & \forall\, i = 1, \ldots, n \end{array}$$

always has a nontrivial solution

Corollary:

For every monotone Boolean function f:

- There exists a $\gamma(f)$-competitive implementation of the $\mathcal{LPA}$.
- $\gamma(f) = \max\{k(f), l(f)\}$.

The HLP_f for studying function evaluation

f: a monotone Boolean function over $\{x_1, \ldots, x_n\}$
Consider the homogeneous linear system HLP_f:

HLP_f

$$
\begin{array}{rcll}
\sum z_i & \leq & k(f) \cdot \sum_P z_i & \forall \text{maxterm } P \text{ of } f, \\
\sum z_i & \leq & l(f) \cdot \sum_P z_i & \forall \text{minterm } P \text{ of } f, \\
z_i & \geq & 0 & \forall\, i = 1, \ldots, n
\end{array}
$$

always has a nontrivial solution

Corollary:

For every monotone Boolean function f:

- There exists a $\gamma(f)$-competitive implementation of the $\mathcal{LPA}$.
- $\gamma(f) = \max\{k(f), l(f)\}$.

The HLP_f for studying function evaluation

f: a monotone Boolean function over $\{x_1, \ldots, x_n\}$
Consider the homogeneous linear system HLP_f:

HLP_f

$$\begin{array}{rcll} \sum z_i & \leq & k(f) \cdot \sum_P z_i & \forall \text{maxterm } P \text{ of } f, \\ \sum z_i & \leq & l(f) \cdot \sum_P z_i & \forall \text{minterm } P \text{ of } f, \\ z_i & \geq & 0 & \forall\, i = 1, \ldots, n \end{array}$$

always has a nontrivial solution

Corollary:

For every monotone Boolean function f:

- There exists a $\gamma(f)$-competitive implementation of the $\mathcal{LPA}$.
- $\gamma(f) = \max\{k(f), l(f)\}$.

Some Open Questions

- Can threshold functions be optimally evaluated in linear time?
- Is the extremal competitive ratio always integer?
- Find the extremal comp. ratio of general Boolean functions.
- Is there a polynomial algorithm with optimal extremal comp. ratio for evaluating monotone Boolean functions (given by an oracle/by the list of minterms)?

Some Open Questions

- Can threshold functions be optimally evaluated in linear time?
- Is the extremal competitive ratio always integer?
- Find the extremal comp. ratio of general Boolean functions.
- Is there a polynomial algorithm with optimal extremal comp. ratio for evaluating monotone Boolean functions (given by an oracle/by the list of minterms)?

Some Open Questions

- Can threshold functions be optimally evaluated in linear time?
- Is the extremal competitive ratio always integer?
- Find the extremal comp. ratio of general Boolean functions.
- Is there a polynomial algorithm with optimal extremal comp. ratio for evaluating monotone Boolean functions (given by an oracle/by the list of minterms)?

Some Open Questions

- Can threshold functions be optimally evaluated in linear time?
- Is the extremal competitive ratio always integer?
- Find the extremal comp. ratio of general Boolean functions.
- Is there a polynomial algorithm with optimal extremal comp. ratio for evaluating monotone Boolean functions (given by an oracle/by the list of minterms)?

The end

THANK YOU