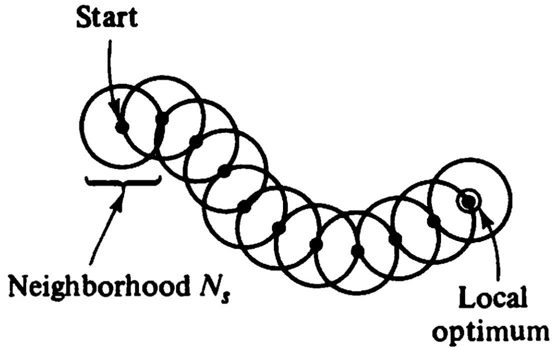
# Дискретная оптимизация

МФТИ, весна 2016

Александр  Дайняк

www.dainiak.com

# Локальный поиск




- Минимизируем целевую функцию $f$ на множестве $\mathcal{S}$.
- Система окрестностей
  - *сильно связная*, если
    $$\forall x', x'' \in \mathcal{S} \quad \exists x_1, x_2, \dots, x_k: \quad x_{i+1} \in \mathcal{N}(x_i), x_1 = x', x_k = x'',$$
    то есть из любой точки $\mathcal{S}$ можно попасть в любую другую, перемещаясь по окрестностям
  - *точная*, если, начав из любого начального приближения, алгоритм локального поиска находит глобальный оптимум
  - *полиномиально обозримая (polynomially searchable)*, если для любого $s \in \mathcal{S}$ существует полиномиальный алгоритм для выбора наилучшего элемента в $\mathcal{N}(s)$

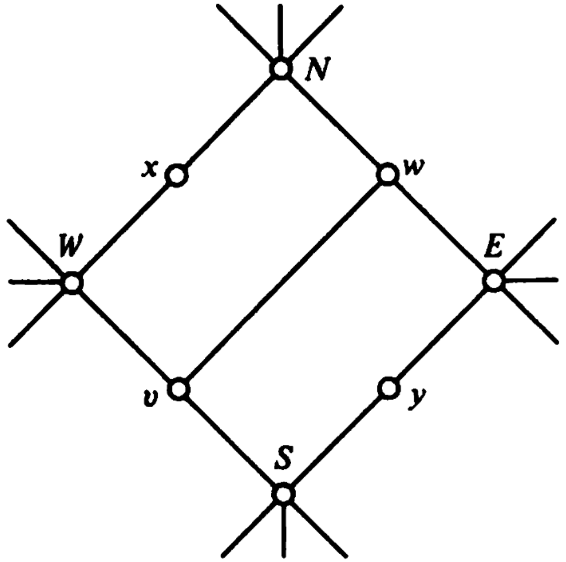
# Локальный поиск в задаче TSP

**Теорема. (Ch. Papadimitriou, K. Steiglitz ’1977)**

Если $\mathrm{P} \neq \mathrm{NP}$, то для задачи TSP не существует системы окрестностей, которая была бы одновременно точна и полиномиально обозрима.

# Подграф-алмаз

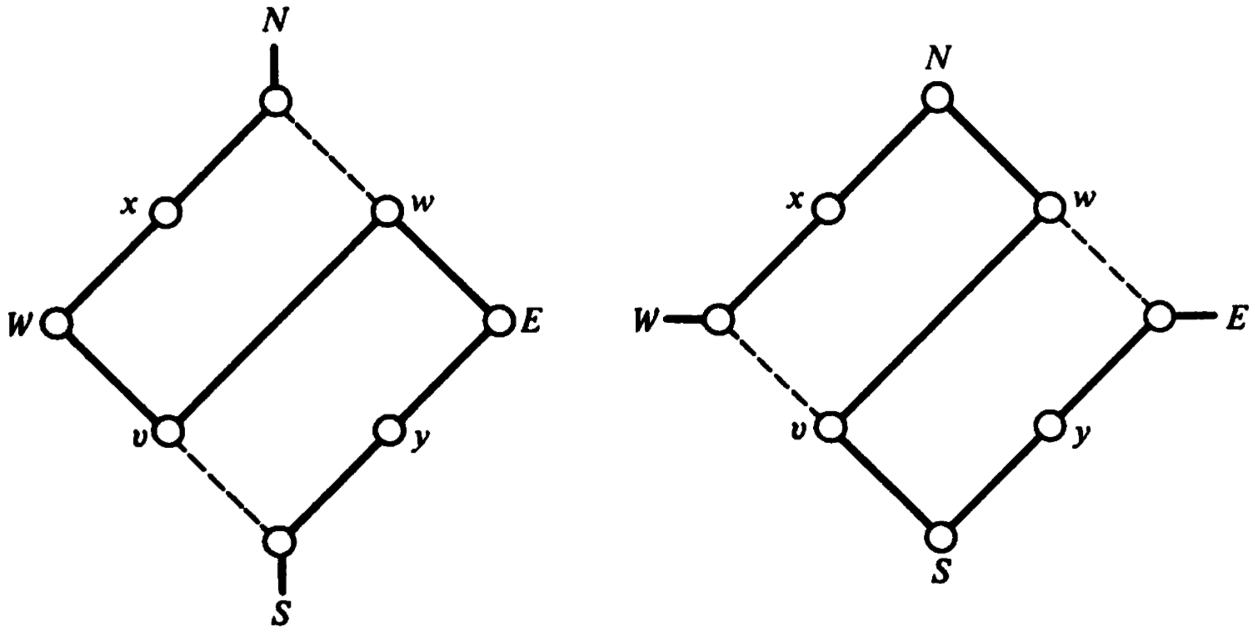
Пусть в некотором графе $G'$ есть такой порождённый подграф:




(Из вершин $x, y, v, w$ не выходят рёбра вовне!)

# Два маршрута через алмаз

Легко проверить: когда гамильтонов цикл в графе $G'$ заходит в подграф-алмаз, он обязан обойти его целиком до выхода из него, причём может быть только два типа обхода подграфа-алмаза: «север-юг» и «восток-запад»:

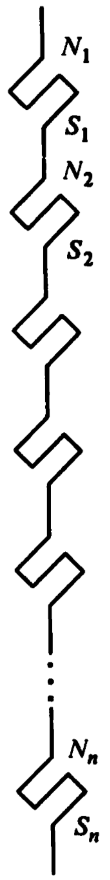


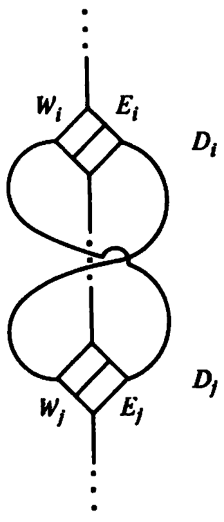
# Переход от обычного графа к «алмазному»

Пусть $G$ — произвольный граф.

Построим по нему граф $G'$:

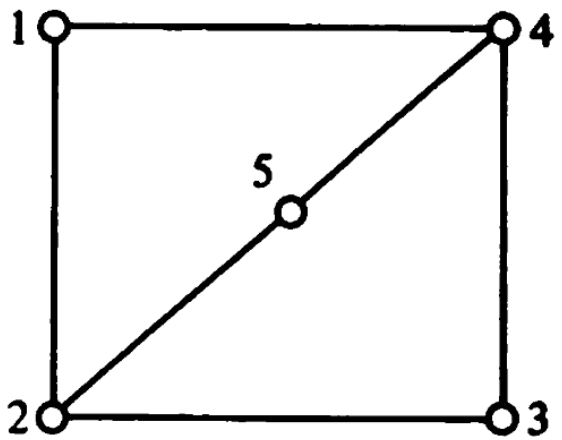
- заменяем каждую вершину $G$ на алмаз,
- все алмазы соединяем последовательно в цепочку, южный полюс алмаза — северный полюс следующего,
- если вершины $u, v$ были смежны в $G$,
то соответствующие алмазы соединяются крест-накрест: запад-восток и восток-запад.



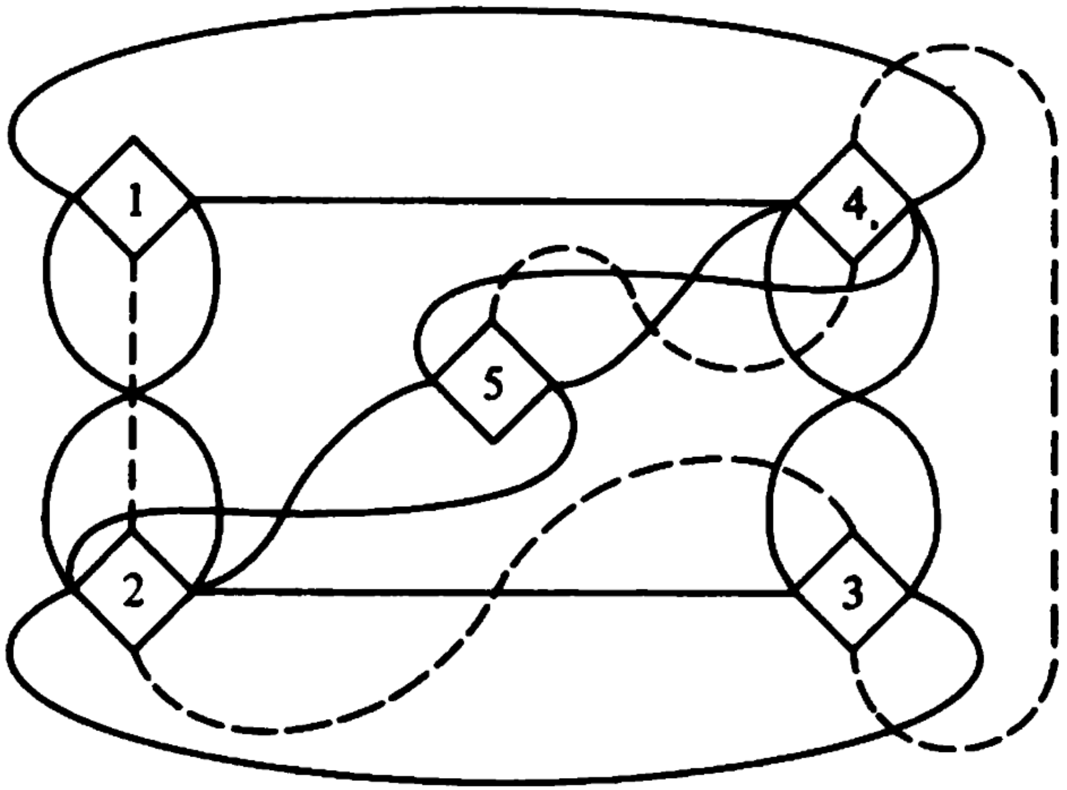



(b)

# Пример перехода от обычного графа к «алмазному»




*G*




*G'*

# Переход от обычного графа к «алмазному»

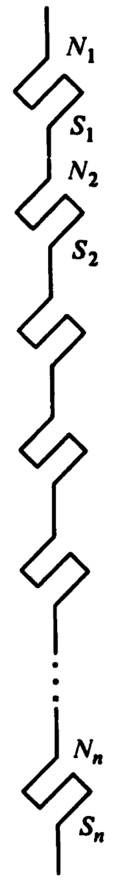
Заметим следующее:

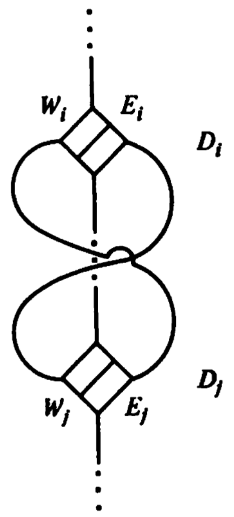
- в $G'$ есть гамильтонова цепь (недаром соединяли все алмазы в цепочку),
- в $G'$ есть г.ц. $\Leftrightarrow$ в $G$ есть г.ц.

Перейдём от $G'$ к полному графу $G''$, положив вес рёбер $E(G')$ равным $1$, а вес остальных рёбер равным 2.

Пусть $n = |G|$. Имеем:

- в $G''$ есть г.ц. веса $n$ $\Leftrightarrow$ в $G$ есть г.ц.,
- в $G''$ в любом случае есть г.ц. веса $n + 1$.




(b)

# Доказательство теоремы Пападимитриу—Стайглица

Допустим, для задачи $\mathrm{TSP}$ есть точная полиномиально обозримая система окрестностей.

Пусть $x$ — гамильтонов цикл в графе $G''$, имеющий вес $(n+1)$.

- Если в $\mathcal{N}(x)$ есть г.ц., вес которого меньше чем у $x$, то в $G$ есть г.ц.
- Если в $\mathcal{N}(x)$ нет г.ц. веса менее $(n+1)$, то (т.к. $\mathcal{N}$ точна) такого г.ц. нет вовсе. Значит, в графе $G$ нет г.ц.

Т.к. $\mathcal{N}$ полиномиально обозрима, то проверка условия

$$\exists y \in \mathcal{N}(x) \quad w(y) < w(x)$$

выполнима за полиномиальное время. Следовательно, за полиномиальное время может быть решена задача о существовании г.ц. в графе $G$ (граф $G$ выбирался произвольно), — противоречие с предположением $\mathrm{P} \neq \mathrm{NP}$.

# Приближённые алгоритмы

Пусть в задаче минимизации $f$ на множестве $\mathcal{S}$

- $x_*$ — оптимальное решение,
- $x_{\mathcal{A}}$— решение, найденное алгоритмом $\mathcal{A}$.

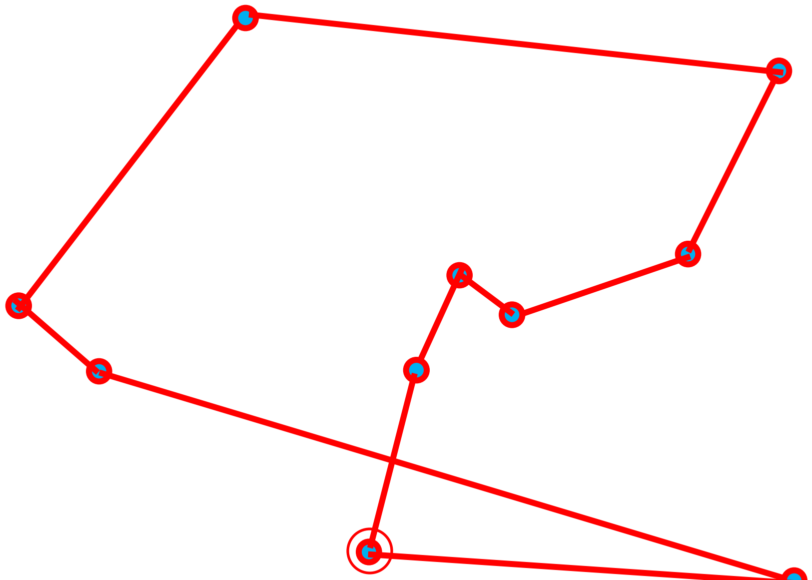
Определим:

- Approximation ratio (показатель качества приближения, показатель аппроксимации, ошибка приближения) алгоритма $\mathcal{A}$:
$$\frac{f(x_{\mathcal{A}})}{f(x_*)}$$
- Domination number (показатель превосходства) алгоритма $\mathcal{A}$:
$$\#\{x \in \mathcal{S} \mid f(x) \geq f(x_{\mathcal{A}})\}$$

# Приближённые алгоритмы

Алгоритм минимизации точный т. и т.т., когда у него

- Approximation ratio $= 1$
- Domination number $= |\mathcal{S}|$



# Алгоритм ближайшего соседа (nearest neighbor, NN) для задачи TSP

1. Начинаем из произвольной вершины.
2. На каждом шаге идём в ближайшую к текущей ещё не посещённую вершину.
3. Когда все вершины пройдены, возвращаемся в стартовую вершину, замыкая цикл.

# Теорема о качестве алгоритма NN

**Теорема [Rosenkrantz, Stearns, & Lewis, 1977].**

Рассматривается только метрическая задача коммивояжёра.

Существуют такие константы $c_1$, $c_2$, что

- для любой задачи с $n$ точками approximation ratio алгоритма NN не превосходит $c_1 \log n$,
- для любого достаточно большого $n$ существует конкретная задача с $n$ точками, для которой approximation ratio алгоритма NN не меньше $c_2 \log n$.

# Верхняя оценка approximation ratio: леммы

Пусть $W^*$ — вес оптимального г.ц. во взвешенном графе $G$, и пусть весовая функция удовлетворяет неравенству треугольника.

**Лемма (о ребре).**
Пусть $e$ — произвольное ребро графа $G$. Тогда

$$w(e) \leq \frac{1}{2} \cdot W^*.$$

**Лемма (об обходе подграфа).**
Пусть $V'$ — произвольное подмножество вершин графа $G$, и пусть $W'$ — вес оптимального г.ц. в графе $G|_{V'}$. Тогда $W' \leq W^*$.

(Упражнение: д-ть, что если нет нер-ва Δ-ка, то леммы неверны.)

# Верхняя оценка approximation ratio: леммы

Для вершины $v$ пусть $L(v)$ — вес того ребра, которое было добавлено в обход алгоритма NN в тот момент, когда вершина $v$ была последней посещённой. Очевидно, $\sum_v L(v) = \text{вес г. ц., построенного NN}$.

**Лемма (о концах ребра).**
Для любого ребра $uv$ выполнено:

$$\left[\begin{array}{l} w(uv) \geq L(u) \\ w(uv) \geq L(v) \end{array}\right.$$

*Доказательство:*

Б.о.о., пусть NN-алгоритм посетил $u$ раньше чем $v$.

Тогда, в силу определения алгоритма, в момент, когда он выбирал, куда пойти из $u$, он выбрал ребро веса $L(u) \leq w(uv)$.

# Верхняя оценка approximation ratio

Пусть $G = (V, E)$ — произвольный (полный) граф с $n$ вершинами, и пусть веса рёбер удовлетворяют неравенству треугольника.

Пусть $W^*$ — вес оптимального г.ц. в $G$.

Пусть $E^{\mathrm{opt}}$ — рёбра оптимального г.ц. в $G$.

Положим

$$\hat{E}^{\mathrm{opt}} \coloneqq \left\{ e \in E^{\mathrm{opt}} \mid w(e) \le \frac{2 \cdot W^*}{|E^{\mathrm{opt}}|} \right\}.$$

Имеем $\left|\hat{E}^{\mathrm{opt}}\right| \ge \frac{1}{2} \cdot |E^{\mathrm{opt}}|$ (если бы это было не так, суммарный вес рёбер в $E^{\mathrm{opt}}$ оказался бы больше $W^*$).

# Верхняя оценка approximation ratio

- $W^*$ — вес оптимального г.ц. в $G$.
- $E^{\mathrm{opt}}$ — рёбра оптимального г.ц. в $G$.
- $\hat{E}^{\mathrm{opt}} \coloneqq \left\{e \in E^{\mathrm{opt}} \mid w(e) \leq \frac{2 \cdot W^*}{|E^{\mathrm{opt}}|}\right\}$.
- $\left|\hat{E}^{\mathrm{opt}}\right| \geq \frac{1}{2} \cdot |E^{\mathrm{opt}}|$.

Для ребра $e = uv$ положим

$$v(e) \coloneqq \begin{cases} u, & \text{если } L(u) \leq w(e), \\ v, & \text{иначе.} \end{cases}$$

(По Лемме о концах ребра, имеем $L\big(v(e)\big) \leq w(e)$.)

Пусть $V_1 \coloneqq \left\{v(e) \mid e \in \hat{E}^{\mathrm{opt}}\right\}$. Имеем

- $|V_1| \geq \frac{1}{2} \cdot \left|\hat{E}^{\mathrm{opt}}\right| \geq \frac{1}{4} \cdot |E^{\mathrm{opt}}| = \frac{1}{4} \cdot |V|$,
- $\sum_{v \in V_1} L(v) \leq \sum_{e \in \hat{E}^{\mathrm{opt}}} w(e) \leq \left|\hat{E}^{\mathrm{opt}}\right| \cdot \frac{2 \cdot W^*}{|E^{\mathrm{opt}}|} \leq 2W^*$.

# Верхняя оценка approximation ratio

Построили множество $V_1 \subseteq V$, такое, что

- $|V_1| \geq \frac{1}{4} \cdot |V|$
- $\sum_{v \in V_1} L(v) \leq 2W^*$

Теперь в графе $G_1 \coloneqq G - V_1$ возьмём оптимальный обход (по Лемме, его вес будет не больше $W^*$), и построим на его основе множество $V_2$, такое что

- $|V_2| \geq \frac{1}{4} \cdot |V(G_1)|$
- $\sum_{v \in V_2} L(v) \leq 2W^*$

Затем в графе $G_2 \coloneqq G_1 - V_2$ выберем множество $V_3$ и т.д.

Будем так делать, пока $|V(G_i)| \geq 3$ (т.е. пока в $G_i$ можно рассматривать г.ц.)

# Верхняя оценка approximation ratio

Пусть построена последовательность множеств $V_1, \ldots, V_k$, такая, что

$$|V(G_k)| < 3.$$

Т.к. $|V(G_{i+1})| \leq \frac{3}{4} \cdot |V(G_i)|$ для каждого $i$, то

$$k \leq \log_{4/3}|V(G)|.$$

Имеем (с учётом Леммы о ребре)

$$\sum_{v \in V(G)} L(v) = \sum_{i=1}^{k} \sum_{v \in V_i} L(v) + \sum_{v \in V(G_k)} L(v) \leq \left(\log_{4/3}|V(G)|\right) \cdot 2W^* + 3 \cdot \tfrac{1}{2} W^*.$$

Отсюда appr. ratio не превосходит $2\left(\log_{4/3} n + 1\right) \lesssim 4.82 \log_2 n$.

# Верхняя оценка approximation ratio

**Упражнение.**

Мы брали в рассуждениях

$$\left\{ e \in E^{\mathrm{opt}} \mid w(e) \leq \frac{K \cdot W^*}{|E^{\mathrm{opt}}|} \right\},$$

где $K = 2$.

Каким нужно взять $K$, чтобы в оценке approximation ratio $4.82 \log_2 n$ получилась константа, меньшая, чем $4.8$?