APPROXIMATION ALGORITHMS

RANDOMIZED ROUNDING OF SEMIDEFINITE PROGRAMS

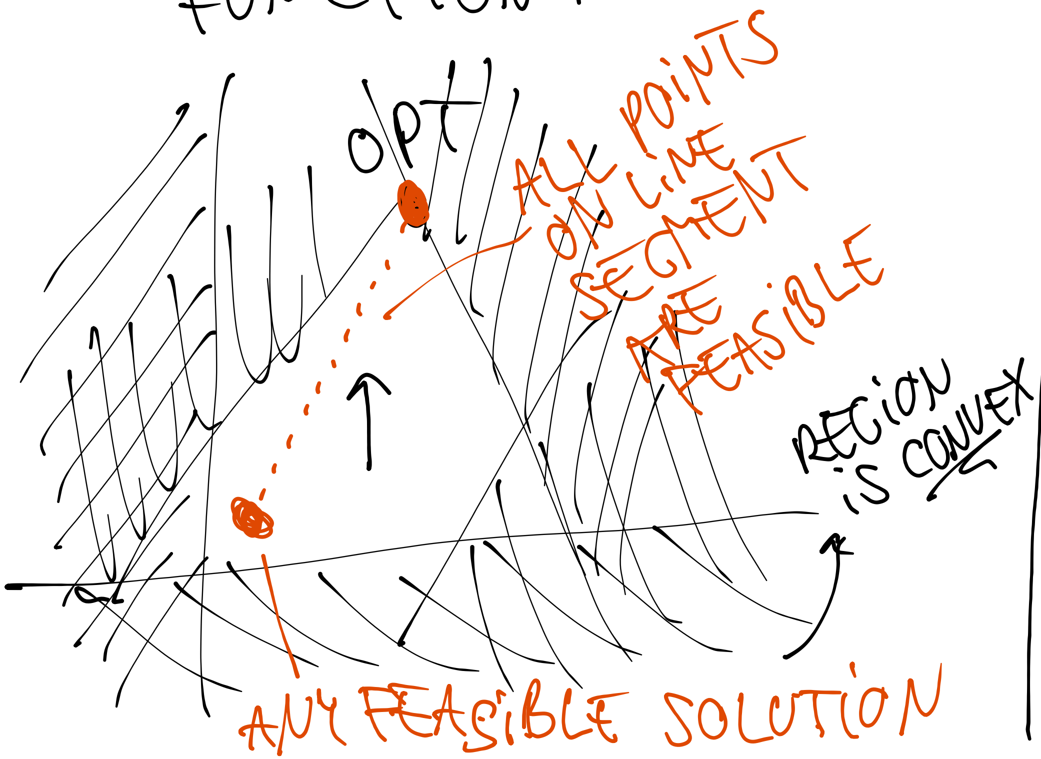
RASMUS PAGH
UNIVERSITY OF COPENHAGEN

TODAY

- CONVEX OPTIMIZATION
  - SEMIDEFINITE PROGRAMMING
  - VECTOR PROGRAMMING
- CASE STUDY: MAX CUT
- SEMIDEFINITE PROGRAMMING IN PYTHON (cvxpy)
- CASE STUDY: CORRELATION CLUSTERING

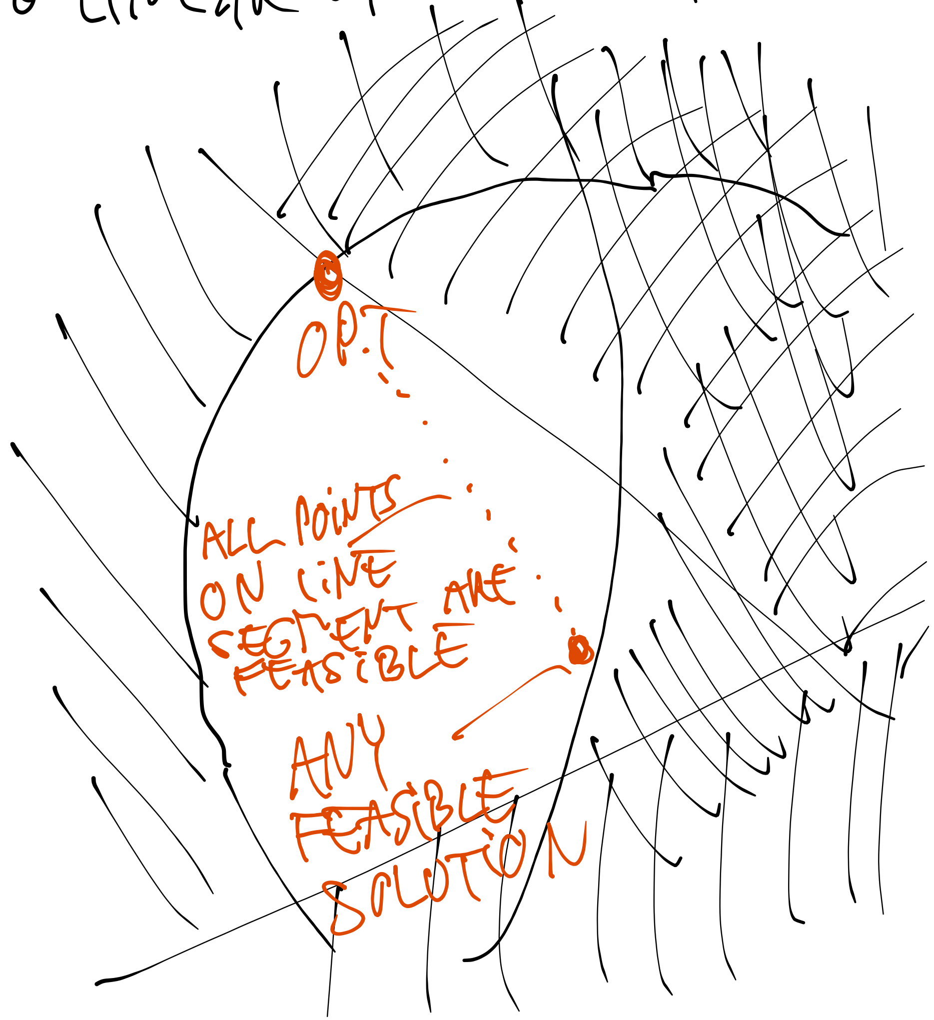
LINEAR PROGRAMMING

- POLYNOMIAL TIME SOLVABLE
- COLLECTION OF HALFSPACE CONSTRAINTS.
- LINEAR OBJECTIVE FUNCTION.



CONVEX OPTIMIZATION ← USE AS BLACK BOX

- "USUALLY" ~~EFFICIENTLY~~ SOLVABLE
- COLLECTION OF CONVEX CONSTRAINTS
- LINEAR OBJECTIVE FUNCTION



SEMIDEFINITE PROGRAMMING

- DECISION VARIABLES FORM A SYMMETRIC MATRIX

$$X = V^T V, \quad \text{WHERE } V \in \mathbb{R}^{n \times n} \leftarrow \text{SET OF } n\text{-BY-}n \text{ REAL MATRICES}$$

$$V = \begin{bmatrix} | & & | \\ v_1 & \cdots & v_n \\ | & & | \end{bmatrix} \quad X = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} \quad x_{ij} = v_i \cdot v_j = \sum_{k=1}^{n} (v_i)_k (v_j)_k$$

- REQUIREMENT ON X (POSITIVE SEMIDEFINITE) WRITTEN AS $X \succeq 0$.

- SEMIDEFINITE PROGRAM: MAXIMIZE $\cancel{\sum_{i,j} c_{ij} x_{ij}}$ $\sum_{i,j} c_{ij} (v_i \cdot v_j)$

VECTOR PROGRAMMING FORM

SUBJECT TO:

$\sum a_{ijk} (v_i \cdot v_j) = \cancel{\sum_{i,j} a_{ijk} x_{ij}} \le b_k$ FOR ALL k

($X \succeq 0$ AND X SYMMETRIC)

$\boxed{\text{CAN SOLVE IN POLY TIME UP TO SMALL ERROR}}$

MAX CUT

$w_{ij} \geq 0$

INPUT: COMPLETE GRAPH ON n VERTICES, WEIGHT MATRIX $(w_{ij})_{ij}$

GOAL: FIND CUT $(S, V\setminus S)$ THAT MAXIMIZES SUM OF WEIGHTS OF EDGES CROSSING THE CUT:

$$\sum_{i \in S} \sum_{j \in V\setminus S} w_{ij}$$

> **PAUSE AND THINK:**
>
> HOW CAN MAX CUT BE FORMULATED USING VECTOR PROGRAMMING WITH INTEGRALITY CONSTRAINTS?

MAX CUT

INPUT: COMPLETE GRAPH ON n VERTICES, WEIGHT MATRIX $(w_{ij})_{ij}$

GOAL: FIND CUT $(S, V\setminus S)$ THAT MAXIMIZES SUM OF WEIGHTS OF EDGES CROSSING THE CUT:

$$\sum_{i\in S}\sum_{j\in V\setminus S} w_{ij}$$

RELAXED
VECTOR PROGRAMMING FORMULATION ~~(ALMOST)~~:

MAXIMIZE $\frac{1}{2}\sum_{i<j} w_{ij}(1-v_i\cdot v_j)$

SIGN INDICATES WHETHER OR NOT IN S.

PADDING WITH ZEROS TO GET n DIM.

SUBJECT TO $\cancel{v_i \in (\pm 1, 0, \dots, 0)}$ FOR ALL i

$\|v_i\|_2^2 = v_i \cdot v_i = 1$ FOR ALL i

WANT:

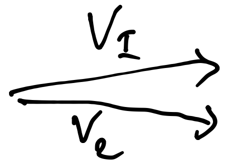
$$v_i \cdot v_j = \begin{cases} 1 & \text{IF ON SAME SIDE OF CUT} \\ 0 & \text{OTHERW} \end{cases}$$

QUESTION: HOW DO WE "ROUND" AN n-DIMENSIONAL VECTOR v_i TO A VALUE IN $\{-1,+1\}$?

"ROUNDING" OF VECTORS

EXTREME CASES:

1)

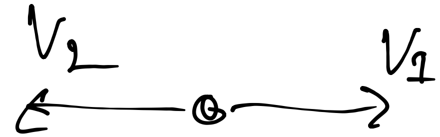


$V_1 \cdot V_2 \approx 1$

$W_{12}(1 - V_1 \cdot V_2) \approx 0$

WANT VERTEX 1 AND 2
ON SAME SIDE OF CUT

2)

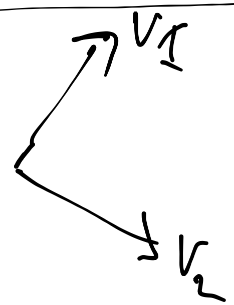


$V_1 \cdot V_2 \approx -1$

$W_{12}(1 - V_1 \cdot V_2) \approx 2W_{12}$

WANT VERTEX 1 AND 2
ON DIFFERENT SIDES OF CUT

MIDDLE CASE:



$V_1 \cdot V_2 \approx 0$

$W_{12}(1 - V_1 \cdot V_2) \approx W_{12}$

WANT VERTEX 1 AND
2 ON DIFFERENT SIDES
WITH PROBABILITY 1/2.

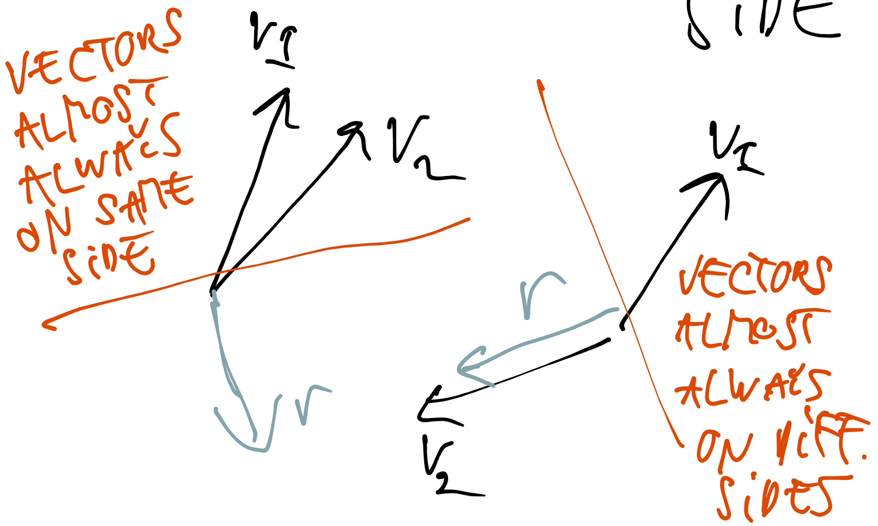
LOCALITY-SENSITIVE HASHING

$\boxed{\text{ASSUME } \|v_1\| = \|v_2\| = 1}$

IDEA: MAP v_1, v_2 TO $h(v_1), h(v_2)$ $\{-1, +1\}$ SUCH THAT:

$$\Pr[h(v_1) = h(v_2)] = \begin{cases} 0 & \text{if } v_1 \cdot v_2 = -1 \\ \frac{1}{2} & \text{if } v_1 \cdot v_2 \approx 0 \\ 1 & \text{if } v_1 \cdot v_2 = 1 \end{cases}$$

CONSTRUCTION: SAMPLE RANDOM HYPERPLANE WITH NORMAL VECTOR $r \in \mathbb{R}^n$, MAP EACH SIDE TO -1 AND $+1$, RESPECTIVELY



$$h(v) = \underbrace{\text{sign}(r \cdot v)}_{\in \{-1, +1\}}$$

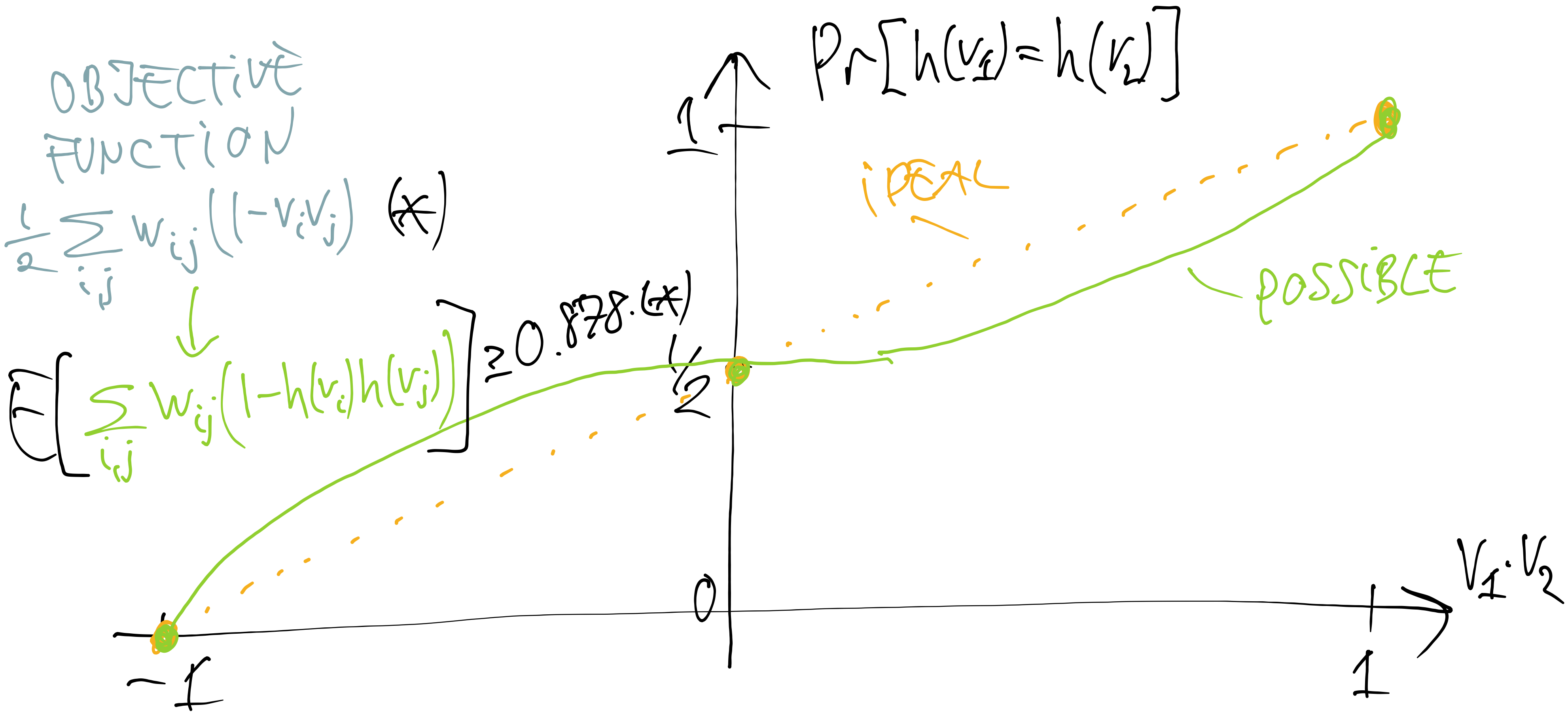
INDEP. OF $\|r\|$

"SIMHASH"

FACT: $r \sim N(0,1)^n$ WORKS

ANALYZING MAX CUT RANDOMIZED ROUNDING

$$\Pr[h(v_1)=h(v_2)] = \begin{cases} 0 & \text{if } v_1 \cdot v_2 = -1 \\ \frac{1}{2} & \text{if } v_1 \cdot v_2 \approx 0 \\ 1 & \text{if } v_1 \cdot v_2 = 1 \end{cases}$$



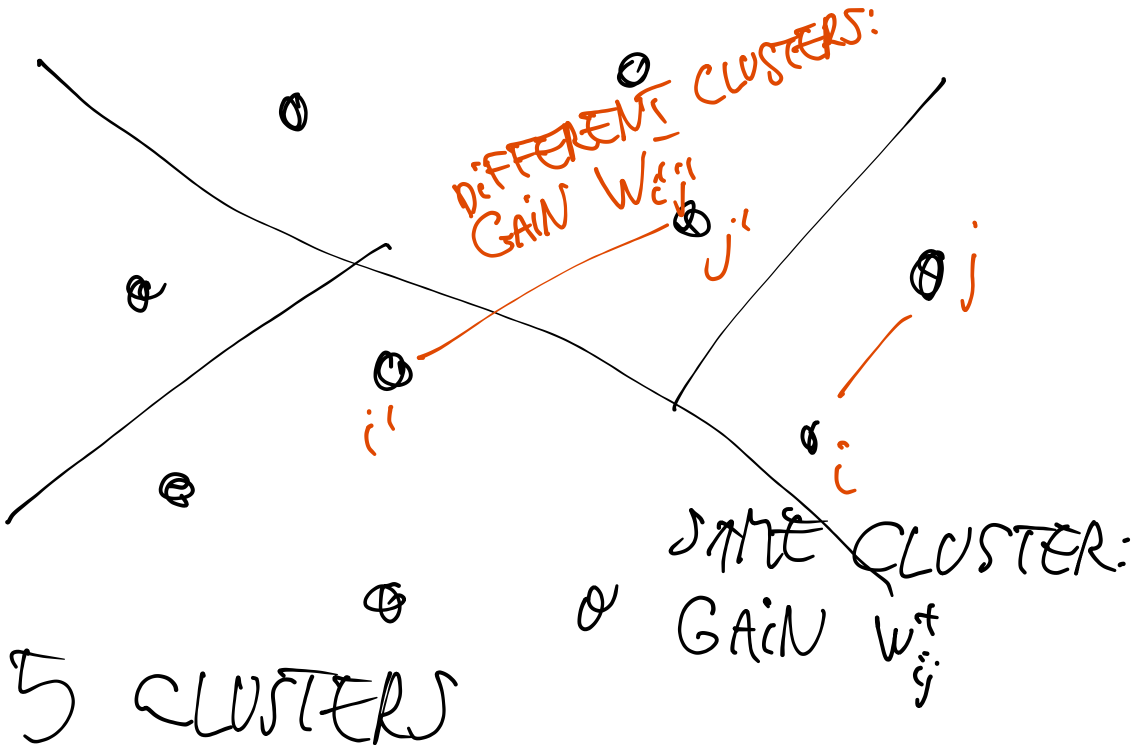
FINDING A CORRELATION CLUSTERING

$w^+_{ij} \geq 0 \quad w^-_{ij} \geq 0$

INPUT: WEIGHT MATRICES $W^+ = [w^+_{ij}]_{ij}$, $W^- = [w^-_{ij}] \in \mathbb{R}^{n \times n}$

OUTPUT: MAPPING INTO CLUSTERS $c(v)$, FOR $v \in \{1, \dots, n\}$

OBJECTIVE: MAXIMIZE $\displaystyle\sum_{i,j : c(i) = c(j)} w^+_{ij} + \sum_{i,j : c(i) \neq c(j)} w^-_{ij}$



PAUSE AND THINK:

HOW CAN WE FORMULATE CORRELATION CLUSTERING USING VECTOR PROGRAMMING?

HINT: NEED $v_i \cdot v_j = 1$ IFF i AND j ARE IN THE SAME CLUSTER

RELAXED ~~VECTOR~~ PROGRAMMING FORMULATION

OBJECTIVE: MAXIMIZE $\sum_{i,j:c(i)=c(j)} w_{ij}^+ + \sum_{i,j:c(i)\neq c(j)} w_{ij}^-$

JUST A CONSTANT, CAN REMOVE W/O CHANGING OPT

$$\sum_{i,j} w_{ij}^+ (v_i \cdot v_j) + \sum_{i,j} w_{ij}^- (1 - v_i \cdot v_j)$$

SUBJECT TO $\cancel{v_i \in \{e_1, \ldots, e_n\}}$

STANDARD BASIS VECTORS

$v_i \cdot v_i = 1, \quad v_i \cdot v_j \geq 0, \quad \forall i, j$

NEXT STEP: MAP VERᴺ INTO CLUSTER $c(v)$.

IDEALLY:

$$\Pr[c(v_i) = c(v_j)] = v_i \cdot v_j$$

$$\Pr[c(v_i) \neq c(v_j)] = 1 - v_i \cdot v_j$$

WOULD GIVE EXPECTED VALUE FOR OBJECTIVE FUNCTION EQUAL TO THE RELAXATION

ACTUAL RANDOM CLUSTERING

TAKE TWO LSH ("SIMHASH") FUNCTIONS h_1, h_2 AS USED IN MAX CUT, AND DEFINE 4 CLUSTERS:

$$c(v) = (h_1(v), h_2(v)) \in \{-1, +1\}^2$$

CAN SHOW THAT

$$E\Big[\sum_{\substack{i,j \\ c(i)=c(j)}} w_{ij}^+ + \sum_{\substack{i,j \\ c(i)\neq c(j)}} w_{ij}^-\Big] \geq \frac{3}{4} \sum_{i,j} w_{ij}^+ (v_i \cdot v_j) + w_{ij}^- (1 - v_i \cdot v_j)$$

$$\geq \frac{3}{4} \, OPT.$$