CMPSCI 711: “Really Advanced Algorithms”

Lecture 1

Andrew McGregor

January 29, 2009

Last Compiled: January 29, 2009

Good News and Bad News...

- In previous years, CMPSCI 711 has alternated between randomized algorithms and parallel algorithms.

Good News and Bad News. . .

- In previous years, CMPSCI 711 has alternated between randomized algorithms and parallel algorithms.
- Good News: This semester it's randomized algorithms!

Good News and Bad News...

- In previous years, CMPSCI 711 has alternated between randomized algorithms and parallel algorithms.
- Good News: This semester it's randomized algorithms!
- (Only bad news if you really wanted parallel algorithms.)

Outline

What's great about randomized algorithms?

- *Simplicity:* Algorithms are often simple and elegant, e.g., randomized quick-sort...

What's great about randomized algorithms?

- *Simplicity:* Algorithms are often simple and elegant, e.g., randomized quick-sort...
- *Speed:* Some randomized algorithms are faster than the best known deterministic algorithms, e.g., checking if a multivariate polynomial is the zero-polynomial...

What's great about randomized algorithms?

- *Simplicity:* Algorithms are often simple and elegant, e.g., randomized quick-sort...
- *Speed:* Some randomized algorithms are faster than the best known deterministic algorithms, e.g., checking if a multivariate polynomial is the zero-polynomial...
- *Defeating Adversaries!* Imagine playing rock, paper, scissors without randomization...

What's not so great about randomized algorithms?

- *Errors:* May return the wrong answer with small probability.

What's not so great about randomized algorithms?

- *Errors:* May return the wrong answer with small probability.
- *Running Time:* May not terminate.

What's not so great about randomized algorithms?

- *Errors:* May return the wrong answer with small probability.
- *Running Time:* May not terminate.
- *Debugging:* Bugs might be hard to reproduce.

Some recurring ideas. . .

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

Some recurring ideas. . .

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

- *Foiling an adversary*, e.g., load balancing.

Some recurring ideas...

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

- *Foiling an adversary*, e.g., load balancing.
- *Random sampling*, e.g., finding the number of ways to satisfy a formula.

Some recurring ideas...

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

- *Foiling an adversary*, e.g., load balancing.
- *Random sampling*, e.g., finding the number of ways to satisfy a formula.
- *Lots of witnesses*, e.g., if the polynomial is non-zero then it's non-zero at many points.

Some recurring ideas. . .

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

- *Foiling an adversary*, e.g., load balancing.
- *Random sampling*, e.g., finding the number of ways to satisfy a formula.
- *Lots of witnesses*, e.g., if the polynomial is non-zero then it's non-zero at many points.
- *Symmetry breaking*, e.g., when two machines start sending packets at the same time, use "exponential back-off".

Some recurring ideas. . .

- *Basic probability is very powerful*, e.g., linearity of expectation

$$\mathbb{E}[A + B] = \mathbb{E}[A] + \mathbb{E}[B]$$

- *Foiling an adversary*, e.g., load balancing.
- *Random sampling*, e.g., finding the number of ways to satisfy a formula.
- *Lots of witnesses*, e.g., if the polynomial is non-zero then it's non-zero at many points.
- *Symmetry breaking*, e.g., when two machines start sending packets at the same time, use "exponential back-off".
- *Derandomization*, e.g., how many coin flips do we need?

Outline

Introduction to Randomized Algorithms

Course Outline and Administrivia

Examples
- Min-Cut
- Max-Cut

Puzzle

Basic Stuff

Lectures are Tuesday and Thursday, 1pm to 2.15pm in CMPS 140.

Lecturer: Professor Andrew McGregor

- Email: mcgregor@cs.umass.edu
- Office: CMPS 334
- Phone: 545-6867
- Office hours: Tuesday 3:00 - 4:00, or by appointment.

Textbooks and Materials

Textbooks:

- R. Motwani and P. Raghavan, Randomized Algorithms. Cambridge University Press, 1995. (Required)
- M. Mitzenmacher and E. Upfal, Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005. (Optional)

Other materials, including lecture slides, will be posted at:

`www.cs.umass.edu/~mcgregor/courses/CS711/index.html` .

Course Outline

- First part of the course (approx. 16 lectures):
  - Selected material from first 7 chapters of [MR] including the probabilistic method; tail inequalities; algebraic methods; random walks; derandomization and limited independence.
- Second part of the course (approx. 12 lectures):
  - Applications to approximation and combinatorial optimization; online and stream computation; metric embeddings, communication theory; and other areas will be discussed as time permits.
  - Some material from later chapters of [MR] but also recent research papers.

Assessment

- *Homeworks:* Approximately 3 assignments will contribute 50% to grade. Collaboration is allowed but solutions must be written separately and mention who you worked with.

Assessment

- *Homeworks:* Approximately 3 assignments will contribute 50% to grade. Collaboration is allowed but solutions must be written separately and mention who you worked with.
- *Exams:* There will be two take home exams. No collaboration is permitted. The exams will contribute 25% to grade.

Assessment

- *Homeworks:* Approximately 3 assignments will contribute 50% to grade. Collaboration is allowed but solutions must be written separately and mention who you worked with.
- *Exams:* There will be two take home exams. No collaboration is permitted. The exams will contribute 25% to grade.
- *Participation:* Remaining 25% of the grade will be based on class participation: active engagement in class and you may be expected to present a paper in second part.

Outline

Introduction to Randomized Algorithms

Course Outline and Administrivia

Examples
- Min-Cut
- Max-Cut

Puzzle

Min-Cut Problem (1/3)

Given an unweighted, multi-graph $G = (V, E)$, we want to partition V into V_1 and V_2 such that $|E \cap (V_1 \times V_2)|$ is minimized.

Min-Cut Problem (1/3)

Given an unweighted, multi-graph $G = (V, E)$, we want to partition V into V_1 and V_2 such that $|E \cap (V_1 \times V_2)|$ is minimized.

Algorithm

- *Contract a random edge.*
- *Repeat until there are only 2 vertices remaining.*
- *Output the number of remaining edges.*

Min-Cut Problem (1/3)

Given an unweighted, multi-graph $G = (V, E)$, we want to partition V into V_1 and V_2 such that $|E \cap (V_1 \times V_2)|$ is minimized.

Algorithm

- *Contract a random edge.*
- *Repeat until there are only 2 vertices remaining.*
- *Output the number of remaining edges.*

Let $|V| = n$ and $|E| = m$.

Min-Cut Problem (2/3)

Theorem

The algorithm is correct with probability at least $2/n^2$.

Min-Cut Problem (2/3)

Theorem

The algorithm is correct with probability at least $2/n^2$.

Proof.

- Let (V_1, V_2) be a minimum cut and let $|E \cap (V_1 \times V_2)| = k$.

$\square$

Min-Cut Problem (2/3)

Theorem

The algorithm is correct with probability at least $2/n^2$.

Proof.

- Let (V_1, V_2) be a minimum cut and let $|E \cap (V_1 \times V_2)| = k$.
- Minimum cut of the graph doesn't decrease.

□

Min-Cut Problem (2/3)

Theorem

The algorithm is correct with probability at least $2/n^2$.

Proof.

- Let (V_1, V_2) be a minimum cut and let $|E \cap (V_1 \times V_2)| = k$.
- Minimum cut of the graph doesn't decrease.
- Let A_i be event that we don't contract min-cut edge at step i.

$$\Pr(\cap_{1 \leq i \leq n-2} A_i) = \Pr(A_1) \Pr(A_2 | A_1) \ldots \Pr(A_{n-2} | \cap_{1 \leq i \leq n-3} A_i)$$

□

Min-Cut Problem (2/3)

Theorem
The algorithm is correct with probability at least $2/n^2$.

Proof.

- Let (V_1, V_2) be a minimum cut and let $|E \cap (V_1 \times V_2)| = k$.
- Minimum cut of the graph doesn't decrease.
- Let A_i be event that we don't contract min-cut edge at step i.

$$\Pr(\cap_{1 \le i \le n-2} A_i) = \Pr(A_1)\Pr(A_2 | A_1) \ldots \Pr(A_{n-2} | \cap_{1 \le i \le n-3} A_i)$$

- Number of edges after i-th step is at least $(n-i)k/2$

□

Min-Cut Problem (2/3)

Theorem

The algorithm is correct with probability at least $2/n^2$.

Proof.

- Let (V_1, V_2) be a minimum cut and let $|E \cap (V_1 \times V_2)| = k$.
- Minimum cut of the graph doesn't decrease.
- Let A_i be event that we don't contract min-cut edge at step i.

$$\Pr(\cap_{1 \le i \le n-2} A_i) = \Pr(A_1) \Pr(A_2 | A_1) \dots \Pr(A_{n-2} | \cap_{1 \le i \le n-3} A_i)$$

- Number of edges after i-th step is at least $(n-i)k/2$
- $\Pr(A_i | A_1 \cap A_2 \cap \dots \cap A_{i-1}) \ge 1 - 2/(n-i)$

□

Min-Cut Problem (3/3)

Theorem

Repeating $\alpha n^2/2$ times (with new random coin flips) and returning smallest cut is correct with probability at least $1 - e^{-\alpha}$.

Min-Cut Problem (3/3)

Theorem

Repeating $\alpha n^2/2$ times (with new random coin flips) and returning smallest cut is correct with probability at least $1 - e^{-\alpha}$.

Proof.

- Because each repeat is independent,

$$\mathbb{P}\left[\text{always fails}\right] = \prod_{1 \le i \le \alpha n^2/2} \mathbb{P}\left[i\text{-th try fails}\right] \le (1 - 2/n^2)^{\alpha n^2/2}$$

- Use fact $1 - x \le e^{-x}$ for $x \ge 0$ and simplify.

□

Max-Cut Problem (1/3)

Given an unweighted, graph $G = (V, E)$, partition V into V_1 and V_2 such that $|E \cap (V_1 \times V_2)|$ is maximized.

Max-Cut Problem (1/3)

Given an unweighted, graph $G = (V, E)$, partition V into V_1 and V_2 such that $|E \cap (V_1 \times V_2)|$ is maximized.

Algorithm

- *Put each node in V_1 with probability 1/2 and in V_2 otherwise.*

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i, j) is cut and $X_{i,j} = 0$ otherwise.

□

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i,j) is cut and $X_{i,j} = 0$ otherwise.
- $X = \sum_{(i,j)\in E} X_{i,j}$ and $\mathbb{E}[X_{i,j}] = 1/2$.

□

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i, j) is cut and $X_{i,j} = 0$ otherwise.
- $X = \sum_{(i,j)\in E} X_{i,j}$ and $\mathbb{E}[X_{i,j}] = 1/2$.
- Result follows by linearity of expectation.

□

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i,j) is cut and $X_{i,j} = 0$ otherwise.
- $X = \sum_{(i,j)\in E} X_{i,j}$ and $\mathbb{E}[X_{i,j}] = 1/2$.
- Result follows by linearity of expectation.

□

Theorem

$\mathbb{P}[X \geq |E|/2] = 2/(2+m).$

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i, j) is cut and $X_{i,j} = 0$ otherwise.
- $X = \sum_{(i,j)\in E} X_{i,j}$ and $\mathbb{E}[X_{i,j}] = 1/2$.
- Result follows by linearity of expectation.

□

Theorem

$\mathbb{P}[X \geq |E|/2] = 2/(2 + m).$

Proof.

- $\mathbb{E}[X] \leq \mathbb{P}[X \leq m/2 - 1](m/2 - 1) + \mathbb{P}[X \geq m/2]\, m$

□

Max-Cut Problem (2/3)

Let X be the number of edges that are cut.

Theorem

$\mathbb{E}[X] = \sum_r r\mathbb{P}[X = r] = m/2.$

Proof.

- Let $X_{i,j} = 1$ if edge (i, j) is cut and $X_{i,j} = 0$ otherwise.
- $X = \sum_{(i,j)\in E} X_{i,j}$ and $\mathbb{E}[X_{i,j}] = 1/2$.
- Result follows by linearity of expectation.

□

Theorem

$\mathbb{P}[X \geq |E|/2] = 2/(2 + m).$

Proof.

- $\mathbb{E}[X] \leq \mathbb{P}[X \leq m/2 - 1](m/2 - 1) + \mathbb{P}[X \geq m/2]\, m$
- Substitute $\mathbb{E}[X] = m/2$ and solve.

□

Max-Cut Problem (3/3)

Theorem

Repeat the algorithm $\alpha(m+2)/2$ times. The largest cut observed is a $1/2$ approximation of MaxCut with probability at least $1 - e^{-\alpha}$.

Max-Cut Problem (3/3)

Theorem

Repeat the algorithm $\alpha(m+2)/2$ times. The largest cut observed is a $1/2$ approximation of MaxCut with probability at least $1 - e^{-\alpha}$.

Proof.

- MaxCut(G) is at most m so $m/2$ edges is a $1/2$ approx.

□

Max-Cut Problem (3/3)

Theorem

Repeat the algorithm $\alpha(m+2)/2$ times. The largest cut observed is a $1/2$ approximation of MaxCut with probability at least $1 - e^{-\alpha}$.

Proof.

- MaxCut(G) is at most m so $m/2$ edges is a $1/2$ approx.
- Boost the probability from $2/(m+2)$ to $1 - e^{-\alpha}$ as before.

□

Max-Cut Problem (3/3)

Theorem

Repeat the algorithm $\alpha(m+2)/2$ *times. The largest cut observed is a* $1/2$ *approximation of MaxCut with probability at least* $1 - e^{-\alpha}$.

Proof.

- MaxCut(G) is at most m so $m/2$ edges is a $1/2$ approx.
- Boost the probability from $2/(m+2)$ to $1 - e^{-\alpha}$ as before.

□

Best result that's possible is .878 approximation.

Outline

A Probability Puzzle...

- One hundred people are boarding a hundred seater plane and have assigned seats.
- Unfortunately the first passenger who boards has lost his boarding pass and just takes a seat at random. Subsequent passengers take their assigned seat if they are available, or take an unoccupied seat at random otherwise.
- What's the probability that the last passenger is sitting in her assigned seat?