Об асимптотически точных приближенных алгоритмах для некоторых трудных задач маршрутизации

Эдуард Хайрутдинович Гимади, Оксана Юрьевна Цидулко

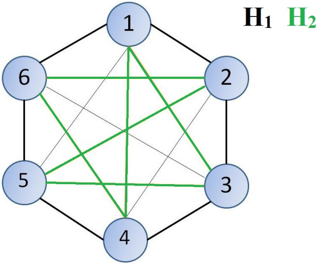
Институт математики им. С.Л. Соболева СО РАН
Новосибирский государственный университет

ММРО-17, Светлогорск
19 Сентября 2015 – 25 Сентября 2015

Contents

m-Peripatetic Salesman Problem

Дан полный граф $G = (V, E)$ и весовые функции ребер $w_i : E \to \mathbf{R}_+$, $i = 1, \ldots, m,$.

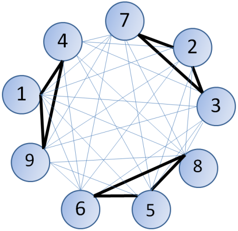


Требуется найти m реберно-непересекающихся гамильтоновых циклов $H_1, \ldots, H_m$, таких что

$$W_1(H_1) + \ldots + W_m(H_m) = \sum_{i=1}^{m} \sum_{e \in H_i} w_i(e) \to \min .$$

Minimum m-cycles cover problem (m-CC-min)

Дан полный граф $G = (V, E)$ и известны веса ребер $w : E \to \mathbf{R}_+$.

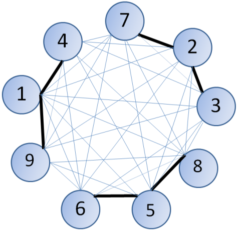


Требуется найти m вершинно-непересекающихся циклов $C_1, \ldots, C_m$, таких что $V(C_1) \cup V(C_2) \cup \ldots \cup V(C_m) = V$ и

$$W(C_1) + \ldots + W(C_m) = \sum_{i=1}^{m} \sum_{e \in C_i} w(e) \to \min$$

Minimum m-chains cover problem (m-ChC-min)

Дан полный граф $G = (V, E)$ и известны веса ребер $w : E \to \mathbf{R}_+$.



Требуется найти m вершинно-непересекающихся простых цепей $C_1, \ldots, C_m$, таких что $V(C_1) \cup V(C_2) \cup \ldots \cup V(C_m) = V$ и

$$W(C_1) + \ldots + W(C_m) = \sum_{i=1}^{m} \sum_{e \in C_i} w(e) \to \min$$

Сложностной статус

Все 3 задачи за полиномиальное время сводятся к TSP$_{min}$ и обратно, поэтому они наследуют ее сложностной и аппроксимационный статус.

- TSP$_{min}$ NP-трудна в сильном смысле (Karp, 1972)
- TSP неприближаема с точностью $O(2^n)$ за полиномиальное время, если $P \neq NP$ (Sahni and Gonzales, 1976)

Приближенные алгоритмы

- I - входные данные задачи
- $F_A(I)$ - значение целевой функции, полученное некоторым алгоритмом А
- $OPT(I)$ – оптимальное значение целевой функции.

Поскольку не существует приближенного полиномиального алгоритма с разумно малой гарантированной оценкой точности $p \geq 1$:

$$F_A(I) \leq pOPT(I),$$

Приближенные алгоритмы

- I - входные данные задачи
- $F_A(I)$ - значение целевой функции, полученное некоторым алгоритмом A
- $OPT(I)$ – оптимальное значение целевой функции.

Поскольку не существует приближенного полиномиального алгоритма с разумно малой гарантированной оценкой точности $p \geq 1$:

$$F_A(I) \leq pOPT(I),$$

предлагается вероятностный подход:

$$Pr\{F_A(I) > (1 + \varepsilon_A)OPT(I)\} \leq \delta_A.$$

Мы хотим ограничить вероятность нежелательного события величиной δ_A

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

- **Вход:** Полный n-вершинный граф $G = (V, E)$ с весовыми функциями $w_i : E \to \mathbf{R}_+,\ i = 1, \ldots, m$, где $m < n/4$
- **Выход:** m реберно-непересекающихся гамильтоновых циклов $H_1, \ldots, H_m$
- **Временная сложность:** $O(mn^2)$

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

- **Вход:** Полный n-вершинный граф $G = (V, E)$ с весовыми функциями $w_i : E \to \mathbf{R}_+,\ i = 1, \ldots, m$, где $m < n/4$
- **Выход:** m реберно-непересекающихся гамильтоновых циклов $H_1, \ldots, H_m$
- **Временная сложность:** $O(mn^2)$
- **Основная идея:** модификация жадного алгоритма, последовательное нахождение гамильтоновых циклов $H_1, \ldots, H_m$.

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

Этап $i = 1, \ldots, m$**.**
На i-ом этапе мы рассматриваем имеющийся граф G с весовой функцией ребер w_i и строим i-ый гамильтонов цикл H_i :

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

Этап $i = 1, \ldots, m$**.**
На i-ом этапе мы рассматриваем имеющийся граф G с весовой функцией ребер w_i и строим i-ый гамильтонов цикл H_i :

Шаг i0

Случайным образом выберем первое ребро $e_1^{(i)}$ для цикла H_i.

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

Этап $i = 1, \ldots, m$**.**
На i-ом этапе мы рассматриваем имеющийся граф G с весовой функцией ребер w_i и строим i-ый гамильтонов цикл H_i :

Шаг i0

Случайным образом выберем первое ребро $e_1^{(i)}$ для цикла H_i.

Шаг i1

Из произвольной вершины ребра $e_1^{(i)}$ построим частичный путь, применяя принцип "иди в ближайшую не пройденную вершину" $n - 4i$ раз.

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

Этап $i = 1, \ldots, m$.
На i-ом этапе мы рассматриваем имеющийся граф G с весовой функцией ребер w_i и строим i-ый гамильтонов цикл H_i :

Шаг i0

Случайным образом выберем первое ребро $e_1^{(i)}$ для цикла H_i.

Шаг i1

Из произвольной вершины ребра $e_1^{(i)}$ построим частичный путь, применяя принцип "иди в ближайшую не пройденную вершину" $n - 4i$ раз.

Шаг i2

Достроим этот частичный путь до гамильтонова цикла H_i с помощью специальной процедуры extension-rotation $\mathbb{P}$.

Алгоритм $\widetilde{A}_1$ для m-PSP$_{min}$

Этап $i = 1, \ldots, m$**.**
На i-ом этапе мы рассматриваем имеющийся граф G с весовой функцией ребер w_i и строим i-ый гамильтонов цикл H_i :

Шаг i0

Случайным образом выберем первое ребро $e_1^{(i)}$ для цикла H_i.

Шаг i1

Из произвольной вершины ребра $e_1^{(i)}$ построим частичный путь, применяя принцип "иди в ближайшую не пройденную вершину" $n - 4i$ раз.

Шаг i2

Достроим этот частичный путь до гамильтонова цикла H_i с помощью специальной процедуры extension-rotation $\mathbb{P}$.

Уберем из графа G все ребра, принадлежащие циклу H_i, для исключения возможности их попадания в следующие циклы $i + 1, \ldots, m$.

Процедура extension-rotation $\mathbb{P}$.

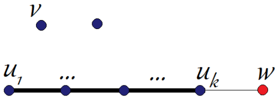
В графе $H = (V_H, E_H)$, с минимальной степенью вершины $> |V_H|/2$, процедура $\mathbb{P}$ строит гамильтонову цепь P с заданными концами u и v за время $O(|V_H|^2)$.

- Пусть построен путь $P = \{u = u_1, \ldots, u_k\}$

Процедура extension-rotation $\mathbb{P}$.

В графе $H = (V_H, E_H)$, с минимальной степенью вершины $> |V_H|/2$, процедура $\mathbb{P}$ строит гамильтонову цепь P с заданными концами u и v за время $O(|V_H|^2)$.

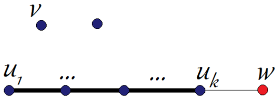
- Пусть построен путь $P = \{u = u_1, \ldots, u_k\}$
- Если возможно, добавим к пути ребро $\{u_k, w\}$, $w \notin P$.



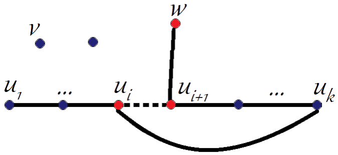
Процедура extension-rotation $\mathbb{P}$.

В графе $H = (V_H, E_H)$, с минимальной степенью вершины $> |V_H|/2$, процедура $\mathbb{P}$ строит гамильтонову цепь P с заданными концами u и v за время $O(|V_H|^2)$.

- Пусть построен путь $P = \{u = u_1, \ldots, u_k\}$
- Если возможно, добавим к пути ребро $\{u_k, w\}$, $w \notin P$.



- Иначе возьмем произвольное $w \notin P$, добавим в P ребра $\{u_k, u_i\}$ и $\{u_{i+1}, w\}$, уберем ребро $\{u_i, u_{i+1}\}$



Алгоритм $\widetilde{A}_2$ для задачи m-CyclesCover

- **Вход:** полный n-вершинный граф $G = (V, E)$ с весами ребер $w : E \to \mathbf{R}_+$, целое $m < n/3$.
- **Выход:** m вершинно-непересекающихся циклов $C_1, \ldots, C_m$
- **Временная сложность:** $O(n^2)$

Алгоритм $\widetilde{A}_2$ для задачи m-CyclesCover

- **Вход:** полный n-вершинный граф $G = (V, E)$ с весами ребер $w : E \to \mathbf{R}_+$, целое $m < n/3$.
- **Выход:** m вершинно-непересекающихся циклов $C_1, \ldots, C_m$
- **Временная сложность:** $O(n^2)$
- **Основная идея:** Жадный алгоритм в предположении, что

$$\#edges(C_i) = \lfloor n/m \rfloor, \quad i = 1, \ldots, m-1$$

$$\#edges(C_m) = n - \sum_{i=1}^{m-1} \#edges(C_i),$$

и первое ребро в каждом цикле выбирается случайно.

Алгоритм $\widetilde{A}_3$ для задачи m-ChainsCover

- **Вход:** полный n-вершинный граф $G = (V, E)$ с весами ребер $w : E \to \mathbf{R}_+$, целое $m < n/2$.
- **Выход:** m вершинно-непересекающихся цепей $C_1, \dots, C_m$
- **Временная сложность:** $O(n^2)$

Алгоритм $\widetilde{A}_3$ для задачи m-ChainsCover

- **Вход:** полный n-вершинный граф $G = (V, E)$ с весами ребер $w : E \to \mathbf{R}_+$, целое $m < n/2$.
- **Выход:** m вершинно-непересекающихся цепей $C_1, \dots, C_m$
- **Временная сложность:** $O(n^2)$
- **Основная идея:** Жадный алгоритм в предположении, что

$$\#edges(C_i) = |n/m| - 1, \quad i = 1, \dots, m-1$$

$$\#edges(C_m) = n - m - \sum_{i=1}^{m-1} \#edges(C_i),$$

Оценки качества алгоритма A на случайных входах

$$\mathbf{Pr}\{F_A > (1 + \varepsilon_A(n))OPT\} \leq \delta_A(n)$$

Оценки качества алгоритма A на случайных входах

$$\mathbf{Pr}\{F_A > (1 + \varepsilon_A(n))OPT\} \leq \delta_A(n)$$

- n – размерность задачи

Оценки качества алгоритма A на случайных входах

$$\mathbf{Pr}\{F_A > (1 + \varepsilon_A(n))OPT\} \leq \delta_A(n)$$

- n – размерность задачи
- $\varepsilon_A(n)$ – оценка относительной погрешности алгоритма

Оценки качества алгоритма A на случайных входах

$$\mathbf{Pr}\{F_A > (1 + \varepsilon_A(n))OPT\} \leq \delta_A(n)$$

- n – размерность задачи
- $\varepsilon_A(n)$ – оценка относительной погрешности алгоритма
- $\delta_A(n)$ – вероятность несрабатывания, т.е. доля случаев, когда алгоритм A не гарантирует относительную погрешность, не превосходящую $\varepsilon_A(n)$.

Оценки качества алгоритма A на случайных входах

$$\mathbf{Pr}\{F_A > (1 + \varepsilon_A(n))OPT\} \leq \delta_A(n)$$

- n – размерность задачи
- $\varepsilon_A(n)$ – оценка относительной погрешности алгоритма
- $\delta_A(n)$ – вероятность несрабатывания, т.е. доля случаев, когда алгоритм A не гарантирует относительную погрешность, не превосходящую $\varepsilon_A(n)$.

Definition

Алгоритм A называется асимптотически точным на классе рассматриваемых задач, если существуют оценки ε_A и δ_A:

$$\varepsilon_A \xrightarrow[n\to\infty]{} 0, \quad \delta_A \xrightarrow[n\to\infty]{} 0.$$

Случайные входы рассматриваемых задач

Входные данные для задачи m-PSP

представим в виде $m \times n \times n$ матрицы весов $C = \left(c_{ijk}\right)$, где c_{ijk} равен значению i-ой весовой функции $w_i(e)$ на ребре $e = (j, k)$, $i = \overline{1, m}$,$j, k = \overline{1, n}$.

Входы для задач m-CyclesCover и m-ChainsCover

представим в виде $n \times n$ матрицы весов $C = \left(c_{ij}\right)$, где c_{ij} равен значению $w(e)$ на ребре $e = (i, j)$, $i, j = \overline{1, n}$.

Случайные входы рассматриваемых задач

Входные данные для задачи m-PSP

представим в виде $m \times n \times n$ матрицы весов $C = \left(c_{ijk}\right)$, где c_{ijk} равен значению i-ой весовой функции $w_i(e)$ на ребре $e = (j, k)$, $i = \overline{1, m}$,$j, k = \overline{1, n}$.

Входы для задач m-CyclesCover и m-ChainsCover

представим в виде $n \times n$ матрицы весов $C = \left(c_{ij}\right)$, где c_{ij} равен значению $w(e)$ на ребре $e = (i, j)$, $i, j = \overline{1, n}$.

Под случайным входом для этих задач

будем понимать соответствующую матрицу весов C, все элементы которой являются независимыми одинаково распределенными случайными величинами.

Функции распределения

Определение

Функция распределения $\widetilde{\mathcal{F}}(x)$ является функцией $\mathcal{F}$-мажорирующего типа, если

$$\widetilde{\mathcal{F}}(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Функции распределения

Определение

Функция распределения $\widetilde{\mathcal{F}}(x)$ является функцией $\mathcal{F}$-мажорирующего типа, если

$$\widetilde{\mathcal{F}}(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Функции распределения

Мы рассматривали случайные входы для наших задач со следующими функциями распределения

Функции распределения

Определение

Функция распределения $\widetilde{\mathcal{F}}(x)$ является функцией $\mathcal{F}$-мажорирующего типа, если

$$\widetilde{\mathcal{F}}(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Функции распределения

Мы рассматривали случайные входы для наших задач со следующими функциями распределения

- UNI$[a_n, b_n]$-мажорирующего типа,
  где UNI$[a_n, b_n]$ – равномерное распределение на $[a_n, b_n]$, $0 < a_n < b_n$;

Функции распределения

Определение

Функция распределения $\widetilde{\mathcal{F}}(x)$ является функцией $\mathcal{F}$-мажорирующего типа, если

$$\widetilde{\mathcal{F}}(x) \geq \mathcal{F}(x) \quad \text{for every x}$$

Функции распределения

Мы рассматривали случайные входы для наших задач со следующими функциями распределения

- UNI$[a_n, b_n]$-мажорирующего типа,
  где UNI$[a_n, b_n]$ – равномерное распределение на $[a_n, b_n]$, $0 < a_n < b_n$;
- $\mathcal{F}_\beta$-мажорирующего типа,
  где $\mathcal{F}_\beta(x)$ – показательное распределение с параметром $\beta = \beta_n$:
  $$\mathcal{F}_\beta(x) = 1 - \exp\left(\frac{x - a_n}{\beta}\right), \; x \geq a_n > 0.$$

Вероятностный анализ. Основные идеи

$H_i = \{e_1^{(i)}, \ldots, e_n^{(i)}\}$ – i-ый построенный гамильтонов цикл в m-PSP
$C_i = \{e_1^{(i)}, \ldots, e_{n_i}^{(i)}\}$ – i-ый построенный цикл(цепь) в задаче m-CC (m-ChC)

Вероятностный анализ. Основные идеи

$H_i = \{e_1^{(i)}, \ldots, e_n^{(i)}\}$ – i-ый построенный гамильтонов цикл в m-PSP
$C_i = \{e_1^{(i)}, \ldots, e_{n_i}^{(i)}\}$ – i-ый построенный цикл(цепь) в задаче m-CC (m-ChC)

Оценки качества для алгоритмов, решающих эти задачи, определяются следующими неравенствами:

m-PSP:
$$Pr\Big\{\sum_{i=1}^{m}\sum_{s=1}^{n} w_i(e_s^{(i)}) > (1+\varepsilon_{\widetilde{A}})OPT\Big\} \le \delta_{\widetilde{A}}.$$

m-CC:
$$Pr\Big\{\sum_{i=1}^{m}\sum_{s=1}^{n_i} w(e_s^{(i)}) > (1+\varepsilon_{\widetilde{A}})OPT\Big\} \le \delta_{\widetilde{A}}.$$

Вероятностный анализ. Основные идеи

$H_i = \{e_1^{(i)}, \ldots, e_n^{(i)}\}$ – i-ый построенный гамильтонов цикл в m-PSP
$C_i = \{e_1^{(i)}, \ldots, e_{n_i}^{(i)}\}$ – i-ый построенный цикл(цепь) в задаче m-CC (m-ChC)

Оценки качества для алгоритмов, решающих эти задачи, определяются следующими неравенствами:

m-PSP: $$Pr\Big\{\sum_{i=1}^{m}\sum_{s=1}^{n} w_i(e_s^{(i)}) > (1+\varepsilon_{\widetilde{A}})OPT\Big\} \le \delta_{\widetilde{A}}.$$

m-CC: $$Pr\Big\{\sum_{i=1}^{m}\sum_{s=1}^{n_i} w(e_s^{(i)}) > (1+\varepsilon_{\widetilde{A}})OPT\Big\} \le \delta_{\widetilde{A}}.$$

Положим $\xi_{is} = w_i(e_s^{(i)})$

Вероятностный анализ. Основные идеи

- Все веса ξ_{is} выбранных алгоритмом ребер – независимые случайные величины.

Вероятностный анализ. Основные идеи

- Все веса ξ_{is} выбранных алгоритмом ребер – независимые случайные величины.
- Вес ξ_{is} ребра, выбранного в жадной части алгоритма, оценивается сверху как минимум из

$$\begin{cases} \text{n-2i-s+2} & \text{for m-PSP} \\ \text{n-(i-1)n/m - s - 1} & \text{for m-CC-problem} \\ \text{n-(i-1)n/m - s} & \text{for m-ChC-problem} \end{cases}$$

элементов матрицы случайного входа.

Вероятностный анализ. Основные идеи

- Все веса ξ_{is} выбранных алгоритмом ребер – независимые случайные величины.
- Вес ξ_{is} ребра, выбранного в жадной части алгоритма, оценивается сверху как минимум из

$$\begin{cases} \text{n-2i-s+2} & \text{for m-PSP} \\ \text{n-(i-1)n/m - s - 1} & \text{for m-CC-problem} \\ \text{n-(i-1)n/m - s} & \text{for m-ChC-problem} \end{cases}$$

  элементов матрицы случайного входа.
- Веса ребер, выбранных не в жадной части алгоритма, имеют то же распределение, что и элементы входа.

Вероятностный анализ. Основные идеи

- Все веса ξ_{is} выбранных алгоритмом ребер – независимые случайные величины.
- Вес ξ_{is} ребра, выбранного в жадной части алгоритма, оценивается сверху как минимум из

$$\begin{cases} \text{n-2i-s+2} & \text{for m-PSP} \\ \text{n-(i-1)n/m - s - 1} & \text{for m-CC-problem} \\ \text{n-(i-1)n/m - s} & \text{for m-ChC-problem} \end{cases}$$

  элементов матрицы случайного входа.
- Веса ребер, выбранных не в жадной части алгоритма, имеют то же распределение, что и элементы входа.
- Используем очевидное неравенство $OPT \geq a_n mn$

Вероятностный анализ. Основные идеи

- Все веса ξ_{is} выбранных алгоритмом ребер – независимые случайные величины.
- Вес ξ_{is} ребра, выбранного в жадной части алгоритма, оценивается сверху как минимум из

$$\begin{cases} \text{n-2i-s+2} & \text{for m-PSP} \\ \text{n-(i-1)n/m - s - 1} & \text{for m-CC-problem} \\ \text{n-(i-1)n/m - s} & \text{for m-ChC-problem} \end{cases}$$

  элементов матрицы случайного входа.
- Веса ребер, выбранных не в жадной части алгоритма, имеют то же распределение, что и элементы входа.
- Используем очевидное неравенство $OPT \geq a_n mn$
- Используем теорему Петрова.

Петров В.В. Предельные теоремы для сумм независимых случайных величин, 1987

Theorem

Пусть $\eta_1, \ldots, \eta_n$ *– независимые случайные величины и* $S = \Sigma_{k=1}^n \eta_k$.
Если для некоторых положительных постоянных $g_1, \ldots, g_n$ *и* T

$$\mathbf{E} e^{t\eta_k} \leq e^{\frac{g_k t^2}{2}},\ 0 \leq t \leq T,\ k = 1, \ldots, n,$$

то

$$\mathbf{Pr}\{S \geq x\} \leq \begin{cases} e^{\frac{-x^2}{2G}}, & 0 \leq x \leq \mathcal{G}T, \\ e^{\frac{-Tx}{2}}, & x \geq \mathcal{G}T \end{cases}$$

Здесь $\mathbf{E}X$ *– мат.ожидание случайной величины* X, *а* $\mathcal{G} = \Sigma_{k=1}^n g_k$.

Вероятностный анализ алгоритмов

Для применения теоремы Петрова к нашим задачам, мы адаптировали постоянные $g_1, \ldots, g_n$ и T из результатов статей:

Вероятностный анализ алгоритмов

Для применения теоремы Петрова к нашим задачам, мы адаптировали постоянные $g_1, \dots, g_n$ и T из результатов статей:

For uniform distribution function:

E. Kh. Gimadi,Yu. V. Glazkov *An asymptotically exact algorithm for one modification of planar three-index assignment problem//* Journal of Applied and Industrial Mathematics December 2007, Volume 1, Issue 4, pp 442-452

Вероятностный анализ алгоритмов

Для применения теоремы Петрова к нашим задачам, мы адаптировали постоянные $g_1, \ldots, g_n$ и T из результатов статей:

For uniform distribution function:

E. Kh. Gimadi,Yu. V. Glazkov *An asymptotically exact algorithm for one modification of planar three-index assignment problem*// Journal of Applied and Industrial Mathematics December 2007, Volume 1, Issue 4, pp 442-452

For exponential distribution function:

E. Kh. Gimadi, A. Le Gallou, A. V. Shakhshneyder, *Probabilistic analysis of an approximation algorithm for the traveling salesman problem on unbounded above instances*// Journal of Applied and Industrial Mathematics April 2009, Volume 3, Issue 2, pp 207-221

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения **UNI**$[a_n, b_n]$-**мажорирующего типа**, $0 < a_n < b_n$, приведенные алгоритмы для данных задач будут асимптотически точными

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения **UNI**$[a_n, b_n]$-**мажорирующего типа**, $0 < a_n < b_n$, приведенные алгоритмы для данных задач будут асимптотически точными

- при $2 \le m \le \ln n$

$$\varepsilon_{\widetilde{A}} = O\left(\frac{b_n/a_n}{n/\ln n}\right), \quad \delta_{\widetilde{A}} = n^{-9},$$

$$\text{если} \quad \frac{b_n}{a_n} = o\Big(\frac{n}{\ln n}\Big);$$

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения **UNI**$[a_n, b_n]$**-мажорирующего типа**, $0 < a_n < b_n$, приведенные алгоритмы для данных задач будут асимптотически точными

- при $2 \le m \le \ln n$

$$\varepsilon_{\widetilde{A}} = O\left(\frac{b_n/a_n}{n/\ln n}\right), \quad \delta_{\widetilde{A}} = n^{-9},$$

$$\text{если} \quad \frac{b_n}{a_n} = o\left(\frac{n}{\ln n}\right);$$

- при $\ln n < m \le n^{1-\theta}, \theta \in (0,1)$

$$\varepsilon_{\widetilde{A}} = O\left(\frac{b_n/a_n}{n^\theta}\right), \quad \delta_{\widetilde{A}} = n^{-9},$$

$$\text{если} \quad \frac{b_n}{a_n} = o(n^\theta).$$

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения, **мажорирующей показательное распределение** $\mathcal{F}_\beta$, приведенные алгоритмы для данных задач будут асимптотически точными

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения, **мажорирующей показательное распределение** $\mathcal{F}_\beta$, приведенные алгоритмы для данных задач будут асимптотически точными

- для $2 \le m \le \ln n$

$$\varepsilon_{\widetilde{A}} = O\left(\frac{\beta/a_n}{n/\ln n}\right), \quad \delta_{\widetilde{A}} = n^{-3m/4},$$

$$\text{если} \quad \frac{\beta}{a_n} = o\left(\frac{n}{\ln n}\right);$$

Вероятностный анализ. Результаты

Для случайных входов с функцией распределения, **мажорирующей показательное распределение** $\mathcal{F}_\beta$, приведенные алгоритмы для данных задач будут асимптотически точными

- для $2 \le m \le \ln n$

$$\varepsilon_{\widetilde{A}} = O\left(\frac{\beta/a_n}{n/\ln n}\right), \quad \delta_{\widetilde{A}} = n^{-3m/4},$$

$$\text{если} \quad \frac{\beta}{a_n} = o\left(\frac{n}{\ln n}\right);$$

- для $\ln n < m \le n^{1-\theta}, \theta \in (0,1)$

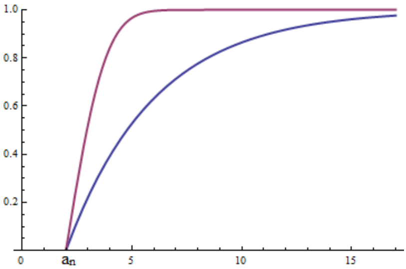
$$\varepsilon_{\widetilde{A}} = O\left(\frac{\beta/a_n}{n^\theta}\right), \quad \delta_{\widetilde{A}} = n^{-3m/4},$$

$$\text{если} \quad \frac{\beta}{a_n} = o(n^\theta).$$

Вероятностный анализ. Результаты

Следствие

Нормальное рспределение с подходящими параметрами мажорирует показательное.
Оценки качества алгоритмов, полученные для случайных входов с **показательным** *распределением с параметром β и средним a_n, будут верны для входов с* **усеченным нормальным** *распределением с параметром $\sigma = \frac{\beta}{2}$ и средним a_n.*

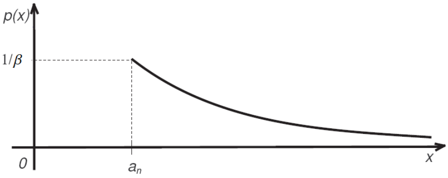


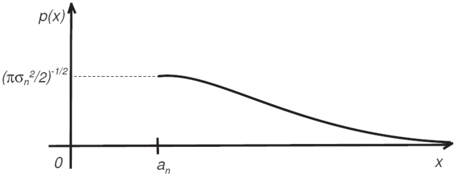
Заключение

1. **Для труднорешаемых задач маршрутизации m-PSP, m-CyclesCover и m-ChainsCover были построены простые полиномиальные алгоритмы.**
2. **Для случайных входов с распределениями, мажорирующими равномерное или показательное, проведен вероятностный анализ алгоритмов и получены оценки их качества.**
3. **Найдены достаточные условия асимптотической точности этих алгоритмов.**

Спасибо за внимание!

Вероятностный анализ. Результаты





Показательное распределение

$$p(x) = \begin{cases} \dfrac{1}{\beta_n} \exp\left(-\dfrac{x-a_n}{\beta_n}\right), & \text{если } a_n \le x \le \infty, \\ 0, & \text{в противном случае.} \end{cases}$$

Усеченное нормальное распределение

$$p(x) = \begin{cases} \dfrac{2}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\dfrac{(x-a_n)^2}{2\sigma_n^2}\right), & \text{если } a_n \le x \le \infty, \\ 0, & \text{в противном случае.} \end{cases}$$

Распределения мажорирующего типа

Оценки качества алгоритма, полученные для входных данных с рассмотренными распределениями, будут верны и для соответствующих распределений мажорирующего типа.

Утверждение 1

Пусть $\xi_1, ..., \xi_k$ независимые случайные величины с распределением $F(x)$, а $\hat{F}(x)$ функция распределения случайной величины $\xi = \min(\xi_1, ..., \xi_k)$,
Пусть $\eta_1, ..., \eta_k$ независимые случайные величины с распределением $G(x)$, а $\hat{G}(x)$ функция распределения случайной величины $\eta = \min(\eta_1, ..., \eta_k)$.

Тогда для всех x

$$F(x) \leq G(x) \Rightarrow \hat{F}(x) \leq \hat{G}(x).$$

Распределения мажорирующего типа

Оценки качества алгоритма, полученные для входных данных с рассмотренными распределениями, будут верны и для соответствующих распределений мажорирующего типа.

Утверждение 1

Пусть $\xi_1, ..., \xi_k$ независимые случайные величины с распределением $F(x)$, а $\hat{F}(x)$ функция распределения случайной величины $\xi = \min(\xi_1, ..., \xi_k)$,
Пусть $\eta_1, ..., \eta_k$ независимые случайные величины с распределением $G(x)$, а $\hat{G}(x)$ функция распределения случайной величины $\eta = \min(\eta_1, ..., \eta_k)$.

Тогда для всех x

$$F(x) \leq G(x) \Rightarrow \hat{F}(x) \leq \hat{G}(x).$$

Распределения мажорирующего типа

Оценки качества алгоритма, полученные для входных данных с рассмотренными распределениями, будут верны и для соответствующих распределений мажорирующего типа.

Утверждение 1

Пусть $\xi_1, ..., \xi_k$ независимые случайные величины с распределением $F(x)$, а $\hat{F}(x)$ функция распределения случайной величины $\xi = \min(\xi_1, ..., \xi_k)$,
Пусть $\eta_1, ..., \eta_k$ независимые случайные величины с распределением $G(x)$, а $\hat{G}(x)$ функция распределения случайной величины $\eta = \min(\eta_1, ..., \eta_k)$.

Тогда для всех x

$$F(x) \le G(x) \Rightarrow \hat{F}(x) \le \hat{G}(x).$$

Доказательство следует из

$$\hat{F}(x) = 1 - (1 - F(x))^k \quad \text{and} \quad \hat{G}(x) = 1 - (1 - G(x))^k.$$

Распределения мажорирующего типа

Утверждение 2

Пусть $P_\xi, P_\eta, P_\zeta, P_\chi$ функции распределения случайных величин ξ, η, ζ, χ, соответственно. И пусть ξ и ζ независимые, η и χ независимые случайные величины. Тогда

$$(\forall x\ P_\xi(x) \leq P_\eta(x)) \wedge (\forall y\ P_\zeta(y) \leq P_\chi(y)) \Rightarrow (\forall z\ P_{\xi+\zeta}(z) \leq P_{\eta+\chi}(z)).$$

Доказательство

$$P_{\xi+\zeta}(x) = \int\limits_{-\infty}^{\infty} P_\xi(x-y)dP_\zeta(y) \leq \int\limits_{-\infty}^{\infty} P_\eta(x-y)dP_\zeta(y)$$

$$= P_{\eta+\zeta}(x) = \int\limits_{-\infty}^{\infty} P_\zeta(x-y)dP_\eta(y) \leq \int\limits_{-\infty}^{\infty} P_\chi(x-y)dP_\eta(y) = P_{\eta+\chi}(x).$$

Распределения мажорирующего типа

- Проведенный анализ содержал только перобразования рассмотренные в утверждениях 1-2

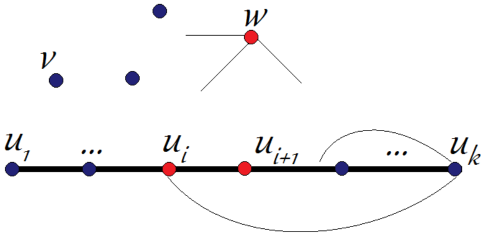
Распределения мажорирующего типа

- Проведенный анализ содержал только перобразования рассмотренные в утверждениях 1-2
- Все веса ребер решения задачи m-PSP независимые случайные величины

Распределения мажорирующего типа

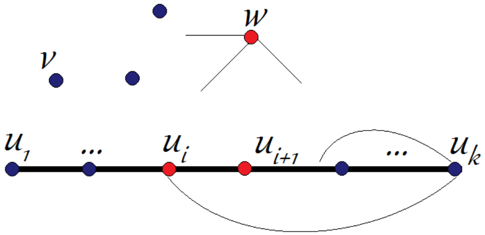
- Проведенный анализ содержал только перобразования рассмотренные в утверждениях 1-2
- Все веса ребер решения задачи m-PSP независимые случайные величины
- Т.о. для входных данных с функциями распределения **UNI**$[a_n, b_n]$-**мажорирующего типа**, и **показательного** $\mathcal{F}_\beta$-**мажорирующего типа**, алгоритм будет асимптотически точен с соответствующими оценками качества.

Correctness of the procedure $\mathbb{P}$



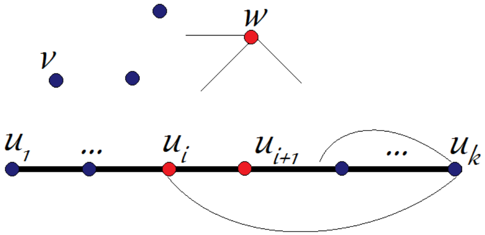
- Suppose, there is no edge $\{u_i, u_{i+1}\} \in P$ such that $\{u_k, u_i\}$ and $\{w, u_{i+1}\} \in E_H$.

Correctness of the procedure $\mathbb{P}$



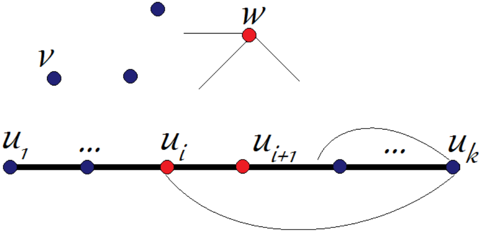
- Suppose, there is no edge $\{u_i, u_{i+1}\} \in P$ such that $\{u_k, u_i\}$ and $\{w, u_{i+1}\} \in E_H$.
- There are $> \hat{n}/2$ vertices adjacent to w.

Correctness of the procedure $\mathbb{P}$



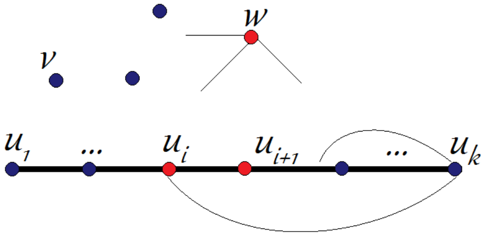
- Suppose, there is no edge $\{u_i, u_{i+1}\} \in P$ such that $\{u_k, u_i\}$ and $\{w, u_{i+1}\} \in E_H$.
- There are $> \hat{n}/2$ vertices adjacent to w.
- Vertices not adjacent to w: w, u_k, and u_{i+1}, where $i : \{u_k, u_i\} \in E_H$.

Correctness of the procedure $\mathbb{P}$



- Suppose, there is no edge $\{u_i, u_{i+1}\} \in P$ such that $\{u_k, u_i\}$ and $\{w, u_{i+1}\} \in E_H$.
- There are $> \hat{n}/2$ vertices adjacent to w.
- Vertices not adjacent to w: w, u_k, and u_{i+1}, where $i : \{u_k, u_i\} \in E_H$.
- So there are $> 1 + 1 + \hat{n}/2 - 2 = \hat{n}/2$ vertices that are not adjacent to w.

Correctness of the procedure $\mathbb{P}$



- Suppose, there is no edge $\{u_i, u_{i+1}\} \in P$ such that $\{u_k, u_i\}$ and $\{w, u_{i+1}\} \in E_H$.
- There are $> \hat{n}/2$ vertices adjacent to w.
- Vertices not adjacent to w: w, u_k, and u_{i+1}, where $i : \{u_k, u_i\} \in E_H$.
- So there are $> 1 + 1 + \hat{n}/2 - 2 = \hat{n}/2$ vertices that are not adjacent to w.

Contradiction.