* Disclaimer: many ideas were implicit earlier: certain approx algo were designed by mathematicians in the 1960's. Lovász Θ function gives early ('79) application of SDP, etc.

computability

1965 — Edmonds: Efficient = polynomial time

1971 — Cook: NP-hardness

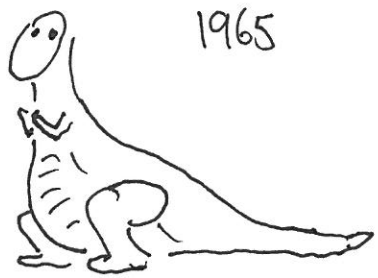
1974 — Johnson: approximation algorithms

1982 — LP-based approx algo (Efficient LP solvers)
Hochbaum, Karmarkar, Karp

1992 — PCP Thm: NP-hardness of approx (IP = PSPACE, MIP = NEXP)
SDP-based approx algo
Goemans-Williamson

1996 — Håstad: optimal inapprox results

2002 — Khot: Unique Games Conj



2002 – 2016: many new algorithms, rounding techniques
better LP solvers, hierarchies of LP/SDP (= better programs), limits of LP/SDP.
implications of UGC (e.g. SDP optimal for CSPs), towards UGC (proof or disproof)
nearly linear reductions, many new hardness results.

Def An algo A gives α-approx for a problem Π if for all inputs x,

$$\alpha \Pi(x) \le A(x) \le \Pi(x) \quad \text{where } \Pi \text{ is a maximization problem}$$

$[0,1]$

$$\Pi(x) \le A(x) \le \alpha \cdot \Pi(x) \quad \text{where } \Pi \text{ is a minimization problem}$$

≥ 1

Other notions: - additive approx $\pm\beta$

- expected approx factor $\underset{r}{\mathbb{E}}\, A(x,r)$

Examples

- Partition Given $s_1 \ldots s_n$ partition $A \dot\cup B = \{1, \ldots, n\}$ to min $\max\{\sum_{i\in A} s_i, \sum_{i\in B} s_i\}$:)

Thm For any $\varepsilon > 0$, there is an efficient $(1+\varepsilon)$-approx algo for Partition ("polynomial time approx scheme").

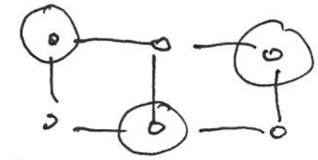
The algo picks the $1/\varepsilon$ largest numbers, partitions them optimally. Partitions rest greedily.

- Clique Given $G=(V,E)$ find the largest subset of vertices s.t. every two vertices have an edge. :(

Thm There is an efficient $\frac{\text{polylg}\, n}{n}$-approx algo for clique.

- Vertex Cover Given $G=(V,E)$ find the smallest subset of vertices that touches all edges.

Thm There is an efficient 2-approx for vertex cover



Pf Pick an edge, take both its endpoints to cover, remove all covered edges.

- Set Cover Given $S_1 \ldots S_m \subseteq U$, $|U|=n$, find smallest family of sets that covers U.

Thm There is an efficient $\ln n^{+1}$-approx for set-cover.

Pf Pick set that covers as many elements of U as possible, remove all covered elements

Lem If the opt cover has k sets then in each iteration there must be a set that covers $\frac{1}{k}$ fraction of remaining elements.

Hence, if U_i = uncovered elements in iteration i.

$|U_i| \leq (1-\frac{1}{k})^i |U|$ when $i > k \ln |U|$, $|U_i| < 1$.

- **Max 3SAT** Given clauses $C_1 \ldots C_m$ each of the form

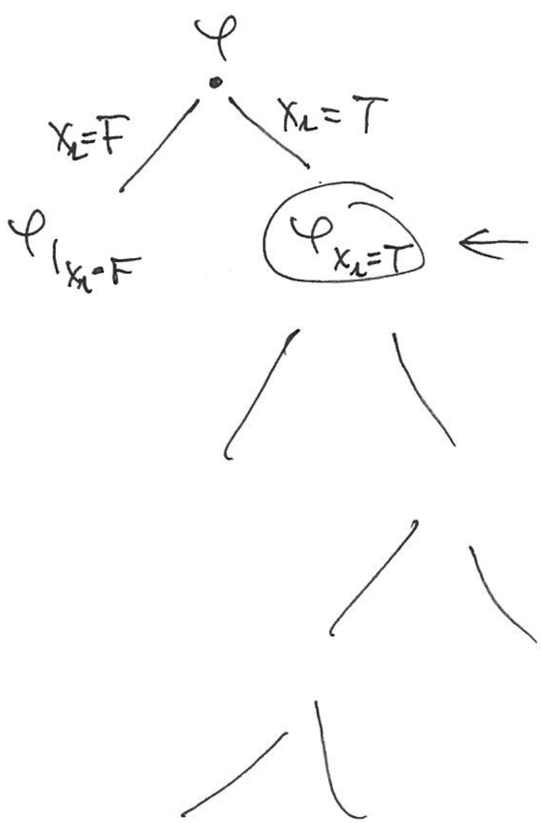
$$(\neg)x_i \vee (\neg)x_j \vee (\neg)x_k$$

find an assignment to variables $x_1 \ldots x_n$ that satisfies as many clauses as possible.

Thm There is a $\frac{7}{8}$-approx algo for Max 3SAT

Lem The expected number of satisfied clauses if we pick the assignment at random is $\frac{7}{8}m$.

Can derandomize using the method of conditional expectation



Can compute efficiently the expected fraction of sat clauses; their average should be $\frac{7}{8}$, continue with the higher