Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 20 сентября. Срок сдачи: 11 октября. Для зачёта по заданию нужно набрать в сумме хотя бы 9 баллов. Любые вопросы задавайте по почте.

1. Поставьте формально задачу оптимизации (укажите множество S и функцию f) для следующей содержательной задачи:

   (a) (2 балла) Есть несколько контейнеров различного объёма и веса. Требуется выбрать из них несколько контейнеров так, чтобы они влезли в грузовик фиксированной вместительности, а суммарный вес их был как можно больше.

   (b) (2 балла) Есть несколько городов, изначально не соединённых дорогами. Известна стоимость прокладки пути между любой парой городов. Нужно определить, между какими из городов нужно проложить дороги, так, чтобы из любого города в любой другой можно было доехать (возможно, через другие города). Хочется потратить на строительство как можно меньше денег из федерального бюджета.

   (c) (2 балла) Та же задача, что в предыдущем пункте, но требуется дополнительно, чтобы возможность доехать из любого города в любой сохранялась даже в случае, когда одна из дорог между произвольной парой городов закрывается на ремонт.

2. (2 балла) Приведите пример какой-нибудь окрестностной функции для задачи о назначениях.

3. Определим k-окрестность гамильтонова цикла C в графе как множество всех циклов, которые можно получить удалением k рёбер из C и добавлением других k рёбер. На лекции было показано, что рассмотрение 2-окрестностей не гарантирует достижение глобального оптимума в задаче коммивояжёра при локальном поиске. Покажите то же для k-окрестностей при

   (a) (2 балла) $k = 3$,

   (b) (4 балла) $k \leqslant n - 3$, где n — количество вершин в графе.

4. (4 балла) Верно ли, что для любых двух гамильтоновых циклов C', C'' в полном графе можно найти последовательность гамильтоновых циклов $C_1, C_2, \ldots, C_k$, такую, что $C_1 = C'$, $C_k = C''$, и для каждого i цикл C_{i+1} принадлежит 2-окрестности цикла C_i?

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 27 сентября. Срок сдачи: 11 октября. Для зачёта по заданию нужно набрать в сумме хотя бы 7 баллов. Любые вопросы задавайте по почте.

1. (1 балл) В доказательстве леммы об изолировании при определении величины $\alpha(s)$ рассматриваются минимумы вида $\min_{A\in\mathcal{F},\, s\notin A} w(A)$. Вопрос: что делать, если таких множеств A в семействе $\mathcal{F}$ не найдётся?

2. (2 балла) Докажите аналог леммы об изолировании для задачи поиска множества с *максимальным* весом.

3. (2 балла) Докажите аналог леммы об изолировании, при условии, что теперь вес множества определяется не суммой, а произведением весов всех его элементов.

4. (2 балла) Пусть $(S',\, \mathcal{F}')$ и S'', $\mathcal{F}''$ — матроиды, причём $S' \cap S'' = \varnothing$. Положим $S = S' \cup S''$, $\mathcal{F} = \{A' \cup A'' \mid A' \in \mathcal{F}',\, A'' \in \mathcal{F}''\}$. Докажите, что пара $(S,\, \mathcal{F})$ также является матроидом.

5. Пусть S — произвольное конечное множество, k — фиксированное натуральное число, $1 \leqslant k \leqslant |S| - 1$. Положим $\mathcal{F} = \{A \subseteq S \mid |A| \leqslant k\}$.

   (a) (1 балл) Докажите, что пара $(S,\, \mathcal{F})$ — матроид.

   (b) (4 балла) Положим $n = |S|$. Пусть $P(n,k,N)$ — вероятность того, что в данном матроиде при случайном выборе весов из множества $\{1,\, \ldots,\, N\}$ в задаче поиска независимого множества максимального веса будет единственное решение. Докажите, что

$$P(n,k,N) = C_n^k \cdot \sum_{i=1}^{N-1} \left(\left(1 - \frac{i}{N}\right)^k - \left(1 - \frac{i+1}{N}\right)^k \right) \left(\frac{i}{N}\right)^{n-k}.$$

6. (2 балла) Пусть S —множество мощности n, а $\mathcal{F}$ —наследственная система подмножеств S. Пусть веса элементов множества S выбираются среди натуральных чисел $\{1,\, \ldots,\, N\}$. Допустим, что пара $(S,\, \mathcal{F})$ не является матроидом. Какое максимальное значение может принимать разность $\max_{A\in\mathcal{F}} w(A) - w(A')$, где A' — множество, построенное жадным алгоритмом? Дайте ответ в виде функции от величин n, N.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 4 октября. Срок сдачи: 1 ноября, 18:10. Для зачёта по заданию нужно набрать в сумме хотя бы 5 баллов. Любые вопросы задавайте по почте.

1. (3 балла) Докажите включение: $RP \cap co-RP \subseteq ZPP$. Иными словами, по двум алгоритмам $\gamma_1 \in RP$ и $\gamma_2 \in co-RP$ постройте алгоритм $\gamma \in ZPP$. В алгоритме γ допускается вызывать любое количество раз алгоритмы γ_1 и γ_2 как подпрограммы (только проследите, чтобы матожидание суммарного времени работы γ было полиномиальным по размеру входных данных).

2. (2 балла) Пусть $\vec{a}, \vec{b} \in \mathbb{R}^n$, и пусть $\vec{x}$ — вектор из нулей и единиц, компоненты которого выбираются случайным образом. Докажите, что при $\vec{a} \neq \vec{b}$ вероятность выполнения равенства $(\vec{a}, \vec{x}) = (\vec{b}, \vec{x})$ не превосходит $\frac{1}{2}$.

3. (2 балла) У Алисы и Боба есть графы на n пронумерованных вершинах (у каждого — свой). Докажите (используя рандомизацию), что, передав суммарно друг другу $O(\log n)$ битов, Алиса и Боб смогут выяснить, совпадают ли их графы, с вероятностью ошибки не более $\frac{1}{16}$.

4. (3 балла) Пусть полиномы P, Q от n переменных имеют степень d. Допустим, что полиномы P и Q заданы нам в виде вычисляющих эти полиномы «чёрных ящиков», причём эти ящики сами иногда работают с ошибками. Мы можем подать на вход чёрного ящика произвольный набор значений переменных, а ящик возвращает нам значение полинома на этом наборе с вероятностью $\frac{3}{4}$, и возвращает 0 с вероятностью $\frac{1}{4}$. Наша задача — проверить, совпадают ли полиномы P и Q, используя приём из теоремы Шварца–Зиппеля: рассмотрев значения полиномов на случайном наборе значений переменных. Докажите, что для получения правильного ответа в задаче проверки совпадения полиномов с вероятностью $\geqslant \frac{5}{8}$ достаточно выбирать значения переменных случайным образом из множества $\{1, 2, \ldots, 3d\}$.

5. (3 балла) С помощью вычисления определителей матриц можно определять, есть ли совершенное паросочетание в произвольном графе (на лекции были рассмотрены только двудольные графы). Определим по графу G матрицу M_G, в которой элемент i–й строки и j–го столбца равен

   - x_{ij}, если в G есть ребро между i-й и j-й вершинами, и $i < j$,
   - $-x_{ji}$, если в G есть ребро между i-й и j-й вершинами, и $i > j$,
   - 0, если в G нет ребра между i-й и j-й вершинами.

   Докажите, что $\det M_G \equiv 0$ тогда и только тогда, когда в G отсутствует совершенное паросочетание.

Задание по лекции 11 октября. Срок сдачи: 1 ноября, 18:10. Для зачёта по заданию нужно набрать в сумме хотя бы 4 балла. Любые вопросы задавайте по почте.

1. (a) (1 балл) Пусть веса на рёбрах полного графа равны 1 или 2. Всегда ли такой граф будет метрическим?

   (b) (1 балл) Пусть веса рёбер полного графа выбираются из множества $\{1, a\}$, где a — положительное действительное число, большее единицы. В каком диапазоне должно лежать a, чтобы любой такой граф являлся метрическим.

   (c) (2 балла) Веса рёбер полного графа на n вершинах выбираются случайным образом из множества $\{1, 2, 3\}$. Доказать, что вероятность того, что полученный граф будет метрическим, не превосходит 0.9^n при всех достаточно больших n.

2. (4 балла) В задаче 5 с прошлой лекции показано, как применять вычисление определителей для нахождения совершенных паросочетаний в произвольных (недвудольных) графах. Используя этот факт, обобщите параллельный алгоритм поиска паросочетания на недвудольные графы.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 18 октября. Срок сдачи: 15 ноября. Для зачёта по заданию нужно набрать в сумме хотя бы 5 баллов. Любые вопросы задавайте по почте.

1. (1 балл) Приведите пример метрического графа, вершины которого нельзя уложить на плоскости так, чтобы евклидовы расстояния между вершинами равнялись весам соответствующих рёбер.

2. (2 балла) Постройте «метрический» граф, на котором алгоритм Кристофидеса даёт приближение, *асимптотически равное* $\frac{3}{2}$OPT.

3. (2 балла) Покажите, что при решении евклидовой задачи коммивояжёра нельзя производить перемасштабирование только по одной координате. Иначе говоря, приведите набор точек на плоскости, такой, что оптимальные гамильтоновы циклы для исходного набора и для набора в перемасштабированной системе координат различны. Под перемасштабированием понимается переход от точек (x, y) к точкам $(\alpha x, y)$, где $\alpha \neq 1$.

4. (2 балла) Докажите, что проделанные на лекции модификации задачи коммивояжёра (заключение точек в минимальный квадрат, переход к целочисленным координатам) сработают с тем же успехом и для евклидовой задачи коммивояжёра в трёхмерном пространстве.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 25 октября. Срок сдачи: 15 ноября. Для зачёта по заданию нужно набрать в сумме хотя бы 4 балла. Любые вопросы задавайте по почте.

1. (3 балла) Проверьте, что для вычисления обратной матрицы для верхнетреугольной $n \times n$-матрицы с единицами на главной диагонали действительно достаточно схем глубины $O((\log n)^2)$. На лекции было дано общее описание подхода. Восстановите детали и оцените глубину полученной схемы. Проследите, какие операции можно выполнять одновременно, а какие приходится выполнять последовательно.

2. (1 балл) Докажите равенство $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

3. (1 балл) Пусть A — конечное множество точек на плоскости. Докажите, что длина гамильтонова цикла через эти точки не меньше, чем периметр выпуклой оболочки множества A.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 15 ноября. Срок сдачи: 19 декабря. Для зачёта по заданию нужно набрать в сумме хотя бы 5 баллов. Любые вопросы задавайте по почте.

1. (5 баллов) Реализуйте на языке C++ генетический алгоритм решения задачи коммивояжёра. В качестве функций мутации и кроссовера можно использовать предложенные на лекции.

2. (5 баллов) Реализуйте на языке C++ «муравьиный» алгоритм решения задачи коммивояжёра.

3. (3 балла) Предложите схему генетического алгоритма для задачи о назначениях. Придумайте подходящие функции мутации и кроссовера.

4. (3 балла) Предложите схему генетического алгоритма для задачи поиска совершенного паросочетания минимального веса в полном двудольном графе. Придумайте подходящие функции мутации и кроссовера.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 22 ноября. Срок сдачи: 19 декабря. Для зачёта по заданию нужно набрать в сумме хотя бы 5 баллов. Любые вопросы задавайте по почте.

1. Представьте следующие задачи оптимизации в виде задач ЦЛП, проследив, чтобы количество переменных и неравенств в задаче ЦЛП не превосходило некоторого полинома от размера входных данных в исходной задаче.

   (a) (1 балл) Поиск остовного дерева минимального веса в графе с положительными весами на рёбрах.

   (b) (1 балл) Поиск паросочетания минимального веса в графе (не обязательно двудольном).

   (c) (1 балл) Поиск двухцветной правильной раскраски вершин графа (предполагается, что такая раскраска существует).

   (d) (2 балла) Поиск набора значений булевых переменных, на котором заданная КНФ обращается в единицу (пример КНФ: $(\overline{x} \vee \overline{y} \vee z)(x \vee \overline{z})$, пример набора, на котором она обращается в единицу: (1, 0, 1)).

2. (5 баллов) Реализуйте метод ветвей и границ для задачи коммивояжёра. Разбиение на подзадачи можно проводить любым удобным вам способом. В качестве нижней оценки стоимости решения в подзадачах можно использовать вес минимального остовного дерева. Протестируйте своё решение на случайно сгенерированном полном десятивершинном взвешенном графе.

Введение в дискретную оптимизацию. Осень 2010.

Задание по лекции 13 декабря. Срок сдачи: 21 декабря. Для зачёта по заданию нужно набрать в сумме хотя бы 5 баллов. Любые вопросы задавайте по почте.

1. Реализуйте на языке C++ в свободной форме алгоритм

   (a) (5 баллов) GREEDY

   (b) (5 баллов) IMBALANCE

   (c) (5 баллов) ROBIN HOOD

   К программе должны прилагаться один-два тестовых файла. Программа должна выдавать последовательность назначений (на какой процессор какая задача назначена) и оценку результирующего расписания (соответственно, длительность или максимальную нагрузку на процессор).