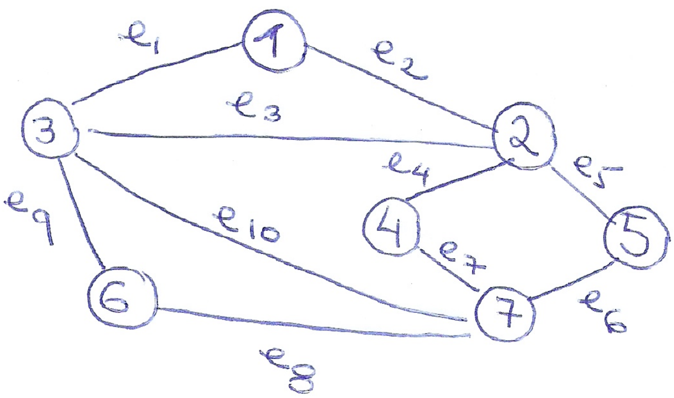
A2-1

Uma cobertura de vértices é um subconjunto $S \subseteq V$ dos vértices do grafo tal que toda aresta do grafo tem pelo menos um extremo em $S$.

Vertex Cover GJ [GT1]

Dado um grafo $G$, encontrar o conjunto de vértices que é uma cobertura de vértices e tem cardinalidade mínima.




$\{2, 3, 7\}$ é uma cobertura mínima

$\{e_1, e_5, e_8\}$ é um emparelhamento máximo

Em um grafo, sempre vale a desigualdade:

$$|M| \leq |S|,$$

onde $M$ é um emparelhamento e $S$ é uma cobertura de vértices.

Parâmetros do grafo:

$\alpha'$ = cardinalidade do emparelhamento máximo.

$\beta$ = cardinalidade da cobertura mínima.

A2-2

Algoritmo Cobertura de Vértices (G)

1. $C \leftarrow \emptyset$
2. $E' \leftarrow E$
3. enquanto $E' \neq \emptyset$
4. sejá $uv \in E'$
5. $C \leftarrow C \cup \{u, v\}$
6. remova de $E'$ toda aresta incidente a $u$ ou a $v$
7. retorne $C$.

Teorema. O algoritmo Cobertura de Vértices é uma 2-aproximação polinomial.

prova: O algoritmo sempre retorna uma cobertura porque executamos até que toda aresta tenha sido coberta pelo conjunto de vértices $C$.

Seja $C^*$ uma cobertura mínima e chame de $A$ o conjunto de arestas escolhidas na linha 4.

O conjunto $C^*$ é uma cobertura e precisa conter pelo menos um extremo de cada aresta escolhida.

O conjunto de arestas escolhidas é um emparelhamento, logo cada aresta de $A$ é coberta por um vértice diferente de $C^*$.

Obtemos o limite inferior $|C^*| \geq |A|$.

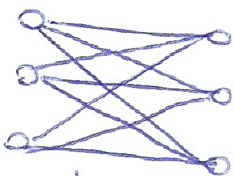
Por outro lado, cada execução da linha 4 agrega dois novos vértices a $C$, obtemos:

$$|C| = 2|A| \leq 2|C^*|.$$

∎

A nossa análise é justa?
Existe instância I cujo valor retornado pelo algoritmo $val(A(I)) = 2\,opt(I)$?

Considere o grafo bipartido completo $K_{n,n}$
Qualquer emparelhamento maximal escolhe n arestas, o que fornece C com 2n vértices e o conjunto de vértices de um lado da bipartição dá uma cobertura mínima com n vértices.



$K_{3,3}$

Para grafos bipartidos, sempre vale $\alpha' = \beta$.

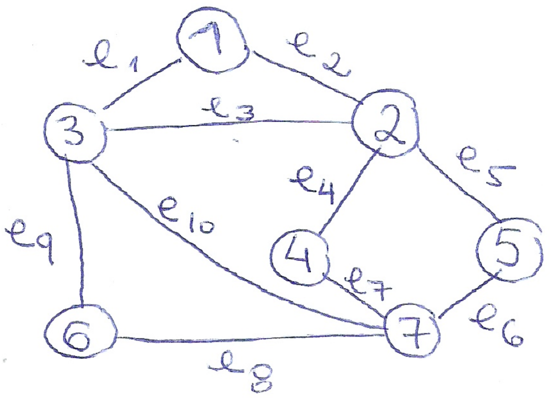
O exemplo mostra que escolher um emparelhamento máximo ao invés de maximal não é vantajoso.

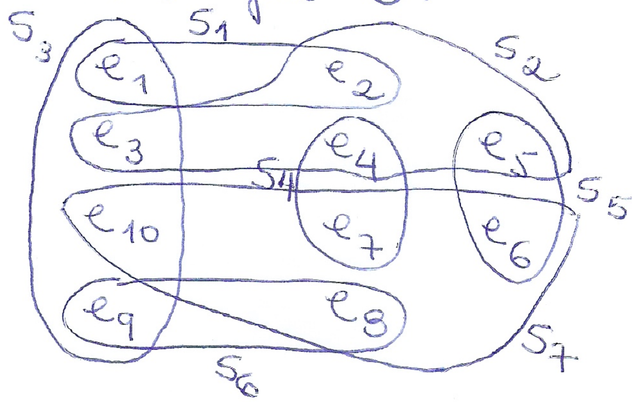
O algoritmo guloso seleciona um vértice a cada iteração.
Será que o algoritmo guloso que seleciona um vértice de grau máximo a cada iteração é melhor?

Uma cobertura de conjuntos é uma coleção de subconjuntos de um conjunto universo cuja união é o conjunto universo.

Minimum set cover:

Dados um conjunto universo $U = \{u_1, \dots, u_m\}$ e dada uma coleção de subconjuntos de $U$ $S_1, \dots, S_n$, selecionar o menor número de subconjuntos cuja união seja $U$.




Redução polinomial de Vertex Cover para Set Cover

G admite cobertura de vértices $\leq k$

se e somente se

U admite cobertura de conjuntos $\leq k$.

Na redução, dado um grafo $G(V,E)$ consideramos $U = E$ e $S_i = \{e \in E : e \text{ incidente a } v_i\}$ para $1 \leq i \leq |V|$.

A entrada de Set Cover construída é particular: todo $u \in U$ pertence a exatamente dois subconjuntos $S_i$.

Toda aresta tem exatamente dois extremos

A2-5

## Algoritmo Greedy Cobertura de Conjuntos $(X, \mathcal{F})$

$U \leftarrow X$
$\mathcal{C} \leftarrow \emptyset$
enquanto $U \neq \emptyset$
    selecione $S \in \mathcal{F}$ que maximiza $|S \cap U|$
    $U \leftarrow U \setminus S$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \{S\}$
retorne $\mathcal{C}$

No exemplo, o algoritmo seleciona $S_2$, $S_7$ e $S_3$.
O guloso seleciona a cada iteração o subconjunto que cobre o maior número de elementos de $U$ ainda não cobertos.

Para estabelecer a razão de aproximação $\alpha$, lembramos que o $d$-ésimo número harmonico

$$H_d = \sum_{i=1}^{d} \frac{1}{i} \quad \text{e} \quad H_0 = 0.$$

As somas parciais têm crescimento logaritmico:

$$\sum_{i=1}^{d} \frac{1}{i} \leq 1 + \ln d$$

Chamaremos $H_d$ de $H(d)$.

A2-6

Teorema O algoritmo Greedy Cobertura de Conjuntos é uma $\rho(n)$-aproximação polinomial, onde $\rho(n) = H(\max\{|S| : S \in \mathcal{F}\})$

prova: O número de iterações é limitado superiormente por $\min(|U|, |\mathcal{F}|)$, o que garante o tempo polinomial $O(|U||\mathcal{F}| \min(|U|, |\mathcal{F}|))$.

Atribua custo 1 a cada conjunto selecionado pelo algoritmo e distribua este custo pelos elementos cobertos pela primeira vez.

Seja $S_i$ o i-ésimo conjunto selecionado e distribua o custo de selecionar $S_i$ igualmente entre os elementos de $U$ cobertos pela primeira vez através de $S_i$.

A2-7

Seja $c_x$ o custo atribuído ao elemento $x$, para cada $x \in U$. Se $x$ é coberto pela primeira vez por $S_i$, então $c_x = \frac{1}{|S_i - (S_1 \cup S_2 \cup \ldots \cup S_{i-1})|}$

Como cada escolha de novo conjunto atribui 1 unidade de custo e cada elemento $x$ só é atribuído custo $c_x$ uma vez: $|\mathcal{C}| = \sum_{x \in U} c_x$.

Cada elemento $x \in U$ pertence a pelo menos um conjunto da cobertura ótima $\mathcal{C}^*$:

$$\sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x \geq \sum_{x \in U} c_x, \text{ portanto}$$

$$|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} \sum_{x \in S} c_x.$$

Provaremos a seguir $\boxed{\sum_{x \in S} c_x \leq H(|S|)}$ e concluímos a prova assim:

$$|\mathcal{C}| \leq \sum_{S \in \mathcal{C}^*} H(|S|) \leq |\mathcal{C}^*| \cdot H(\max_{S \in \mathcal{F}} |S|) \quad \blacksquare$$

Resta provar $\sum_{x \in S} c_x \leq H(|S|)$.

A2-8

Seja $S \in \mathcal{F}$ e $i \in \{1, 2, \dots, |\mathcal{C}|\}$.
Seja $u_i = |S - (S_1 \cup S_2 \cup \dots \cup S_i)|$, o número de elementos de $S$ ainda não cobertos, e $u_0 = |S|$ porque inicialmente estão todos descobertos. Temos $u_{i-1} \geq u_i$ e $u_{i-1} - u_i$ elementos de $S$ cobertos pela primeira vez através de $S_i$, para $i = 1, 2, \dots, k$, onde $k$ é o menor índice tal que $u_k = 0$.

$$\sum_{x \in S} c_x = \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})|}.$$

Note que $|S_i - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| \geq |S - (S_1 \cup S_2 \cup \dots \cup S_{i-1})| = u_{i-1}$
pela escolha gulosa de $S_i$, sabemos que $S$ não cobre mais que $S_i$.

Portanto, $\sum_{x \in S} c_x \leq \sum_{i=1}^{k} (u_{i-1} - u_i) \cdot \frac{1}{u_{i-1}} = \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{u_{i-1}}$

$$\leq \sum_{i=1}^{k} \sum_{j=u_i+1}^{u_{i-1}} \frac{1}{j} \quad (\text{porque } j \leq u_{i-1})$$

$$= \sum_{i=1}^{k} \left( \sum_{j=1}^{u_{i-1}} \frac{1}{j} - \sum_{j=1}^{u_i} \frac{1}{j} \right) = \sum_{i=1}^{k} (H(u_{i-1}) - H(u_i)) =$$

$$= H(u_0) - H(u_k) = H(u_0) - H(0) = H(u_0) = H(|S|).$$