Остовные деревья

Лекция 4

Задача «Минимальное остовное дерево»

- *Дано*: Граф G, веса c: $E(G) \to \mathbf{R}$.
- *Найти* остовное дерево в G наименьшего веса или определить, что G — несвязный.

Задача «Максимальный взвешенный лес»

Дано: Граф G, веса c: $E(G) \to \mathbf{R}$.
Найти лес в G наибольшего веса.

Эквивалентные задачи

- Будем говорить, что задача P **линейно сводится** к задаче Q, если существуют функции f и g, вычислимые за линейное время, такие что f преобразует частную задачу x из P в частную задачу y из Q, и g преобразует решение $f(x)$ в решение x.
- Если P линейно сводится к Q и Q линейно сводится к P, то обе задачи называются **эквивалентными**.

Эквивалентность

Предложение 4.1

Задача «Минимальное остовное дерево» и задача «Максимальный взвешенный лес» эквивалентны.

Доказательство

- (G,c) — исходный пример задачи «Максимальный взвешенный лес».
- Удалим все ребра отрицательного веса.
- Положим $c'(e) = -\, c(e)$.
- Добавим минимальное множество ребер F, так чтобы полученный граф G' стал связным. (Веса можно взять любые.)
- Решим задачу «Минимальное остовное дерево» на примере (G',c').
- Удалив из решения множество ребер F, получим решение исходной задачи.
- (G,c) — исходный пример задачи «Минимальное остовное дерево».
- Положим $c'(e) = K - c(e)$, где $K = 1 + \max_{e \in E(G)} c(e)$.
- Решение задачи «Максимальный взвешенный лес» на примере (G',c') дает решение задачи «Минимальное остовное дерево» на исходном примере.

Условия оптимальности

Теорема 4.2

Пусть (G ,c) — пример задачи «Минимальное остовное дерево», и пусть T — остовное дерево в G.
Тогда следующие условия эквивалентны:

a) T — оптимум.

b) Для любого $e = \{x, y\} \in E(G) \setminus E(T)$, все ребра из x-y-пути в T не дороже чем e.

- Для любого $e \in E(T)$, e — ребро наименьшей стоимости из $\delta(V(C))$, где C — связная компонента на $T - e$.

(c)$\Rightarrow$(a)

- (c) Пусть T такое, что для любого $e \in E(T)$, e — ребро наименьшей стоимости из $\delta(V(C))$, где C — связная компонента на $T - e$.
- Пусть T^* оптимальное решение, такое что $E(T) \cap E(T^*)$ максимально возможное. Покажем, что $T = T^*$.
- Пусть $e = \{x,y\} \in E(T) \setminus E(T^*)$.
- Пусть C — связная компонента на $T - e$.
- $T^* + e$ содержит цикл D. Так как $e \in E(D) \cap \delta(C)$, то существует $f \neq e, f \in E(D) \cap \delta(C)$.
- Тогда $(T^* + e) - f$ является остовным деревом.
- T^* оптимум $\Rightarrow c(e) \geq c(f)$ и (c) $\Rightarrow c(f) \geq c(e)$.
- $c(f) = c(e)$ и $(T^* + e) - f$ является оптимальным остовным деревом.
- Противоречие, так как в $(T^* + e) - f$ больше на одно общее ребро с T, чем в T^* .

Алгоритм Краскала (1956)

Input: Связный граф G, веса c: $E(G) \to \mathbf{R}$.
Output: Остовное дерево T наименьшего веса.

1) Сортируем ребра так, что $c(e_1) \le c(e_2) \le \ldots \le c(e_m)$.
2) **Set** $T := (V(G), \varnothing)$.
3) **For** $i := 1$ **to** m **do**:
   If $T+e_i$ не содержит цикла **then** $T := T+e_i$.

Алгоритм Краскала (2)

Теорема 4.3

Алгоритм Краскала находит оптимальное решение за $O(mn)$.

Алгоритм Краскала (3)

Теорема 4.4

Алгоритм Краскала можно реализовать за $O(m \log n)$.

Алгоритм Краскала (1956)

Input: Связный граф G, веса c: $E(G) \to \mathbf{R}$.
Output: Остовное дерево T наименьшего веса.

1) Сортируем ребра так, что $c(e_1) \le c(e_2) \le \ldots \le c(e_\mathrm{m})$.
2) **Set** $T := (V(G), \varnothing)$.
3) **For** $i := 1$ **to** m **do**:
   If $T+e_i$ не содержит цикла **then** $T := T+e_i$.

Как улучшить шаг 3

- Основная цель шага 3 проверить не приведет ли добавление ребра $e_i = \{v,w\}$ к образованию цикла.
- Это эквивалентно проверки лежат ли v и w в одной связной компоненте.
- По ходу алгоритма будем строить дополнительный орлес B с $V(B) = V(G)$, такой что связные компоненты B индуцированы на тех же вершинах, что и связные компоненты T.

Проверка

- Первоначально, $B = (V(G), \varnothing)$ и $h(v) = 0$, для всех $v \in V(G)$, где $h(v)$ длина максимального пути из v в B.

3.1 Для ребра $e_i = \{v,w\}$ находим корни r_v и r_w ордеревьев в B, содержащих v и w.

3.2 Если $r_v = r_w$, то переходим к следующему ребру и идем на 3.1.

3.3 Если $r_v \neq r_w$, то добавляем e_i к T.

3.4. Если $h(r_v) \geq h(r_w)$, то добавляем дугу (r_v, r_w) к B, иначе добавляем дугу (r_w, r_v) к B.

Время работы шага 3

- Время работы пропорционально длине r_v-v-пути в B.
- Покажем, что любое ордерево в B с корнем r имеет по крайней мере $2^{h(r)}$ вершин.
- Когда $B = (V(G), \varnothing)$ утверждение очевидно.
- Пусть алгоритм добавляет дугу (x,y) в B. Получаем новое ордерево с корнем в x и числом вершин $\geq 2^{h(x)} + 2^{h(y)}$.
- Если $h(x) > h(y)$, то значение $h(x)$ не меняется и утверждение справедливо.
- Если $h(x) = h(y)$, то значение $h(x)$ увеличивается на 1. Число вершин в новом ордереве $\geq 2^{h(x)} + 2^{h(y)} = 2^{h(x)+1}$.
- $h(r) \leq \log n$, и трудоемкость шага 3 $\leq m\log n$.

Алгоритм Прима (1957)

Input: Связный граф G, веса c: $E(G) \to \mathbf{R}$.
Output: Остовное дерево T минимального веса.

1) Выбрать $v \in V(G)$. $T := (\{v\}, \varnothing)$.
2) **While** $V(T) \neq V(G)$ **do**:
   Выбрать ребро $e \in \delta_G(V(T))$ минимальной стоимости. $T := T + e$.

Алгоритм Прима (2)

Теорема 4.5

Алгоритм Прима находит решение за $O(n^2)$ элементарных операций.

Как реализовать шаг 2

While $V(T) \neq V(G)$ **do**:
Выбрать ребро $e \in \delta_G(V(T))$ минимальной стоимости. $T := T + e$.

- Для каждой вершины $v \notin V(T)$ будем хранить самое дешевое ребро (кандидата) из v в $V(T)$.
- Тогда выбор ребра минимальной стоимости и замена кандидатов может быть выполнена за $O(n)$ элементарных операций.

Задача «Максимальный взвешенный ориентированный лес»

- *Дано*: Орграф G,

  веса $c: E(G) \to \mathbf{R}$.

- *Найти* ориентированный лес в G наибольшего веса.

Задача «Минимальное остовное ориентированное дерево»

- *Дано*: Орграф G,

  веса c: $E(G) \to \mathbf{R}$.

- *Найти* остовное ориентированное дерево в G наименьшего веса или определить, что оно не существует.

Задача «Минимальное остовное корневое ориентированное дерево»

- *Дано*: Орграф G, вершина $r \in V(G)$,

  веса c: $E(G) \to \mathbf{R}$.
- *Найти* остовное ориентированное дерево с корнем r в G наименьшего веса или определить, что оно не существует.

Эквивалентность трех задач

Предложение 4.6.

Задачи «Максимальный взвешенный ориентированный лес», «Минимальное остовное ориентированное дерево» и «Минимальное остовное корневое ориентированное дерево» эквивалентны.

Упражнение 4.1

Доказать предложение 4.6 .

Ориентированный лес

Орграф называется **ориентированным лесом**, если соответствующий ему граф является лесом и каждая вершина v имеет не более одного входящего ребра.

Ориентированный лес и циклы

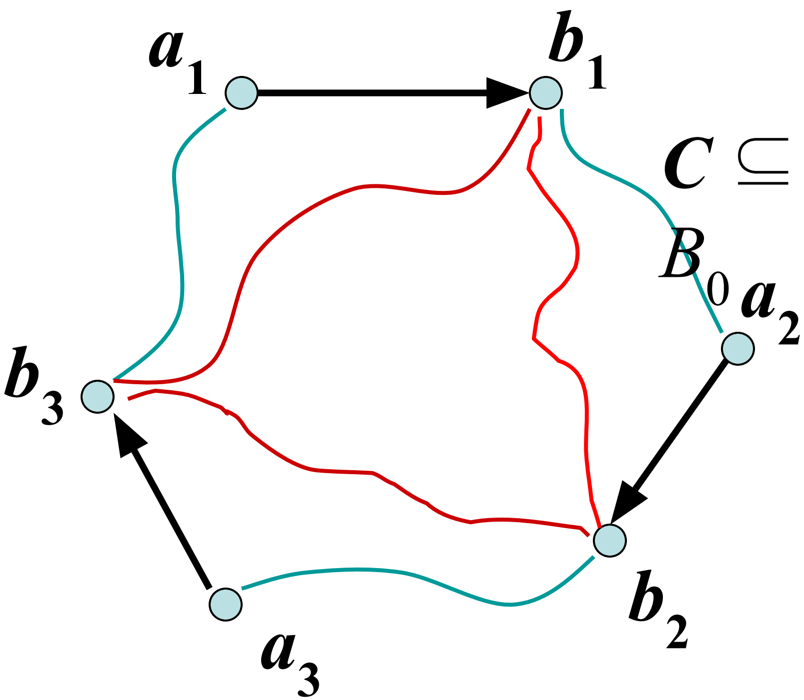
Предложение 4.7.

Пусть B — орграф с $\left|\delta_B^-(x)\right| \le 1$ для всех $x \in V(B)$. Тогда B имеет цикл тогда и только тогда, когда соответствующий ему граф имеет цикл.

Лемма 4.8. (Karp [1972])

Пусть B_0 — подграф G максимального веса с $\left|\delta_{B_0}^-(v)\right| \le 1$ для всех $v \in V(B_0)$. Тогда существует оптимальный ориентированный лес B в G такой, что для каждого цикла C в B_0, $|E(C) \setminus E(B)| = 1$.

Доказательство леммы

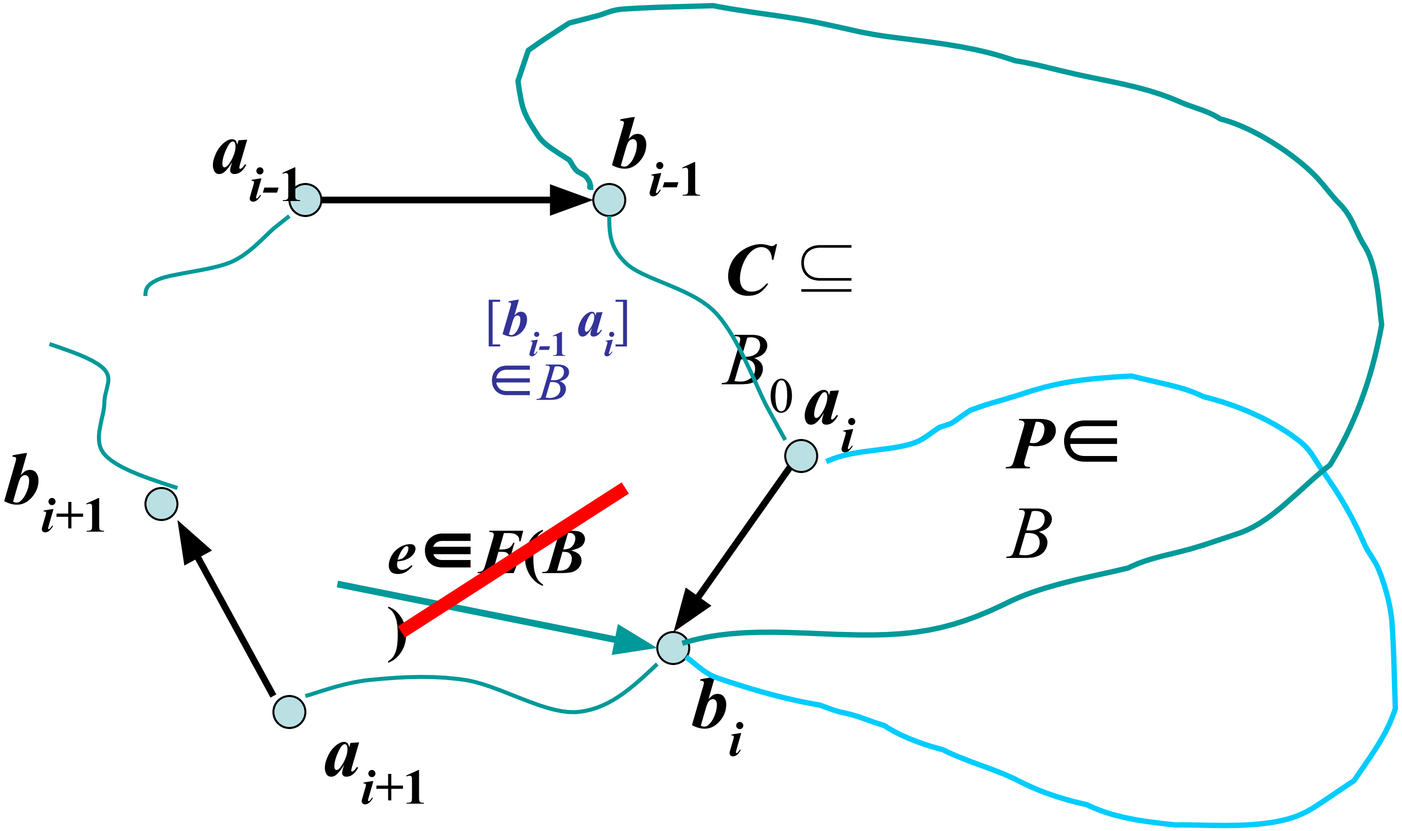


Пусть B – оптимальный орлес в G имеющий макси -мально много ребер из B_0.

$$E(C)\setminus E(B)=\{(a_1, b_1),\ldots, (a_k, b_k)\}$$

Покажем, что в B есть b_i-b_{i-1}-путь для всех i.

Покажем, что в B есть b_i-b_{i-1}-путь для всех i.



$E(B') := \{(x,y) \in E(B)\} \setminus \{e\} \cup \{(a_i, b_i)\}$

B' — не орлес.

Основная идея

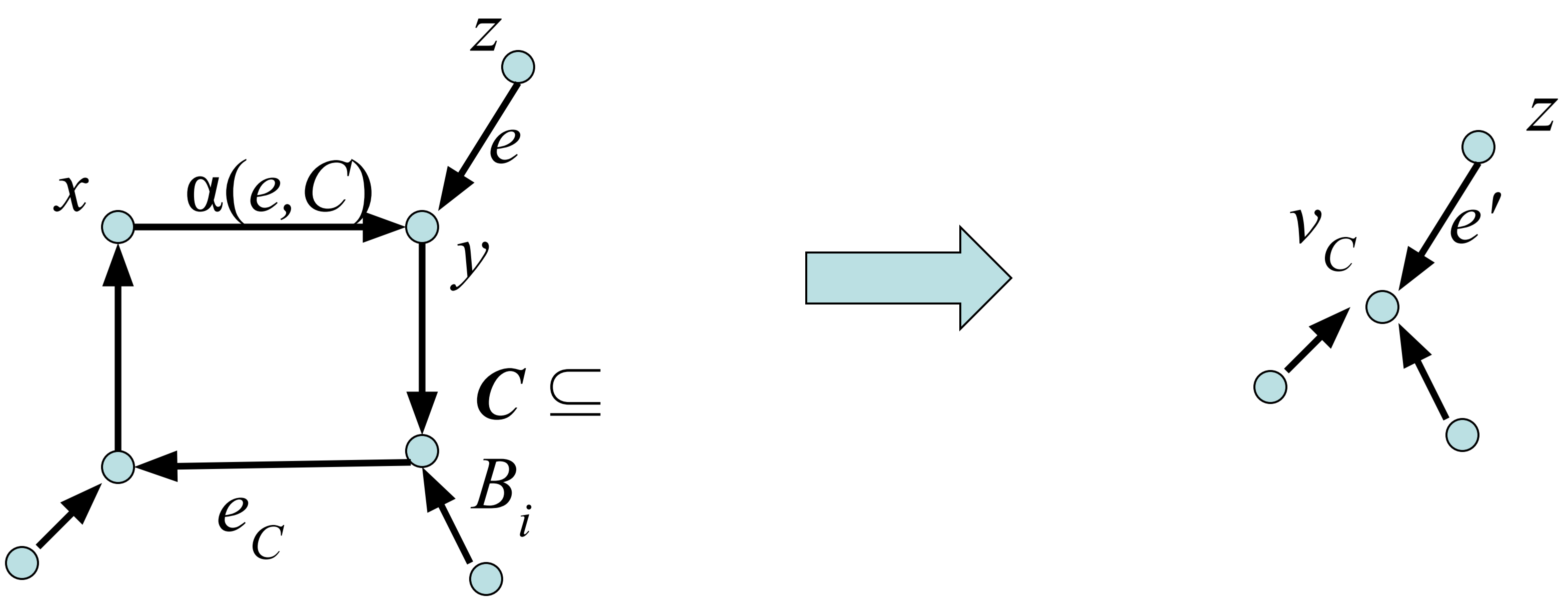
- Найти B_0 ориентированный подграф G максимального веса, в котором в каждую вершину входит не больше одной дуги.
- Стянуть каждый цикл в B_0 в вершину.
- Перераспределить веса в новом графе G_1, так чтобы любой оптимальный орлес в G_1 соответствовал оптимальному орлесу в G.

Алгоритм Эдмондса *построения ориентированного леса максимального веса* (1967)

Input: орграф G, веса c: $E(G) \to \mathbf{R}_+$.
Output: орлес максимального веса B of G.

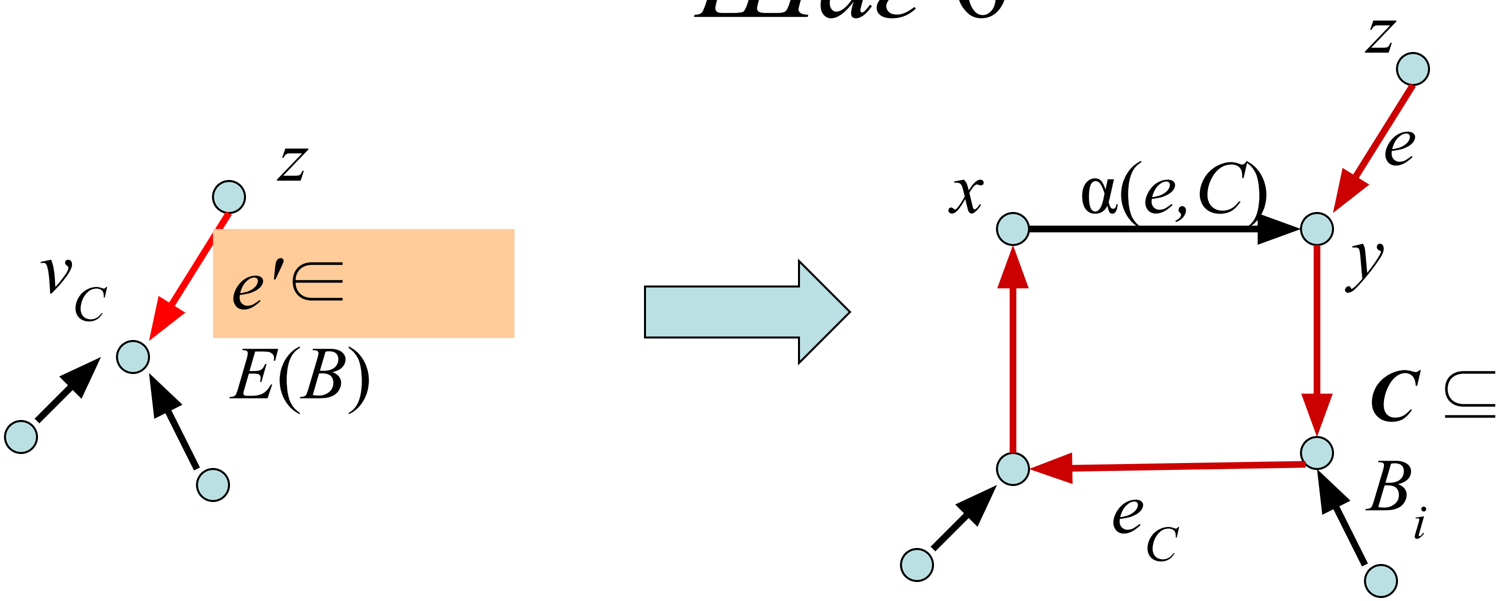
1) **Set** $i := 0$, $G_0 := G$, и $c_0 := c$.
2) Пусть B_i подграф G максимального веса с $\left|\delta^-_{B_i}(v)\right| \le 1$ для всех $v \in B_i$.
3) **If** B_i не содержит циклов **then** $B := B_i$ и **go to (**5).
4) Построим (G_{i+1}, c_{i+1}) из (G_i, c_i): **do** для каждого цикла C из B_i.
Стянем C к одной вершине v_C в G_{i+1}.
For каждого ребра $e = (z, y) \in E(G_i)$ с $z \notin V(C)$, $y \in V(C)$ **do**:
Set $c_{i+1}(e') := c_i(e) - c_i(\alpha(e,C)) + c_i(e_C)$ и $\Phi(e') := e$,
где $e' := (z, v_C)$, $\alpha(e,C) = (x,y) \in E(C)$, и e_C самое дешевое ребро C.
1) **If** $i = 0$ **then stop.**
2) **For** каждого цикла C из B_{i-1} **do:**
If есть ребро $e' = (z, v_C) \in E(B)$
then $E(B) := (E(B)\setminus\{e'\}) \cup \Phi(e') \cup (E(C)\setminus\{\alpha(\Phi(e'),C)\})$
else $E(B) := E(B) \cup (E(C)\setminus\{e_C\})$.
Set $V(B) := V(G_{i-1})$, $i := i-1$ и **go to** (5).

Шаг 4

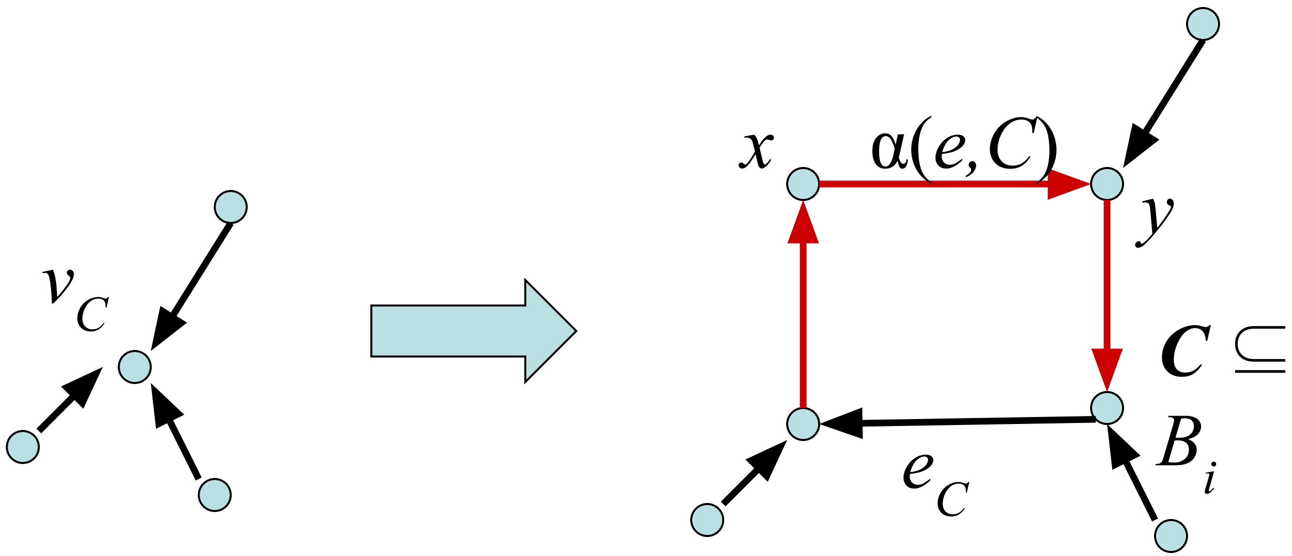


- **For** каждого ребра $e = (z, y) \in E(G_i)$ с $z \notin V(C)$, $y \in V(C)$ **do**:
Set $c_{i+1}(e') := c_i(e) - c_i(\alpha(e,C)) + c_i(e_C)$ и $\Phi(e') := e$,
где $e' := (z, v_C)$, $\alpha(e,C) = (x,y) \in E(C)$, и e_C самое дешевое ребро C.

Шаг 6



z
v_C
$e' \in$
$E(B)$
x
$\alpha(e,C)$
z
e
y
$C \subseteq$
B_i
e_C



v_C
x
$\alpha(e,C)$
y
$C \subseteq$
B_i
e_C

Алгоритм Эдмондса

Теорема 4.9

Алгоритм Эдмондса находит оптимальное решение.

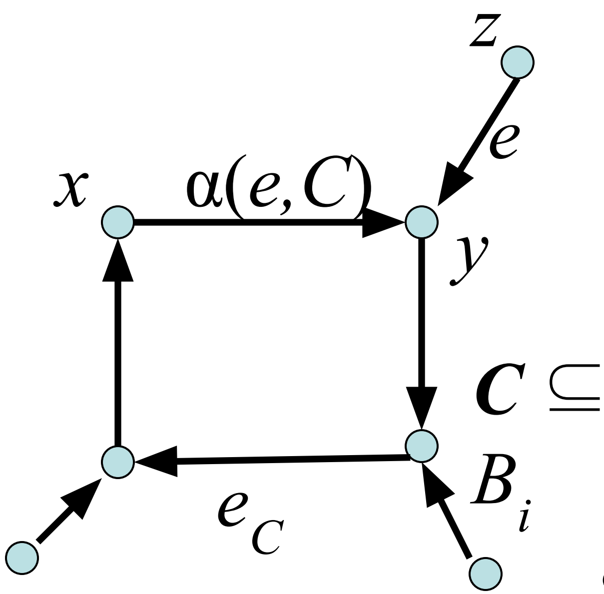
Доказательство

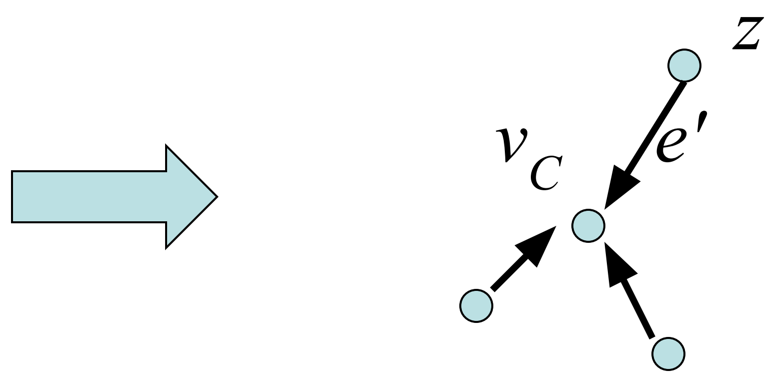
- Последовательно применяя шаг 4 алгоритма, мы получили последовательность (G_i, c_i), $i = 0,\ldots, k$.
- Ясно, что полученный на последнем применении шага 4 орлес B является оптимальным для (G_k, c_k).
- Покажем, что на шаге 6 мы последовательно строим оптимальные решения для (G_i, c_i), $i = k-1,\ldots,0$.
- Мы хотим показать, что шаг 6 переводит ориентированный лес наибольшего веса B для G_i в ориентированный лес наибольшего веса B^* для G_{i-1}.

Доказательство (2)

- Пусть B'_{i-1} — произвольный орлес в G_{i-1}, такой что $|E(C) \setminus E(B'_{i-1})| = 1$ для любого C из B_{i-1}.
- Пусть B'_i получается из B'_{i-1} стягиванием циклов в B_{i-1}.
- Тогда B'_i орлес в G_i.

Шаг 4





Set $c_{i+1}(e') := c_i(e) - c_i(\alpha(e,C)) + c_i(e_C)$

$$c_{i-1}\left(B'_{i-1}\right) \le c_i\left(B'_i\right) + \sum_{C \in B_{i-1}} \left(c_{i-1}\left(C\right) - c_{i-1}\left(e_C\right)\right)$$

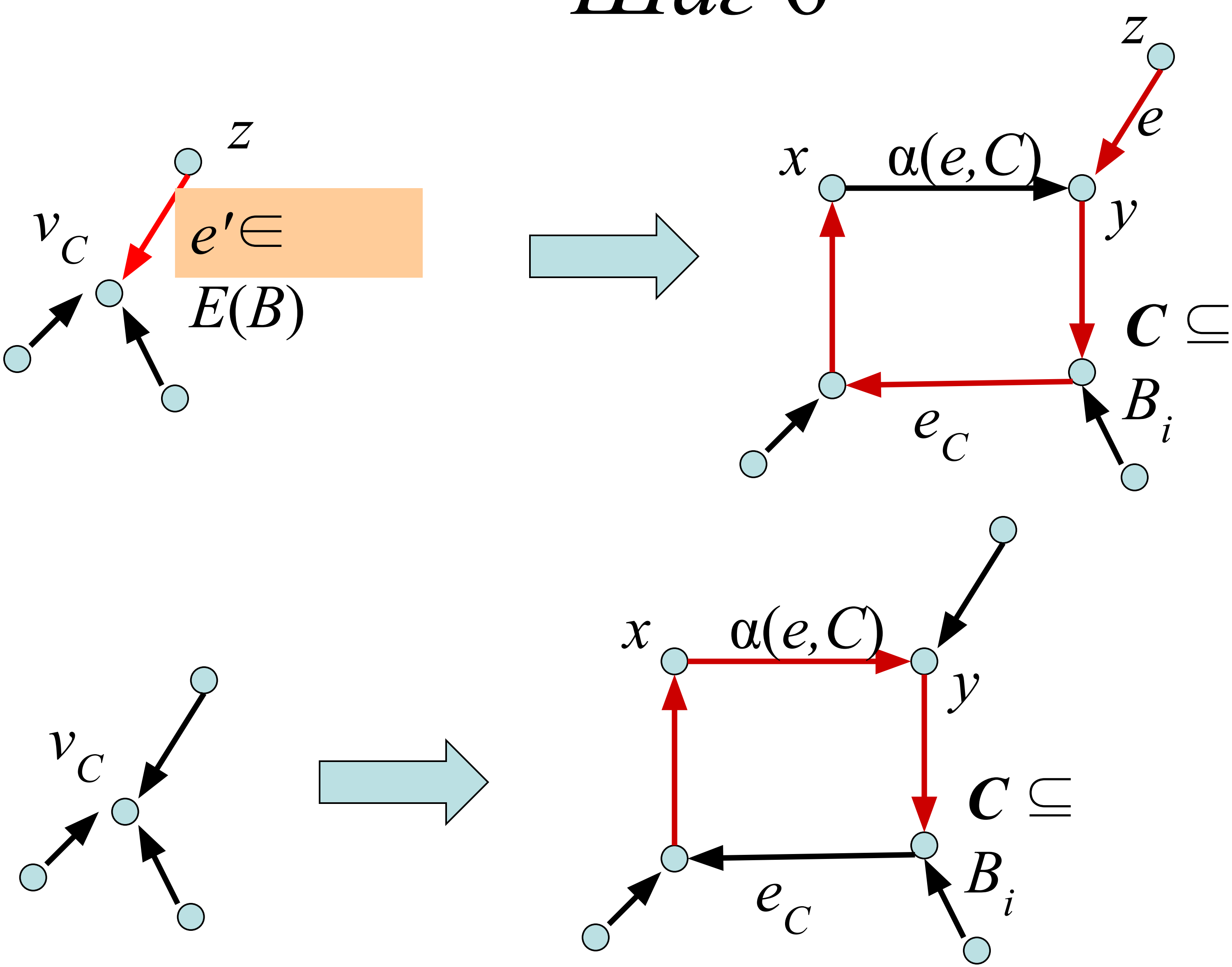
Индукция

- По индукции, B — ориентированный лес наибольшего веса в G_i.
- $c_i(B) \geq c_i(B'_i)$

$$c_{i-1}\left(B'_{i-1}\right) \leq c_i\left(B'_i\right) + \sum_{C \in B_{i-1}} \left(c_{i-1}\left(C\right) - c_{i-1}\left(e_C\right)\right)$$

$$\leq c_i\left(B\right) + \sum_{C \in B_{i-1}} \left(c_{i-1}\left(C\right) - c_{i-1}\left(e_C\right)\right) = c_{i-1}\left(B^*\right)$$

Шаг 6



z
v_C
$e' \in E(B)$
x
$\alpha(e,C)$
z
e
y
$\boldsymbol{C} \subseteq B_i$
e_C
v_C
x
$\alpha(e,C)$
y
$\boldsymbol{C} \subseteq B_i$
e_C

Упражнение 4.2

- Пусть (V,T_1) и (V,T_2) два дерева на одном множестве вершин V. Доказать, что для любого ребра $e \in T_1$ существует ребро $f \in T_2$ такое, что $(V,(T_1 \setminus \{e\}) \cup \{f\})$ и $(V,(T_2 \setminus \{f\}) \cup \{e\})$ — деревья.

Упражнение 4.3

- Дан граф G с произвольными весами $c\colon E(G) \to \mathbf{R}$.
- Найти остовный подграф в G наименьшего веса.

Упражнение 4.4

- Дан граф G с произвольными весами $c\colon E(G) \to \mathbf{R}$.
- Найти остовное дерево T в G, такое что вес максимального ребра в T наименьший $(\max\{c(e) | e \in T\} \to \min)$.