

- 教材讨论
 - JH第4章第3节第1、2、3小节

问题1: greedy vs. local search

- greedy和local search有什么异同?

问题2: MIN-VCP

- 这个算法的基本思路是什么？
- 近似比是多少，为什么？
- 时间复杂度是多少？为什么？用什么数据结构来实现？

Algorithm 4.3.2.1. Input: A graph $G = (V, E)$.

Step 1: $C := \emptyset$ {during the computation $C \subseteq V$, and at the end C should contain a vertex cover};
 $A := \emptyset$ {during the computation $A \subseteq E$ is a matching, and at the end A is a maximal matching};
 $E' := E$ {during the computation $E' \subseteq E$, E' contains exactly the edges that are not covered by the actual C , and at the end $E' = \emptyset$ }.

Step 2: **while** $E' \neq \emptyset$
 do begin choose an arbitrary edge $\{u, v\}$ from E' ;
 $C := C \cup \{u, v\}$;
 $A := A \cup \{\{u, v\}\}$;
 $E' := E' - \{\text{all edges incident to } u \text{ or } v\}$
 end

Output: C .

问题2: MIN-VCP (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为2的例子吗?

Algorithm 4.3.2.1. Input: A graph $G = (V, E)$.

Step 1: $C := \emptyset$ {during the computation $C \subseteq V$, and at the end C should contain a vertex cover};

$A := \emptyset$ {during the computation $A \subseteq E$ is a matching, and at the end A is a maximal matching};

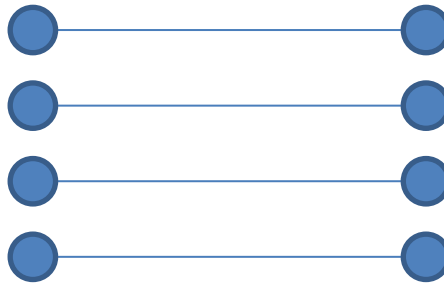
$E' := E$ {during the computation $E' \subseteq E$, E' contains exactly the edges that are not covered by the actual C , and at the end $E' = \emptyset$ }.

Step 2: **while** $E' \neq \emptyset$
 do begin choose an arbitrary edge $\{u, v\}$ from E' ;
 $C := C \cup \{u, v\}$;
 $A := A \cup \{\{u, v\}\}$;
 $E' := E' - \{\text{all edges incident to } u \text{ or } v\}$
 end

Output: C .

问题2: MIN-VCP (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为2的例子吗?



问题3: SCP

- 这个算法的基本思路是什么？
- 近似比是多少？
- 时间复杂度是多少？为什么？用什么数据结构来实现？

Algorithm 4.3.2.11.

Input: (X, \mathcal{F}) , where X is a finite set, $\mathcal{F} \subseteq \mathcal{Pot}(X)$ such that $X = \bigcup_{Q \in \mathcal{F}} Q$.

Step 1: $C := \emptyset$ {during the computation $C \subseteq \mathcal{F}$ and at the end C is a set cover of (X, \mathcal{F}) };

$U := X$ {during the computation $U \subseteq X$, $U = X - \bigcup_{Q \in C} Q$ for the actual C , and at the end $U = \emptyset$ }.

Step 2: **while** $U \neq \emptyset$
 do begin choose an $S \in \mathcal{F}$ such that $|S \cap U|$ is maximal;
 $U := U - S$;
 $C := C \cup \{S\}$

end

Output: C .

问题3: SCP (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为 $\Omega(\ln n)$ 的例子吗?

Algorithm 4.3.2.11.

Input: (X, \mathcal{F}) , where X is a finite set, $\mathcal{F} \subseteq \mathcal{Pot}(X)$ such that $X = \bigcup_{Q \in \mathcal{F}} Q$.

Step 1: $C := \emptyset$ {during the computation $C \subseteq \mathcal{F}$ and at the end C is a set cover of (X, \mathcal{F}) };

$U := X$ {during the computation $U \subseteq X$, $U = X - \bigcup_{Q \in C} Q$ for the actual C , and at the end $U = \emptyset$ }.

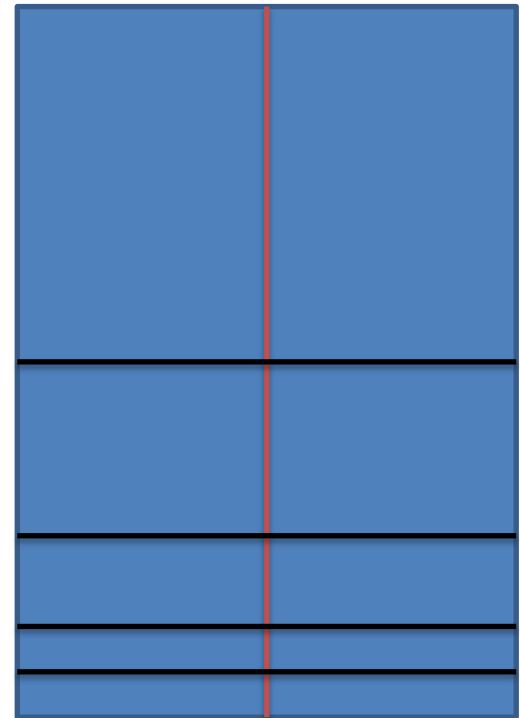
Step 2: **while** $U \neq \emptyset$
 do begin choose an $S \in \mathcal{F}$ such that $|S \cap U|$ is maximal;
 $U := U - S$;
 $C := C \cup \{S\}$

end

Output: C .

问题3: SCP (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为 $\Omega(\ln n)$ 的例子吗?
 - $U = \{ (x,y) \mid 1 \leq x,y \leq n \}$
 - $S_1 = \{ (x,y) \mid x \leq n/2 \}$
 - $S_2 = \{ (x,y) \mid x > n/2 \}$
 - $T_1 = \{ (x,y) \mid n/2 < y \leq n \}$
 - $T_2 = \{ (x,y) \mid n/4 < y \leq n/2 \}$
 - $T_3 = \{ (x,y) \mid n/8 < y \leq n/4 \}$
 -



问题3： SCP (续)

- 和MIN-VCP相比， SCP到底难在什么地方？

问题4: MAX-CUT

- 这个算法的基本思路是什么？
- 近似比是多少，为什么？
- 时间复杂度是多少？为什么？用什么数据结构来实现？

Algorithm 4.3.3.1.

Input: A graph $G = (V, E)$.

Step 1: $S = \emptyset$

{the cut is considered to be $(S, V - S)$; in fact S can be chosen arbitrarily in this step};

Step 2: **while** there exists such a vertex $v \in V$ that the movement of v from one side of the cut $(S, V - S)$ to the other side of $(S, V - S)$ increases the cost of the cut.

do begin take a $u \in V$ whose movement from one side of $(S, V - S)$ to the other side of $(S, V - S)$ increases the cost of the cut, and move this u to the other side.

end

Output: $(S, V - S)$.

问题4: MAX-CUT (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为2的例子吗?

Algorithm 4.3.3.1.

Input: A graph $G = (V, E)$.

Step 1: $S = \emptyset$

{the cut is considered to be $(S, V - S)$; in fact S can be chosen arbitrarily in this step};

Step 2: **while** there exists such a vertex $v \in V$ that the movement of v from one side of the cut $(S, V - S)$ to the other side of $(S, V - S)$ increases the cost of the cut.

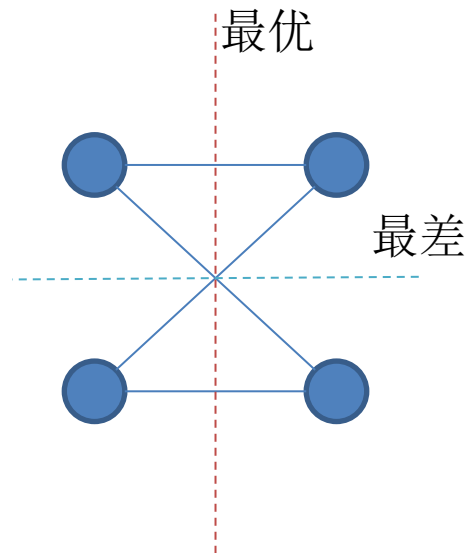
do begin take a $u \in V$ whose movement from one side of $(S, V - S)$ to the other side of $(S, V - S)$ increases the cost of the cut, and move this u to the other side.

end

Output: $(S, V - S)$.

问题4: MAX-CUT (续)

- 你能构造出一些近似比为1的例子吗?
- 你能构造出一些近似比为2的例子吗?



问题5: greedy和local search

- 任选1个问题，分别给出一种greedy算法和一种local search算法，并尽力给出近似比（或者给出一些坏例子）
 - longest simple path
 - MAX-SAT
 - MAX-CL