بسم الله الرحمن الرحیم

برنامه‌ریزی نیمه معین برای طراحی الگوریتم‌های تقریبی

جلسه دوم: الگوریتم تقریبی برای برش بیشینه

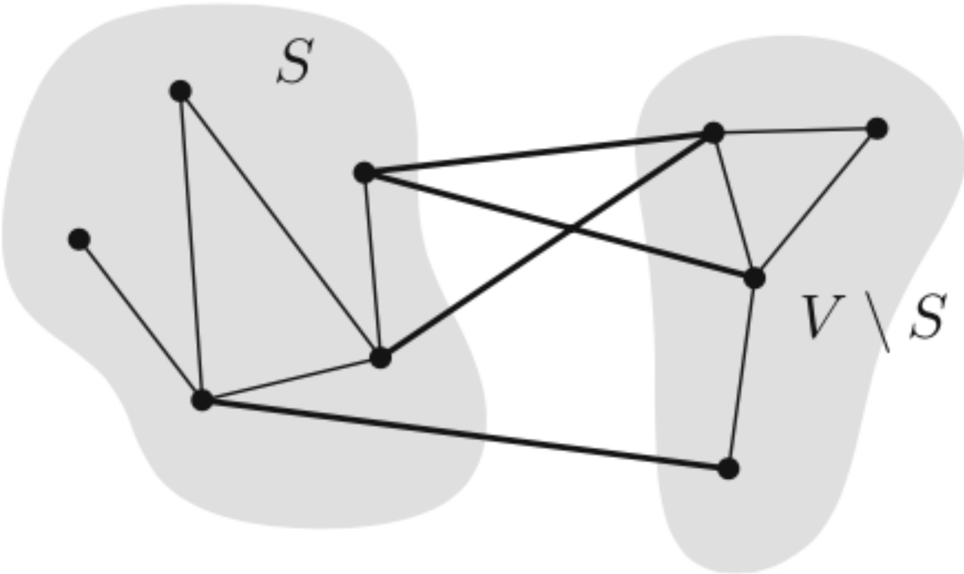
Project: People

حل تمرین

- خانم مائده حشمتی

الگوریتم تقریبی برای مسئله برش بیشینه

مسئله «برش بیشینه»



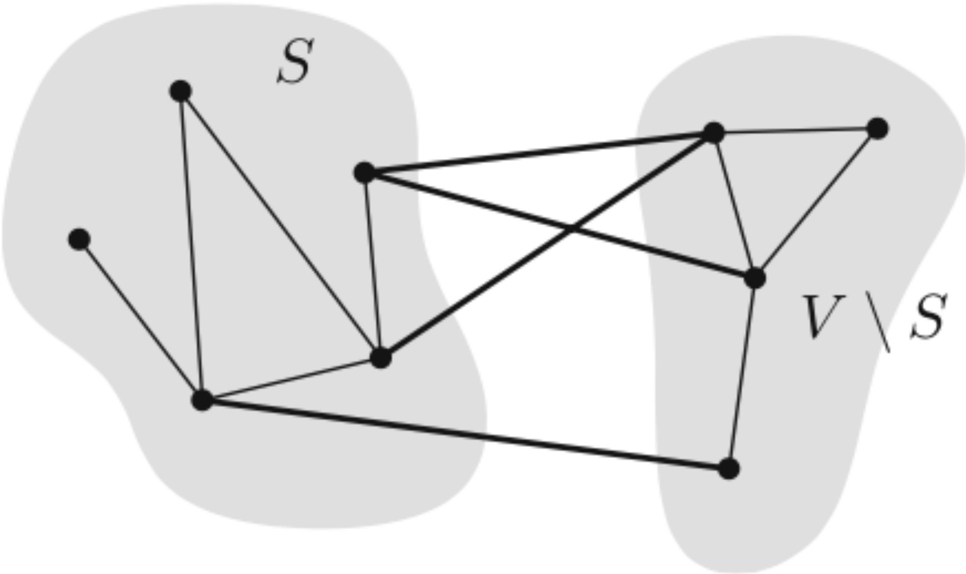
- ورودی: گراف
- خروجی: برش (S, V/S)
- هدف: بیشینه کردن

$$E(S, V \setminus S) =$$

$$\{e \in E : \ |e \cap S| = |e \cap (V \setminus S)| = 1\}$$

مسئله «برش بیشینه»

- ورودی: گراف
- خروجی: برش (S, V/S)
- هدف: بیشینه کردن



$$E(S, V \setminus S) =$$

$$\{e \in E : |e \cap S| = |e \cap (V \setminus S)| = 1\}$$

سخت!

NP ـ سخت

الگوریتم تقریبی

- مسئله بیشینه‌سازی:
  - $\mathcal{I}$: مجموعه نمونه‌ها
  - $I \in \mathcal{I}$: یک نمونه
  - $F(I)$: جواب‌های شدنی
  - $\omega(s) \geq 0$: ارزش یک جواب شدنی s
  - $\mathsf{Opt}(I) = \sup_{s \in F(I)} \omega(s) \in \mathbb{R}_+ \cup \{-\infty, \infty\}$

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δـ تقریب) :
  - ۱) زمان اجرا چند جمله‌ای باشد
    - زمان اجرا $\leq p(|I|)$
      - که $|I|$: اندازه ورودی
  - ۲) ضریب تقریب حداکثر δ باشد
    - برای همه $I \in \mathscr{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
      - $f(I)$: یکی از خصوصیات ورودی

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δـ تقریب) :
  - ۱) زمان اجرا چند جمله‌ای باشد
    - زمان اجرا $\geq p(|I|)$
      - که $|I|$: اندازه ورودی
  - ۲) ضریب تقریب حداکثر δ باشد
    - برای همه $I \in \mathscr{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
      - $f(I)$: یکی از خصوصیات ورودی
- الگوریتم تقریبی تصادفی

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δـ تقریب) :
  - ۱) زمان اجرا چند جمله‌ای باشد
    - زمان اجرا $\geq p(|I|)$
      - که $|I|$: اندازه ورودی
  - ۲) ضریب تقریب حداکثر δ باشد
    - برای همه $I \in \mathcal{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
      - $f(I)$: یکی از خصوصیات ورودی
- الگوریتم تقریبی تصادفی
  - امید زمان اجرا چند جمله‌ای

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δ_ تقریب) :
    - ۱) زمان اجرا چند جمله‌ای باشد
        - زمان اجرا $\geq p(|I|)$
            - که $|I|$: اندازه ورودی
    - ۲) ضریب تقریب حداکثر δ باشد
        - برای همه $I \in \mathcal{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
            - $f(I)$: یکی از خصوصیات ورودی
- الگوریتم تقریبی تصادفی
    - امید زمان اجرا چند جمله‌ای
    - ضریب تقریب،

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δ_ تقریب) :
  - ۱) زمان اجرا چند جمله‌ای باشد
    - زمان اجرا $\leq p(|I|)$
      - که $|I|$: اندازه ورودی
  - ۲) ضریب تقریب حداکثر δ باشد
    - برای همه $I \in \mathcal{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
      - $f(I)$: یکی از خصوصیات ورودی
- الگوریتم تقریبی تصادفی
  - امید زمان اجرا چند جمله‌ای
  - ضریب تقریب،
    - ؟

الگوریتم تقریبی

- الگوریتم (غیر تصادفی) A یک الگوریتم تقریبی با ضریب تقریب δ (الگوریتم δ_تقریب) :
  - ۱) زمان اجرا چند جمله‌ای باشد
    - زمان اجرا $\geq p(|I|)$
      - که $|I|$: اندازه ورودی
  - ۲) ضریب تقریب حداکثر δ باشد
    - برای همه $I \in \mathcal{I}$ داشته باشیم $w(A(I)) \geq \delta(f(I)) \cdot OPT(I)$
      - $f(I)$: یکی از خصوصیات ورودی
- الگوریتم تقریبی تصادفی
  - امید زمان اجرا چند جمله‌ای
  - ضریب تقریب،
    - ؟
    - برای همه $I \in \mathcal{I}$ داشته باشیم $E[w(A(I))] \geq \delta(f(I)) \cdot OPT(I)$

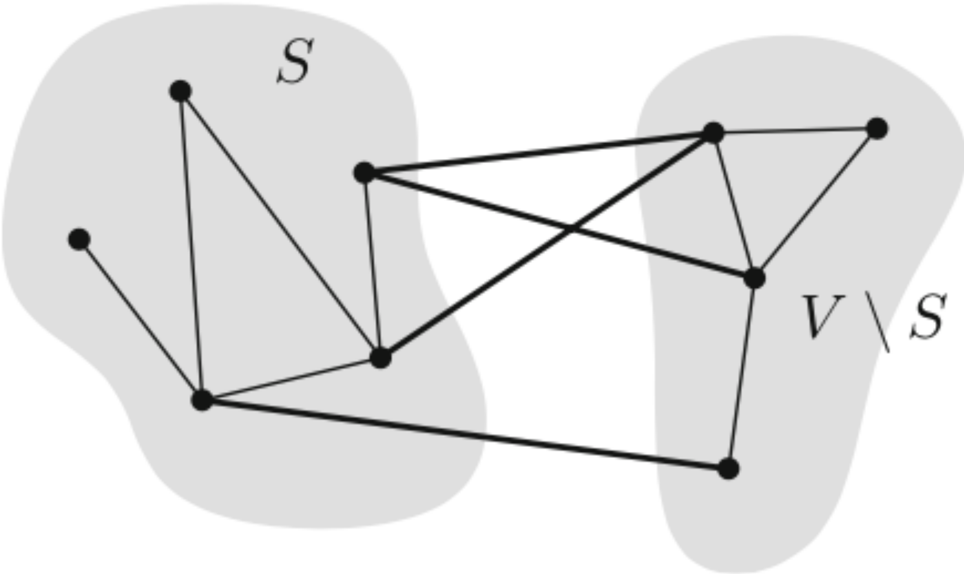
کدام ماشین؟

کدام ماشین؟

- محاسبه با اعداد حقیقی
  - کامپیوترهای موجود
  - ماشین تورینگ
  - ماشین RAM حقیقی
    - خیلی قوی‌تر
    - نباید از قوی بودن آن استفاده کنیم!

الگوریتم ۱/۲ ـ تقریب

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

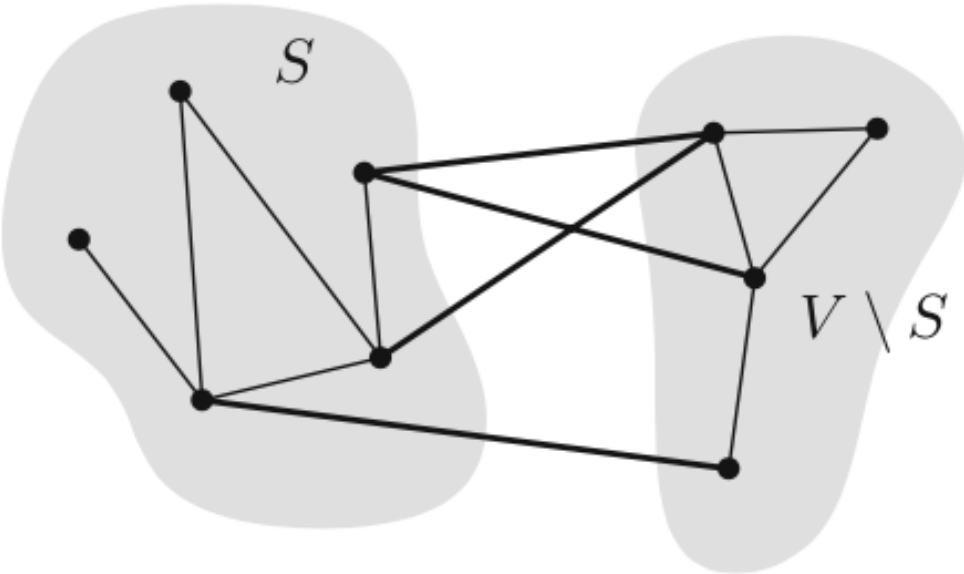


- ورودی: گراف
- خروجی: برش (S, V/S)
- هدف: بیشینه کردن

$$E(S, V \setminus S) =$$

$$\{e \in E : |e \cap S| = |e \cap (V \setminus S)| = 1\}$$

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه



- ورودی: گراف
- خروجی: برش (S, V/S)
- هدف: بیشینه کردن

$$E(S, V \setminus S) =$$
$$\{e \in E : |e \cap S| = |e \cap (V \setminus S)| = 1\}$$

- الگوریتم تصادفی:
  - هر راس را به احتمال ۱/۲ در S قرار بده

الگوریتم $\frac{1}{2}$-تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$-تقریب برای مسئله برش بیشینه است.

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:

الگوریتم $\frac{1}{2}$-تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$-تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:
    - امید تعداد یال‌های برش

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:
    - امید تعداد یال‌های برش

$$\mathbf{E}\left[\omega(\mathtt{RandomizedMaxCut}(G))\right] = \mathbf{E}\left[|E(S, V \setminus S)|\right]$$

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:
    - امید تعداد یال‌های برش

$$\begin{aligned}
\mathbf{E}\left[\omega(\mathtt{RandomizedMaxCut}(G))\right] &= \mathbf{E}\left[|E(S, V\setminus S)|\right] \\
&= \sum_{e\in E} \mathrm{Prob}[e \in E(S, V\setminus S)]
\end{aligned}$$

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:
    - امید تعداد یال‌های برش

$$\begin{aligned} \mathbf{E}\left[\omega(\texttt{RandomizedMaxCut}(G))\right] &= \mathbf{E}\left[|E(S, V \setminus S)|\right] \\ &= \sum_{e \in E} \mathrm{Prob}[e \in E(S, V \setminus S)] \\ &= \sum_{e \in E} \tfrac{1}{2} = \tfrac{1}{2}|E| \end{aligned}$$

الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه

- الگوریتم RandomizedMaxCut:
  - هر راس را به احتمال ۱/۲ در S قرار بده

- قضیه: الگوریتم بالا، یک الگوریتم $\frac{1}{2}$ـ تقریب برای مسئله برش بیشینه است.
- اثبات:
  - زمان اجرا
  - ضریب تقریب:
    - امید تعداد یال‌های برش

$$
\begin{aligned}
\mathbf{E}\left[\omega(\texttt{RandomizedMaxCut}(G))\right] &= \mathbf{E}\left[|E(S, V \setminus S)|\right] \\
&= \sum_{e \in E} \mathrm{Prob}[e \in E(S, V \setminus S)] \\
&= \sum_{e \in E} \tfrac{1}{2} = \tfrac{1}{2}|E| \geq \tfrac{1}{2}\mathsf{Opt}(G).
\end{aligned}
$$

الگوریتم گومنز_ ویلیامسون

- مرحله ۱: برنامه‌ریزی ریاضی برای مسئله
- مرحله ۲: آرام‌سازی
- مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین
- مرحله ۴: گرد کردن (محاسبه جواب نهایی)
- مرحله ۵: محاسبه اتلاف گرد کردن

پیش‌نیاز

- ماتریس نیمه معین
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر
    - به ازای هر x داشته باشیم $x^\top A x \geq 0$

پیش‌نیاز

- ماتریس نیمه معین
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر
    - به ازای هر x داشته باشیم $\mathrm{x}^\top A \mathrm{x} \geq 0$
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر و فقط اگر
    - بتوان ماتریس U را یافت که $A = U^\top U$

پیش‌نیاز

- ماتریس نیمه معین
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر
    - به ازای هر x داشته باشیم $\mathrm{x}^\top A \mathrm{x} \geq 0$
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر و فقط اگر
    - بتوان ماتریس U را یافت که $A = U^\top U$
- برنامه‌ریزی نیمه‌معین را می‌توان سریع با تقریب حل کرد

2.4.1 Definition. *A semidefinite program in equational form is the following kind of optimization problem:*

$$\begin{array}{ll} \text{Maximize} & \sum_{i,j=1}^{n} c_{ij} x_{ij} \\ \text{subject to} & \sum_{i,j=1}^{n} a_{ijk} x_{ij} = b_k, \quad k = 1, \ldots, m, \\ & X \succeq 0, \end{array} \tag{2.2}$$

where the x_{ij}, $1 \leq i, j \leq n$, are n^2 variables satisfying the symmetry conditions $x_{ji} = x_{ij}$ for all i, j, the c_{ij}, a_{ijk} and b_k are real coefficients, and

$$X = (x_{ij})_{i,j=1}^{n} \in \mathrm{SYM}_n.$$

پیش‌نیاز

- ماتریس نیمه معین
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر
    - به ازای هر x داشته باشیم $\mathrm{x}^\top A \mathrm{x} \geq 0$
  - ماتریس حقیقی و متقارن A مثبت نیمه‌معین است، اگر و فقط اگر
    - بتوان ماتریس U را یافت که $A = U^\top U$
- برنامه‌ریزی نیمه‌معین را می‌توان سریع با تقریب حل کرد

جواب ما $\geq \mathrm{OPT} - \epsilon$

2.4.1 Definition. *A semidefinite program in equational form is the following kind of optimization problem:*

$$\begin{array}{ll} \text{Maximize} & \sum_{i,j=1}^{n} c_{ij} x_{ij} \\ \text{subject to} & \sum_{i,j=1}^{n} a_{ijk} x_{ij} = b_k, \quad k = 1, \ldots, m, \\ & X \succeq 0, \end{array} \tag{2.2}$$

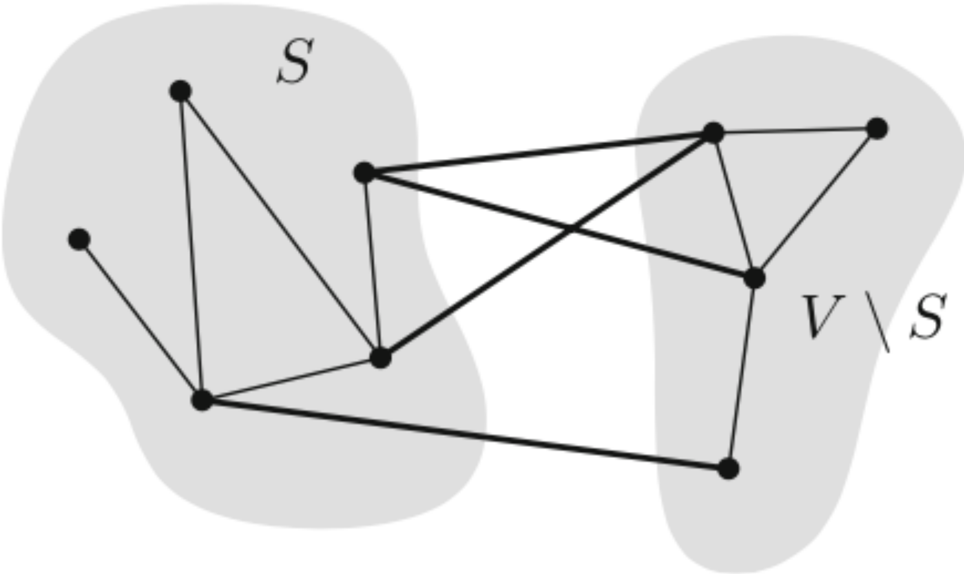
where the x_{ij}, $1 \leq i, j \leq n$, are n^2 variables satisfying the symmetry conditions $x_{ji} = x_{ij}$ for all i, j, the c_{ij}, a_{ijk} and b_k are real coefficients, and

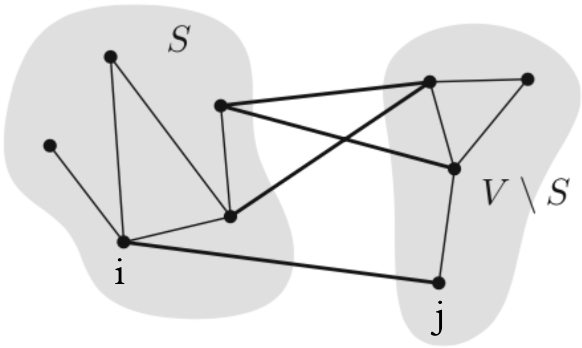
$$X = (x_{ij})_{i,j=1}^{n} \in \mathrm{SYM}_n.$$

مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

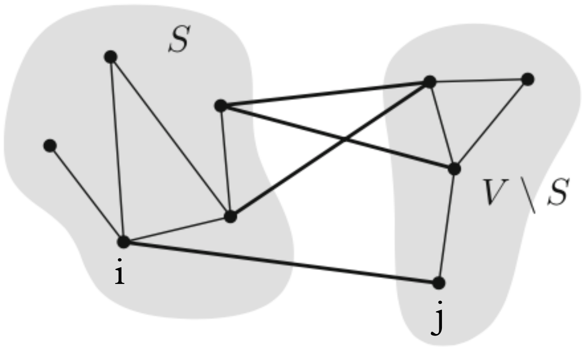
- متغیر
- قید
- تابع هدف





مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

- متغیر
- قید
- تابع هدف



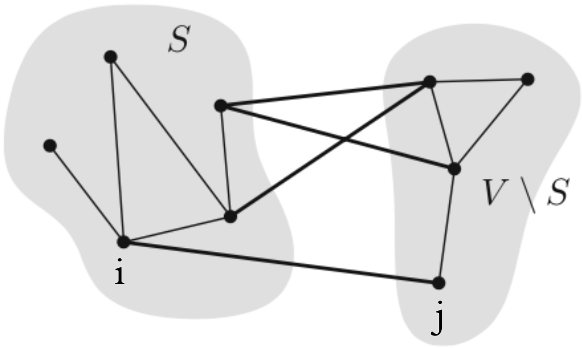
مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

- متغیر

$$z_1, z_2, \ldots, z_n$$

- قید

- تابع هدف



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

- متغیر

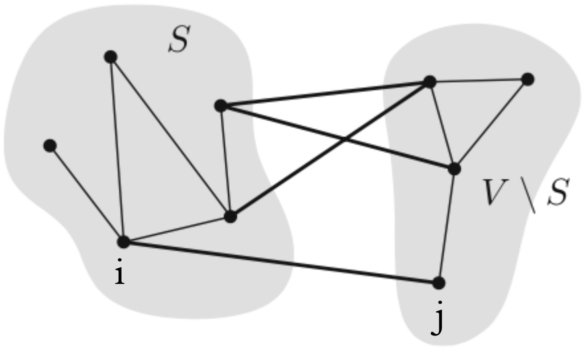
$$z_1, z_2, \ldots, z_n$$

- قید

$$z_1, z_2, \ldots, z_n \in \{-1, 1\}$$

- تابع هدف

مرحله ۱: برنامه‌ریزی ریاضی برای مسئله



- متغیر

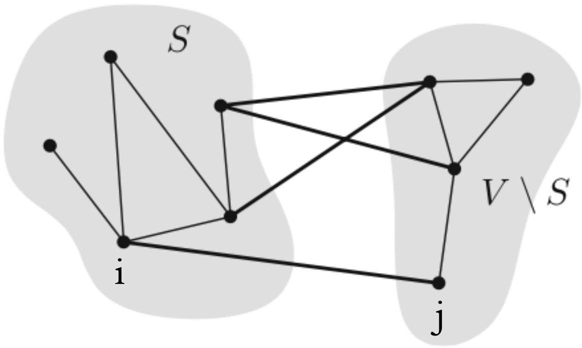
$$z_1, z_2, \ldots, z_n$$

- قید

$$z_1, z_2, \ldots, z_n \in \{-1, 1\}$$

- تابع هدف

$$z_i z_j$$



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

- متغیر

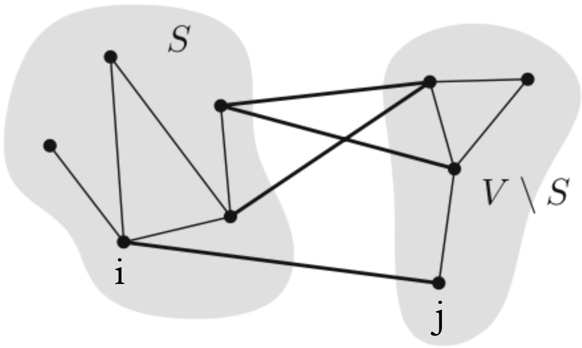
$$z_1, z_2, \ldots, z_n$$

- قید

$$z_1, z_2, \ldots, z_n \in \{-1, 1\}$$

- تابع هدف

$$\frac{1 - z_i z_j}{2}$$



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

- متغیر

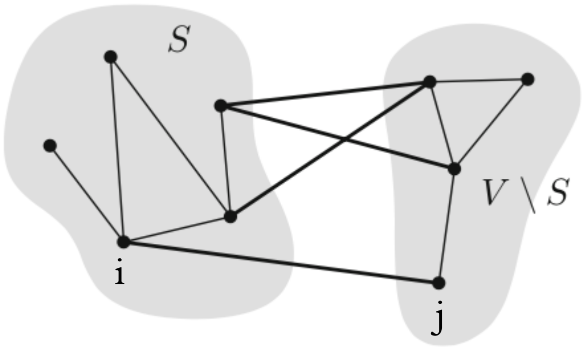
$$z_1, z_2, \dots, z_n$$

- قید

$$z_1, z_2, \dots, z_n \in \{-1, 1\}$$

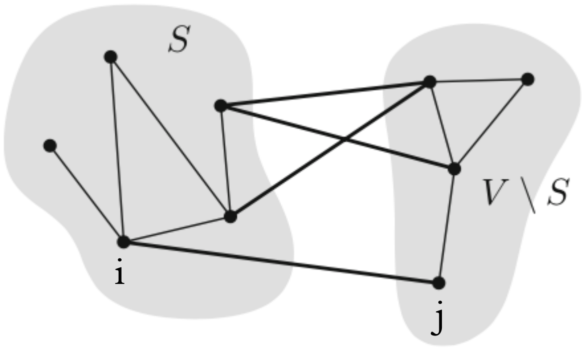
- تابع هدف

$$\sum_{\{i,j\}\in E} \frac{1 - z_i z_j}{2}$$



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

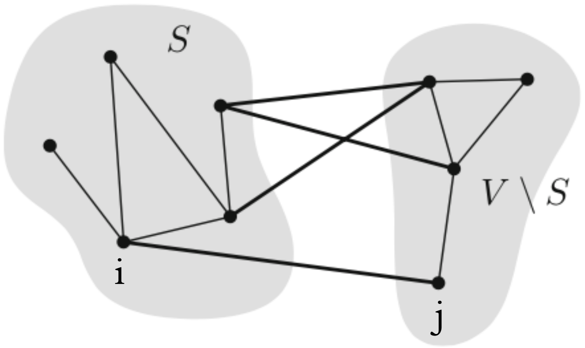
$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\ \text{subject to} & z_i \in \{-1, 1\}, \quad i = 1, \ldots, n. \end{array}$$



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\ \text{subject to} & z_i \in \{-1, 1\}, \quad i = 1, \ldots, n. \end{array}$$

جواب بهینه =
Opt(G)



مرحله ۱: برنامه‌ریزی ریاضی برای مسئله

$$
\begin{array}{ll}
\text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\
\text{subject to} & z_i \in \{-1, 1\}, \quad i = 1, \ldots, n.
\end{array}
$$

جواب بهینه =
Opt(G)

سخت!
NP-سخت

مرحله ۲:

آرام‌سازی

مرحله ۲ : آرام‌سازی

$$
\begin{array}{ll}
\text{Maximize} & \sum_{\{i,j\}\in E} \frac{1 - z_i z_j}{2} \\
\text{subject to} & z_i \in \{-1, 1\}, \quad i = 1, \ldots, n.
\end{array}
$$

- گسسته –> قابل حل
- جواب‌های شدنی جدید (آرام‌سازی شده) شامل جواب‌های شدنی اصلی

$$z_i$$

$$\mathbf{u}_i \in S^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$$

مرحله ۲ : آرام‌سازی

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\ \text{subject to} & z_i \in \{-1,1\}, \quad i=1,\ldots,n. \end{array}$$

- گسسته -> قابل حل
- جواب‌های شدنی جدید (آرام‌سازی شده) شامل جواب‌های شدنی اصلی
- ؟؟؟

$$z_i$$

$$\Downarrow$$

$$\mathbf{u}_i \in S^{n-1} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| = 1\}$$

مرحله ۲ : آرام‌سازی

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\ \text{subject to} & z_i \in \{-1,1\}, \quad i=1,\ldots,n. \end{array}$$

- گسسته -> قابل حل
- جواب‌های شدنی جدید (آرام‌سازی شده) شامل جواب‌های شدنی اصلی
- ؟؟؟
- ایده:
-

$$\boxed{\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T \mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i=1,2,\ldots,n. \end{array}}$$

مرحله ۲ : آرام‌سازی

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-z_i z_j}{2} \\ \text{subject to} & z_i \in \{-1,1\}, \quad i=1,\ldots,n. \end{array}$$

- گسسته ــ> قابل حل
- جواب‌های شدنی جدید (آرام‌سازی شده) شامل جواب‌های شدنی اصلی
- ؟؟؟
- ایده:
  -

$$\boxed{\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T \mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i=1,2,\ldots,n. \end{array}}$$

- آرام‌سازی است ⟵ جواب آرام‌سازی $\leq \mathrm{OPT}(G)$

مرحله ۳:

تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$x_{ij} = \mathbf{u}_i^T \mathbf{u}_j$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$x_{ij} = \mathbf{u}_i^T\mathbf{u}_j$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$x_{ij} = \mathbf{u}_i^T \mathbf{u}_j$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

- **قضیه**: مقدار هر دو برنامه‌ریزی مساوی است

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

$$x_{ij} = \mathbf{u}_i^T\mathbf{u}_j$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

- **قضیه**: مقدار هر دو برنامه‌ریزی مساوی است

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T \mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- جواب اولی ‎->‎ جواب دومی: $x_{ij} = \mathbf{u}_i^T \mathbf{u}_j$
    - X مثبت نیمه‌معین است $(X = U^\top U)$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

- **قضیه**: مقدار هر دو برنامه‌ریزی مساوی است

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i=1,2,\ldots,n. \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii}=1, \quad i=1,2,\ldots,n, \\ & X \succeq 0. \end{array}$$

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

- **قضیه**: مقدار هر دو برنامه‌ریزی مساوی است

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- جواب دومی ‪–>‬ جواب اول:
  - X مثبت نیمه‌معین است ($X = U^\top U$) ‪–>‬ ماتریس U ‪–>‬ ستون‌های U

مرحله ۳: تبدیل آرام‌سازی به برنامه‌ریزی نیمه‌معین

- **قضیه**: مقدار هر دو برنامه‌ریزی مساوی است

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- جواب دومی <-> جواب اول.

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$
\begin{array}{ll}
\text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\
\text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\
& X \succeq 0.
\end{array}
$$

- پاسخ تقریبی را محاسبه می‌کنیم

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- پاسخ تقریبی را محاسبه می‌کنیم

$$\mathrm{OPT}_{\mathrm{SDP}} - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$
\begin{array}{ll}
\text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\
\text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\
 & X \succeq 0.
\end{array}
$$

- پاسخ تقریبی را محاسبه می‌کنیم

$$\mathrm{OPT}_{\mathrm{SDP}} - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

$$\mathrm{OPT}(G) \leq \mathrm{OPT}_{\mathrm{SDP}}$$

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- پاسخ تقریبی را محاسبه می‌کنیم

$$\mathrm{OPT}_{\mathrm{SDP}} - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - \mathbf{u}_i^{*T} \mathbf{u}_j^*}{2}$$

$$\mathrm{OPT}(G) \leq \mathrm{OPT}_{\mathrm{SDP}}$$

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\} \in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- پاسخ تقریبی را محاسبه می‌کنیم

$$\mathrm{OPT}_{\mathrm{SDP}} - \epsilon \le \sum_{\{i,j\} \in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

$$\mathrm{OPT}(G) \le \mathrm{OPT}_{\mathrm{SDP}}$$

$$\mathrm{OPT}(G) - \epsilon \le \sum_{\{i,j\} \in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

تاکنون

- ورودی: گراف G
- برنامه‌نویسی نیمه‌معین زیر را می‌سازیم

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-x_{ij}}{2} \\ \text{subject to} & x_{ii} = 1, \quad i = 1, 2, \ldots, n, \\ & X \succeq 0. \end{array}$$

- پاسخ تقریبی را محاسبه می‌کنیم

$$\mathrm{OPT}_{\mathrm{SDP}} - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

$$\mathrm{OPT}(G) \leq \mathrm{OPT}_{\mathrm{SDP}}$$

$$\Leftarrow$$

$$\mathrm{OPT}(G) - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

- مرحله بعد: محاسبه جواب (برش)

مرحله ۴: گرد کردن)محاسبه جواب نهایی(

مرحله ۴: گرد کردن (محاسبه جواب نهایی)

- داریم:

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^{*T}\mathbf{u}_j^*}{2} \geq \mathsf{Opt}(G) - \varepsilon$$

- می‌خواهیم: تبدیل u به ۱+ یا ۱-

چگونه؟

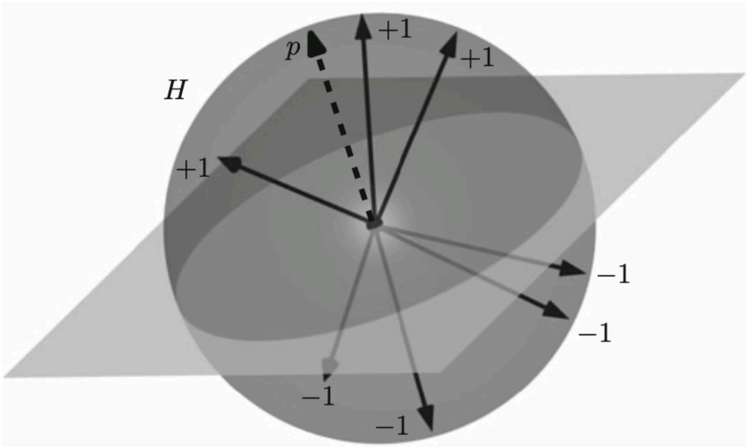
مرحله ۴: گرد کردن (محاسبه جواب نهایی)

- داریم:

$$\begin{array}{ll} \text{Maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T \mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

$$\sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2} \geq \mathsf{Opt}(G) - \varepsilon$$

- می‌خواهیم: تبدیل u به ۱+ یا ۱-
  - انتخاب صفحه تصادفی روی کره
  - دو طرف صفحه --> برش
  - چرا خوب است؟



- انتخاب صفحه: انتخاب بردار p عمود بر صفحه
- دو طرف صفحه:

$$\mathbf{u} \mapsto \begin{cases} 1 & \text{if } \mathbf{p}^T\mathbf{u} \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

- انتخاب بردار تصادفی روی کره با توزیع یکنواخت؟
  - توزیع چند متغیره نرمال، نرمال‌شده
  - D: توزیع نرمال چند متغیره مستقل
  - $\tilde{p} \sim D$
  - $f(\tilde{p}) = \frac{1}{(2\pi)^{-k/2}} \exp(\|\tilde{p}\|^2)$
  - $p = \tilde{p}/\|p\|$

1. Compute an almost optimal solution $\mathbf{u}_1^*, \mathbf{u}_2^*, \ldots, \mathbf{u}_n^*$ of the vector program

$$\begin{array}{ll} \text{maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

   This is a solution that satisfies

$$\sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^{*T}\mathbf{u}_j^*}{2} \geq \mathsf{SDP}(G) - 5\cdot 10^{-4} \geq \mathsf{Opt}(G) - 5\cdot 10^{-4},$$

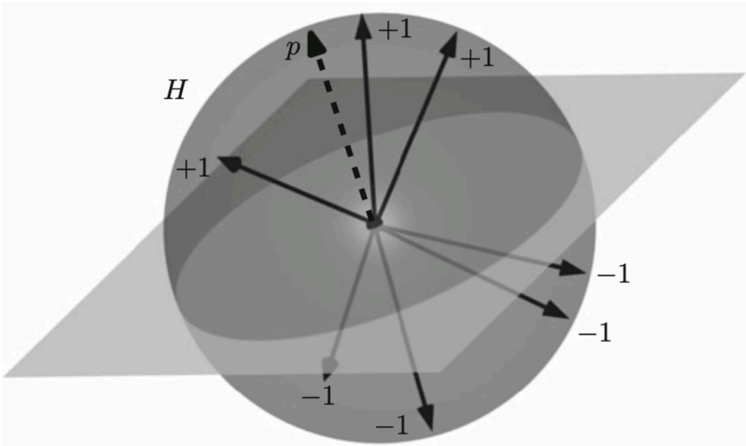
   and it can be found in polynomial time by semidefinite programming and Cholesky factorization.
2. Choose $\mathbf{p} \in S^{n-1}$ uniformly at random, and output the cut induced by

$$S := \{i \in \{1, 2, \ldots, n\} \colon \mathbf{p}^T\mathbf{u}_i^* \geq 0\}.$$

- انتخاب صفحه: انتخاب بردار p عمود بر صفحه
- دو طرف صفحه:

$$\mathbf{u} \mapsto \begin{cases} 1 & \text{if } \mathbf{p}^T\mathbf{u} \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

-



ما = $\sum_{\{i,j\}\in E} \mathbb{P}[\text{sgn}(pu_i) \neq \text{sgn}(pu_i')]$

$$\text{OPT}(G) - \epsilon \leq \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}$$

مرحله ۵: محاسبه اتلاف گرد کردن

مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that* (1.5) *maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

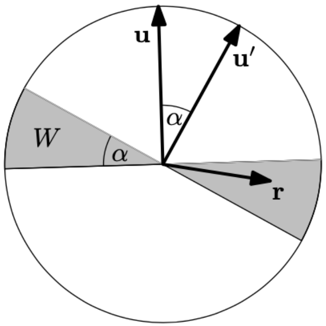
$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that* (1.5) *maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

- اثبات:
  - r: تصویر p روی صفحه u و u’



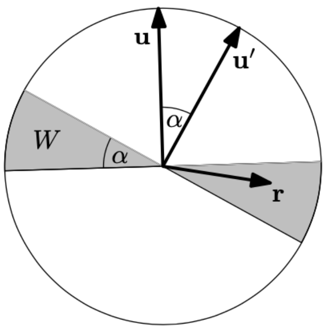
مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$*. The probability that (1.5) maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

- اثبات:
  - r: تصویر p روی صفحه u و u’

$$\mathbf{u} \quad \mapsto \quad \begin{cases} 1 & \text{if } \mathbf{p}^T \mathbf{u} \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$



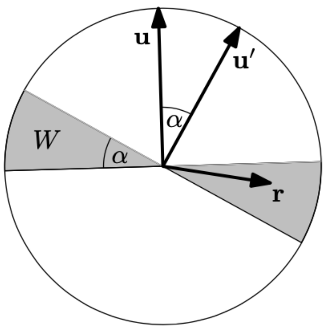
مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that* (1.5) *maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

- اثبات:
  - r: تصویر p روی صفحه u و u’
  - pu و ru هم علامتند
  - pu’ و ru’ هم علامتند

$$\mathbf{u} \mapsto \begin{cases} 1 & \text{if } \mathbf{p}^T \mathbf{u} \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$



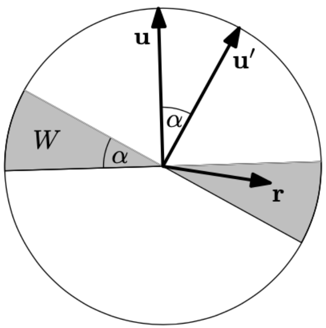
مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that (1.5) maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

- اثبات:
  - r: تصویر p روی صفحه u و u’
  - pu و ru هم علامتند
  - pu’ و ru’ هم علامتند
  - r: زاویه‌ای با توزیع یکنواخت دارد.

$$\mathbf{u} \mapsto \begin{cases} 1 & \text{if } \mathbf{p}^T\mathbf{u} \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$



مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that (1.5) maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

ما = $\sum_{\{i,j\} \in E} \mathbb{P}[\operatorname{sgn}(pu_i) \neq \operatorname{sgn}(pu_i')] = \sum_{\{i,j\} \in E} \frac{\arccos {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{\pi}$

$$\sum_{\{i,j\} \in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2} \geq \mathsf{Opt}(G) - \varepsilon$$

مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that* (1.5) *maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

ما = $\sum_{\{i,j\} \in E} \mathbb{P}[\mathrm{sgn}(pu_i) \neq \mathrm{sgn}(pu_i')]$ = $\sum_{\{i,j\} \in E} \frac{\arccos \mathbf{u}_i^{*T} \mathbf{u}_j^*}{\pi}$

؟

$$\sum_{\{i,j\} \in E} \frac{1 - \mathbf{u}_i^{*T} \mathbf{u}_j^*}{2} \geq \mathsf{Opt}(G) - \varepsilon$$

مرحله ۵: محاسبه اتلاف گرد کردن

1.4.1 Lemma. *Let* $\mathbf{u}, \mathbf{u}' \in S^{n-1}$. *The probability that* (1.5) *maps* $\mathbf{u}$ *and* $\mathbf{u}'$ *to different values is*

$$\frac{1}{\pi} \arccos \mathbf{u}^T \mathbf{u}'.$$

ما = $\sum_{\{i,j\}\in E} \mathbb{P}[\mathrm{sgn}(pu_i) \neq \mathrm{sgn}(pu'_i)]$ $= \sum_{\{i,j\}\in E} \frac{\arccos {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{\pi}$

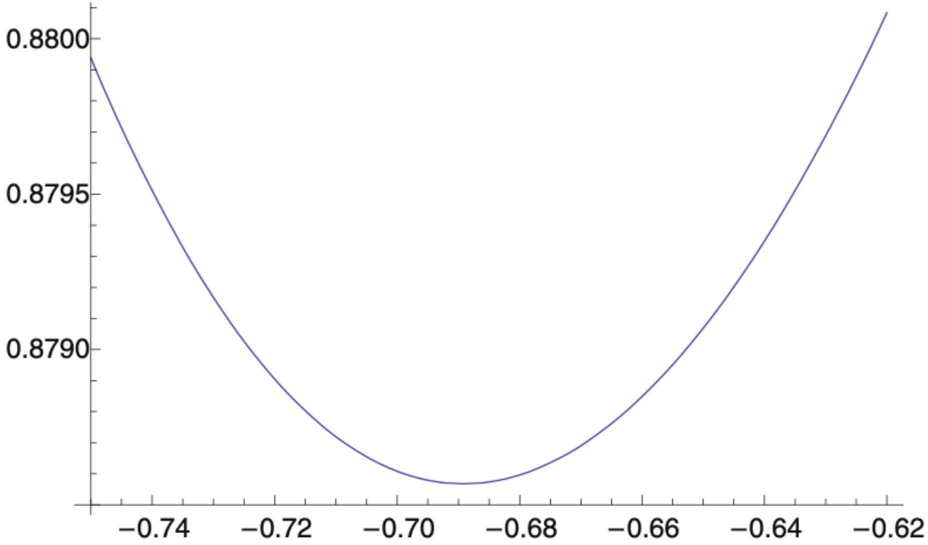
$$\sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2} \geq \mathsf{Opt}(G) - \varepsilon$$

؟

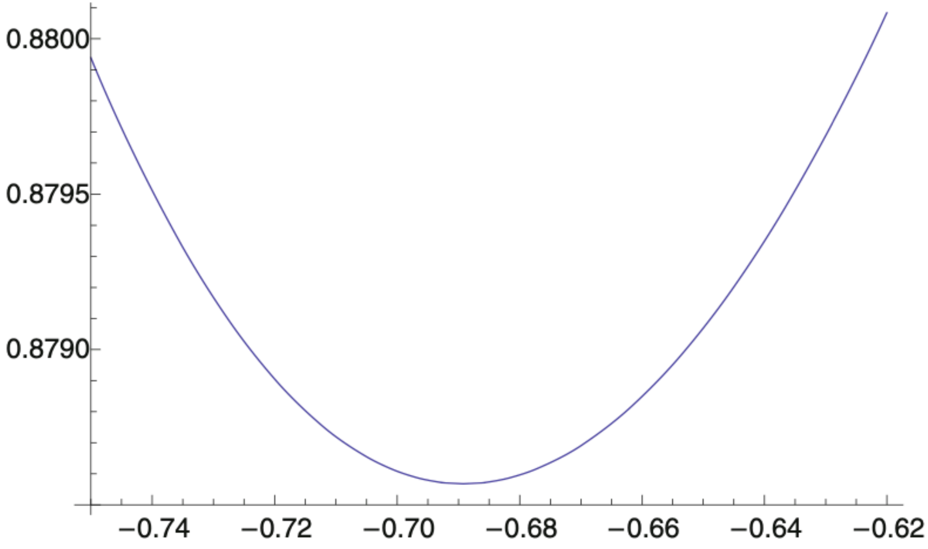
$$\frac{\arccos(z)}{\pi} \geq \delta \; \frac{1-z}{2}.$$

$$z \in [-1, 1]$$

$$f(z) = \frac{2\arccos(z)}{\pi(1-z)}$$



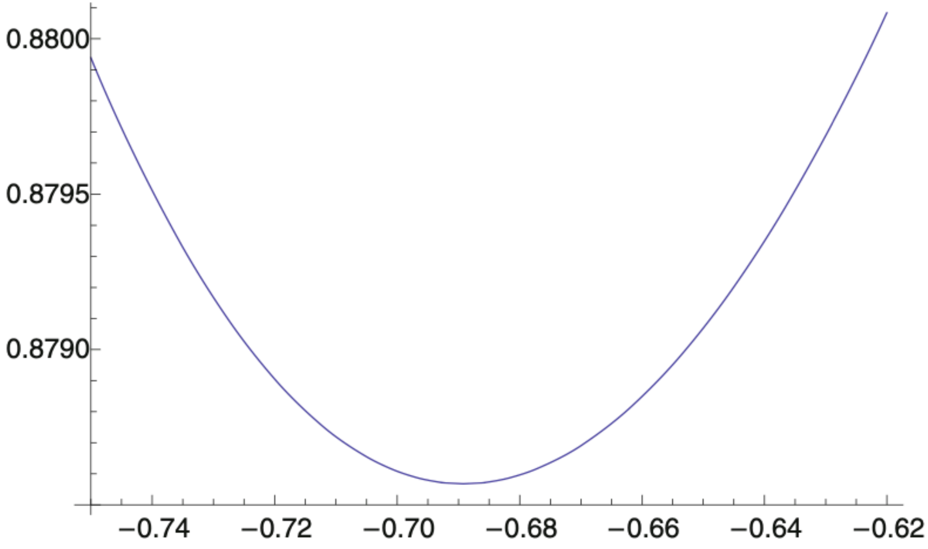
$$f(z) = \frac{2\arccos(z)}{\pi(1-z)}$$



$$f(z) = \frac{2\arccos(z)}{\pi(1-z)}$$

$$\boxed{z^* \approx -0.68915773665}$$

$$f(z^*) \approx 0.87856720578 > 0.8785672$$



$$f(z) = \frac{2\arccos(z)}{\pi(1-z)}$$

$$\boxed{z^* \approx -0.68915773665}$$

$$f(z^*) \approx 0.87856720578 > 0.8785672$$

1.4.2 Lemma. *For all* $z \in [-1, 1]$,

$$\frac{\arccos(z)}{\pi} \geq 0.8785672 \, \frac{1-z}{2}.$$

$$\text{ما} = \sum_{\{i,j\}\in E} \frac{\arccos \mathbf{u}_i^{*T} \mathbf{u}_j^*}{\pi} \geq 0.8785672 \sum_{\{i,j\}\in E} \frac{1 - \mathbf{u}_i^{*T} \mathbf{u}_j^*}{2}$$

$$
\text{ما} = \sum_{\{i,j\}\in E} \frac{\arccos {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{\pi} \geq 0.8785672 \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2}
$$

$$
\geq 0.8785672(\mathsf{Opt}(G) - \varepsilon)
$$

$$
\begin{aligned}
\text{ما} = \sum_{\{i,j\}\in E} \frac{\arccos {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{\pi} &\geq 0.8785672 \sum_{\{i,j\}\in E} \frac{1 - {\mathbf{u}_i^*}^T \mathbf{u}_j^*}{2} \\
&\geq 0.8785672(\mathsf{Opt}(G) - \varepsilon) \\
&\geq 0.878\mathsf{Opt}(G),
\end{aligned}
$$

جمع‌بندی

1.4.3 Theorem. *Algorithm* `GWMaxCut` *is a randomized* 0.878*-approximation algorithm for the* MaxCut *problem.*

1. Compute an almost optimal solution $\mathbf{u}_1^*, \mathbf{u}_2^*, \ldots, \mathbf{u}_n^*$ of the vector program

$$\begin{array}{ll} \text{maximize} & \sum_{\{i,j\}\in E} \frac{1-\mathbf{u}_i^T\mathbf{u}_j}{2} \\ \text{subject to} & \mathbf{u}_i \in S^{n-1}, \quad i = 1, 2, \ldots, n. \end{array}$$

   This is a solution that satisfies

$$\sum_{\{i,j\}\in E} \frac{1-{\mathbf{u}_i^*}^T\mathbf{u}_j^*}{2} \geq \mathsf{SDP}(G) - 5\cdot 10^{-4} \geq \mathsf{Opt}(G) - 5\cdot 10^{-4},$$

   and it can be found in polynomial time by semidefinite programming and Cholesky factorization.
2. Choose $\mathbf{p} \in S^{n-1}$ uniformly at random, and output the cut induced by

$$S := \{i \in \{1, 2, \ldots, n\} : \mathbf{p}^T\mathbf{u}_i^* \geq 0\}.$$

پایان