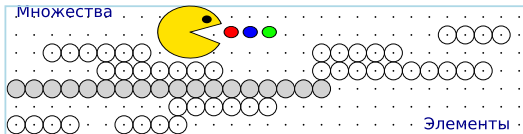


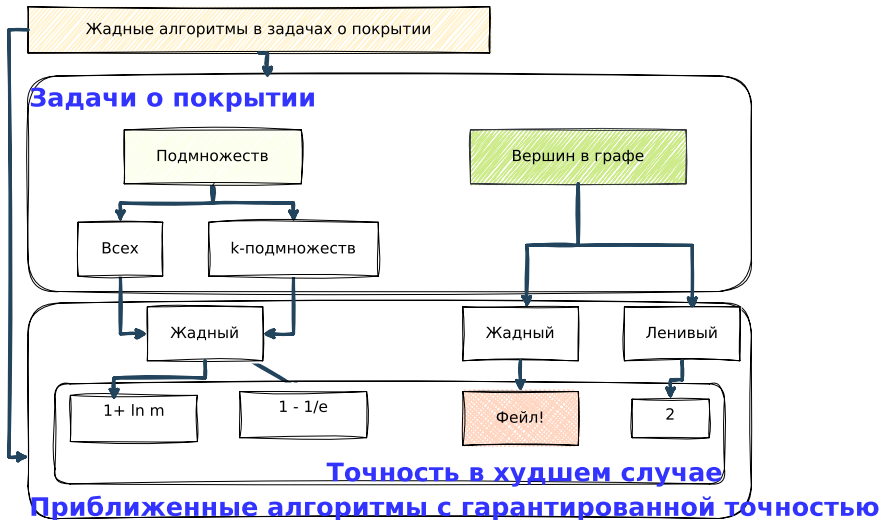
Жадные алгоритмы в задачах о покрытии

Н. Н. Кузюрин С. А. Фомин

20 января 2024 г.



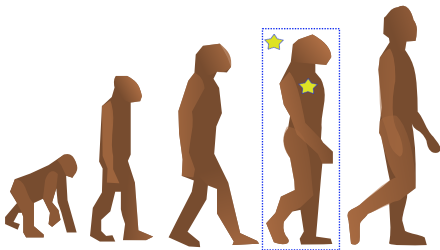
«Карта памяти» лекции



Приближенный алгоритм с гарантированной точностью

Определение

Алгоритм называется ***C-приближенным***, если при любых исходных данных он находит допустимое решение со значением целевой функции, отличающимся от оптимума не более чем в C раз.



Покрытие множества

Задача

«Покрытие множества»^a.

- Множество X , $|X| = m$.
- Семейство подмножеств $\{S_1, \dots, S_n\}$, $S_j \subseteq X$.
- $X = \cup_{j=1}^n S_j$ (покрытие гарантировано).

Найти минимальное по мощности множество подмножеств, покрывающее X :

$$|J| \rightarrow \min,$$

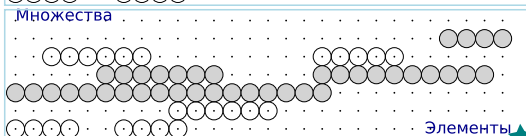
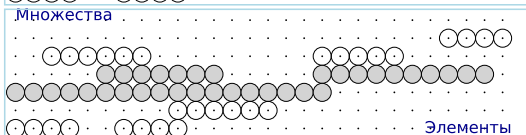
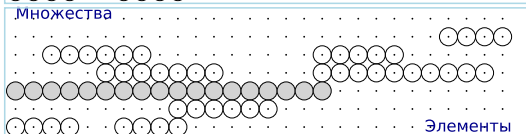
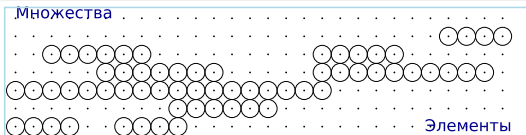
$$J \subseteq \{1, 2, \dots, n\},$$

$$X = \cup_{j \in J} S_j.$$

^aВ англоязычной литературе — *Set Cover*.

Жадный алгоритм в задаче о покрытии

Выбор на каждом шаге подмножества, покрывающего **максимальное число еще непокрытых элементов**:



Покрытие на каждом шаге

Лемма

X_k — число непокрытых элементов после k -го шага.

M — размер минимального покрытия.

$$X_{k+1} \leq X_k - \frac{X_k}{M} = X_k(1 - 1/M). \quad (1)$$

Доказательство.

- 1 Пусть X непокрытых покрываются минимум M' подмножествами.
- 2 Алгоритм за шаг покрывает не меньше $\frac{X}{M'}$.
- 3 Иначе, оптимальное покрытие должно покрывать меньше $M' \cdot \frac{X}{M'}$ элементов.
- 4 Т.к $M' \leq M$, получаем доказательство (1).



Точность жадного алгоритма: верхняя оценка

Теорема

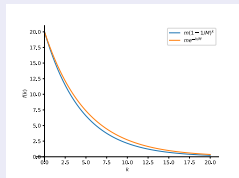
Размер покрытия, построенного жадным алгоритмом, превосходит минимальное не более чем в $1 + \ln m$ раз.

Доказательство.

X_i — непокрытых элементов после i -го шага.

M — размер минимального покрытия.

- 1 $X_0 = X, |X_0| = |X| = m$
- 2 $X_k \leq m(1 - 1/M)^k \leq m \exp(-\frac{k}{M})$
- 3 Пусть k_0 - последний шаг, когда $m \exp(-\frac{k_0}{M}) \geq 1$.
- 4 Тогда при $k_1 = k_0 + 1$ будет $X_{k_1} < 1$.
- 5 $k_1/M \leq 1 + \ln m$



```
import sympy as sm
```

```
x, M, k, m = sm.symbols('x, M, k, m')
```

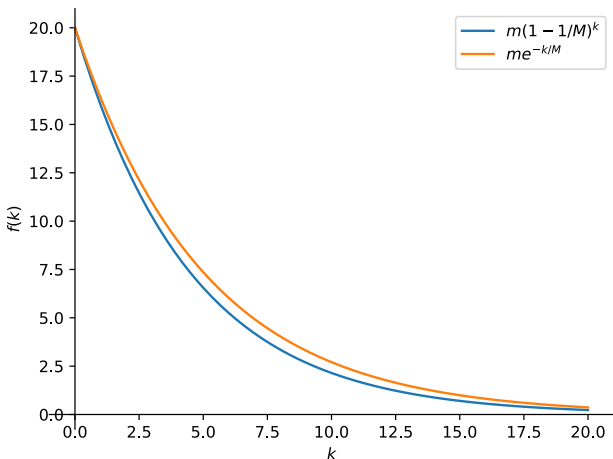
```
exprs = [m * (1 - 1/M)**k, m * sm.exp(-k/M)]
```

```
sample = [(m, 20), (M, 5)]
```

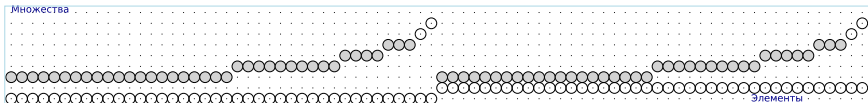
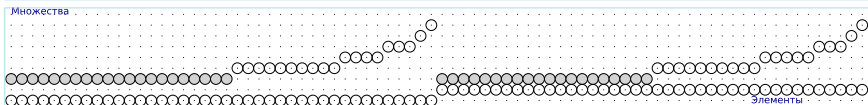
```
arange = (k, 0, 20)
```

```
plot = sm.plot(exprs[0].subs(sample), arange, legend=True, label=exprs[0], show=False)
```

```
plot.append( sm.plot(exprs[1].subs(sample), arange, label=exprs[1], show=False)[0] )
```



Как обмануть жадный алгоритм? Нижняя оценка.



k -покрытие множества

Задача

« K -покрытие»

Задано:

- Множество X , $|X| = m$.
- Семейство подмножеств $\{S_1, \dots, S_n\}$, $S_j \subseteq X$.

Выбрать такие k подмножеств, чтобы мощность их объединения была максимальна:

$$\begin{aligned} &|\cup_{j \in J} S_j| \rightarrow \max, \\ &J \subseteq \{1, 2, \dots, n\}, \\ &|J| = k. \end{aligned}$$

$(1 - e^{-1})$ -приближенный жадный для « k -покрытия»

Доказательство.

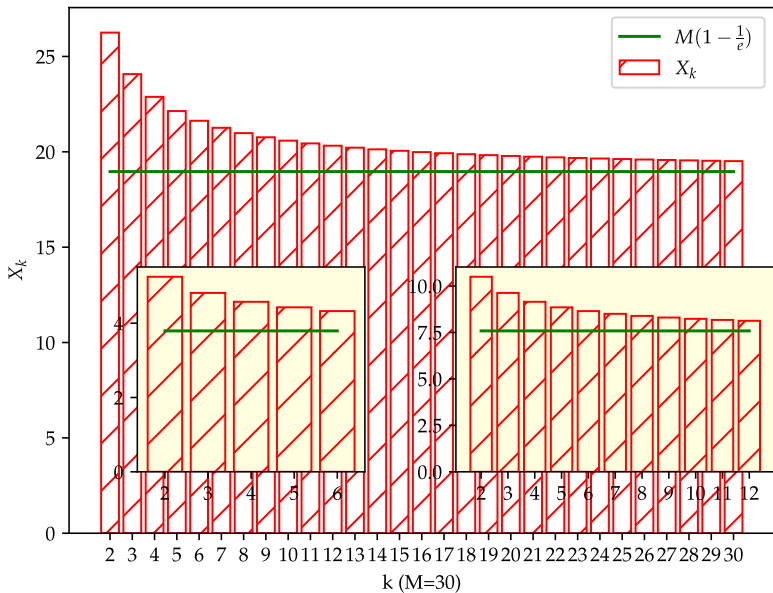
X_i — число покрытых элементов после i -го шага.

M — оптимум.

$$\begin{aligned} X_{i+1} &\geq X_i + \frac{M - X_i}{k} = \frac{M}{k} + X_i \left(1 - \frac{1}{k}\right) \\ &\geq \frac{M}{k} + \left(\frac{M}{k} + X_{i-1} \left(1 - \frac{1}{k}\right)\right) \left(1 - \frac{1}{k}\right). \end{aligned}$$

$$\begin{aligned} X_k &\geq \frac{M}{k} \left[1 + \left(1 - \frac{1}{k}\right) + \left(1 - \frac{1}{k}\right)^2 + \dots + \left(1 - \frac{1}{k}\right)^{k-1} \right] \\ &= \frac{M}{k} \left[\frac{1 - \left(1 - \frac{1}{k}\right)^k}{\frac{1}{k}} \right] = M \left[1 - \left(1 - \frac{1}{k}\right)^k \right] \geq M(1 - e^{-1}). \end{aligned}$$





Вершинное покрытие

Определение

«Вершинное покрытие»^a.

Для неориентированного графа $G = (V, E)$ подмножество вершин $V' \subseteq V$ называется **вершинным покрытием**, если каждое ребро из E содержит хотя бы одну вершину из V' .

^aВ англоязычной литературе — *Vertex Cover*.

Задача

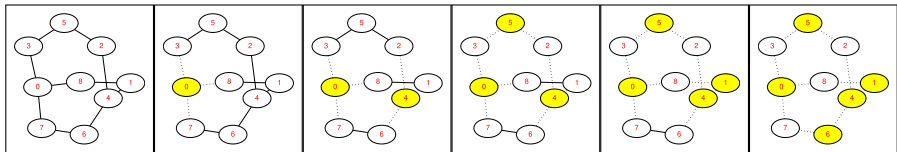
«Минимальное вершинное покрытие»^a.

Дан неориентированный граф $G = (V, E)$ (множество вершин V , множество ребер E). Найти вершинное покрытие минимальной мощности.

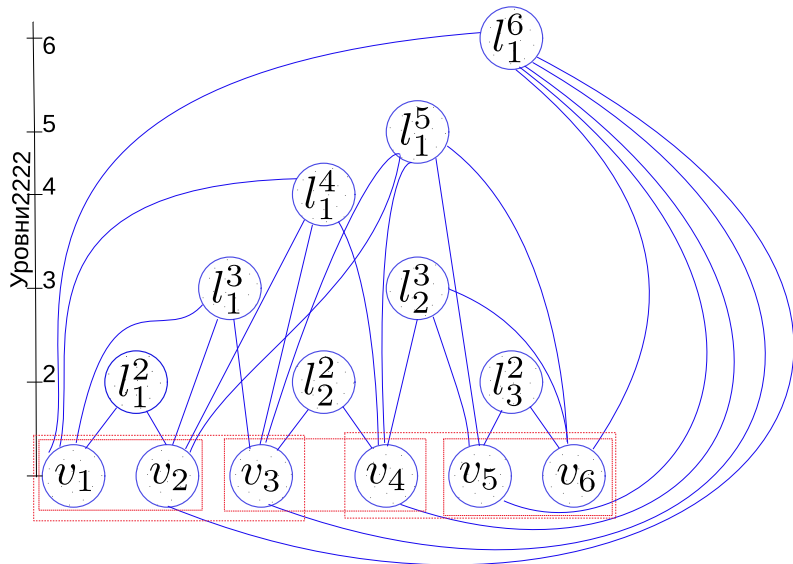
^aВ англоязычной литературе — *Min Vertex Covering*.

Жадный алгоритм для вершинного покрытия

```
def greedyVertexCovering(G):  
    C = set()  
    while G.number_of_edges() > 0: # цикл сложности  $O(|E|)$   
        nodeMaxDegree, maxDegree = None, None  
        for n in G.nodes(): # цикл сложности  $O(|V|)$   
            if not maxDegree or G.degree(n) > maxDegree:  
                nodeMaxDegree = n  
                maxDegree = G.degree(n)  
        C.add(nodeMaxDegree)  
        for e in list(G.edges(nodeMaxDegree)):  
            G.remove_edge(*e) #удаляем покрытые ребра  
    return C
```



«Жадный» не обеспечивает даже константной точности



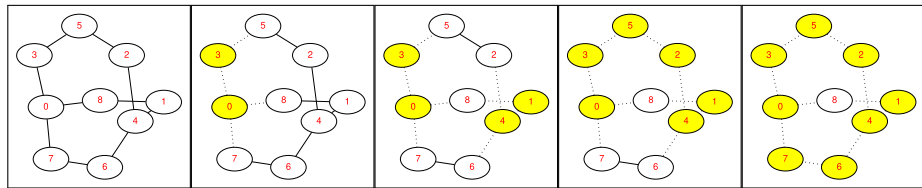
Определение

Паросочетание^a — подмножество ребер графа, такое, что никакие два ребра из этого подмножества не инцидентны какой-либо одной вершине.

^a*Matching* в англоязычной литературе.

Ленивый алгоритм для вершинного покрытия

```
def lazyVertexCovering(G):  
    C = set()  
    while G.number_of_edges() > 0:  
        #первое появившееся ребро  
        (u, v) = list(G.edges())[0]  
        C.add(u)  
        C.add(v)  
        for e in list(G.edges(u)):  
            G.remove_edge(*e)  
        for e in list(G.edges(v)):  
            G.remove_edge(*e)  
    return C
```



Теорема

«Ленивый» алгоритм для задачи 3 «Min Vertex Covering» гарантирует точность 2.

Доказательство.

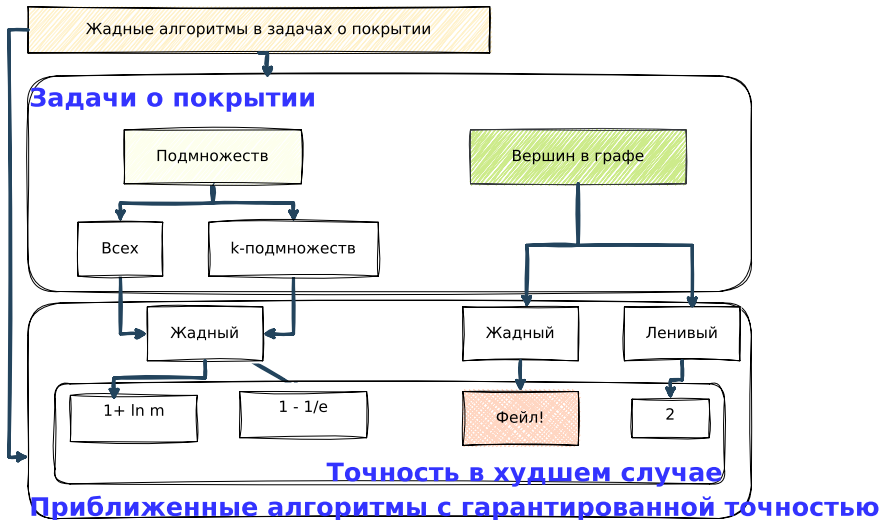
- 1 E^* — ребра, «попавшиеся алгоритму».
- 2 $|E^*| = |C|/2$
- 3 $OPT \geq |E^*|$: \forall вершинное покрытие должно содержать по крайней мере одну вершину для каждого ребра из E^* .

Получаем:

$$|C| \leq 2 \cdot OPT.$$



«Карта памяти» лекции



<http://discopal.ispras.ru/>

Вопросы?