CS369H: Hierarchies of Integer Programming Relaxations Spring 2016-2017

Lecture 1: Introduction and Overview

Professor Moses Charikar *Scribes: Paris Syminelakis*

Overview We provide a short introduction to the topics of the course and go on to introduce the *Lovasz-Schrijver* and *Sherali-Adams* Hierarchies. A good reference is the review by Chlamtac and Tulsiani [CT12].

1 Introduction

1.1 Integer linear programs and Polytopes

Many combinatorial optimization problems of interest can be formulated as *Integer Linear Programs* (ILP). Given a matrix $A \in \mathbb{R}^{m\times n}$ and vectors $b \in \mathbb{R}^m,\ w \in \mathbb{R}^n$, we define the following problem

$$\begin{aligned} \min \quad & w^\top x \\ \text{s.t} \quad & Ax \le b \\ & x \in \{0,1\}^n \end{aligned} \tag{1}$$

That is we seek to optimize a linear function over the intersection of the (convex) polytope $Ax \le b$ and the Boolean hypercube $\{0,1\}^n$. For any such ILP, we have an associated (convex) polytope

$$P_n := \mathrm{conv}(\{x \in \{0,1\}^n | Ax \le b\}) \tag{2}$$

by taking the convex combination of all feasible solutions. Due to the Fundamental Theorem of Linear Programming [BT97] the optimization problem (1) is equivalent to the following *linear program* (LP)

$$\min\{w^\top x | x \in P_n\} \tag{3}$$

Hence, the study of integer linear programs is intimately related to properties of polytopes. Before, providing some examples we note that binary constraints can be generalized to constraints of higher arity in a straightforward manner, so in this course we will focus only on the binary case. In this course, we will explore the *power of certain broad algorithmic frameworks* in solving (1).

1.2 Linear Programming and Polyhedral Lifts

The prototypical source of ILP examples are *graph problems*. By enumerating all the $n := \binom{N}{2}$ edges, we can encode any graph $G(V,E)$ on N vertices by the indicator function $\mathbf{1}_E \in \{0,1\}^n$. Conversely, given $x \in \{0,1\}^n$ we may define a graph by considering the set of edges $E_x := \{e | x_e = 1\}$. This correspondence allows us to encode different problems as an ILP by A, b, w appropriately. We give below a few such examples.

Definition 1.1 (TSP). *Given N cities $\{1,\dots,N\}$ and costs $\{c_{ij}\}_{i,j\le N}$ for travelling between cities i and j, find the permutation (bijection) $\pi : [N] \to [N]$ that minimizes: $c_{\pi(1)\pi(2)} + c_{\pi(2)\pi(3)} + \dots c_{\pi(n)\pi(1)}$.*

This problem is known as the *Traveling Salesman Problem*. In order to turn this problem into an ILP, we observe that any such permutation π corresponds to a *tour* of the N cities, i.e., *a spanning* 2*-regular subgraph*. Let $\mathrm{TSP}_n := \mathrm{conv}(\{\, x \in \{0,1\}^n | \, E_x \text{ is a tour}\})$.

Exercise 1.1. *Show that the constraint* $\{E_x \textit{is a valid tour}\}$ *can be expressed using linear constraints for* $x \in \{0,1\}^n$.

TSP is known to be NP-Complete and yet we are able to express it as a linear program for which we have polynomial time algorithms! The catch is that TSP_n requires *exponentially many constraints* to be expressed as $Ax \leq b$. However, this does not imply that there is no LP of polynomial size that can be used to solve TSP, but only that this particular way of formulating the problem does not lead to efficient algorithms. For example, consider the *Minimum Spanning Tree* problem.

Definition 1.2 (MST)**.** *Given* n *cities* $\{1, \ldots, n\}$ *and costs* $c_{ij} \geq 0$ *between cities* i *and* j*, find a spanning tree of minimum cost.*

We associate with the MST problem the following polytope $\mathrm{ST}_n := \mathrm{conv}(\{\mathbf{1}_\tau \in \{0,1\}^n | \tau \text{ is a spanning tree}\})$. Again, this formulation requires an exponential number of constraints, and yet we do have efficient algorithms for MST. It turns out that there is another polytope L (from "lift") in n^3 dimensions such that $\mathrm{ST}_n = L \cap \{A^{'}y \leq b^{'}\}$ can be expressed as the intersection (shadow) of the polytope L with an affine subspace $A^{'}y \leq b^{'}$ using a polynomial number of constraints [Mar91]. Thus, linear programming can be used to provide a polynomial algorithm for MST. Such lifts are called *polyhedral* as it involve only linear constraints. We therefore have the following fundamental question:

When can we express a complicated polytope as the "shadow" of a simple one?

This question has a rich history and we only give here an (incomplete) list of a few of the highlights.

[1988] Yanakakkis [Yan88] introduces the model and a key tool, *non-negative matrix factorization*. Shows that any *symmetric linear program* that can be used to express the TSP or Matching polytopes must have *exponential size*.

[2011] Rothvoss [Rot13] shows using a probabilistic argument that there exist polytopes in $\mathbb{R}^n$ that require linear programs of size $2^{\Omega(\frac{n}{2})}$.

[2012] Fiorini et al. [FMP$^+$15] prove that the TSP and Stable Set[1] polytopes require linear programs (not necessarily symmetric) of size $2^{\Omega(\sqrt{n})}$.

[2014] Rothvoss [Rot14] shows that the matching polytope requires exponential $2^{\Omega(n)}$ sized linear programs and, using a reduction from Yanakkakis [Yan88], that also the TSP polytope requires linear programs of exponential size.

After almost 30 years of research we have finally ruled out the possibility of using linear programs to get *exact* solutions to NP-complete problems. Thus, showing an unconditional lower bound to a large class of algorithms.

[1]Think of it as the convex hull of the indicator vectors of independent sets.

1.3 Semidefinite Programs and Spectrahedral Lifts

Another rich class of algorithms are ones based on *Semidefinite programming* (SDP). Given a set of matrices $C, A_1, \ldots, A_m \in \mathbb{R}^{n\times n}$ and a vector $b \in \mathbb{R}^m$, we define the following problem:

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \langle A_i, X\rangle \leq b_i,\ \forall i = 1, \ldots, m \\ & X \succeq 0 \end{aligned} \tag{4}$$

where $\langle A, B\rangle = \sum_{ij} A_{ij}B_{ij}$. The set $S^n_+ := \{X \in \mathbb{R}^{n\times n} | X \succeq 0\}$ of semidefinite matrices is a cone. A *spectahedron* is the intersection of an affine subspace (linear inequalities) with the S^n_+. To motivate the use of spectrahedron in optimization, we will use the example of the *Max Cut* problem.

Definition 1.3 (MC)**.** *Given un undirected graph with non-negative weights $c_{ij} \geq 0$ for each edge (i, j), find a set $S \subset V$ such that the cost $c(\partial S) := \sum_{i\in S, j\notin S} c_{ij}$ of the edges crossing S is maximized.*

We can write down the following integer optimization problem:

$$\begin{aligned} \max \quad & \frac{1}{4}\sum_{ij} c_{ij}(1 - x_i x_j) \\ \text{s.t.} \quad & x_i \in \{-1, 1\},\ i = 1, \ldots, n \end{aligned}$$

It is an easy exercise to see that this is an exact formulation. Given the solution x^* to the problem we can recover the optimal set by $S_{x^*} = \{i | x_i = 1\}$. The idea of SDP is to find values X_{ij} for each pair of variables to "model" $x_i \cdot x_j$ directly and encode the fact that X_{ij} should be an inner product by setting $X \succeq 0$. Then try to express the remaining constraints as linear inequalities on X. For the MC problem we have the following SDP relaxation:

$$\begin{aligned} \max \quad & \frac{1}{4}\sum_{ij} c_{ij}(1 - X_{ij}) \\ \text{s.t.} \quad & X_{ii} = 1,\ i = 1, \ldots, n \\ & X \succeq 0 \end{aligned}$$

A breakthrough result by Goemanns and Williamson showed how one can use SDP's to provide approximation algorithms using the cholesky-decomposition of X and randomized rounding. Since there has been a lot of effort to understand the power of algorithms that

(i) encode all the *instance specific information* in the solution X^* of an SDP.

(ii) use X^* as well as *problem specific information* to get a feasible solution $\tilde{x}$.

A general technique to formulate an SDP relaxation to an ILP problem is the Lasserre-Parillo Hierarchy [Las01, Par03]. A few of the landmark results are:

[1994] Goemans and Williamson introduce the Hyperplane-Rounding technique for rounding SDP's and give an 0.8371 approximation algorithm for MC.

[2005] Khot [Kho05] introduces the Unique Games Conjecture (UGC).

[2008] Raghavendra [Rag08] shows that under the UGC, the Goemans-Williamson technique is optimal for any 2-variable Constraint Satisfaction Problem (CSP).

[2011] Barak, Raghavendra, Steurer [BRS11] show how to use the Lassere Hierarchy to get algorithms for 2-variables CSP's and provide approximation guarantees. Similar results where obtained independently by Guruswami and Sinop [GS11].

[2013] Fawzi et al [FSP15] and Lee et al. [LRST14] show that among symmetric SDP relaxations, Lasserre Hierarchy is optimal and that symmetric SDP relaxations require exponential size to exactly capture the Cut [Lau03] or Parity polytope.

[2015] Lee, Raghavendra, Steurer [LRS15] show that any (not necessarily symmetric) spectrahedral lift of the Cut, TSP and stable set polytopes must have size at least $2^{\Omega(n^\delta)}$ for some $\delta > 0$. They show that essentially the Lasserre Hierarchy is optimal among Semidefinite Programs.

After, 20 years of research we know have a better understanding of the power of SDP relaxations. However, there are many open questions left in terms of using this Hierarchy in designing approximation algorithms for specific problems and more generally providing limits in terms of approximation.

1.4 Main Themes of the Course

1. *Hierarchies*: we will introduce the Lovasz-Schrivjer, Sherali-Adams and Lasserre-Parillo Hiereachies.
2. *Applications:* we will study problems coming from approximation algorithms, signal processing and machine learning and show how the above techniques are applied.
3. *Lower Bounds*: we will study the limitations of these techniques from different perspectives, i.e. extension complexity, approximability.

2 Lovasz-Schrijver Hierarchy

The *Lovasz-Schrijver* (LS) hierarchy is a procedure to start from a basic polytope $\tilde{P}$ that is convex relaxation of some complicated polytope P_n (see (2)), i.e. the convex hull of all integral feasible solutions, and produce a tightened polytope $N(\tilde{P})$. This is done by first adding variables and constraints (*lift*), and then projecting onto the original set of variables (*project*). $N(\tilde{P})$ is the result of lift and project. Similarly, if we also add positive semi-definite constraints before projecting, we get $N_+(\tilde{P})$ and the resulting hierarchy is denoted by LS_+. It is called a hierarchy because both operators are *completely mechanical and can be applied in sequence*. Let $t \geq 1$, then the t-th level of the Lovasz-Schrijver Hierarchy gives the polytope

$$N^t(\tilde{P}) := N(N^{t-1}(\tilde{P})) = \underbrace{N(N(\cdots(N(\tilde{P}))))}_{t \text{ times}} \tag{5}$$

similarly for $N_+^t(\tilde{P})$. One of the key properties of LS (and other) hierarchy is that after n levels we recover the exact polytope, i.e., $N^n(\tilde{P}) = P_n$. In order to optimize over $N^t(\tilde{P})$ or $N_+^t(\tilde{P})$ we need time $n^{O(t)}$, therefore the hope is that after a few levels of repeating this procedure the extra structure can be algorithmically exploited to give improved rounding algorithms. This is the high-level description of the procedure, we next present it in more detail and use the *Maximum Independent Set* problem as concrete example.

Definition 1.4 (MIS). *Given N vertices and a graph $G(V, E)$ find a maximal set of vertices S such that there is no edge with both ends in S.*

We can express this problem in terms of the *Stable Set* polytope as $\max\{\mathbf{1}^\top x | x \in \text{Stable}(G)\}$, where

$$\text{Stable}(G) := \text{conv}(\{x \in \{0,1\}^n | x_i + x_j \leq 1,\ \forall ij \in E(G)\}) \tag{6}$$

Relaxing the integrality constraints we get the following Linear Programming relaxation of the MIS problem $\max\{\mathbf{1}^\top x | x \in \tilde{P}\}$ where $\tilde{P} := \{x \in \mathbb{R}^n | x_i + x_j \leq 1,\ \forall ij \in E(G)\}$.

2.1 Lifting into a Cone

We are given a basic convex set $\tilde{P}$ (coming from some basic relaxation of P). We first convert into the cone:

$$C(\tilde{P}) := \Big\{ (\lambda, \lambda y_1 \cdots, \lambda y_n) \in \mathbb{R}^{n+1} \Big| \, \lambda \geq 0,\ y \in \tilde{P} \Big\} \tag{7}$$

In the case of the MIS problem, the resulting cone would be

$$C_{\text{MIS}} = \big\{ (y_0, y_1, \ldots, y_n) \in \mathbb{R}^{n+1} \big| \, y_i + y_j \leq y_0, \forall ij \in E,\ y_0 \geq 0 \big\} \tag{8}$$

2.2 Encoding products of variables

The heart of this method is to add constraints that also constraint the product of two variables. This is especially relevant as boolean variables satisfy $x_i \cdot x_i = x_i$. We do this by adding new variables Y_{ij} for all $(i, j) \in [n]^2$ and use these variables to impose additional constraints. on $y \in C(\tilde{P})$. Let $K = C(\tilde{P})$ and $N(K)$ be the cone in $\mathbb{R}^{n+1}$ defined by $y \in N(K)$ iff there exists a matrix $Y \in \mathbb{R}^{(n+1)\times(n+1)}$ such that

- Y is symmetric (commutativity of multiplication)
- $Y_{ii} = y_i$ for $i = 1, \ldots, n$ (boolean variables)
- $Y_{0i} = y_i$ for $i = 1, \ldots, n$ (we can recover original variables from $Y \in \mathbb{R}^{(n+1)^2}$).
- $Y_i \in K$ and $Y_0 - Y_i \in K$ for $i = 1, \ldots, n$ (encoding products $x_i x_j$ as well as $(1 - x_i)x_j$).

The cone $N_+(K)$ is exactly the same but with the additional constraint $Y \succeq 0$. The matrix Y is called *protection matrix*. The term stems from a game-theoretic interpretation of the LS hierarchy.

We have two players the prover and the verifier. The goal of the prover is to convince the verifier that a certain vector y belongs to the set $N(K)$. For any such vector the verifier can ask the prover to produce a matrix Y that certifies that fact. For higher values of $t \geq 2$, if the prover wants to convince the verifier that $y^t \in N^t(K)$ then she needs to produce a protection matrix $Y(t)$. Subsequently, the verifier would pick an index i and ask for the prover to prove that $Y_i(t) \in N^{t-1}(K)$ or $Y_0(t) - Y_i(t) \in N^{t-1}(K)$, where $Y_i(t)$ denotes the i-th row of $Y(t)$. Hence, the prover needs to provide a strategy that succeeds for every sequence of questions that the verifier must ask. This interpretation is useful in proving limitations of the LS hierarchy, i.e. by constructing a valid fooling strategy.

The final convex relaxation based on LS^t for the MIS problem is given by:

$$\begin{aligned}
\max \quad & \sum_{i=1}^{n} y_i \\
\text{s.t} \quad & (y_0, y_1, \ldots, y_n) \in N^t(C_{\text{MIS}}) \\
& y_0 = 1
\end{aligned} \tag{9}$$

Exercise 1.2. *Show that after n-rounds $N^n(K) = K \cap \{0,1\}^n$.*

2.3 Weak Separation Oracle

One benefit of the recursive definition of this procedure is that we can use this even if we are not given K or $\tilde{P}$ explicitly but only through a *weak separation oracle* (WSO). Before, we give the definition we need to define the notion of a *polar cone*. Given a convex cone $K \in \mathbb{R}^{n+1}$, let $K^* = \left\{ w \in \mathbb{R}^{n+1} \middle| \, w^\top x \geq 0, \forall x \in K \right\}$ denote the polar cone of K.

Definition 1.5 (WSO)**.** *A weak separation oracle for a convex cone K takes as input a vector $x \in \mathbb{Q}^{n+1}$ and a rational number $\epsilon > 0$, and it either certifies that the euclidean distance of x from K is at most ϵ, or it returns a vector w such that $\|W\| \geq 1$, $w^\top x \leq \epsilon$ and the euclidean distance of w from K^* is at most ϵ.*

Exercise 1.3. *Given a weak separation oracle for a convex cone K, show that one can construct a weak separation oracle for $N(K)$ and $N_+(K)$.*

3 Sherali-Adams Hierarchy

We motivate the Sherali-Adams (SA) relaxation by looking into Lovasz-Schrijver. Given a vector $y \in N^2(K)$ let $Y(2)$ be the corresponding protection matrix. For any index i, we know that $Y_i(2) \in N(K)$ and hence there exists a corresponding protection matrix $Y^{'}$ for this vector. Looking at the jk element of this matrix, if the vector y was integral we should have that $Y^{'}_{jk} = y_i \cdot y_j \cdot y_k$. If we first picked j and then ik we would get a different matrix $Y^{''}$ and the ik-th element would be respectively $Y^{''}_{ik} = y_j \cdot y_i \cdot y_k$. A deficiency of LS is that there is no guarantee that $Y^{'}_{jk} = Y^{''}_{ik}$.

Sherali-Adams relaxation addresses this by adding constraints to enforce that all products evaluate to the same quantity. The main idea in t-th level relaxation of the Sherali-Adams (SA^t) is to introduce a variable Y_S for each subset $S \subset [n]$ of $t+1$ variables with the aim that $Y_S = \prod_{i \in S} y_i$ for integral vectors and when the vector is non-integral one may use this variables to define a locally-consistent distribution (pseudo-distributions).

3.1 Enforcing local consistency

Let $a^\top y \leq b$ be one of the constraints define the convex polytope $\tilde{P}$. Sherali-Adams adds constraints that would be equivalent to the following in the case of boolean vector y, $\forall S, T \subset [n]$ such that $|S| + |T| \leq t$

$$(a^\top y - b) \cdot \prod_{i \in S} y_i \prod_{j \in T} (1 - y_i) \leq 0 \tag{10}$$

Carrying out the multilpication and exploiting the fact that the variables are boolean, we can express these constraints using our variables $\{Y_S\}_{|S|\leq t+1}$.

$$\sum_{T'\subset T}(-1)^{|T'|}\left(\sum_{i=1}^{n}a_iY_{S\cup T'\cup\{i\}}-bY_{S\cup T'}\right)\leq 0 \tag{11}$$

The number of new variables and constraint added are $O(n^{O(t)})$ and the resulting LP can be solved in $n^{O(t)}$ time. The work of Guruswami and Sinop [GS12] among other things shows that one can optimize over the t-th level of the Sherali-Adams hierarchy using only a weak separation oracle and in certain cases can solve such problems in time $n2^{O(t)}$ instead of $n^{O(t)}$.

As an example the SA^t relaxation for MIS is:

$$\max \quad \sum_{i=1}^{n}Y_{\{i\}} \tag{12}$$

$$\text{s.t.} \quad \sum_{T'\subset T}(-1)^{|T'|}\left(Y_{S\cup T'\cup\{i\}}+Y_{S\cup T'\cup\{i\}}-Y_{S\cup T'}\right)\leq 0, \qquad \forall|S|+|T|\leq t, \forall ij\in E(G) \tag{13}$$

$$0\leq\sum_{T'\subseteq T}(-1)^{|T'|}Y_{S\cup T'\cup\{i\}}\leq\sum_{T'\subseteq T}(-1)^{|T'|}Y_{S\cup T'}, \qquad \forall|S|+|T|\leq t, i\in[n] \tag{14}$$

The last set of constraints come from requiring that $y_i\in[0,1]$. These constraints where redundant in the simple LP forumlation for MIS.

Exercise 1.4. *Show that the constraints in* (14) *are necessary.*

3.2 Local Distribution viewpoint

We first not that since SA^t is a convex relaxation, then any convex combination of $0/1$ solutions is a feasible for the Sherali-Adams relaxation. Is the converse true? The answer is that this is true only *locally*.

Lemma 1.1. *Any feasible solution to the t-th level Sherali-Adams relaxation is equivalent to a family of distributions $\{\mathcal{D}(S)\}_{|S|\leq t+1}$ such that they are* locally consistent.

In the interest of time, we do not prove the lemma here. Extending this intuition further we define the notion of a *partial assignment*. Let $\alpha\in\{0,1\}^S$ and define $Y_{s,\alpha}=\mathbb{E}[\prod_{i\in\alpha^{-1}(1)}y_i\prod_{j\in\alpha^{-1}(0)}(1-y_j)]$. Using the variables in Sherali-Adams we can express this probability as $Y_{S,\alpha}=\sum_{T\subseteq\alpha^{-1}(0)}(-1)^{|T|}Y_{\alpha^{-1}(1)\cup T}$.

References

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 472–481. IEEE, 2011.

[BT97] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[CT12] Eden Chlamtac and Madhur Tulsiani. Convex relaxations and integrality gaps. In *Handbook on semidefinite, conic and polynomial optimization*, pages 139–169. Springer, 2012.

[FMP$^+$15] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald De Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *Journal of the ACM (JACM)*, 62(2):17, 2015.

[FSP15] Hamza Fawzi, James Saunderson, and Pablo A Parrilo. Equivariant semidefinite lifts and sum-of-squares hierarchies. *SIAM Journal on Optimization*, 25(4):2212–2243, 2015.

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 482–491. IEEE, 2011.

[GS12] Venkatesan Guruswami and Ali Kemal Sinop. Faster sdp hierarchy solvers for local rounding algorithms. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 197–206. IEEE, 2012.

[Kho05] Subhash Khot. On the unique games conjecture. In *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, page 3. IEEE, 2005.

[Las01] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[Lau03] Monique Laurent. Lower bound for the number of iterations in semidefinite hierarchies for the cut polytope. *Mathematics of operations research*, 28(4):871–883, 2003.

[LRS15] James R Lee, Prasad Raghavendra, and David Steurer. Lower bounds on the size of semidefinite programming relaxations. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 567–576. ACM, 2015.

[LRST14] James R Lee, Prasad Raghavendra, David Steurer, and Ning Tan. On the power of symmetric lp and sdp relaxations. In *Computational Complexity (CCC), 2014 IEEE 29th Conference on*, pages 13–21. IEEE, 2014.

[Mar91] R Kipp Martin. Using separation algorithms to generate mixed integer model reformulations. *Operations Research Letters*, 10(3):119–128, 1991.

[Par03] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 245–254. ACM, 2008.

[Rot13] Thomas Rothvoß. Some 0/1 polytopes need exponential size extended formulations. *Mathematical Programming*, 142(1-2):255–268, 2013.

[Rot14] Thomas Rothvoss. The matching polytope has exponential extension complexity. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*, STOC '14, pages 263–272, New York, NY, USA, 2014. ACM.

[Yan88] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 223–228, New York, NY, USA, 1988. ACM.