CMPSCI 711: “Really Advanced Algorithms”

Lecture 6 & 7 – Wiring and Maximum Satisfiability

Andrew McGregor

Last Compiled: February 17, 2009

Outline

Wiring Problem

Maximum Satisfiability

Puzzle

Global Wiring in Gate-Arrays

- $\sqrt{n} \times \sqrt{n}$ array of adjacent square gates.
- t pairs of terminals need connected.
- Add a wire to connect each pair.
- Constraint: Wires only travel horizontally and/or vertically and may only "bend" once.
- Problem: Want to minimize the maximum number of wires that cross the border between any two adjacent gates.

Let's see a picture...

- How many borders?
- How many possible wirings?

Reformulate...

Define

$$x_{i0} = \begin{cases} 1 & \text{if wire } i \text{ is first routed horizontally} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{i1} = \begin{cases} 1 & \text{if wire } i \text{ is first routed vertically} \\ 0 & \text{otherwise} \end{cases}$$

$$T_{b0} = \{i : \text{ wire } i \text{ is routed through border } b \text{ if } x_{i0} = 1\}$$

$$T_{b1} = \{i : \text{ wire } i \text{ is routed through border } b \text{ if } x_{i1} = 1\}$$

Reformulate...

Define

$$x_{i0} = \begin{cases} 1 & \text{if wire } i \text{ is first routed horizontally} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{i1} = \begin{cases} 1 & \text{if wire } i \text{ is first routed vertically} \\ 0 & \text{otherwise} \end{cases}$$

$$T_{b0} = \{i : \text{ wire } i \text{ is routed through border } b \text{ if } x_{i0} = 1\}$$

$$T_{b1} = \{i : \text{ wire } i \text{ is routed through border } b \text{ if } x_{i1} = 1\}$$

We want to minimize:

$$\max_b \left(\sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} \right)$$

Formulate as 0/1 Linear Program

$$
\begin{aligned}
\text{minimize} \quad & w \\
\text{where} \quad & x_{i0}, x_{i1} \in \{0,1\} \text{ for all } i \in [t] \\
\text{subject to} \quad & x_{i0} + x_{i1} = 1 \text{ for all } i \in [t] \\
& \sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} \leq w \text{ for all } b
\end{aligned}
$$

Formulate as 0/1 Linear Program

$$\begin{aligned}
\text{minimize} \quad & w \\
\text{where} \quad & x_{i0}, x_{i1} \in \{0,1\} \text{ for all } i \in [t] \\
\text{subject to} \quad & x_{i0} + x_{i1} = 1 \text{ for all } i \in [t] \\
& \sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} \leq w \text{ for all } b
\end{aligned}$$

- Unfortunately 0/1 linear programming is NP hard.

Formulate as 0/1 Linear Program

$$\begin{aligned} \text{minimize} \quad & w \\ \text{where} \quad & x_{i0}, x_{i1} \in \{0,1\} \text{ for all } i \in [t] \\ \text{subject to} \quad & x_{i0} + x_{i1} = 1 \text{ for all } i \in [t] \\ & \sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} \leq w \text{ for all } b \end{aligned}$$

- Unfortunately 0/1 linear programming is NP hard.
- One option is to relax the program to get a linear program:

$$\text{replace } x_{i0}, x_{i1} \in \{0,1\} \text{ by } 0 \leq x_{i0}, x_{i1} \leq 1$$

Formulate as 0/1 Linear Program

$$
\begin{aligned}
\text{minimize} \quad & w \\
\text{where} \quad & x_{i0}, x_{i1} \in \{0,1\} \text{ for all } i \in [t] \\
\text{subject to} \quad & x_{i0} + x_{i1} = 1 \text{ for all } i \in [t] \\
& \sum_{i \in T_{b0}} x_{i0} + \sum_{i \in T_{b1}} x_{i1} \leq w \text{ for all } b
\end{aligned}
$$

- Unfortunately 0/1 linear programming is NP hard.
- One option is to relax the program to get a linear program:

  replace $x_{i0}, x_{i1} \in \{0,1\}$ by $0 \leq x_{i0}, x_{i1} \leq 1$

- Can efficiently find a solution to the relaxed program but how can we use it to get a solution to the original problem?

A factor 2 approximation

- Let $\hat{w}$ be optimum solution to the relaxed program and let w_0 be optimum solution to the original program:

$$\hat{w} \leq w_0$$

A factor 2 approximation

- Let $\hat{w}$ be optimum solution to the relaxed program and let w_0 be optimum solution to the original program:

$$\hat{w} \leq w_0$$

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}$ be solution to relaxed program:

$$\max(\hat{x}_{i0}, \hat{x}_{i1}) \geq 1/2$$

A factor 2 approximation

- Let $\hat{w}$ be optimum solution to the relaxed program and let w_0 be optimum solution to the original program:

$$\hat{w} \leq w_0$$

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}$ be solution to relaxed program:

$$\max(\hat{x}_{i0}, \hat{x}_{i1}) \geq 1/2$$

- Round the larger up to 1 and smaller down to 0.

A factor 2 approximation

- Let $\hat{w}$ be optimum solution to the relaxed program and let w_0 be optimum solution to the original program:

$$\hat{w} \leq w_0$$

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}$ be solution to relaxed program:

$$\max(\hat{x}_{i0}, \hat{x}_{i1}) \geq 1/2$$

- Round the larger up to 1 and smaller down to 0.
- Let $\tilde{x}_{i0}, \tilde{x}_{i1}$ be the new values.

A factor 2 approximation

- Let $\hat{w}$ be optimum solution to the relaxed program and let w_0 be optimum solution to the original program:

$$\hat{w} \leq w_0$$

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}$ be solution to relaxed program:

$$\max(\hat{x}_{i0}, \hat{x}_{i1}) \geq 1/2$$

- Round the larger up to 1 and smaller down to 0.
- Let $\tilde{x}_{i0}, \tilde{x}_{i1}$ be the new values.
- Then, for all borders b:

$$\sum_{i \in T_{b0}} \tilde{x}_{i0} + \sum_{i \in T_{b1}} \tilde{x}_{i1} \leq 2 \left(\sum_{i \in T_{b0}} \hat{x}_{i0} + \sum_{i \in T_{b1}} \hat{x}_{i1} \right) \leq 2\hat{w} \leq 2w_0$$

Randomized Rounding (1/2)

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}, \hat{w}$ be solution to relaxed program.

Randomized Rounding (1/2)

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}, \hat{w}$ be solution to relaxed program.
- With prob. $\hat{x}_{i0}$, let $\tilde{x}_{i0} = 1$ & $\tilde{x}_{i1} = 0$. Otherwise, set $\tilde{x}_{i0} = 0$ & $\tilde{x}_{i1} = 1$

Randomized Rounding (1/2)

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}, \hat{w}$ be solution to relaxed program.
- With prob. $\hat{x}_{i0}$, let $\tilde{x}_{i0} = 1$ & $\tilde{x}_{i1} = 0$. Otherwise, set $\tilde{x}_{i0} = 0$ & $\tilde{x}_{i1} = 1$
- For any border b, let

$$\tilde{w}(b) = \sum_{i \in T_{b0}} \tilde{x}_{i0} + \sum_{i \in T_{b1}} \tilde{x}_{i1}$$

Randomized Rounding (1/2)

- For $i \in [t]$, let $\hat{x}_{i0}, \hat{x}_{i1}, \hat{w}$ be solution to relaxed program.
- With prob. $\hat{x}_{i0}$, let $\tilde{x}_{i0} = 1$ & $\tilde{x}_{i1} = 0$. Otherwise, set $\tilde{x}_{i0} = 0$ & $\tilde{x}_{i1} = 1$
- For any border b, let

$$\tilde{w}(b) = \sum_{i \in T_{b0}} \tilde{x}_{i0} + \sum_{i \in T_{b1}} \tilde{x}_{i1}$$

- Then,

$$\mathbb{E}\left[\tilde{w}(b)\right] = \sum_{i \in T_{b0}} \hat{x}_{i0} + \sum_{i \in T_{b1}} \hat{x}_{i1} \leq \hat{w}$$

Randomized Rounding (2/2)

- Note that $\tilde{w}(b)$ is the sum of independent poisson trials.

Randomized Rounding (2/2)

- Note that $\tilde{w}(b)$ is the sum of independent poisson trials.
- Using the Chernoff bound and $\mathbb{E}\left[\tilde{w}(b)\right] \leq w_0$:

$$\mathbb{P}\left[\tilde{w}(b) \geq (1+\delta)w_0\right] \leq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{w_0}$$

Randomized Rounding (2/2)

- Note that $\tilde{w}(b)$ is the sum of independent poisson trials.
- Using the Chernoff bound and $\mathbb{E}\left[\tilde{w}(b)\right] \leq w_0$:

$$\mathbb{P}\left[\tilde{w}(b) \geq (1+\delta)w_0\right] \leq \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{w_0}$$

- Apply the union bound:

$$\mathbb{P}\left[\forall b : \tilde{w}(b) \geq (1+\delta)w_0\right] < 2n\left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{w_0}$$

Randomized Rounding (2/2)

- Note that $\tilde{w}(b)$ is the sum of independent poisson trials.
- Using the Chernoff bound and $\mathbb{E}[\tilde{w}(b)] \leq w_0$:

$$\mathbb{P}[\tilde{w}(b) \geq (1+\delta)w_0] \leq \left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^{w_0}$$

- Apply the union bound:

$$\mathbb{P}[\forall b : \tilde{w}(b) \geq (1+\delta)w_0] < 2n\left[\frac{e^\delta}{(1+\delta)^{1+\delta}}\right]^{w_0}$$

- With probability at least $1-\epsilon$

$$\max_b(\tilde{w}(b)) \leq \left(1+\sqrt{\frac{4\ln(2n/\epsilon)}{w_0}}\right)w_0$$

if $w_0 \geq \omega(\log(n/\epsilon))$

Outline

Maximum Satisfiability

- Input: m clauses in n Boolean variables $x_1, \ldots, x_n$, e.g.,

$$x_1 \vee x_2 \vee \bar{x}_3 \ , \quad \bar{x}_1 \vee \bar{x}_4 \ , \quad \ldots \ , \quad x_9 \vee x_{10} \vee x_{21}$$

  where $\bar{x}_i = 1 - x_i$.
- Goal: Maximize the number of clauses that are satisfied.

Maximum Satisfiability

- Input: m clauses in n Boolean variables $x_1, \ldots, x_n$, e.g.,

$$x_1 \vee x_2 \vee \bar{x}_3 \ , \quad \bar{x}_1 \vee \bar{x}_4 \ , \quad \ldots \ , \quad x_9 \vee x_{10} \vee x_{21}$$

  where $\bar{x}_i = 1 - x_i$.
- Goal: Maximize the number of clauses that are satisfied.

Notes:

- Problem is NP-hard so will consider approximation algorithms.

Maximum Satisfiability

- Input: m clauses in n Boolean variables $x_1, \ldots, x_n$, e.g.,

$$x_1 \vee x_2 \vee \bar{x}_3 \ , \quad \bar{x}_1 \vee \bar{x}_4 \ , \quad \ldots \ , \quad x_9 \vee x_{10} \vee x_{21}$$

  where $\bar{x}_i = 1 - x_i$.
- Goal: Maximize the number of clauses that are satisfied.

Notes:

- Problem is NP-hard so will consider approximation algorithms.
- A randomized algorithm is an α-approximations for a maximization problem if it returns X with $\mathbb{E}[X]/\text{OPT} \geq \alpha$.

Maximum Satisfiability

- Input: m clauses in n Boolean variables $x_1, \ldots, x_n$, e.g.,

$$x_1 \vee x_2 \vee \bar{x}_3 \ , \quad \bar{x}_1 \vee \bar{x}_4 \ , \quad \ldots \ , \quad x_9 \vee x_{10} \vee x_{21}$$

  where $\bar{x}_i = 1 - x_i$.
- Goal: Maximize the number of clauses that are satisfied.

Notes:

- Problem is NP-hard so will consider approximation algorithms.
- A randomized algorithm is an α-approximations for a maximization problem if it returns X with $\mathbb{E}\left[X\right]/\text{OPT} \geq \alpha$.
- Call x_i and $\bar{x}_i$ *literals*. Assume x_i and $\bar{x}_i$ don't appear in same clause.

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Proof.

- Independently set each x_i to 0 with prob. 1/2 and 1 otherwise.

□

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Proof.

- Independently set each x_i to 0 with prob. $1/2$ and 1 otherwise.
- Let $Z_i = 1$ if i-th clause is satisfied and 0 otherwise.

□

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Proof.

- Independently set each x_i to 0 with prob. $1/2$ and 1 otherwise.
- Let $Z_i = 1$ if i-th clause is satisfied and 0 otherwise.
- If i-th clause has k literals,

$$\mathbb{P}[Z_i = 1] = 1 - 1/2^k \geq 1/2$$

□

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Proof.

- Independently set each x_i to 0 with prob. $1/2$ and 1 otherwise.
- Let $Z_i = 1$ if i-th clause is satisfied and 0 otherwise.
- If i-th clause has k literals,

$$\mathbb{P}\left[Z_i = 1\right] = 1 - 1/2^k \geq 1/2$$

- Expected number of clauses satisfied is $\mathbb{E}\left[\sum_{i \in [m]} Z_i\right] \geq m/2$

□

Easy 1/2-approximation

Theorem

A truth assignment chosen uniformly at random is a 1/2 approx.

Proof.

- Independently set each x_i to 0 with prob. 1/2 and 1 otherwise.
- Let $Z_i = 1$ if i-th clause is satisfied and 0 otherwise.
- If i-th clause has k literals,

$$\mathbb{P}\left[Z_i = 1\right] = 1 - 1/2^k \geq 1/2$$

- Expected number of clauses satisfied is $\mathbb{E}\left[\sum_{i \in [m]} Z_i\right] \geq m/2$

□

Example of the Probabilistic Method: Above prove shows that there always exists a way to satisfy $m/2$ clauses. This is tight.

Formulate as Linear Program

Define the following sets:

$$
\begin{aligned}
C_j &= \{i : \; x_i \text{ or } \bar{x}_i \text{ appears in } j\text{-th clause}\} \\
C_j^+ &= \{i : \; x_i \text{ appears in } j\text{-th clause}\} \\
C_j^- &= \{i : \; \bar{x}_i \text{ appears in } j\text{-th clause}\}
\end{aligned}
$$

Formulate as Linear Program

Define the following sets:

$$\begin{aligned} C_j &= \{i : \ x_i \text{ or } \bar{x}_i \text{ appears in } j\text{-th clause}\} \\ C_j^+ &= \{i : \ x_i \text{ appears in } j\text{-th clause}\} \\ C_j^- &= \{i : \ \bar{x}_i \text{ appears in } j\text{-th clause}\} \end{aligned}$$

Consider the following program:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{m} z_j \\ \text{where} \quad & x_i, z_j \in \{0,1\} \text{ for all } i \in [n], j \in [m] \\ \text{subject to} \quad & \sum_{i \in C_j^+} x_i + \sum_{i \in C_j^-} (1 - x_i) \geq z_j \text{ for all } j \in [m] \end{aligned}$$

Formulate as Linear Program

Define the following sets:

$$\begin{aligned} C_j &= \{i: \; x_i \text{ or } \bar{x}_i \text{ appears in } j\text{-th clause}\} \\ C_j^+ &= \{i: \; x_i \text{ appears in } j\text{-th clause}\} \\ C_j^- &= \{i: \; \bar{x}_i \text{ appears in } j\text{-th clause}\} \end{aligned}$$

Consider the following program:

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{m} z_j \\ \text{where} \quad & x_i, z_j \in \{0,1\} \text{ for all } i \in [n], j \in [m] \\ \text{subject to} \quad & \sum_{i \in C_j^+} x_i + \sum_{i \in C_j^-} (1 - x_i) \geq z_j \text{ for all } j \in [m] \end{aligned}$$

Relax by replacing "$x_i, z_j \in \{0,1\}$" by "$0 \leq x_i, z_j \leq 1$"

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.
- With probability $\hat{x}_i$, set $x_i = 1$ and $x_i = 0$ otherwise.

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.
- With probability $\hat{x}_i$, set $x_i = 1$ and $x_i = 0$ otherwise.
- By independence,

$$\mathbb{P}\left[Z_j = 1\right] \;=\; 1 - \prod_{i \in C_j^+} (1 - \hat{x}_i) \cdot \prod_{i \in C_j^-} \hat{x}_i$$

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.
- With probability $\hat{x}_i$, set $x_i = 1$ and $x_i = 0$ otherwise.
- By independence,

$$\mathbb{P}[Z_j = 1] \;=\; 1 - \prod_{i \in C_j^+} (1 - \hat{x}_i) \cdot \prod_{i \in C_j^-} \hat{x}_i$$

- Since $\sum_{i \in C_j^+} \hat{x}_i + \sum_{i \in C_j^-} (1 - \hat{x}_i) \geq \hat{z}_j$, if C_j has k literals

$$\mathbb{P}[Z_j = 1] \geq 1 - (1 - \hat{z}_j/k)^k$$

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.
- With probability $\hat{x}_i$, set $x_i = 1$ and $x_i = 0$ otherwise.
- By independence,

$$\mathbb{P}\left[Z_j = 1\right] \;=\; 1 - \prod_{i \in C_j^+} (1 - \hat{x}_i) \cdot \prod_{i \in C_j^-} \hat{x}_i$$

- Since $\sum_{i \in C_j^+} \hat{x}_i + \sum_{i \in C_j^-} (1 - \hat{x}_i) \geq \hat{z}_j$, if C_j has k literals

$$\mathbb{P}\left[Z_j = 1\right] \geq 1 - (1 - \hat{z}_j/k)^k$$

- For $\beta_k = 1 - (1 - 1/k)^k$:

$$\mathbb{P}\left[Z_j = 1\right] \geq 1 - (1 - \hat{z}_j/k)^k \geq \beta_k \hat{z}_j \geq (1 - 1/e)\hat{z}_j$$

Randomized Rounding

- Let $\hat{x}_i, \hat{z}_i$ be solutions to the relaxed program.
- With probability $\hat{x}_i$, set $x_i = 1$ and $x_i = 0$ otherwise.
- By independence,

$$\mathbb{P}[Z_j = 1] = 1 - \prod_{i \in C_j^+} (1 - \hat{x}_i) \cdot \prod_{i \in C_j^-} \hat{x}_i$$

- Since $\sum_{i \in C_j^+} \hat{x}_i + \sum_{i \in C_j^-} (1 - \hat{x}_i) \geq \hat{z}_j$, if C_j has k literals

$$\mathbb{P}[Z_j = 1] \geq 1 - (1 - \hat{z}_j / k)^k$$

- For $\beta_k = 1 - (1 - 1/k)^k$:

$$\mathbb{P}[Z_j = 1] \geq 1 - (1 - \hat{z}_j / k)^k \geq \beta_k \hat{z}_j \geq (1 - 1/e) \hat{z}_j$$

- $\mathbb{E}\left[\sum_{j \in [m]} Z_j\right] \geq (1 - 1/e) \sum_{j \in [m]} \hat{z}_j \geq (1 - 1/e)\text{OPT}$

Recap

- First algorithm:
  - Set each variable independently to 0 or 1 with equal probability.
  - Gave $1/2$ approximation.
  - Would have been better if all clauses had many literals.

Recap

- First algorithm:
  - Set each variable independently to 0 or 1 with equal probability.
  - Gave $1/2$ approximation.
  - Would have been better if all clauses had many literals.
- Second algorithm:
  - Set each variable independently to 0 or 1 with probability based on the linear program solution.
  - Gave $(1 - 1/e)$ approximation.
  - Would have been better if all clauses had few literals.

Recap

- First algorithm:
  - Set each variable independently to 0 or 1 with equal probability.
  - Gave $1/2$ approximation.
  - Would have been better if all clauses had many literals.
- Second algorithm:
  - Set each variable independently to 0 or 1 with probability based on the linear program solution.
  - Gave $(1 - 1/e)$ approximation.
  - Would have been better if all clauses had few literals.
- What would be a good third algorithm?

$3/4$ Approximation

Theorem

Running both algorithms and returning best solution gives $3/4$ *approximation.*

3/4 Approximation

Theorem

Running both algorithms and returning best solution gives 3/4 *approximation.*

Proof.

3/4 Approximation

Theorem

Running both algorithms and returning best solution gives 3/4 *approximation.*

Proof.

- Let A and B be number of clauses satisfied by first and second algorithms: $\mathbb{E}[\max(A, B)] \geq \mathbb{E}[A]/2 + \mathbb{E}[B]/2$.

3/4 Approximation

Theorem

Running both algorithms and returning best solution gives 3/4 approximation.

Proof.

- Let A and B be number of clauses satisfied by first and second algorithms: $\mathbb{E}\left[\max(A, B)\right] \geq \mathbb{E}\left[A\right]/2 + \mathbb{E}\left[B\right]/2$.
- Let $S_k = \{j : \; j\text{-th clause contains exactly } k \text{ literals}\}$

3/4 Approximation

Theorem

Running both algorithms and returning best solution gives 3/4 *approximation.*

Proof.

- Let A and B be number of clauses satisfied by first and second algorithms: $\mathbb{E}[\max(A, B)] \geq \mathbb{E}[A]/2 + \mathbb{E}[B]/2$.
- Let $S_k = \{j : j\text{-th clause contains exactly } k \text{ literals}\}$
- From the analysis of the previous algorithms:

$$A = \sum_k \sum_{j \in S_k} (1 - 2^{-k}) \quad \text{and} \quad B \geq \sum_k \sum_{j \in S_k} \beta_k \hat{z}_j$$

3/4 Approximation

Theorem

Running both algorithms and returning best solution gives 3/4 approximation.

Proof.

- Let A and B be number of clauses satisfied by first and second algorithms: $\mathbb{E}[\max(A,B)] \geq \mathbb{E}[A]/2 + \mathbb{E}[B]/2$.
- Let $S_k = \{j : j\text{-th clause contains exactly } k \text{ literals}\}$
- From the analysis of the previous algorithms:

$$A = \sum_k \sum_{j \in S_k} (1 - 2^{-k}) \quad \text{and} \quad B \geq \sum_k \sum_{j \in S_k} \beta_k \hat{z}_j$$

- Therefore, using $1 - 2^{-k} + \beta_k \geq 3/2$ for all k:

$$\mathbb{E}[A]/2 + \mathbb{E}[B]/2 \geq \sum_k \sum_{j \in S_k} \frac{1 - 2^{-k} + \beta_k}{2} \hat{z}_j \geq 0.75\textsc{opt}$$

Outline

Puzzle

- You lose the unbiased coin with which you were planning to do the homework.
- A friend lends you a biased coin but doesn't tell you the bias.
- Without trying to estimate the bias, how can you use the biased coin to simulate a perfectly unbiased coin?