

# SDP Relaxation of MAX CUT

## Integer quadratic programming formulation of MAX CUT

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{ij \in E} w_{ij} (1 - y_i y_j) \\ \text{s.t.} \quad & y_i \in \{-1, 1\} \quad \text{for all } i \in V. \end{aligned}$$

## Semidefinite programming relaxation of MAX CUT

$$\begin{aligned} \max \quad & \frac{1}{2} \sum_{ij \in E} w_{ij} (1 - v_i^T \cdot v_j) \\ \text{s.t.} \quad & v_i^T \cdot v_i = 1 \quad \text{for all } i \in V, \\ & v_i \in \mathbb{R}^n \quad \text{for all } i \in V. \end{aligned}$$

93

## Randomized Rounding of Vector Program

- 1 compute (near-)optimal solution ( $v^*$ ) to SDP relaxation;
- 2 pick a random vector  $r = (r_1, \dots, r_n)^T$  by drawing each component from  $\mathcal{N}(0, 1)$ , the normal distribution with mean 0 and variance 1;
- 3 for  $i = 1, \dots, n$ : if  $r^T \cdot v_i^* \geq 0$  then put  $i$  in  $S$ ;

### Remarks.

- ▶ The hyperplane orthogonal to  $r$  partitions the  $n$ -dimensional unit sphere into two halves, corresponding to  $S$  and  $V \setminus S$ .
- ▶ The normalization  $r/\|r\|$  of  $r$  is uniformly distributed over the  $n$ -dimensional unit sphere.
- ▶ The projections of  $r$  onto two unit vectors  $e_1, e_2$  are independent and normally distributed if and only if  $e_1$  and  $e_2$  are orthogonal.

### Corollary 6.4.

Let  $r'$  the projection of  $r$  onto a 2-dimensional plane. The normalization  $r'/\|r'\|$  of  $r'$ , is uniformly distributed on a unit circle in the plane.  $\square$

94

## Analysis of the SDP-based Algorithm

### Lemma 6.5.

The probability that edge  $ij \in E$  is in the cut is  $\frac{1}{\pi} \arccos(v_i^T \cdot v_j)$ .

Proof:...



### Lemma 6.6.

For  $x \in [-1, 1]$  it holds that  $\frac{1}{\pi} \arccos(x) \geq 0.878 \cdot \frac{1}{2}(1 - x)$ .



### Theorem 6.7.

SDP-based randomized rounding is a 0.878-approximation algorithm for MAX CUT.

Proof:...



**Remark.** The algorithm can be derandomized by using a sophisticated application of the method of conditional expectations.

95

## Inapproximability Results for MAX CUT

We state the following results without proof.

### Theorem 6.8.

If there is an  $\alpha$ -approximation algorithm for MAX CUT with  $\alpha > 16/17 \approx 0.941$ , then  $P = NP$ .



### Theorem 6.9.

Given the *Unique Games Conjecture* there is no  $\alpha$ -approximation algorithm for MAX CUT with constant

$$\alpha > \min_{-1 \leq x \leq 1} \frac{\frac{1}{\pi} \arccos(x)}{\frac{1}{2}(1 - x)} \approx 0.878$$

unless  $P = NP$ .



96

## Coloring 3-Colorable Graphs

### Observation 6.10.

A graph of maximum degree  $\Delta$  can be  $(\Delta + 1)$ -colored in polynomial time.

Simple algorithm for coloring 3-colorable graphs:

- 1 while there is a node  $v$  of degree  $\geq \sqrt{n}$
- 2     color nodes in  $\{v\} \cup N(v)$  with 3 new colors and delete them;
- 3     color the remaining nodes with  $\sqrt{n}$  new colors;

### Theorem 6.11.

The algorithm above colors any 3-colorable graph with at most  $4\sqrt{n}$  colors.

Proof:...

□

97

## Semidefinite Programming Relaxation

Idea:

Associate nodes with unit vectors such that adjacent nodes are far apart:

$$\begin{array}{ll} \min & \lambda \\ \text{s.t.} & v_i^T \cdot v_j \leq \lambda \quad \text{for all } ij \in E, \\ & v_i^T \cdot v_i = 1 \quad \text{for all } i \in V, \\ & v_i \in \mathbb{R}^n \quad \text{for all } i \in V. \end{array}$$

### Lemma 6.12.

If  $G = (V, E)$  is 3-colorable, there is feasible SDP solution with  $\lambda \leq -\frac{1}{2}$ .

Proof:...

□

98

# Randomized Algorithm for Semicoloring

## Definition 6.13.

A **semicoloring** is a coloring of nodes such that there are at most  $n/4$  monochromatic edges (i. e., feasible coloring for subgraph on  $n/2$  nodes.)

## Randomized SDP-based algorithm

- 1 compute optimal solution  $v^*$  to the semidefinite program;
- 2 choose  $t := 2 + \log_3 \Delta$  random vectors  $r_1, \dots, r_t$ ;  
// (draw each component of  $r_i$  from  $\mathcal{N}(0, 1)$ )
- 3 color nodes in  $V$  with at most  $2^t$  colors such that  $i, j \in V$  get the same color if and only if  $\text{sgn}(v_i^T \cdot r_k) = \text{sgn}(v_j^T \cdot r_k)$  for  $k = 1, \dots, t$ ;

## Theorem 6.14.

The algorithm colors the nodes with at most  $4\Delta^{\log_3 2}$  colors. This is a semicoloring with probability at least  $1/2$ .

Proof:...

□<sub>99</sub>

## Iterative Coloring Algorithm

Let  $g(n) \in \tilde{O}(f(n))$  if  $g(n) \in O(f(n) \log^c n)$  for some constant  $c \geq 0$ .

Applying the above algorithm iteratively ( $O(\log n)$  times) yields  $\tilde{O}(n^{\log_3 2})$ -coloring. Since  $\log_3 2 \approx 0.631$ , this is worse than  $\sqrt{n}$ .

## Improved Algorithm

- 1 while there is a node  $v$  of degree  $\geq \sigma$
- 2 color nodes in  $\{v\} \cup N(v)$  with 3 new colors and delete them;
- 3 use above algorithm to  $O(\sigma^{\log_3 2} \log n)$ -color remaining nodes;

## Theorem 6.15.

For  $\sigma = n^{\log_6 3}$ , the algorithm colors a 3-colorable graph with  $\tilde{O}(n^{0.387})$  colors.

Proof:...

□