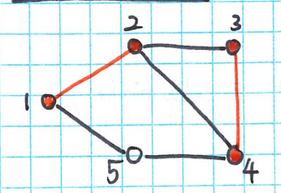
Vertex Cover problem is NP-hard.

Algo. A: Find a maximal matching M，将 M 中所有端点(边的)添加到 I.

Theorem 1.1 Algo. A is a 2-approximation algo.



红色边组成 maximal matching. $M=\{(1,2),(3,4)\}$.

Algo. A 给出的 solution $I=\{1,2,3,4\}$.

Proof. 1. Polynomial. one by one 选边，直到不能再添加.

2. Feasible. Assume edge e is not covered, 将 e 加入 M 将得到一个 bigger matching →← M is maximal.

3. $\leq$ 2-opt. ① 显然得 $|I|=2|M|$.

② 任何解都少有 M 中每条边的至少一个端点.

含opt.	$|M|$	$\geq$	$\cdot 1$
$\frac{1}{2}$opt.			

$\Rightarrow opt \geq |M| \cdot 1$

$\Rightarrow I = 2|M| \leq 2opt.$

ILP Formulation (NP-hard)

$\min Z=\sum_{j=1}^{m} x_j$

s.t. $x_i+x_j \geq 1 \quad \forall (i,j)\in E$

$x_j \in \{0,1\} \quad \forall j \in V.$

LP-relaxation (can be solved efficiently)

$\min Z=\sum_{j=1}^{m} x_j$

s.t. $x_i+x_j \geq 1 \quad \forall (i,j)\in E$

$x_j \geq 0 \quad \forall j \in V.$

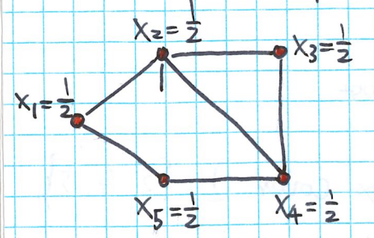
For minimization problem, $Z_{LP} \leq Z_{ILP}$.

solve LP then round that solution in a feasible solution for IP.

This technique is called LP-rounding

Algo. B: step 1: Solve the LP. $\rightarrow Z^*_{LP}=(x_1^*, x_2^*, \cdots, x_m^*)$

step 2: Add j to solution I if $x_j^* \geq \frac{1}{2}$



An opt LP-solution is $x_j^*=\frac{1}{2}$ for all j. $Z^*=2.5$.

which is strictly smaller than opt ILP-solution = 3.

Algo. B 给出的 solution $I=\{1,2,3,4,5\}$

approximation ratio $=\frac{5}{2}$.

Theorem 1.2 Algo. B is a 2-approximation algo.

Proof. 1. Polynomial. LP can be solved in polynomial time.

2. Feasible. To show 每条边都有一个端点在 I 中.

考虑任一条边 (i,j), $x_i^*+x_j^* \geq 1$. Either $x_i^* \geq \frac{1}{2}$ or $x_j^* \geq \frac{1}{2}$ (or both), which means either i or j or both is added to I.

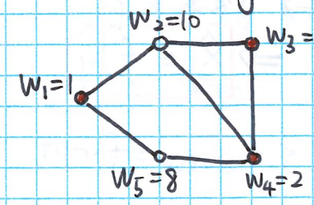
3. $\leq$ 2-opt. Denote rounded solution $\hat{x}_j=\begin{cases}1, & \text{if } x_j^* \geq \frac{1}{2} \\ 0, & \text{otherwise.}\end{cases} \Rightarrow \hat{x}_j \leq 2x_j^*$

$$|I|=\sum_{j=1}^{m} \hat{x}_j \leq 2\sum_{j=1}^{m} x_j^* = 2Z^*_{LP} \leq 2Z^*_{ILP} = 2opt.$$

Weighted case.

each vertex j has a given weight $w_j>0$ and goal is to minimize the total weight of the cover.

Algo. A does not go through, however Algo. B does apply with only some minor changes: an extra term w_j.



(LP) $\min Z=\sum_{j=1}^{m} w_j x_j$

s.t. $x_i+x_j \geq 1 \quad \forall (i,j)\in E$

$x_j \geq 0 \quad \forall j \in V.$

$opt = 1+1+2 = 4$

$$\sum_{j\in I} w_j = \sum_{j=1}^{m} w_j \hat{x}_j \leq 2\sum_{j=1}^{m} w_j x_j^* = 2Z^*_{LP} \leq 2Z^*_{ILP} = 2opt.$$

1.3 A deterministic rounding algorithm.

现在考虑 Set Cover problem.

Set Cover

Instance: Set of elements $E=\{e_1, \cdots, e_n\}$

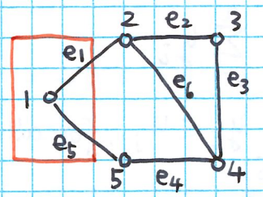
subset $S_1, \cdots S_m \subseteq E$, and

weights $w_1, \cdots, w_m \geq 0$.

Solution: $I \subseteq \{1,2,\cdots,m\}$, s.t each element is covered: $\bigcup_{j\in I} S_j = E$.

Cost: Weight of the cover: $\sum_{j\in I} w_j$

Goal: Find a solution of min cost.



Graph G is an instance of VC, and we can write it as a SC.

For each vertex j there is a set S_j containing the adjacent edges $S_1=\{e_1,e_5\}$, $S_2=\{e_1,e_2,e_6\}$, $S_3=\{e_2,e_3\}$, $S_4=\{e_3,e_4,e_6\}$, and $S_5=\{e_4,e_5\}$.

Vertex Cover 是 Set Cover 的一个特例 (~~each~~ each element e_i appears in at most 2 sets.

each element is in at most f sets

Assume. 在SC中，每个element最多出现在 f 个set. f 是常数

Lp-rounding.

(ILP) $\min Z = \sum_{j=1}^{m} w_j x_j$

s.t. $\sum_{j: e_i \in S_j} x_j \geq 1$, for all $i=1,\cdots,n$

$x_j \in \{0,1\}$, for all $j=1,\cdots,m$.

Lp-relaxation

(LP) $\min Z = \sum_{j=1}^{m} w_j x_j$

s.t. $\sum_{j: e_i \in S_j} x_j \geq 1$, for all $i=1,\cdots,n$

$x_j \geq 0$, for all $j=1,\cdots,m$

Algo. 1 step 1: solve the LP. $Z^*_{LP} = (x_1^*, x_2^*, \cdots, x_m^*)$

step 2: Add j to solution I if $x_j^* \geq \frac{1}{f}$

Theorem 1.3 Algo.1 is an f-approximation algo for Set Cover.

Proof 1. Polynomial. S1: LP, polynomial, S2: linear time.

2. Feasible. 每个元素最多出现在 f 个集合中

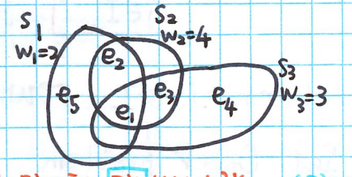
⇒对于 e_i 的constraint中最多有 f 个变量，其中至少一个变量 $\geq \frac{1}{f}$

3. $\leq f$-opt. Rounded solution $\hat{x}_j = \begin{cases} 1, & \text{if } x_j^* \geq 1/f \\ 0, & \text{otherwise.} \end{cases}$ ⇒ $\hat{x}_j \leq f x_j^*$

x-values are increased by at most a factor f

$$\sum_{j\in I} w_j = \sum_{j=1}^{m} \hat{x}_j w_j \leq f \sum_{j=1}^{m} w_j x_j^* = f Z^*_{LP} \leq f Z^*_{ILP} = f \cdot opt.$$

1.4. Rounding a dual solution



Dual of LP for set cover.

(D) $\max Z = \sum_{i=1}^{n} y_i$

s.t. $\sum_{i: e_i \in S_j} y_i \leq w_j$, for all $j=1,\cdots,m$

$y_i \geq 0$, for all $i=1,\cdots,n$.

(LP) $\min\ 2x_1+4x_2+3x_3$

s.t. $x_1+x_2+x_3 \geq 1$ (set 1)

$x_1+x_2 \geq 1$ (set 2)

$x_2+x_3 \geq 1$ (set 3)

$x_3 \geq 1$

$x_1 \geq 1$

$x_{1,2,3} \geq 0$

(D) $\max\ y_1+y_2+y_3+y_4+y_5$

set 1 $y_1+y_2+y_5 \leq 2$

set 2 $y_1+y_2+y_3 \leq 4$

set 3 $y_1+y_3+y_4 \leq 3$

$y_{1,2,3,4,5} \geq 0$

Algo. 2 step 1: solve the dual. $Z_D^* = (y_1^*, y_2^*, \cdots, y_n^*)$

step 2: Add j to solution I if the dual constraint for S_j is tight: $\sum_{i: e_i \in S_j} y_i^* = w_j$.

Theorem 1.4 Algo. 2 is an f-approximation algo. for SC.

Proof 1. Polynomial. ✓

2. Feasible. Assume not, that means 有些元素 e_i 没有被覆盖.

But none of the dual constraints j for which $i \in S_j$ is tight. Hence, 给 y_i^* 增加一点点, maintain a feasible dual sol.

Increase the dual objective value ↛ dual sol. is opt.

3. $\leq f$-opt. $\sum_{j\in I} w_j = \sum_{j\in I} \sum_{i: e_i \in S_j} y_i^* \leq f \sum_{i=1}^{n} y_i^* = f Z_D^* = f Z_{LP}^* \leq f \cdot opt.$

① ② ③

Remark: in the proof, optimality of y^* is not really needed.

Note that for any feasible solution y, $\sum_{i=1}^{n} y_i \leq Z_D^* \leq OPT$.

① only tight set S_j were picked by the algo: $\sum_{i: e_i \in S_j} y_i^* = w_j$.

② each y_i^* appears at most f times in the summation.

③ Strong duality.

Algo.1 is not worse than Algo.2.

Theorem. Let I_1, I_2 be solutions by Algo.1 and Algo.2

Proof. $j \in I_1 \Rightarrow x_j^* \geq 1/f \Rightarrow x_j^* > 0 \Rightarrow \sum_{i: e_i \in S_j} y_i^* = w_j \Rightarrow j \in I_2$

1.5 The primal-dual method.

We simply start with $y_i = 0$ for all i and then increase the dual value one by one.

Algo. 3 step 1: $I = \phi$, $y_i = 0$ for all i.

step 2: 只要有 e_i is uncovered, increase y_i until some dual constraint l become tight and add j to the solution I for which the corresponding dual constraint becomes tight.

- Start with $I = \{\}$ (infeasible) and $y = 0$ (feasible)
- For $i = 1, \cdots, n$

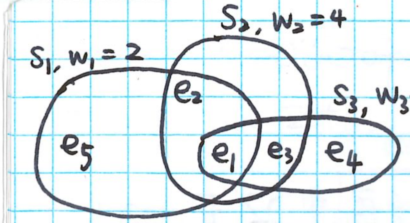
  increase y_i until some set becomes tight
- Take all tight sets in the solution.

Theorem 1.5 Algo. 3 is an f-approximation algo. for SC.

Proof 1. Polynomial. $O(fn)$: each e_i in at most f sets ✓

2. Feasible. each element in a tight set.

3. $\leq f$-opt. $\sum_{j\in I} w_j = \sum_{j\in I} \sum_{i: e_i \in S_j} y_i \leq f \sum_{i=1}^{n} y_i = f Z_D \leq f Z_D^* = f Z_{LP}^* \leq f opt.$



(D)
$$\max\ y_1+y_2+y_3+y_4+y_5$$
$$\text{s.t.}\quad y_1+y_2+y_5 \le 2$$
$$y_1+y_2+y_3 \le 4$$
$$y_1+y_3+y_4 \le 3$$
$$y_{1,2,3,4,5} \ge 0$$

剩余 (remaining): $i=1$: 0, 2, 1; $i=3$: 0, 1, 0

	y_1	y_2	y_3	y_4	y_5
$i=0$	0	0	0	0	0
$i=1$	2	0	0	0	0
$i=2$	2	0	0	0	0
$i=3$	2	0	1	0	0
$i=4$	2	0	1	0	0
$i=5$	2	0	1	0	0

与 Algo.2 的区别：不需要 optimal dual value.

But that is no problem since $Z_D \le Z_D^*$.

1.6 A greedy algo.

Here, the greedy set cover algo. picks in each step the set for which the cost per newly covered element is minimum.

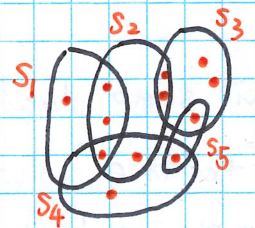
Lemma 1 Let $A_1, A_2, \cdots, A_q$ be finite sets. Then $\sum_j |A_j| \ge |\bigcup_j A_j|$

Algo. 4 (unweighted) step 1: $I=\phi$, $\hat{S}_j = S_j$ for all j.

step 2: As long as some e_i is uncovered, add that l to I for which $\hat{S}_l$ is maximal. Update $\hat{S}_j$ for all j.

Note: Let $\hat{S}_j$ be the set of uncoverd items in S_j during some state of the algo.

idea: add sets one by one, and each time take the 'cheapest' option



covered in each time	6	3	2	1

$I = \{S_2, S_3, S_4, S_1\}$.

Theorem 1.6 (unweighted) Algo. 4 is an H_n-approximation algo. for SC.

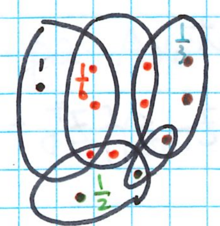
($H_n = \frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{n} \approx \ln n$ is the n-th harmonic number).

Note: Ratio depends on n, not f.

- If $f < H_n$, then Algo. 1,2,3 better than Algo. 4
- If $f > H_n$, then Algo. 1,2,3 worse than Algo. 4

Proof. 1. Divide total cost over items

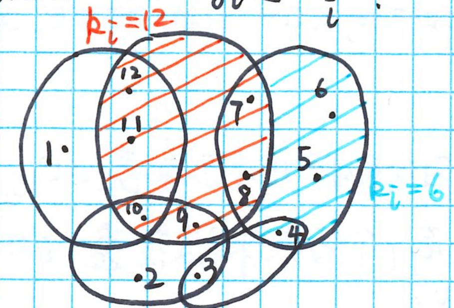
cost is $|I| = y_1 + y_2 + \cdots + y_n$, where y_i is cost for item e_i.



when a new set S_l is added to the solution, then the cost for this set (which is 1) is divided over all newly covered items. i.e. $y_i = \frac{1}{|\hat{S}_l|}$

2. Relabel, s.t. items are covered in order $e_n, e_{n-1}, \cdots, e_1$.

Show that $y_i \le \frac{opt}{i}$.



- Let k_i 表示 e_i 被覆盖前，未被覆盖的元素的数量.

  $\to k_i \ge i$.
- The k_i items can be covered by at most opt sets

  $\to$ One of these opt sets has at least k_i/opt uncovered items.①
- Greedy picks a set with at least k_i/opt uncovered items ②

  $\to y_i \le \frac{1}{k_i/opt} = opt/k_i \le opt/i$. (# of newly covered items.)

Note: ① Assume that 6 elements are covered by 3 sets, then there is a set with at least $\frac{6}{3}=2$ elements.

② eg. $e_6, e_5, e_4 = \frac{1}{3}$.

> 另一种表述. $y_i \le \frac{OPT}{k_i}$
>
> 假设任一元素 e_i 在 Algo.4 中的第二步被集合 S_l 覆盖
>
> 由算法的描述，我们可得 $|\hat{S}_l| \ge |\hat{S}_j|$，对于所有 j. (since the algo. picks the set with the max number of uncovered items.)
>
> 令 I' 为一个 optimal solution.
>
> $$OPT = |I'| = \sum_{j\in I'} \underset{\text{weight}=1}{1} \ge \sum_{j\in I'} \frac{|\hat{S}_j|}{|\hat{S}_l|} = \frac{1}{|\hat{S}_l|}\sum_{j\in I'}|\hat{S}_j| \overset{③}{\ge} \frac{1}{|\hat{S}_l|} k_i = y_i k_i$$
>
> 注：③ The inequality holds from Lemma 1: $\sum_{j\in I'}|\hat{S}_j| \ge |\bigcup_{j\in I'} \hat{S}_j| = k_i$.

3. 综上，$|I| = \sum_{i=1}^{n} y_i \le OPT \cdot \sum_{i=1}^{n} \frac{1}{k_i} \le OPT \cdot H_n$.

Now, the weighted version. The changes are minimal.

Algo. 4 (weighted) step 1: $I=\phi$, $\hat{S}_j = S_j$ for all j.

step 2: As long as some e_i is uncovered, add that l to I for which $w_l/\hat{S}_l$ is minimal.

Theorem 1.7 (weighted) Algo. 4 is an H_n-approximation algo. for SC.

Proof (sketch) $y_i = \frac{w_l}{|\hat{S}_l|}$, $\sum_{j\in I} w_j = \sum_{i=1}^{n} y_i$, $OPT = \sum_{j\in I'} w_j \ge \frac{w_l}{|\hat{S}_l|}\sum_{j\in I'}|\hat{S}_j| \ge \frac{w_l}{|\hat{S}_l|} k_i = y_i k_i$

Alternative analysis (Dual fitting)

By using the dual LP formulation of problem, one can prove an even better bound.

考虑任一集合 $S_j = \{e_1, e_2, \cdots, e_{|S_j|}\}$，假设按相反的顺序被覆盖：$e_{|S_j|}, \cdots, e_1$

考虑任一元素 e_i，$i \le |S_j|$，在 e_i 被覆盖前，有 i 个元素未被覆盖，i.e. $|\bar{S}_j| \ge i$.

假设 e_i 由 S_l 覆盖，则 $y_i = \frac{w_l}{|\bar{S}_l|} \le \frac{w_j}{|\bar{S}_j|} \le \frac{w_j}{i}$

令 $g = \max_j |S_j|$，则 $\sum_{i: e_i \in S_j} y_i \le \sum_{i=1}^{|S_j|} \frac{w_j}{i} = w_j H_{|S_j|} \le w_j H_g$.

令 $y_i' = y_i / H_g$，则 y_i' 是对偶可行解

$$\sum_{i=1}^{n} y_i = H_g \sum_{i=1}^{n} y_i' \le H_g Z_D^* = H_g Z_{LP}^* \le H_g \cdot OPT.$$

Note that this is better (or at least not worse) than the bound $H_n OPT$ since $H_g \le H_n$.

1.7. A Randomized Rounding Algo.

Algo. 5 step 1: solve LP. $Z_{LP}^* = (x_1^*, x_2^*, \cdots, x_n^*)$.

step 2: Add j to solution I with probability $\min\{1, K x_j^*\}$

Theorem 1.8 For $K = c \ln n$ and $c \ge 2$, algo. 5 is a $2c \ln n$ — approximation with high probability.

Proof. Consider an arbitrary element e_i. Let $K x_j^* \ge 1$ for some j with $e_i \in S_j$, then e_i will be covered.

Otherwise, $Pr(e_i \text{ not covered}) = \prod_{j: e_i \in S_j} (1 - K x_j^*) \le \prod_{j: e_i \in S_j} e^{-K x_j^*} = e^{-\sum_{j: e_i \in S_j} K x_j^*}$

$\le e^{-K}$ $\sum_{j: e_i \in S_j} x_j^* \ge 1$, feasible for LP. ($1 - x \le e^{-x}$)

if $K = c \ln n$, then $e^{-K} = n^{-c} \le \frac{1}{n^2}$, for $c \ge 2$

$Pr(\text{some } e_i \text{ is not covered}) \le n \cdot \frac{1}{n^2} = \frac{1}{n}$ (Union bound)

$\Rightarrow Pr(\text{solution is feasible}) \ge 1 - \frac{1}{n}$

Note: Union Bound

For event A_1 and A_2

$Pr(A_1 \cup A_2) \le Pr(A_1) + P_2(A_2)$

The expected value of the solution is

$$\sum_{j=1}^{m} Pr(j \in I)\, w_j \le \sum_{j=1}^{m} K x_j^* w_j = K Z_{LP}^*.$$

However, this expectation is also over events that the solution is infeasible. Let F denote the event that the solution is feasible and let Cost denote the cost of the solution by the algo. Then,

$$E(Cost) = E(Cost|F)\, Pr(F) + E(Cost|\text{not } F)\, Pr(\text{not } F)$$

$$\ge E(Cost|F)\, Pr(F)$$

For $n \ge 2$, $Pr(F) \ge 1 - \frac{1}{n} = \frac{1}{2}$

Thus $E(Cost|F) \le 2E(Cost) \le 2K Z_{LP}^* \le 2K OPT = 2c \ln n\, OPT$

Chapter 2. Greedy and Local Search.

2.0 Short introduction into scheduling theory.

Many scheduling problems can be described by a three field notation $\alpha|\beta|\gamma$, where

α: machine environment

β: job characteristics

γ: objective criterion to be minimized (or max.)

注意 $\sum C$ 的计算 $= C_1 + C_2 + \cdots + C_n$. 不要与 $\sum p$

① $C_1 = p_1$, $C_2 = p_1 + p_2$, $C_3 = p_1 + p_2 + p_3$, $\cdots$, $C_n = p_1 + p_2 + \cdots + p_n$. $\sum C_j = n p_1 + (n-1) p_2 + \cdots + 1 \cdot p_n$.

Some easy scheduling problems:

① $1||\cdot|\sum_j C_j$, shortest processing time (SPT) rule is optimal.

② $w_1 p_1 = w_1 \cdot 1$, $w_2(p_1+p_2) = w_2 \cdot 2$, $\cdots$, $w_n \cdot \sum p_n = w_n \cdot n$. $\sum w_j C_j = 1 \cdot w_1 + 2 \cdot w_2 + \cdots + n \cdot w_n$.

② $1||p_j = 1|\sum_j w_j C_j$, decreasing order of weights is optimal.

③ $1||\cdot|\sum_j w_j C_j$, Smith's rule (decreasing order of w_j/p_j) is optimal.

$1|r_j, pmtn|\sum_j C_j$, smallest remaining processing time (SRPT) rule is optimal

③ Assume not in Smith's order $w_1/p_1 < w_2/p_2$

swap the jobs

Then, the increase in total weighted completion time is $w_1 p_2 - w_2 p_1 < 0$.