АКАДЕМИЯ НАУК РОССИИ
СИБИРСКОЕ ОТДЕЛЕНИЕ
ИНСТИТУТ МАТЕМАТИКИ им. С.Л.СОБОЛЕВА

На правах рукописи

СВИРИДЕНКО Максим Иванович

АЛГОРИТМЫ С ОЦЕНКАМИ ДЛЯ ДИСКРЕТНЫХ ЗАДАЧ РАЗМЕЩЕНИЯ

(01.01.09 – математическая кибернетика)

Диссертация на соискание
ученой степени кандидата
физико-математических наук

Научный руководитель
д.ф.-м.н., профессор
В.Л.Береснев

Новосибирск – 1998 г.

Оглавление

Введение

За последние десятилетия достигнуты значительные успехи в построении и развитии математического аппарата для обоснования решений в самых различных областях человеческой деятельности. Одним из таких интенсивно развивающихся направлений исследований является теория дискретных задач размещения, выделившаяся в самостоятельную научную дисциплину около тридцати лет назад.

Дискретные задачи размещения моделируют ситуации, в которых требуется в конечной области разместить конечное число объектов так, чтобы достичь наибольшего эффекта или получить наименьшие издержки. В реальной жизни в качестве размещаемых объектов могут выступать промышленные предприятия, различные средства обслуживания, складские системы, объекты оборонного назначения и многое другое.

В дискретных задачах размещения множество допустимых решений конечно, поэтому заранее известно, что оптимальное решение существует. Более того существует универсальный способ отыскания такого решения посредством полного перебора всех допустимых решений. Однако такой алгоритм поиска применим только в тех исключительных случаях, когда мощность множества допустимых решений сравнительно невелика. В интересных с практической точки зрения задачах, как правило, мощность множества допустимых решений быстро (например, экспоненциально) растет с увеличением размерности (объема исходных данных) задачи. Это приводит к тому, что в задачах реальной размерности количество допустимых решений становится величиной астрономического порядка, что делает перебор практически невозможным ни в настоящем времени, ни в обозримом будущем. Назначение теории дискретных задач размещения – создание работоспособных в практическом смысле алгоритмических инструментов решения задач. К сожалению, почти все естественные и интересные постановки задач размещения оказываются NP-трудными, поэтому попытки построения точных и эффективных (быстрых) алгоритмов для этих задач вероятней всего обречены на неудачу. В силу этого для дискретных задач размещения, как и для других NP-трудных задач дискретной оптимизации, можно выделить следующие основные направления исследований:

– выделение эффективно решаемых подзадач и построение для них точных эффективых алгоритмов с рекордными оценками трудоемкости;

– адаптация универсальных схем неявного перебора к конкретным задачам и построение работоспособных алгоритмов неявного перебора;

– построение эффективных приближенных алгоритмов с априорными оценками точности.

Число различных математических постановок, которые можно отнести к дискретным задачам размещения достаточно велико. Об этом свидетельствуют, в частности, огромное число публикаций на эту тему. Однако среди множества задач, встречающихся в литературе, можно выделить следующие четыре основных: простейшую задачу размещения, задачу о р-медиане, задачу о р-центре и квадратичную задачу о назначениях. Эти задачи составляют основу практически всех интересных математических моделей размещения и аккумулируют в себе большинство математических трудностей, с которыми приходится сталкиваться при исследовании различных моделей размещения. Эти задачи, которые можно назвать классическими, относятся к хорошо изученным задачам дискретной оптимизации, которые привлекали и привлекают к себе внимание большого числа исследователей. Простейшая задача размещения, вероятно, самая известная и изученная задача из указанных четырех, наряду с задачей коммивояжера, является полигоном для испытания новых методов и подходов в дискретной оптимизации.

Основное содержание настоящей диссертационной работы составляют результаты исследований по третьему из указанных выше направлений применительно к четырем перечисленным "классическим" задачам размещения и их обобщениям.

Приведем математические формулировки этих задач и дадим краткие обзоры основных известных результатов их исследования.

Пусть имеются конечные множества $I = \{1, \ldots, m\}$ и $J = \{1, \ldots, n\}$, первое из которых задает множество возможных пунктов размещения предприятий для выпуска некоторого однородного продукта, а второе – множество потребителей этого продукта. В некоторых задачах множества I и J могут совпадать. Пусть, кроме того, заданы величины c_i, $i \in I$, определяющие начальные затраты на размещение предприятия в пункте i и величины c_{ij}, $i \in I, j \in J$, равные затратам на доставку продукта от предприятия, размещенного в пункте i, до потребителя j. Требуется выбрать из множества возможных пунктов размещения предприятий подмножество пунктов S, в которых следует разместить предприятия, таким образом, чтобы сумма начальных затрат плюс сумма наименьших затрат на доставку продукта каждому потребителю была минимальной. Формально простейшая задача размещения (ПЗР) в комбинаторной постановке записывается следующим образом:

$$\min_{S \subseteq I} \left\{ \sum_{j \in J} \min_{i \in S} c_{ij} + \sum_{i \in S} c_i \right\}.$$

Наряду с комбинаторной постановкой часто рассматривается постановка ПЗР в виде задачи линейного целочисленного программирования:

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij} + \sum_{i \in I} c_i x_i,$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J,$$

$$x_{ij} \le x_i, \quad i \in I, j \in J,$$

$$x_{ij}, x_i \in \{0, 1\}, \quad i \in I, j \in J.$$

Используемые в этой задаче переменные имеют следующий смысл: $x_i = 1$, если в пункте i размещается предприятие, $x_i = 0$, в противном случае; $x_{ij}=1$, если пункт потребления j обслуживается предприятием, размещенном в пункте i, и $x_{ij}=0$, в противном случае.

Как уже отмечалось ПЗР является достаточно хорошо изученной задачей. Она относится к числу NP-трудных задач поскольку содержит в себе задачу о покрытии конечного множества системой конечных подмножеств. Исследования этой задачи велись по разным направлениям по каждому из которых были достигнуты значительные успехи. Так в работах Береснева [8], Береснева, Гимади, Деменьтева [11] и Erlenkotter [66] описаны алгоритмы решения ПЗР с помощью алгоритмов неявного перебора. Эти методы основаны на применении известной схемы метода ветвей и границ.

Целый ряд результатов связан с рассмотрением частных случаев задачи. Было обнаружено, что если матрица (c_{ij}) удовлетворяет различным специфическим требованиям, то ПЗР полиномиально разрешима. Известно, что ПЗР принадлежит классу P если матрица односвязна (Береснев, Гимади, Дементьев[11]), двусвязна (Береснев [9]), квазивыпукла или обобщенно-квазивыпукла (Гимади [20]). В работе Трубина [30] приводится полиномиальный алгоритм для ПЗР на дереве, с трудоемкостью $O(m^3)$ (см. также [31]). Гимади [16] предложил алгоритм с существенно лучшей трудоемкостью $O(m^2)$. Агеев [34] установил, что для очень широкого класса задач (с матрицами порожденными частичными q-деревьями) существует полиномиальный алгоритм с трудоемкостью $O(nm^{q+1})$. ПЗР и ее обобщения также изучались в работах [2, 10, 21, 15, 17, 19, 25, 18, 5, 6, 7, 14, 22, 26, 33] как с точки зрения полиномиально разрешимых подклассов так и с точки зрения других видов анализа NP-трудных задач (см. также обзор по полиномиально разрешимым подклассам для ПЗР [27]). Кроме того, Береснев [9] установил эквивалентность ПЗР и задачи минимизации булевых полиномов, что позволило выделить новые классы эффективно решаемых задач [9, 2, 4, 3]. В работе [35] Агеевым был получен отрицательный результат: ПЗР оказалась NP-трудной, даже если матрица (c_{ij}) порождена расстояниями на плоской решетке.

Близкой к ПЗР является так называемая простейшая задача размещения на максимум (ПЗРМ), которая в комбинаторной постановке записывается следующим образом:

$$\max_{S \subseteq I} \left\{ \sum_{j \in J} \max_{i \in S} f_{ij} - \sum_{i \in S} c_i \right\}.$$

В этой задаче величины f_{ij}, $i \in I, j \in J$ интерпретируются, как прибыль, получаемая от обслуживания потребителя j предприятием, размещенным в пункте i. При этом задача состоит в отыскании такого подмножества S пунктов размещения предприятий, чтобы сумма максимальной прибыли от обслуживаня потребителей минус сумма начальных затрат была максимальной.

Задачи ПЗР и ПЗРМ эквивалентны в том смысле, что любой точный алгоритм одной задачи применим и к другой. Однако с точки зрения приближенных алгоритмов эти задачи отличаются. Историческая справка и обзор по ПЗР и ПЗРМ могут быть найдены в [62].

Другой классической задачей размещения является задача о p-медиане, комбинаторная формулировка которой имеет следующий вид:

$$\min_{S\subseteq I}\left\{\sum_{j\in J}\min_{i\in S}c_{ij}:|S|\le p\right\}.$$

Задача о p-медиане моделирует ситуацию, при которой в возможных пунктах размещения нужно разместить не более p предприятий так, чтобы суммарные затраты на доставку продукта потребителям были минимальными.

Задача о p-медиане также как и ПЗР может быть записана в виде задачи линейного целочисленного программирования

$$\min\sum_{i\in I}\sum_{j\in J}c_{ij}x_{ij},$$

$$\sum_{i\in I}x_{ij}=1,\quad j\in J,$$

$$x_{ij}\le x_i,\quad i\in I, j\in J,$$

$$\sum_{i\in I}x_i\le p,$$

$$x_{ij},x_i\in\{0,1\},\quad i\in I, j\in J.$$

Kariv и Hakimi [96] показали, что задача о p-медиане является NP-трудной. В этой же работе показано, что если матрица (c_{ij}) есть матрица кратчайших расстояний на дереве, то задача полиномиально разрешима. В обзоре [110] можно найти список основных результатов для этой задачи.

Задача о p-медиане может рассматриваться и как задача на максимум. Комбинаторная формулировка такой задачи имеет вид

$$\max_{S\subseteq I}\left\{\sum_{j\in J}\max_{i\in S}c_{ij}:|S|\le p\right\}.$$

Эта задача эквивалентна задаче о р-медиане с точки зрения нахождения точного решения, но так же как и в случае ПЗР, постановки задач о р-медиане отличаются с точки зрения нахождения приближенных решений.

Третьей рассматриваемой в работе задачей является задача о р-центре. Комбинаторная формулировка этой задачи выглядит следующим образом:

$$\min_{S \subseteq I} \left\{ \max_{i \in I \setminus S} \min_{j \in S} c_{ij} : |S| = p \right\}.$$

В этой задаче нужно выбрать пункты размещения ровно p предприятий таким образом, чобы наибольшие затраты на доставку продукта потребителю были наименьшими. Такую задачу называют еще задачей о расширении узкого места.

Задача о р-центре NP-трудна [93]. Это следует из того, что известная задача о доминирующем множестве может быть сведена к задаче о р-центре. Эффективный алгоритм для задачи о р-центре существует (см. [56]) при тех же самых предположениях, что и для задачи о р-медиане, когда матрица (c_{ij}) есть матрица кратчайших расстояний на дереве. Исторические сведения, а также перечень результатов для задачи о р-центре приводится в обзорах [83], [129].

Одной из наиболее трудных для исследований задач размещения является квадратичная задача о назначении (КЗН). В виде задачи целочисленного программирования она записывается следующим образом:

$$\min \sum_{i\in I}\sum_{p\in J}\left(\sum_{j\in I} w_{ij} \sum_{q\in J} d_{pq} y_{jq}\right) y_{ip},$$
$$\sum_{p\in J} y_{ip} = 1, \quad i \in I,$$
$$\sum_{i\in I} y_{ip} \leq 1, \quad p \in J,$$
$$y_{ip} \in \{0,1\}, \quad i \in I, p \in J.$$

Содержательно КЗН можно проинтерпретировать следующим образом. Пусть $J = \{1,\ldots,n\}$ – множество возможных пунктов размещения предприятий, а $I = \{1,\ldots,m\}$ $(m \leq n)$ – множество размещаемых предприятий. Пусть величины d_{pq}, $p, q \in J$, задают расстояние между пунктами p и q, а величины w_{ij} $i, j \in I$, являются удельными (на единицу расстояния) затратами на перевозку грузов между предприятиями i и j. Определим переменные y_{ip}, $i \in I, p \in J$ следующим образом:

$$y_{ip} = \begin{cases} 1 & \text{если предприятие } i \text{ размещается в пункте } p, \\ 0 & \text{иначе.} \end{cases}$$

Тогда задача КЗН может рассматриваться как задача, в которой требуется разместить m предприятий в n пунктах таким образом, чтобы суммарные затраты на транспортировку грузов между предприятиями были наименьшими.

КЗН относится к числу NP-трудных задач, так как задача коммивояжера и задача о максимальной клике являются частыми случаями КЗН [54]. Несмотря на то, что КЗН является одной из самых изученных задач комбинаторной оптимизации (см. обзор [54] и обзор по полиномиально разрешимым подклассам [55]) точные решения удается получить для задач размерности не более 30×30.

Как уже отмечалось выше интересующие нас задачи являются NP-трудными. Это означает, что построение точных эффективных (полиномиальных) алгоритмов скорее всего невозможно (подробную дискуссию по теории NP-полноты см. в [28]). Одним из основных направлений исследования таких задач является построение приближенных эфективных алгоритмов с априорными оценками их точности. Такие алгоритмы не для всякой индивидуальной задачи находят точное решение, но всегда отыскивают допустимое решение задачи, которое удовлетворяет некоторым требованиям по точности, установленными еще до получения приближенного решения.

Идея построения приближенного алгоритма настолько естественна, что невозможно проследить, кто первым ее придумал и стал использовать. Считается, что первый приближенный алгоритм и его анализ был дан в очень известной работе [80] по теории расписаний. Другими фундаментальными работами в этой области являются работы [94, 23], в которых вводятся понятия приближенного алгоритма и его точности в худшем случае.

Существуют два наиболее распространенных подхода к анализу приближенных алгоритмов и определению их точности: анализ поведения в худшем случае и вероятностный анализ поведения алгоритма. При анализе в худшем случае оценивается погрешность алгоритма на наихудшей из возможных индивидуальной задаче, а в случае вероятностного анализа предполагается, что на множестве индивидуальных задач задано некоторое вероятностное распределение и оценивается математическое ожидание погрешности алгоритма на этом распределении.

Существуют также и другие интересные подходы к вероятностной оценке точности алгоритма. Среди таких подходов следует выделить подход Перепелицы и Гимади [29]. В этой работе анализируется алгоритм "иди в ближайший еще не пройденный город"для задачи коммивояжера на различных вероятностных распределениях входных данных.

Настоящая работа посвящена построению приближенных алгоритмов для задач размещения и анализу их точности в худшем случае. Напомним определения основных понятий, используемых при таком анализе. Пусть имеется оптимизационная задача и приближенный алгоритм A для ее решения. Обозначим через I индивидуальную оптимизационную задачу, $F(I)$ значение целевой функции на приближенном решении, полученном алгоритмом, а через $F^*(I)$ значение целевой функции на оптимальном решении. Существует несколько способов измерения точности приближенного алгоритма. Наиболее распространенной и используемой величиной, характеризующей точность приближенного алгоритма, является так называемая относительная точность, под которой понимается функция $\tau_A(I) = F(I)/F^*(I)$. Оценкой (априорной) относительной точности приближенного алгоритма A решения оптимизационной задачи на максимум (минимум) является величина R_A такая, что для любой индивидуальной задачи I выполняется неравенство

$$\tau_A(I) \geq R_A \quad (\tau_A(I) \leq R_A)\,.$$

Такую оценку R_A, не зависящую от размерности индивидуальной задачи I, называют константной.

Другими важными понятиями комбинаторной оптимизации, связанными с приближенными алгоритмами, являются понятия аппроксимационной схемы и полной аппроксимационной схемы. Аппрокимационной схемой для некоторой оптимизационной задачи называется алгоритм, который для любого фиксированного $\varepsilon > 0$ находит допустимое решение задачи с относительной оценкой погрешности $1 - \varepsilon$ (для задачи на максимум) и трудоемкость алгоритма полиномиальна при фиксированном $\varepsilon > 0$. Схема называется полной, если трудоемкость алгоритма ограничена полиномом от $1/\varepsilon$ и длины входных данных.

В конце семидесятых годов вышло много важных работ по приближенным алгоритмам и, как следствие этого, появилось несколько обзоров [72, 122, 123, 13] в которых суммируются результаты, полученные за эти годы. Затем наступило затишье и на протяжении более десяти лет количество опубликованных статей по приближенным алгоритмам было невелико. С конца восьмидесятых годов произошел взрыв популярности исследований в области приближенных алгоритмов. Очень важным открытием стало понимание того, что для получения оценок значений целевых функций исследуемых задач можно использовать различные полиномиально разрешимые релаксации этих задач. Первыми, кто использовал линейное програмирование при построении и анализе приближенных алгоритмов с относительной оценкой точности были Raghavan, Tompson [120], а после работы [77] использование линейного програмирования стало основным методом построения приближенных алгоритмов. Одним из основных методов используемых в диссертации является метод округления линейных релаксаций. Суть метода заключается в следующем. Предположим, что исследуемую оптимизационную задачу можно сформулировать как задачу линейного булевого программирования. Рассмотрим линейную релаксацию этой задачу, получаемую заменой булевых переменных на непрерывные переменные из отрезка $[0, 1]$. Эта задача является задачей линейного программирования, которая как известно полиномиально разрешима [32]. Понятно, что оптимальное значение целевой функции релаксированной задачи является верхней оценкой (для задачи на максимум) оптимального значения целевой функции исходной задачи. В результате округления оптимального решения релаксированной задачи можно получить приближенное решение исходной задачи, а воспользовавшись верхней оценкой отимального значения ее целевой функции получить и оценку точности построенного приближенного решения. Важнейшей составной частью этого подхода является алгоритм округления оптимального решения релаксированной задачи. Существует широкий спектр техник, которые могут применятся при округлении. Особенно известной стала техника вероятностного округления [120]. В диссертационной работе используются некоторые из известных техник округления, в частности, вероятностное округление. Также предлагается новый метод округления для задач с ограничением на мощность.

В начале 90-х произошел другой прорыв в понимании приближенных алгоритмов, обусловленный появлением теоретических основ для доказательства отрицательных результатов. Бурное развитие в этой области было инициировано двумя работами [70, 117]. В первой работе впервые был получен отрицательный результат для задачи о клике, и предложена техника, которая в дальнейшем усовершенствовалась и оттачивалась в целом ряде работ. Во второй работе предложено синтаксическое определение некоторого класса сложности, который назвали MAXSNP, и введено понятие L-сводимости. Оказалось, что почти все естественные задачи комбинаторной оптимизации лежат в этом классе и многие из них оказались полными в этом классе относительно L-сводимости. Papadimitrou и Yannakakis предположили, что для задач полных в классе MAXSNP не существует аппроксимационной схемы. Эта гипотеза была доказана в работе [46] при условии, что $P \neq NP$. Хорошими введениями в эту бурно развивающуюся область являются диссертация Arora [38] и обзор Arora и Lund [45], а также обзоры [126, 87] о последних достижениях в области приближенных алгоритмов.

Изложим кратко результаты диссертационной работы, состоящей из введения, шести глав, заключения, списка используемой литературы и списка публикаций автора по теме диссертации.

Как уже отмечалось, настоящая диссертация посвящена построению и анализу приближенных алгоритмов для дискретных задач размещения. В первой главе рассматривается простейшая задача размещения на максимум. Для этой задачи строится приближенный алгоритм с оценкой точности $2(\sqrt{2}-1) \approx 0.828$, что лучше оценки точности жадного алгоритм [61]. Основу предложенного алгоритма составляет процедура вероятностного округления оптимального решения линейной релаксации задачи. Показано, что в случае использования такой релаксации, полученная оценка не может быть улучшена. Кроме того, доказано, что для простейшей задачи размещения на максимум не существует аппроксимационной схемы при условии, что $P \neq NP$.

Вторая глава посвящена исследованию обобщений задачи о p-медиане на максимум. Рассматривается обобщенная задача о p-медиане с двумя видами ограничений: с обычным мощностным ограничением $|S| \leq p$ и с более общим ограничением "ранцевого"вида $\sum_{i \in S} d_i \leq D$. Целевая функция задачи определяется как оптимальное значение целевой функции некоторой оптимизационной задачи. Эта оптимизационная задача является более общей, чем, например, рассмотренная в [132], и имеет k дополнительных ресурсных ограничений. Для обеих задач о p-медиане (с двумя видами ограничений) получены оценки точности жадных алгоритмов. При $k = 0$ эти оценки совпадают с оценками из [61, 132]. В этой главе рассматривается также еще одно обобщение задачи о p-медиане на максимум так называемая динамическая задача о p-медиане. Динамическая задача описывает ситуацию, когда размещение предприятий производится в течение некоторого периода времени, состоящего из нескольких единичных отрезков (лет). Требуется в каждом году разместить ограниченное число предприятий так, чтобы максимизировать суммарную прибыль за весь рассматриваемый период времени. Для этой задачи предлагается приближенный алгоритм с оценкой точности $(1 - e^{-1})$. При построении алгоритма используется техника вероятностного округления решения линейной релаксации задачи.

В третьей главе изучается задача о максимальном покрытии p множествами, которая является частным случаем задачи о p-медиане на максимум, а также некоторые близкие к ней задачи. Предлагается новая техника округления решений линейных релаксаций оптимизационных задач с ограничением на мощность. С использованием этой техники, названной процедурой перекачки, для задачи о максимальном покрытии p множествами построен приближенный алгоритм с оценкой точности лучше, чем известная оценка точности жадного алгоритма. Другой рассматриваемой в главе задачей с мощностным ограничением является задача о максимальном разрезе с ограничением на мощность. Для этой задачи предлагается приближенный алгоритм с константной оценкой точности равной $1/2$. В данной главе рассматриваются также обобщения двух указанных задач, для которых с использованием техники перекачки удается построить приближенные алгоритмы с такими же оценками точности, что и для исходных задач.

В четвертой главе исследуется задача выполнимости на максимум с мощностным ограничением. Для этой задачи предлагается алгоритм с оценкой точности равной $1-e^{-1}$. Алгоритм построен с использованием техники вероятностного округления решения линейной релаксации. Полученная оценка является неулучшаемой при условии, что $P \neq NP$ [68].

В пятой главе рассматривается квадратичная задача о назначениях. Для этой задачи дается положительный ответ на поставленный в [118] вопрос о существовании приближенного алгоритма с константной оценкой точности для частного случая задачи, когда матрица (d_{pq}) удовлетворяет неравенству треугольника, а все элементы матрицы (w_{ij}) равны единице. Для указанного частного случая задачи построен алгоритм с оценкой точности равной 2.

В шестой главе предлагается приближенный алгоритм для задачи о p-центре с ранцевым ограничением. Оценка точности алгоритма равна 3 и совпадает с лучшей известной, однако оценка временной сложности предложенного алгоритма лучше известных.

В заключении приводятся основные результаты диссертационной работы.

Основные результаты диссертации опубликованы и докладывались на 2-ом Сибирском конгрессе по индустриальной и прикладной математике (Новосибирск, 1996), Международной конференции "Проблемы оптимизации и экономические приложения" (Омск, 1997), 16-ом Симпозиуме по математическому программированию (Лозанна, Швейцария, 1997), Сибирской коференции по исследованию операций (Новосибирск, 1998), 1-ой Международной конференции по приближенным алгоритмам (Ольбо, Дания, 1998), на научных семинарах Института математики СО РАН и семинаре Центра по Фундаментальным Исследованиям в Области Компьютерных Наук (Оргус, Дания, 1998).

Результаты первой и третьей главы были получены совместно с А. А. Агеевым.

Глава 1

Простейшая задача размещения на максимум

Введение

В настоящей главе рассматривается простейшая задача размещения на максимум

$$\max_{S\subseteq I}\left\{\sum_{j\in J}\max_{i\in S} b_{ij}-\sum_{i\in S} c_i\right\}. \tag{1.1}$$

Для близкой к ней простейшей задачи размещения на минимум (ПЗР) известен [85] приближенный алгоритм с оценкой точности равной $\ln n+1$. Эта оценка совпадает с оценкой точности жадного алгоритма для задачи о покрытии [58, 128].

Задача о покрытии в течение долгого времени была предметом интенсивного исследования с целью построения приближенных алгоритмов [52, 86, 88]. Для этой задачи Feige [68], используя идеи Lund и Yannakakis [109] и мощный результат Raz [121] из теории сложности, доказал, что для любого $\varepsilon > 0$ нельзя построить приближенный алгоритм с оценкой точности $(1-\varepsilon)\ln n$ при условии, что NP-трудные задачи не имеют субэкспоненциальных точных алгоритмов. Так как задача о покрытии является частным случаем ПЗР, то этот отрицательный результат переносится и на ПЗР и, следовательно, маловероятно, что для нее существует приближенный алгоритм с оценкой точности существенно лучшей, чем $\ln n$.

В такой ситуации естественно обратися к анализу частных случаев ПЗР. В [127] рассматривается ПЗР с неравенством треугольника, т.е. задача, в которой элементы матрицы (c_{ij}) удовлетворяют следующему условию: для любых i, j, k выполняется неравенство $c_{ij} + c_{jk} \geq c_{ik}$. Для такой задачи Shmoys, Tardos и Aardal [127], используя фильтрационную технику Lin и Vitter [106, 107], построили приближенный алгоритм с оценкой точности 3,16, а Guha и Khuller [82] улучшили алгоритм и получили оценку равную 2,3. В настоящее время лучшую оценку точности равную $1 + 2/e \approx 1,7$ имеет алгоритм, полученный Chudak и Shmoys [59]. С другой стороны, Guha и Khuller [82], используя отрицательные результаты Feige [68], доказали, что для ПЗР с неравенством треугольника нельзя построить приближенный алгоритм с оценкой точности лучше, чем $\beta \approx 1.43$, где β корень уравнения $1 + 2e^{-x} = x$. Таким образом, существует разрыв между положительными и отрицательными результатами, касающихся оценок точности приближенных алгоритмов для ПЗР с неравенством треугольника. Это позволяет надеятся на усиление полученной для данной задачи резльтатов.

Для более узкого класса задач ПЗР с матрицей евклидовых расстояний Arora, Raghavan и Rao [44] построили полиномиальную аппроксимационную схему, используя технику, развитую Arora [39, 40]. Однако до сих пор остается открытым вопрос о существовании аппроксимационной схемы для ПЗР с матрицей кратчайших расстояний на планарном графе, хотя многие MAXSNP-трудные задачи комбинаторной оптимизации становятся проще на планарных графах и имеют аппроксимационные схемы [44, 50, 81].

Так как целевая функция простейшей задачи размещения на максимум

$$Z(S) = \sum_{j\in J} \max_{i\in S} b_{ij} - \sum_{i\in S} c_i$$

может принимать как положительные, так и отрицательные значения, то возникают трудности с определением оценки относительной точности приближенного алгоритма. Этих трудностей можно избежать, если перейти к рассмотрению задачи с целевой функцией $(Z(S) - Z_R)$, где $Z_R = \sum_{j\in J} \min_{i\in I} b_{ij} - \sum_{i\in I} c_i$ – тривиальная нижняя оценка для значений функции $Z(S)$. Cornuejols, Fisher и Nemhauser [61] доказали, что для ПЗР на максимум со сдвинутой целевой функцией и дополнительным ограничением $|S| \leq K$ простой жадный алгоритм находит допустимое решение S, для которого

$$(Z(S) - Z_R)/(Z^* - Z_R) \geq 1 - e^{-1} \approx 0.632,$$

где Z^* – значение целевой функции на оптимальном решении задачи.

В настоящей главе предлагается полиномиальный приближенный алгоритм для ПЗР на максимум со сдвинутой целевой функцией с оценкой точности $2(\sqrt{2}-1) \approx 0.828$, что существенно лучше указанной выше оценки. Алгоритм основан на идее вероятностного округления, принадлежащей Goemans и Williamson [77]. Кроме того показано, что существует такое $\varepsilon > 0$, что задача отыскания приближенного решения S для которого $(Z(S) - Z_R)/(Z^* - Z_R) \geq 1 - \varepsilon$ является NP-трудной. Доказательство основывается на том факте, что для ПЗР на максимум со сдвинутой целевой функцией возможно ее двухстороннее сведение к специальному случаю задачи выполнимости на максимум (MAX SAT*). Особенности MAX SAT* позволяют сконструировать приближенный алгоритм с лучшей оценкой точности чем оценки точности алгоритмов, известные для общего случая задачи выполнимости на максимум.

1.1 Двухстороннее сведение ПЗР на максимум в сдвинутой форме к задаче MAX SAT*.

Пусть $x_1, x_2, \ldots, x_m$ – булевые переменные. *Литерал* – это либо переменная x, либо ее отрицание $\overline{x}$. Задача выполнимости на максимум (MAX SAT) формулируется следующим образом: дано множество дизъюнкций $\mathcal{C}$, где каждая дизъюнкция состоит из литералов переменных $\{x_1, x_2, \ldots, x_m\}$ и функция весов $w : \mathcal{C} \to \mathbf{R}_+$, требуется найти назначение значений ИСТИНА на переменные $x_1, x_2, \ldots, x_m$, максимизирующее общий вес выполненых дизъюнкций (дизъюнкция является выполненной, если она содержит хотя бы один истинный литерал). Для краткости мы будем обозначать значения ИСТИНА и ЛОЖЬ через 1 и 0 соответственно.

Рассмотрим индивидуальную задачу ПЗР на максимум. Для любого $j \in J$, определим перестановку $(i_1(j), i_2(j), \ldots, i_m(j))$ множества I такую, что $b_{i_1(j),j} \geq b_{i_2(j),j} \geq \ldots \geq b_{i_m(j),j}$. Для любых $k = 1, \ldots, m$ и $j \in J$, определим множества $C_{kj} = \{i_1(j), i_2(j), \ldots, i_k(j)\}$ и веса $w_{kj} = b_{i_k(j),j} - b_{i_{k+1}(j),j}$, где $b_{i_{m+1}(j),j}$ полагается равным $b_{i_m(j),j}$ для всех j. Рассмотрим индивидуальную задачу MAX SAT, где множество дизъюнкций определяется следующим образом: $\{\bigvee_{i \in C_{kj}} x_i : k = 1, \ldots, m, j \in J\} \cup \{\overline{x_i} : i \in I\}$ и функция весов w определяется равенствами $w(\bigvee_{i \in C_{kj}} x_i) = w_{kj}$ и $w(\overline{x_i}) = c_i$ для всех $k = 1, \ldots, m$, $i \in I$ и $j \in J$.

Возьмем произвольное множество $S \subseteq I$ и пусть $x = (x_1, x_2, \ldots, x_m)$ – вектор инцинденций множества S. Тогда

$$\begin{aligned} \sum_{j \in J} \max_{i \in S} b_{ij} - \sum_{i \in S} c_i &= \\ \sum_{j \in J} \Big(\sum_{k=1}^{m} w_{kj} \big(\bigvee_{i \in C_{kj}} x_i \big) + \min_{i \in I} b_{ij} \Big) + \sum_{i \in I} c_i \overline{x_i} - \sum_{i \in I} c_i &= \\ \sum_{k=1}^{m} \sum_{j \in J} w_{kj} \big(\bigvee_{i \in C_{kj}} x_i \big) + \sum_{i \in I} c_i \overline{x_i} + \sum_{j \in J} \min_{i \in I} b_{ij} - \sum_{i \in I} c_i. \end{aligned}$$

Откуда следует, что

$$Z(S) - Z_R = \sum_{k=1}^{m} \sum_{j \in J} w_{kj} \big(\bigvee_{i \in C_{kj}} x_i \big) + \sum_{i \in I} c_i \overline{x_i}. \tag{1.2}$$

Следовательно, значение целевой функции на множестве $S \subseteq I$ индивидуальной задачи ПЗР на максимум в сдвинутой форме равно весу выполненных дизъюнкций в соответствующем назначении на переменные истинностных значений в определенной выше индивидуальной задачи MAX SAT.

Равенство (1.2) показывает, что ПЗР на максимум сводится к специальному случаю задачи MAX SAT, которая может быть записана в следующей общей форме:

$$\max_{x\in\{0,1\}^m} \sum_{l=1}^{L} f_l(\bigvee_{i\in C_l} x_i) + \sum_{i=1}^{m} g_i\overline{x_i}, \qquad (1.3)$$

где $C_l \subseteq \{1,\ldots,m\}$ и $f_l, g_i \geq 0, l = 1,\ldots,L, i = 1,\ldots,m$. В дальнейшем мы будем называть задачу (1.3) – MAX SAT*.

Обратное сведение MAX SAT* к ПЗР на максимум выглядит следующим образом. Пусть (1.3) является индивидуальной задачей MAX SAT*. Рассмотрим индивидуальную задачу (1.1) ПЗР на максимум с $I = \{1,\ldots,m\}$, $J = \{1,\ldots,L\}$, $c_i = g_i$, $i = 1,\ldots,m$, и

$$b_{ij} := \begin{cases} f_l & \text{если } i \in C_l, \\ 0 & \text{в противном случае .} \end{cases}$$

Заметим, что если мы применим к этой индивидуальной задаче сведение, описанное выше, мы получим первоначальную задачу MAX SAT*. Значение целевой функции на множестве $S \subseteq I$ в задаче ПЗР равно весу выполненных дизъюнкций на векторе инцинденций множества S в первоначальной задаче MAX SAT*. Следовательно, *описанное сведение между MAX SAT* и ПЗР на максимум сохраняет относительную оценку точности на соответствующих друг другу допустимых решениях.*

1.2 Приближенный алгоритм для задачи MAX SAT*

В настоящем разделе мы построим приближенный алгоритм для решения MAX SAT*.

Рассмотрим формулировку задачи (1.3) как задачи целочисленного линейного програмирования:

$$\max \sum_{l=1}^{L} f_l z_l + \sum_{i=1}^{m} g_i(1-y_i),$$

$$\sum_{i \in C_l} y_i \geq z_l, \quad l = 1, \ldots, L,$$

$$y_i \in \{0,1\}, \quad i = 1, \ldots, m,$$

$$0 \leq z_l \leq 1, \quad l = 1, \ldots, L.$$

Обозначим эту задачу целочисленного линейного программирования через (IP). Аналогичная формулировка рассматривалась в [77] при построении приближенных алгоритмов для задачи MAX SAT.

Полагая $y_i = x_i$ и $z_l = \bigvee_{i \in C_l} x_i$, $i = 1, \ldots, m$, $l = 1, \ldots, L$, мы устанавливаем взаимнооднозначное соответствие между оптимальными решениями двух задач. Заметим, что оптимальное значение F^*_{IP} задачи (IP) совпадает с оптимальным значением задачи MAX SAT*. Обозначим через (LP) линейную релаксацию задачи (IP), полученную заменой ограничений $y_i \in \{0,1\}$ на ограничения $0 \leq y_i \leq 1$. Очевидно, $F^*_{IP} \leq F^*_{LP}$.

Наш алгоритм является прямой модификацией вероятностного алгоритма с оценкой $(1-1/e)$ для задачи MAX SAT, описанного в работе [77](глава 3). Алгоритм состоит из двух шагов. На первом шаге алгоритм находит оптимальное решение (y_i^*, z_l^*) линейной программы (LP). На втором шаге независимо полагаем для каждого i, $y_i := 1$ с вероятностью $p_i = 1 - \lambda^* + \lambda^* y_i^*$ и $y_i := 0$ с вероятностью $1 - p_i$, где $\lambda^* = 2(\sqrt{2} - 1)$. Детерменированный алгоритм получается прямым использованием метода условных вероятностей (см. [37]). Отметим, что описанный алгоритм отличается от алгоритма Goemans и Williamson только на втором шаге, на котором их алгоритм выбирает индекс i с вероятностью $p_i = y_i^*$.

Чтобы упростить понимание используемого метода, рассмотрим общий случай с $p_i = 1 - \lambda + \lambda y_i^*$, $\lambda \in [0, 1]$, который включает оба алгоритма. Обозначим через $W(\lambda)$ математическое ожидание значения целевой функции на решении задачи (IP), полученном вероятностным алгоритмом для некоторого фиксированного λ.

Для доказательства оценки алгоритма мы будем сравнивать $W(\lambda)$ с F_{LP}^*. В доказательстве ниже используется лемма 3.1 из работы [77]. Очевидно, что

$$\begin{aligned} W(\lambda) &= \sum_{l=1}^{L} f_l(1 - \prod_{i \in C_l}(1 - p_i)) + \sum_{i=1}^{m} g_i(1 - p_i) \\ &= \sum_{l=1}^{L} f_l(1 - \lambda^{|C_l|} \prod_{i \in C_l}(1 - y_i^*)) + \lambda \sum_{i=1}^{m} g_i(1 - y_i^*). \end{aligned} \tag{1.4}$$

Мы доказываем, что

$$1 - \lambda^k \prod_{i \in C_l}(1 - y_i^*) \geq \Delta_k(\lambda) z_l^*, \tag{1.5}$$

где $k = |C_l|$ и $\Delta_k(\lambda) = 1 - \lambda^k(1 - 1/k)^k$. Действительно, используя неравенство о среднем геометрическом и среднем арифметическом и ограничение на z_l, мы получаем, что

$$1 - \lambda^k \prod_{i \in C_l} (1 - y_i^*) \geq 1 - \lambda^k (1 - \frac{\sum_{i \in C_l} y_i^*}{k})^k \geq 1 - \lambda^k (1 - \frac{z_l^*}{k})^k .$$

Неравенство (1.5) следует из того факта, что $f(z_l) = 1 - \lambda^k (1 - z_l/k)^k$ – вогнутая функция и $f(0) = 1 - \lambda^k$, $f(1) = \Delta_k(\lambda)$.

Используя (1.4) и (1.5), получаем

$$W(\lambda) \geq \min\{ \min_{k \geq 1} \Delta_k(\lambda), \lambda\} F_{LP}^* \geq \min\{ \min_{k \geq 1} \Delta_k(\lambda), \lambda\} F_{IP}^* .$$

Используя метод условных вероятностей, имеем, что

$$\varphi(\lambda) = \min\{ \min_{k \geq 1} \Delta_k(\lambda), \lambda\}$$

есть оценка точности дерандомизированного алгоритма. Докажем, что λ^* является наилучшим параметром λ в алгоритме, то есть $\varphi(\lambda) \leq \varphi(\lambda^*)$ для всех $\lambda \in [0,1]$. Заметим, что

$$\min\{\Delta_2(\lambda), \lambda\} \leq \min\{\Delta_2(\lambda^*), \lambda^*\} = \lambda^*,$$

где $\lambda^* = 2(\sqrt{2} - 1)$ корень уравнения $\Delta_2(\lambda) = \lambda$. Заметим далее, что $\Delta_k(\lambda^*) \geq \Delta_2(\lambda^*)$ для всех $k \geq 1$. Действительно, для $k = 1, 3, 4$ это неравенство может быть проверено прямым вычислением. Для оставшихся $k \geq 5$ оно следует из неравенств

$$\Delta_k(\lambda^*) \geq 1 - \lambda^{*k} e \geq 1 - \lambda^{*5} e > 0.85 > \Delta_2(\lambda^*).$$

Следовательно,

$$\varphi(\lambda) \leq \min\{\Delta_2(\lambda), \lambda\} \leq \min\{\Delta_2(\lambda^*), \lambda^*\} = \lambda^* = \varphi(\lambda^*).$$

Таким образом, мы доказали следующее утверждение.

Теорема 1 *Для задачи $MAXSAT^*$ существует вероятностный приближенный алгоритм с оценкой точности* $2(\sqrt{2}-1)$.

Мы можем дерандомизировать только что описанный алгоритм следующим образом. Дерандомизация состоит из n шагов, на шаге $i, i = 1, \ldots, n$ переменные $x_1, \ldots, x_{i-1}$ уже зафиксированы. Переменная x_i определяется по следующему правилу: вычисляются условные математические ожидания значения целевой функции на решении полченном вероятностным алгоритмом при условии, что некоторые переменные зафиксированы

$$E_0 = E(x_1, \ldots, x_{i-1}, x_i = 0) \text{ и } E_1 = E(x_1, \ldots, x_{i-1}, x_i = 1).$$

Если $E_1 \geq E_0$, то мы полагаем $x_i = 1$, в противном случае полагаем $x_i = 0$. Так как

$$\max\{E_1, E_0\} \geq E(x_1, \ldots, x_{i-1}),$$

то мы имеем к концу n-го шага решение x со значением целевой функции, которое не меньше математического ожидания $W(\lambda)$.

1.3 Сложность аппроксимации

В этом разделе мы покажем, что для любого $\varepsilon > 0$ существование $(1-\varepsilon)$-приближенного алгоритма для решения MAX SAT* влечет NP = P. Из этого факта и из сведения, описанного в разделе 1.2 следует, что MAX SAT* также, как и ПЗР в сдвинутой форме, не имеет полиномиальной аппроксимационной схемы.

Пусть SAT обозначает множество выполненных формул в задаче распознавания ВЫПОЛНИМОСТЬ (которую мы также будем обозначать SAT). Пусть MAX 3SAT(5) обозначает специальный случай MAX SAT, в котором каждая дизъюнкция имеет длину 3 и единичный вес, и каждая переменная входит точно в 5 дизъюнкций.

Для любой индивидуальной задачи I задачи MAX SAT обозначим через $w^*(I)$ суммарный вес дизъюнкций, выполненных оптимальным назначением в задаче I. Определим $r(I)$ как $w^*(I)$ деленное на общий вес дизъюнкций в I. Наше доказательство основывается на следующем результате.

Теорема 2 (Arora,Lund [45], теорема 10.2) *Существует полиномиальное сведение τ от задачи SAT к задаче MAX 3SAT(5) такое, что для некоторого фиксированного $\varepsilon > 0$ и всех индивидуальных задач I задачи SAT выполняется:*

$$I \in SAT \Rightarrow r(\tau(I)) = 1,$$

$$I \notin SAT \Rightarrow r(\tau(I)) < 1 - \varepsilon.$$

Заметим, что так как $\frac{5m}{3}$ является суммарным весом дизъюнкций в MAX 3SAT(5), то $w^*(\tau(I)) = \frac{5m}{3} r(\tau(I)$.

Мы опишем сейчас полиномиальное сведение σ от задачи MAX 3-SAT(5) к задаче MAX SAT*. Пусть $\mathcal{C} = \{C_l : l = 1, \ldots, L\}$ $(L = 5m/3)$ – множество дизъюнкций примера I задачи MAX 3-SAT(5). Пример $\sigma(I)$ задачи MAX SAT* определяется следующим образом. Множеством переменных задачи $\sigma(I)$ является $\{y_1, \ldots, y_m\} \cup \{z_1, \ldots, z_m\}$.

Множество дизъюнкций является объединением трех множеств $\mathcal{D}$, $\mathcal{E}$, и $\mathcal{C}'$, где $\mathcal{D} = \{y_i \vee z_i : i = 1, \ldots, m\}$, $\mathcal{E} = \{\overline{y_i} : i = 1, \ldots, m\} \cup \{\overline{z_i} : i = 1, \ldots, m\}$, и $\mathcal{C}'$ состоит из дизъюнкций $\{C_l : l = 1, \ldots, L\}$, в которых каждая переменная x_i заменяется на y_i и ее отрицание $\overline{x_i}$ – на z_i. Дизъюнкции в $\mathcal{D}$ имеют вес 11, дизъюнкции в $\mathcal{E}$ – 5 и в $\mathcal{C}'$ – 1.

Утверждение 1 *Если (y^*, z^*) является оптимальным решением в $\sigma(I)$, то $y_i^* + z_i^* = 1$ для всех $i = 1, \ldots, m$.*

Доказательство. Предположим, что $y_i^* + z_i^* = 0$ для некоторого i. Заменяя $y_i^* = 0$ на 1 в назначении (y^*, z^*), мы увеличиваем его вес по меньшей мере на 1, что противоречит оптимальности (y^*, z^*). Предположим, что $y_i^* + z_i^* = 2$ для некоторого i. Вспомним, что переменные y_i^*, z_i^* входят (в сумме) точно в 5 дизъюнкций в C_l. Тогда по меньшей мере одна из них, скажем y_i^*, входит меньше, чем в 5 дизъюнкций в C_l. Заменяя $y_i^* = 1$ на 0 в решении (y^*, z^*) мы увеличиваем вес, по меньшей мере, на 1, что опять противоречит оптимальности (y^*, z^*).

Следующее утверждение является непосредственным следствием Утверждения 1 и описания сведения σ.

Утверждение 2 *Вектор x^* является оптимальным в примере I задачи MAX 3SAT(5) тогда и только тогда, когда (y^*, z^*), где $y_i^* = x_i^*$, $z_i^* = 1 - x_i^*$, $i = 1, \ldots, m$, является оптимальным решением примера $\sigma(I)$. Кроме того, $w^*(I) = w^*(\sigma(I)) - 16m$.*

Рассмотрим полиномиальное сведение τ' от SAT к MAX SAT*, которое является суперпозицией двух сведений τ и σ. Используя Теорему

2 и Утверждение 2, мы получаем следующее:

Если $I \in \text{SAT}$, тогда $w^*(\tau'(I)) = \frac{5m}{3} + 16m$, и поэтому общий вес дизъюнкций в $\tau'(I)$ равен $\frac{5m}{3} + 16m$, $r(\tau'(I)) = 1$. Пусть $I \notin \text{SAT}$, тогда $w^*(\tau'(I)) = w^*(\tau(I)) + 16m = \frac{5m}{3} r(\tau(I)) + 16m < \frac{5m}{3}(1-\varepsilon) + 16m$. Следовательно,

$$r(\tau'(I)) < \frac{\frac{5m}{3}(1-\varepsilon) + 16m}{\frac{5m}{3} + 16m} = \frac{53 - 5\varepsilon}{53} = 1 - \frac{5}{53}\varepsilon.$$

Поэтому, для любого $\varepsilon' \leq \frac{5}{53}\varepsilon$, существование $(1-\varepsilon')$ приближенного алгоритма для решения MAX SAT* влекло бы P=NP. Результат этого раздела (сложность аппроксимации) можно вывести из более общего результата [101], где получена характеризация сложности различных максимизационных задач.

1.4 Разрыв двойственности

В настоящем разделе мы представим последовательность индивидуальных задач MAX SAT* с F^*_{IP}/F^*_{LP}, стремящихся к $2(\sqrt{2}-1)$, когда длина входа задачи стремится к ∞. Это означает, что оценка точности $2(\sqrt{2}-1)$ неулучшаема, более того, округление, описанное в разделе 1.3, наилучшее среди всех возможных округлений оптимального решения линейной релаксации (LP).

Рассмотрим следующую индивидуальную задачу (IP): $\mathcal{C} = \mathcal{C}' \cup \mathcal{C}''$, где $\mathcal{C}' = \{\bar{x}_i : 1 \leq i \leq m\}$, $\mathcal{C}'' = \{x_i \vee x_j : 1 \leq i < j \leq m\}$, и $w(\bar{x}_i) = 1$, $w(x_i \vee x_j) = \beta/(m-1)$, где $\beta \geq 1$ и $1 \leq i \leq j \leq m$. Заметим, что $|\mathcal{C}'| = m$, $|\mathcal{C}''| = m(m-1)/2$.

Наша задача оценить величину F^*_{IP}/F^*_{LP}.

Так как $(1/2,\ldots,1/2)$ допустимое решение (LP), мы получаем

$$F^*_{LP} \geq \frac{\beta}{m-1} \cdot \frac{m(m-1)}{2} + \frac{m}{2} = m\frac{\beta+1}{2}. \tag{1.6}$$

Пусть $L_m = \{l \in [0,1] : lm$ целое число $\}$. Для любого $l \in L_m$, пусть F^l_{IP} обозначает общий вес выполненных дизъюнкций в $\mathcal{C}''$, где точно lm переменных имеет значение "ЛОЖЬ". Тогда, используя симметрию задачи, мы получаем:

$$\begin{gathered}
F^*_{IP} = \max_{l \in L_m} F^l_{IP} \\
\leq \max_{l\in[0,1]} \Big\{ \Big(\frac{m(m-1)}{2} - \frac{lm(lm-1)}{2} \Big) \frac{\beta}{m-1} + lm \Big\} \\
= \max_{l\in[0,1]} \{ \frac{\beta m}{2} + lm - \frac{\beta lm(lm-1)}{2(m-1)} \} \\
(\text{ так как } \frac{m(lm-1)}{m-1} \geq lm - \frac{m}{m-1}) \\
\\
\leq \max_{l\in[0,1]} \{ m(\frac{\beta}{2} + l - \frac{\beta l^2}{2} + \frac{\beta l}{2(m-1)}) \} \\
\leq \max_{l\in[0,1]} \{ m(\frac{\beta}{2} + l - \frac{\beta l^2}{2} + \frac{\beta}{2(m-1)}) \} \\
(\text{ так как } \beta \geq 1, \text{ максимум достигается при } l = 1/\beta) \\
= m(\frac{\beta^2+1}{2\beta} + \frac{\beta}{2(m-1)}).
\end{gathered}$$

Полагая $\beta = 1 + \sqrt{2}$ и используя доказанные неравенства, получаем

$$\frac{F^*_{IP}}{F^*_{LP}} \leq \frac{\beta^2+1}{\beta(\beta+1)} + \frac{\beta}{(m-1)(\beta+1)}$$

$$
\begin{aligned}
&= \frac{2(2+\sqrt{2})}{(1+\sqrt{2})(2+\sqrt{2})} + \frac{\beta}{(m-1)(\beta+1)} \\
&= 2(\sqrt{2}-1) + \frac{\beta}{(m-1)(\beta+1)} \quad \to \quad 2(\sqrt{2}-1),
\end{aligned}
$$

при m стремящемся к ∞.

Глава 2

Задача о p-медиане на максимум и ее обобщения

Введение

Для задачи о p-медиане на максимум

$$\max_{S\subseteq I}\{\sum_{j\in J}\max_{i\in S} f_{ij} : |S| \le p\}. \tag{2.1}$$

известно [61, 74], что обычный жадный алгоритм имеет оценку точности равную $1-e^{-1}$. Аналогичный результат имеет место для рассмотренного в [132] естественного обобщения задачи о p-медиане вида

$$\max_{S\subseteq I}\{\sum_{j\in J}\max_{i\in S} f_{ij} : \sum_{i\in S} d_i \le D\}.$$

Простая модификация жадного алгоритма позволяет получить алгоритм с константной оценкой точности. В настоящей главе исследуются следующие обобщения указанных выше задач

$$\max_{S\subseteq I}\{f(S) : |S| \le p\}, \tag{2.2}$$

$$\max_{S\subseteq I}\{f(S):\sum_{i\in S} d_i \le D\}, \tag{2.3}$$

где $f(S)$ — функция, заданная на подмножествах множества I и принимающая действительные значения.

В работах [60, 61, 62, 63, 74, 114, 115, 131, 132] развиты методы, позволяющие получать оценки точности жадных алгоритмов решения этих задач в случаях, когда функция $f(S)$ удовлетворяет следующим условиям:

(а) $f(\emptyset) = 0$;

(б) $f(S)$ — неубывающая функция, т. е. если $S \subseteq T$, то $f(S) \le f(T)$;

(в) для любых $S, T \subseteq I$ выполняется неравенство

$$f(T) \le f(S) + \sum_{p\in T\setminus S} \left(f(S\cup\{p\}) - f(S)\right).$$

Последнее свойство может быть заменено известным свойством субмодулярности. Функция $f(S)$ называется субмодулярной, если для любых $S_1, S_2 \subseteq I$ выполняется неравенство

$$f(S_1) + f(S_2) \ge f(S_1 \cup S_2) + f(S_1 \cap S_2).$$

В [113] стр.662 показано, что неубывающая функция $f(S)$ является субмодулярной тогда и только тогда, когда выполняется условие (в). Если функция $f(S)$ удовлетворяет условиям (а)-(в), то жадный алгоритм [115] для задачи (2.2) имеет оценку точности равную $1 - e^{-1} \approx 0,63$, а аналог жадного алгоритма [132] решает задачу (2.3) с оценкой точности $1 - e^{-\beta} \approx 0,35$ где β — корень уравнения $e^x = 2 - x$.

В [132] рассматривается частный случай задач (2.2) и (2.3) с функцией $f(S)$, определяемой как оптимальное значение целевой функции следующей задачи:

$$\max \sum_{i\in S}\sum_{j\in J} f_{ij}x_{ij}; \tag{2.4}$$

$$\sum_{i\in S} x_{ij} \le A_j, \quad j\in J; \tag{2.5}$$

$$\sum_{j\in J} x_{ij} \le B_i, \quad i\in S; \tag{2.6}$$

$$x_{ij} \ge 0 \quad i\in S, j\in J. \tag{2.7}$$

Показано, что так определенная функция $f(S)$ является субмодулярной и, следовательно, удовлетворяет условию (в). Таким образом, жадные алгоритмы решения задач (2.2), (2.3) с такой функцией $f(S)$ имеют указанные выше оценки точности.

В настоящей главе изучаются задачи (2.2) и (2.3) с более общей функцией $f(S)$, определяемой как оптимальное значение целевой функции задачи вида

$$\max \sum_{i\in S}\sum_{j\in J} f_{ij}x_{ij}; \tag{2.8}$$

$$\sum_{i\in S} x_{ij} \le A_j, \quad j\in J; \tag{2.9}$$

$$\varphi_i(x_{i1},\dots,x_{in}) \le 0, \quad i\in S; \tag{2.10}$$

$$\sum_{i\in S}\sum_{j\in J} c_{ijl}x_{ij} \le C_l, \quad l=1,\dots,k; \tag{2.11}$$

$$x_{ij} \ge 0 \quad i\in S, j\in J, \tag{2.12}$$

где $\varphi_i(x_{i1},\dots,x_{in})$, $i\in I$, – непрерывные функции, удовлетворяющие условиям:

1. $\varphi_i(0,\dots,0)\le 0$,
2. если $x_j \le x'_j$, $j\in J$, то $\varphi_i(x_1,\dots,x_n) \le \varphi_i(x'_1,\dots,x'_n)$.

В приведенной задаче ограничения (2.10) являются более общими по сравнению с ограничениями (2.6). Кроме того, в задаче присутствуют k дополнительных ресурсных ограничений.

Будем предполагать, что все числовые параметры исследуемых задач (2.1) и (2.2) с функцией $f(S)$ задаваемой задачей (2.8)-(2.12) являются неотрицательными. Показано, что для любых $S, T \subseteq I$ функция $f(S)$ удовлетворяет неравенству

$$f(T) \leq (k+1)f(S) + \sum_{p \in T \setminus S} (f(S \cup \{p\}) - f(S)).$$

Следовательно, при $k = 0$ функция $f(S)$ является субмодулярной. В общем случае при $k \geq 1$ эта функция не субмодулярная и, следовательно, для жадных алгоритмов решения задач (2.2) и (2.3) с данной функцией $f(S)$ приведенные выше оценки точности не имеют места.

В настоящей главе для жадных алгоритмов решения задач (2.2) и (2.3) с функцией $f(S)$, определяемой задачей (2.8)-(2.12) доказывается, что в случае задачи (2.2) имеет место оценка точности

$$r_k = \frac{1 - e^{-(1+k)}}{k+1},$$

а задачи (2.3)

$$R_k = \frac{1 - e^{-\beta_k \cdot (1+k)}}{k+1},$$

где β_k — корень уравнения $e^{x \cdot (1+k)} = (k+1)(1-x) + 1$. Понятно, что при $k = 0$ приведенные оценки совпадают с оценками из [61, 74, 132]. Показано также, что числа r_k и R_k при $k \geq 0$ удовлетворяют следующим неравенствам:

$$k + 1 < r_k^{-1} \leq k + r_0^{-1},$$

$$k + 2 < R_k^{-1} \leq k + R_0^{-1},$$

где $r_0^{-1} \approx 1,58$ и $R_0^{-1} \approx 2,8$.

В данной главе изучается также динамическая задача о р-медиане на максимум. Эта задача является обобщением задачи о р-медиане и описывает ситуацию, когда размещение предприятий производится не одновременно, а в течение некоторого периода времени, состоящего из нескольких единичных отрезков (лет). Требуется в каждом году разместить ограниченное число предприятий так, чтобы максимизировать суммарную прибыль за весь рассматриваемый период времени.

Динамические задачи в последнее время становятся объектом пристального внимания. Тем не менее для динамических задач размещения не известны результаты по оценкам точности, аналогичные результатам полученных для классических задач размещения. Полученную в данной главе оценку точности алгоритма решения динамической задачи о р-медиане, вероятно, можно отнести к первым результатам в этой области.

2.1 Свойства функции $f(S)$

Положим $f(\emptyset) = 0$ и убедимся в том, что функция $f(S)$ удовлетворяет свойству (б). Пусть $S \subseteq T$. Тогда из любого допустимого решения (x_{ij}), где $i \in S$ и $j \in J$, задачи (2.8)–(2.12) с множеством S, полагая $x_{ij} = 0$ при $i \in T \setminus S$ и $j \in J$, получаем допустимое решение задачи (2.8)–(2.12) с множеством T и тем же значением целевой функции. Так как задача (2.8)–(2.12) — задача на максимум, то $f(S)$ — удовлетворяет свойству (б). Убедимся в корректности определения $f(S)$.

Утверждение 3 . *Определение функции $f(S)$ корректно, т. е. для всех $S \subseteq I$ существует действительное значение функциии $f(S)$.*

Доказательство. Так как линейная функция при максимизации на замкнутом множестве всегда имеет конечный или бесконечный максимум, то для доказательства утверждения достаточно показать, что множество допустимых решений задачи (2.8)–(2.12) непусто и замкнуто.

Известно, что если $\varphi(X)$ — непрерывная функция, то $\{X \in R^n : \varphi(X) \leq 0\}$ является замкнутым множеством. Используя этот факт и то, что пересечение замкнутых множеств замкнуто, получаем, что множество $X \in R^{|S|m}$, удовлетворяющее ограничениям (2.10)–(2.12), является замкнутым. Утверждение 1 доказано.

Утверждение 4 . *Если $f(S) = +\infty$, то существует элемент $i \in S$ такой, что $f(\{i\}) = +\infty$.*

Доказательство. Если $f(S) = +\infty$, то найдется последовательность $X_1 = (x_{ij}^1), \ldots, X_k = (x_{ij}^k), \ldots$ допустимых решений задачи (2.8)–(2.12)

таких, что
$$\lim_{k\to+\infty}\sum_{i\in S}\sum_{j\in J} f_{ij}x_{ij}^k = +\infty.$$

Существует такая пара (i, j), что $f_{ij} \neq 0$ и $\lim_{k\to+\infty} x_{ij}^k = +\infty$, так как в противном случае значение целевой функции на матрице X_k при $k \to +\infty$ было бы ограничено. Определим вектор x^k следующим образом:
$$x_l^k = \begin{cases} 0 \text{ при } l \neq j, \\ x_{ij}^k \text{ при } \quad l = j. \end{cases}$$
Вектор x^k удовлетворяет ограничениям (2.10)–(2.12) при $S = \{i\}$ и $\lim_{k\to+\infty}\sum_{j\in J} f_{ij}x_j^k = +\infty$. Утверждение 2 доказано.

Далее будем предполагать, что $f(S^*) \neq +\infty$, так как в противном случае точное решение можно вычислить за $O(\sum_{i\in I} T_i)$ действий, где T_i — количество действий, необходимых для вычисления функции $f(\{i\})$.

Лемма 1 . *Для любых $S_1, S_2 \subseteq I$ справедливо неравенство $f(S_1) + f(S_2) \geq f(S_1 \cup S_2)$.*

Доказательство. Обозначим через x_{ij}^*, где $i \in S_1 \cup S_2$ и $j \in J$, оптимальное решение задачи (2.8)–(2.12) с множеством $S = S_1 \cup S_2$. Заметим, что для x_{ij}^*, где $i \in S_1$ и $j \in J$, справедливы неравенства (2.10) – (2.12), т. е. x_{ij}^*, где $i \in S_1$ и $j \in J$, — допустимое решение задачи (2.8)–(2.12) с множеством $S = S_1$. Поэтому выполняется неравенство
$$f(S_1) \geq \sum_{i\in S_1}\sum_{j\in J} f_{ij}x_{ij}^*.$$

Аналогично убеждаемся в том, что

$$f(S_2) \geq \sum_{i \in S_2} \sum_{j \in J} f_{ij} x_{ij}^*.$$

Следовательно,

$$f(S_1) + f(S_2) \geq \sum_{i \in S_1} \sum_{j \in J} f_{ij} x_{ij}^* + \sum_{i \in S_2} \sum_{j \in J} f_{ij} x_{ij}^* \geq f(S_1 \cup S_2).$$

Лемма 1 доказана.

Лемма 2 . *Пусть $g(B; x_1, \ldots, x_n)$ — значение целевой функции непрерывной задачи о ранце:*

$$\sum_{i=1}^{n} g_i x_i,$$

$$\sum_{i=1}^{n} c_i x_i \leq B,$$

$$0 \leq x_i \leq 1.$$

Тогда для любого $\Delta \in [0, B]$ найдется вектор $\bar{x} \in R^n$ такой, что

$$g(B - \Delta; \bar{x}_1, \ldots, \bar{x}_n)/g(B; x_1, \ldots, x_n) \geq 1 - \Delta/B,$$

и $\bar{x}_i \leq x_i$ для всех $i = 1, \ldots, n$.

Доказательство. Рассмотрим следующую оптимизационную задачу

$$G(B) = \max \sum_{i=1}^{n} g_i y_i; \tag{2.13}$$

при ограничениях

$$\sum_{i=1}^{n} c_i y_i \leq B; \tag{2.14}$$

$$0 \le y_i \le x_i, \tag{2.15}$$

Пусть $\bar{x}(B)$ — оптимальное решение задачи (2.13)–(2.15). Используя выпуклость функции $G(B)$ и то, что $G(0) = 0$, имеем

$$\frac{g(B-\Delta; \bar{x}_1(B-\Delta), \dots, \bar{x}_n(B-\Delta))}{B-\Delta} = \frac{G(B-\Delta)}{B-\Delta} \ge$$

$$\ge \frac{G(B)}{B} = \frac{g(B; x_1, \dots, x_n)}{B}.$$

Лемма 2 доказана.

Обозначим через $f(S; A_1, \dots, A_n; C_1, \dots, C_k)$ функцию (2.8)–(2.12). Иногда некоторые переменные в этом обозначении для краткости будем опускать.

Лемма 3 . *Если* $\Delta \in [0, C_1]$, *то*

$$1 \ge \frac{f(S; C_1 - \Delta, C_2, \dots, C_k)}{f(S; C_1, \dots, C_k)} \ge 1 - \frac{\Delta}{C_1}.$$

Справедливость леммы следует из того, что если $X = (x_{ij})$ удовлетворяет одному из ограничений (2.10)–(2.12), то $Y \le X$ тоже удовлетворяет этому ограничению. Отсюда из леммы 2 получаем утверждение леммы.

В следующей теореме обобщается известное свойство субмодулярных функций.

Теорема 3 . *Для любых* $S, T \subseteq I$ *справедливо неравенство*

$$f(T) \le (k+1) f(S) + \sum_{p \in T \setminus S} \left(f(S \cup \{p\}) - f(S) \right), \tag{2.16}$$

где k *— число ограничений* (2.12).

Доказательство. Неравенство (2.16) эквивалентно неравенству

$$f(T) + |T| f(S) \le (k+1) f(S) + \sum_{p \in T} f(S \cup \{p\}).$$

Пусть $x^*_{ij}(T)$, где $i \in T$ и $j \in J$, — оптимальное решение задачи (2.8)–(2.12) с множеством T. Для каждого $p \in T$ построим допустимое решение $x_{ij}(S \cup \{p\})$, где $i \in S \cup \{p\}$ и $j \in J$, задач (2.8)–(2.12) с множеством $S \cup \{p\}$ по следующему правилу. Пусть $x_{ij}(S \cup \{p\}) = x^*_{ij}(T)$ при $i = p$. Для любого $i \in S$ полагаем $x_{ij}(S \cup \{p\}) = x_{ij}$, где матрица (x_{ij}), $i \in S$ и $j \in J$, — оптимальное решение задачи

$$\max_{x_{ij} \ge 0} \sum_{i \in S} \sum_{j \in J} f_{ij} x_{ij}$$

при ограничениях:

$$\sum_{i \in S} x_{ij} \le A_j - x^*_{pj}(T), \quad j \in J;$$

$$\varphi_i(x_{i1}, \ldots, x_{in}) \le 0, \quad i \in S;$$

$$\sum_{i \in S} \sum_{j \in J} c_{ijl} x_{ij} \le C_l - \sum_{j \in J} c_{pjl} x^*_{pj}(T), \quad 1 \le l \le k.$$

Пусть $x^*_{pj}(T) = \delta_{pj}$ и $\sum_{j \in J} c_{pjl} x^*_{pj}(T) = \Delta_{pl}$. Из допустимости (x^*_{ij}) следует, что $\sum_{p \in T} \delta_{pj} \le A_j$ и $\sum_{p \in T} \Delta_{pl} \le C_l$ для всех $j \in J$ и $1 \le l \le k$. Неравенство

$$(k+1) f(S) + \sum_{p \in T} f(S \cup \{p\}) \ge (k+1) f(S) +$$

$$+ \sum_{p \in T} \sum_{i \in S \cup \{p\}} \sum_{j \in J} f_{ij} x_{ij}(S \cup \{p\}) \ge (k+1) f(S) + f(T) +$$

$$+ \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_k - \Delta_{pk}).$$

следует из того, что $(x_{ij}(S \cup \{p\}))$ — допустимое решение задачи (2.8)–(2.12). Утверждение теоремы вытекает из следующего неравенства

$$(k+1)f(S) + \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_k - \Delta_{pk}) \geq$$
$$\geq |T| f(S). \tag{2.17}$$

Докажем неравенство (2.17). Используя лемму 3, получаем

$$f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_k - \Delta_{pk}) \geq$$

$$f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_{k-1} - \Delta_{pk-1}, C_k) -$$
$$-\frac{\Delta_{pk}}{C_k} f(S). \tag{2.18}$$

Суммируя неравенства (2.18) по $p \in T$ и используя тот факт, что $\sum_{p \in T} \Delta_{pk} \leq C_k$, имеем

$$f(S) + \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_k - \Delta_{pk}) \geq$$

$$\sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_{k-1} - \Delta_{pk-1}, C_k).$$

Повторив аналогичную процедуру k раз, получаем

$$kf(S) + \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1 - \Delta_{p1}, \ldots, C_k - \Delta_{pk}) \geq$$

$$\sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}; C_1, \ldots, C_k).$$

Для доказательства теоремы осталось установить справедливость следующего неравенства:

$$f(S) + \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}) \geq |T| f(S). \tag{2.19}$$

Пусть $X^*(S) = (x^*_{ij}(S))$, $i \in S$, $j \in J$, — оптимальное решение задачи (2.8)–(2.12) с правыми частями $A_1, \dots, A_n; C_1, \dots, C_k$. Построим допустимые решения задач из левой части неравенства (2.19), на которых оно будет выполняться.

Пусть $A^{'}_j = \sum_{i\in S} x^*_{ij}(S)$. Так как $f(S; A_1, \dots, A_n) = f(S; A^{'}_1, \dots, A^{'}_n)$ и выполняется неравенство

$$f(S; A_1, \dots, A_n) + \sum_{p\in T} f(S; A_1 - \delta_{p1}, \dots, A_n - \delta_{pn}) \geq$$

$$f(S; A^{'}_1, \dots, A^{'}_n) + \sum_{p\in T} f(S; A^{'}_1 - \delta_{p1}, \dots, A^{'}_n - \delta_{pn}),$$

то достаточно рассмотреть случай, когда $A_j = \sum_{i\in S} x^*_{ij}(S)$. Рассмотрим j-й столбец матрицы $X^*(S)$, $x^*_j = (x^*_{ij}(S))$, где $i \in S$. Для каждого $j \in J$ определим множество векторов $\{y^p_j | p \in T\}$, таких, что

1. Для всех $j \in J$ и $p \in T$ размерность вектораy^p_j равна $|S|$.

2. Для всех $i \in S$ и $j \in J$ выполняется неравенство $\sum_{p\in T} y^p_{ij} \leq x^*_{ij}$.

3. Для всех $p \in T$ и $j \in J$ выполняется равенство $\sum_{i\in S} y^p_{ij} = \delta_{pj}$.

Для каждого $j \in J$ вычисляем $\{y^p_j | p \in T\}$ по следующему правилу:

1. Полагаем p равным минимальному индексу в T и $x_{ij} = x^*_{ij}(S)$.

2. Для $i \in S$ в порядке возрастания индекса i вычисляем i-ю компоненту вектора y^p_j:

$$y^p_{ij} = \min\left\{x_{ij}, \delta_{pj} - \sum_{l=1}^{i-1} y^p_{lj}\right\}. \qquad (2.20)$$

3. Если p — максимальный индекс в T, то алгоритм останавливается, в противном случае полагаем $x_{ij} := x_{ij} - y^p_{ij}$, $p := \min_{i\in T}\{i | i > p\}$, и повторяем шаг 2.

Заметим, что построенная система векторов $\{y_j^p | p \in T\}$ удовлетворяет первому и второму свойствам. Убедимся в том, что выполняется третье свойство. По построению для всех $p \in T$ выполняется неравенство $\sum_{i \in S} y_{ij}^p \leq \delta_{pj}$. Допустим, что для некоторого $l \in T$ выполняется строгое неравенство $\sum_{i \in S} y_{ij}^l < \delta_{lj}$. Тогда из (2.20) следует, что $\sum_{p \in T} y_{ij}^p = x_{ij}^*$ для всех $i \in S$. Значит,

$$A_j = \sum_{i \in S} x_{ij}^* = \sum_{i \in S} \sum_{p \in T} y_{ij}^p < \sum_{p \in T} \delta_{pj} \leq A_j.$$

Противоречие. Таким образом, $\{y_j^p : p \in T\}$ удовлетворяет свойству 3.

По свойству 3 матрица $x_{ij}^p = x_{ij}^*(S) - y_{ij}^p$, $i \in S$, $j \in J$, является допустимым решением задачи (2.8)–(2.12) с множеством $S \cup \{p\}$ и правыми частями $A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}$ неравенств (2.10). Используя свойство 2, получаем

$$f(S) + \sum_{p \in T} f(S; A_1 - \delta_{p1}, \ldots, A_n - \delta_{pn}) \geq$$

$$\sum_{i \in S} \sum_{j \in J} f_{ij} (\sum_{p \in T} y_{ij}^p) + \sum_{p \in T} \sum_{i \in S} \sum_{j \in J} f_{ij} x_{ij}^p = |T| f(S),$$

что доказывает неравенство (2.19). Теорема 3 доказана.

2.2 Оценки точности жадных алгоритмов

Приведем описание жадного алгоритма для решения задачи (2.3). Алгоритм состоит из двух этапов:

1-й этап. Положим $S^0 = \emptyset, t = 0$.

Имеем S^t на шаге $t+1$, находим величину

$$\theta_{t+1} = \max_{i \in I \setminus S^t} \frac{\rho_i(S^t)}{d_i} = \frac{\rho_{i_{t+1}}}{d_{i_{t+1}}},$$

где $\rho_i(S^t) = f(S^t \cup \{i\}) - f(S^t)$.
Если $\sum_{i \in S^t} d_i + d_{i_{t+1}} \leq D$, то полагаем $S^{t+1} = S^t \cup \{i_{t+1}\}$ и $t = t+1$. В противном случае полагаем $r = t$ и переходим на второй этап.

2-й этап. Пусть $k \in I \setminus S^r$ — элемент, на котором достигается $\max_{i \in I \setminus S^r} f(\{i\})$. Если $f(S^r) \leq f(\{k\})$, то полагаем $\tilde{S} = \{k\}$. В противном случае $\tilde{S} = S^r$.

В [131] описанный алгоритм применялся к задаче (2.3) с субмодулярной целевой функцией. Жадный алгоритм для задачи (2.2) состоит только из первого этапа с учетом того, что $D = p$ и $d_i = 1$ для всех i. Сложность алгоритма оценивается сверху величиной $O(|I|^2 \cdot T_f)$, где T_f — количество элементарных операций необходимых для вычисления функции $f(S)$. Например, если φ_i — линейные функции, то $f(S)$ можно вычислять с полиномиальной временной сложностью алгоритмом эллипсоидов [32] или алгоритмом Кармаркара [98]. Так как $f(S^r) = 0$ $(+\infty) \iff f(S^*) = 0$ $(+\infty)$, где S^* — оптимальное решение задачи (2.3), то при доказательстве оценок будем считать, что $0 < f(S^*) < +\infty$. Для обоснования оценок точности используется

Лемма 4 . *[132] Пусть k, n и d — произвольные натуральные числа, $\rho_i, 1 \leq i \leq n$, — произвольные действительные неотрицательные* *числа и $\rho_1 > 0$. Тогда*

$$\frac{\rho_1 + \dots + \rho_n}{\min\limits_{t=1,\dots,n}(k \cdot \sum_{i=1}^{t-1} \rho_i + d\rho_t)} \geq \frac{1 - e^{-\frac{k \cdot n}{d}}}{k}.$$

Теперь докажем теорему о свойстве допустимого решения S^r, полученного после первого этапа.

Теорема 4 . *Пусть $\lambda \in [0, 1]$ и $\sum_{i \in S^r} d_i = \lambda D$. Тогда*

$$f(S^r)/f(S^*) \geq (1 - e^{-\lambda(k+1)})/(k+1). \tag{2.21}$$

Доказательство почти совпадает с доказательством аналогичной теоремы для субмодулярных функций, приведенным в [132]. Пусть $S^t, 0 \leq t \leq r$, множества, полученные в результате работы алгоритма. Используя теорему 1 и определение величин $\rho_i(S^t)$, при любом t, $0 \leq t \leq r-1$ получаем

$$f(S^*) \leq (k+1)f(S^t) + \sum_{i \in T \setminus S^t} \rho_i(S^t) \leq (k+1)f(S^t) + D\theta_{t+1}.$$

Второе неравенство следует из того, что $\rho_i(S^t) \leq d_i\theta_{t+1}$ и $\sum_{i \in T} d_i \leq D$.

Положим $n_0 = 0$ и $n_t = \sum_{j=1}^{t} d_{i_j}$, где $1 \leq t \leq r$. Определим величины $\rho_i, i = 1, \dots, \lambda D$, по следующему правилу: $\rho_i = \theta_t$ для всех $i = n_{t-1} + 1, \dots, n_t$.

Так как $f(S^t) = \sum_{i=1}^{n_t} \rho_i$, то для каждого s, $n_t + 1 \leq s \leq n_{t+1}$, имеем

$$(k+1)f(S^t) + D\theta_{t+1} \leq (k+1)\sum_{i=1}^{s-1} \rho_i + D\rho_s.$$

Используя лемму 4 и то, что $f(S^r) = \sum_{i=1}^{\lambda D} \rho_i$, получаем

$$f(S^r)/f(S^*) \geq \frac{\sum_{i=1}^{\lambda D} \rho_i}{\min_{s=1,\dots,\lambda D}\{(k+1)\sum_{i=1}^{s-1} \rho_i + D\rho_s\}} \geq$$

$$\geq \frac{1-e^{-\lambda(k+1)}}{k+1}.$$

Теорема 4 доказана.

Из теоремы 4 вытекает оценка точности решений, получаемых жадным алгоритмом, для задачи (2.2), так как в этом случае $\lambda = 1$ и поэтому

$$\frac{f(S^r)}{f(S^*)} \geq (1-e^{-(k+1)})/(k+1). \tag{2.22}$$

Докажем оценку точности решений задачи (2.3).

Теорема 5 . *Пусть* β_k *является корнем уравнения* $e^{x\cdot(1+k)} = (k+1)(1-x)+1$. *Тогда*

$$f(\tilde{S})/f(S^*) \geq (1-e^{-\beta_k(k+1)})/(k+1).$$

Доказательство. Так как $\sum_{i\in S^r} d_i = \lambda D$ и $d_{i_{r+1}} > (1-\lambda)D$, то, используя лемму 1, получаем

$$\rho_{r+1}(S^r) = f(S^r \cup \{i_{r+1}\}) - f(S^r) \leq f(\{i_{r+1}\}).$$

Следовательно,

$$f(S^*) \leq (k+1)f(S^r) + D\theta_{r+1} <$$
$$< (k+1)f(S^r) + \frac{d_{i_{r+1}}}{1-\lambda}\theta_{r+1} \leq (k+1)f(S^r) + \frac{f(\{i_{r+1}\})}{1-\lambda} \leq$$
$$\leq \left[\left(\frac{1}{1-\lambda}\right) + k + 1\right] \cdot \max\{f(S^r), f(\{i_{r+1}\})\}.$$

Применяя (2.21), получаем неравенство

$$\frac{f(\tilde{S})}{f(S^*)} \geq \frac{\max\{f(S^r), f(\{i_{r+1}\})\}}{f(S^*)} \geq$$

$$\geq \min_{0\leq\lambda\leq 1} \max\left\{\frac{1-e^{-\lambda(k+1)}}{k+1}, \left(\frac{1}{1-\lambda}+k+1\right)^{-1}\right\}.$$

Так как на отрезке [0,1] функция $\frac{1-e^{-\lambda(k+1)}}{k+1}$ строго возрастает, а функция $(\frac{1}{1-\lambda}+k+1)^{-1}$ строго убывает, то максимум достигается в точке пересечения функций при $\lambda = \beta_k$, где β_k — корень уравнения $e^{x\cdot(1+k)} = (k+1)(1-x)+1$. Теорема 5 доказана.

Отметим без доказательства некоторые свойства последовательности β_k:

1. $\{\beta_k\}$ является убывающей последовательностью.
2. $\beta_0 \approx 0,44 < 0,5$ и $\beta_k > 0$для всех $k \geq 0$.
3. $\{\beta_k\} \to 0$ при $k \to +\infty$.

Теорема 6 . *Если $k \geq 0$, то справедливы следующие неравенства*

$$k+1 < \frac{k+1}{1-e^{-(1+k)}} \leq k + r_0^{-1}, \tag{2.23}$$

$$k+2 < \frac{k+1}{1-e^{-\beta_k\cdot(1+k)}} \leq k + R_0^{-1}. \tag{2.24}$$

Доказательство. Первое неравенство в (2.23) очевидно, второе прямым преобразованием приводится к виду

$$e^k \geq k(1-e^{-1})+1,$$

которое обращается в равенство при $k = 0$ и является строгим при $k \geq 1$. Подставляя в (2.24) вместо $e^{\beta_k\cdot(1+k)}$ величину $(k+1)(1-\beta_k)+1$ и делая преобразования, получаем эквивалентные неравенства

$$k+2 < k+1+\frac{1}{1-\beta_k} \leq k+1+\frac{1}{1-\beta_0} = k+R_0^{-1},$$

которые выполняются, так как β_k — убывающая последовательность и $0 < \beta_k < 0,5$. Теорема 6 доказана.

Приведем пример исходных данных задачи (2.2), на которых доказанная оценка достигается асимптотически. В [61] приводится пример исходных данных, на которых оценка достигается при $k = 0$. Рассмотрим задачу (2.2) с функцией $f(S)$, которая является частным случаем функции (2.8)–(2.12).

$$\max_{x_{ij} \geq 0, S \subseteq I} \sum_{i \in S} \sum_{j \in J} f_{ij} x_{ij}$$

при ограничениях:

$$\sum_{i \in S} x_{ij} \leq 1 \quad j \in J;$$

$$\sum_{i \in S} \sum_{j \in J} c_{ijl} x_{ij} \leq 1 \quad l = 1, \ldots, k;$$

$$|S| \leq p.$$

Пусть $|I| = 2k + 2$ и $p = |J| = k + 1$. Обозначим через E_k единичную матрицу порядка k, через O_k нулевую матрицу порядка k и через O_k^l квадратную матрицу того же порядка с единственным ненулевым элементом $a_{ll} = 1$. Определим матрицу $F_k = (f_{ij}^k)$ размера $(2k+2) \times (k+1)$

следующим образом:

$$F_k = \begin{pmatrix} 1+\varepsilon & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \varepsilon \cdot E_k & \\ 0 & & & \\ 0 & & & \\ \vdots & & E_k & \\ 0 & & & \end{pmatrix}.$$

Для каждого k, $k = 1, 2, \ldots$ определим k матриц $C_k^l = (c_{ijl}^k)$, $1 \leq l \leq k$, размера $(2k+2) \times (k+1)$ следующим образом:

$$C_k^l = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & O_k & \\ 0 & & & \\ 0 & & & \\ \vdots & & O_k^l & \\ 0 & & & \end{pmatrix}.$$

Оптимальным решением S^* при достаточно малом ε будет множество $\{2, k+3, \ldots, 2k+2\}$. В результате работы жадного алгоритма получим $\tilde{S} = \{1, 3, \ldots, k+2\}$. Следовательно, $f(\tilde{S})/f(S^*) = (1 + (k+1)\varepsilon)/(k+1) = \varepsilon + (1/(k+1))$. В силу произвольности ε оценка точности решений не превосходит величины $1/(k+1)$, что в асимптотике при $k \to +\infty$ совпадает с оценкой (2.22). Вопрос о достижимости оценки при фиксированном $k \geq 1$ остается открытым.

2.3 Динамическая задача о p-медиане на максимум

Пусть $I = \{1, \ldots, m\}$ – множество возможных пунктов размещения предприятий, $J = \{1, \ldots, n\}$ – множество потребителей продукта, произведенного предприятиями, $T = \{1, \ldots, \mathbf{T}\}$ – плановый период, состоящий из единичных отрезков, которые будем называть годами планового периода. Математическая постановка задачи записывается следующим образом:

$$\max \sum_{t \in T} \sum_{j \in J} \sum_{i \in I} f_{ijt} x_{ijt}, \tag{2.25}$$

$$\sum_{i \in I} x_{ijt} = 1, \quad j \in J, t \in T, \tag{2.26}$$

$$x_{ijt} \leq \sum_{\tau=1}^{t} x_{i\tau}, \quad i \in I, j \in J, t \in T, \tag{2.27}$$

$$\sum_{i \in T} x_{it} \leq P_t, \quad t \in T, \tag{2.28}$$

$$x_{ijt}, x_{it} \in \{0, 1\}, \quad i \in I, j \in J, t \in T. \tag{2.29}$$

Коэффициенты f_{ijt}, P_t имеют следующий смысл: $f_{ijt} \geq 0$ – прибыль, получаемая при обслуживании j-го потребителя предприятием, размещенном в i-м пункте в t-м году, $P_t \geq 0$ – целое число ограничивающее количество предприятий, размещенных в t-м году. Переменные задачи x_{ijt}, x_{it} определяются следующим образом:

$$x_{ijt} = \begin{cases} 1, & \text{если предприятие, размещенное в } i\text{-ом пункте,} \\ & \text{обслуживает } j\text{-го потребителя в } t\text{-м году,} \\ 0, & \text{в противном случае,} \end{cases}$$

$$x_{it} = \begin{cases} 1, & \text{если } i\text{-м пункте размещается предприятие в } t\text{-м году,} \\ 0, & \text{в противном случае.} \end{cases}$$

Целевая функция (2.25) имеет смысл суммарной прибыли, получаемой за весь плановый период. Ограничение (2.26) гарантирует, что все потребители будут обслужены. Ограничение (2.27) означает, что в t-м году j-й потребитель может быть обслужен любым предприятием, размещенным к данному моменту времени. Условия (2.28) ограничивают число предприятий, размещаемых в t-м году.

Рассматриваемая задача отличается от задачи о p-медиане на максимум тем, что описывает ситуацию, в которой размещение предриятий производится не в один момент времени, а в течение планового промежутка, состоящего из некоторого числа единичных отрезков времени. Для каждого такого отрезка заданный спрос потребителей должен быть удовлетворен предприятиями, уже размещенными к этому моменту времени. При этом число предприятий, размещаемых на каждом отрезке времени, ограничено.

Для рассматриваемой динамической задачи предлагается полиномиальный приближенный алгоритм с оценкой относительной погрешности, совпадающей с оценкой из [61] для задачи статической задачи о p-медиане на максимум . Предлагаемый алгоритм получен в результате дерандомизации вероятностного алгоритма построения приближенного решения. Построенный вероятностный алгоритм представляет собой вероятностную процедуру округления оптимального решения вспомогательной задачи линейного програмирования. При дерандомизации алгоритма используется метод условных вероятностей [37].

2.4 Эквивалентная формулировка динамической задачи о p-медиане на максимум

Сопоставим каждой паре (j,t), где $j \in J$, $t \in T$, перестановку $\phi_{jt} = \{i_1(j,t), \ldots, i_n(j,t)\}$ такую, что $f_{i_1(j,t)jt} \geq f_{i_2(j,t)jt} \geq \ldots \geq f_{i_n(j,t)jt}$. В дальнейшем для краткости вместо $i_k(j,t)$ будем писать i_k. Для каждого p, $1 \leq p \leq m$, полагаем $\Delta^p_{jt} = f_{i_p jt} - f_{i_{p+1} jt}$, где $f_{i_{m+1} jt}$ считаем равным нулю. Заметим, что $f_{i_k jt} = \sum_{p=k}^{m} \Delta^p_{jt}$. Используя введенные обозначения, запишем цепочку равенств:

$$\sum_{i \in I} f_{ijt} x_{ijt} = \sum_{k=1}^{m} f_{i_k jt} x_{i_k jt} = \sum_{k=1}^{m} (\sum_{p=k}^{m} \Delta^p_{jt}) x_{i_k jt} = \sum_{p=1}^{m} \Delta^p_{jt} (\sum_{k=1}^{p} x_{i_k jt}).$$

Рассмотрим следующую оптимизационную задачу:

$$\max \sum_{j \in J} \sum_{t \in T} \sum_{p=1}^{m} \Delta^p_{jt} z^p_{jt}; \tag{2.30}$$

$$\sum_{\tau=1}^{t} \sum_{k=1}^{p} y_{i_k \tau} \geq z^p_{jt}, \quad j \in J, t \in T, 1 \leq p \leq m; \tag{2.31}$$

$$\sum_{i \in I} y_{it} \leq P_t, \quad t \in T; \tag{2.32}$$

$$y_{it}, z^p_{jt} \in \{0,1\}, j \in J, t \in T, 1 \leq p \leq m. \tag{2.33}$$

Если $(x_{ijt}), (x_{it})$ – допустимое решение задачи (2.25)–(2.29), то, полагая,

$y_{it} = x_{it}$, $j \in J, t \in T$,

$z_{jt}^p = \sum_{k=1}^{p} x_{i_k jt}$, $j \in J, t \in T, 1 \leq p \leq m$,

получаем допустимое решение задачи (2.30)–(2.33) (ограничения (2.31) следуют из ограничений (2.27)) с тем же значением целевой функции.

Обратно, пусть $(z_{jt}^p), (y_{it})$ допустимое решение задачи (2.30)–(2.33), обладающее свойством максимальности, т.е. если $\sum_{\tau=1}^{t}\sum_{k=1}^{p} y_{i_k\tau} \geq 1$, то $z_{jt}^p = 1$ Любое допустимое решение можно преобразовать к такому виду, не уменьшая при этом значения целевой функции. Пусть $r(j,t) = \min\{p | 1 \leq p \leq n, z_{jt}^p = 1\}$. Заметим, что из свойства максимальности следует, что $z_{jt}^p = 1$ для всех $p \geq r(j,t)$. Положим для всех $j \in J, t \in T, 1 \leq p \leq m$

$x_{it} = y_{it}$, $i \in I, t \in T$,

$$x_{ijt} = \begin{cases} 1, & i = i_{r(j,t)}, \\ 0, & \text{в противном случае}, \end{cases} \qquad i \in I, j \in J, t \in T,$$

Докажем, что определенное таким образом решение $(x_{it}), (x_{ijt})$ задачи (2.25)–(2.29) является допустимым и что значение целевой функции (2.30) совпадает со значением целевой функции (2.25). Действительно,

$$\sum_{t\in T}\sum_{j\in J}\sum_{i\in I} f_{ijt}x_{ijt} = \sum_{t\in T}\sum_{j\in J} f_{i_{r(j,t)}jt} =$$

$$\sum_{t\in T}\sum_{j\in J}(\sum_{p=r(j,t)}^{m} \Delta_{jt}^p) = \sum_{t\in T}\sum_{j\in J}\sum_{p=1}^{m} \Delta_{jt}^p z_{jt}^p.$$

Из определения величин x_{ijt} и x_{it} видно, что ограничения (2.26), (2.28) выполняются. Докажем, что выполняются ограничения (2.27). Для этого достаточно показать, что $1 \leq \sum_{\tau=1}^{t} x_{i_{r(j,t)}\tau}$ для всех $j \in J, t \in T$. В силу максимальности решения $(z_{jt}^p), (y_{jt})$ и определения номера $r(j,t)$

выполняются неравенства

$$\sum_{\tau=1}^{t}\sum_{k=1}^{r(j,t)-1} y_{i_k\tau} = 0 \text{ и } \sum_{\tau=1}^{t}\sum_{k=1}^{r(j,t)} y_{i_k\tau} \geq 1.$$

Следовательно,

$$\sum_{\tau=1}^{t} y_{i_{r(j,t)}\tau} = \sum_{\tau=1}^{t}\sum_{k=1}^{r(j,t)} y_{i_k\tau} - \sum_{\tau=1}^{t}\sum_{k=1}^{r(j,t)-1} y_{i_k\tau} \geq 1.$$

Из доказанного также следует, что значения целевых функций (2.25)–(2.29) и (2.30)-(2.33) на оптимальных решениях совпадают. Так как любое допустимое решение задачи (2.30)-(2.33) можно преобразовать в допустимое решение задачи (2.25)-(2.29), не уменьшая при этом значения целевой функции, получаем, что приближенный алгоритм для решения задачи (2.30)-(2.33) можно преобразовать в приближенный алгоритм для решения задачи (2.25)-(2.29) с той же оценкой относительной погрешности.

2.5 Приближенный алгоритм и его анализ

Задача (2.30)–(2.33) является частным случаем следующей задачи:

$$\max \sum_{j\in J} f_j z_j; \tag{2.34}$$

$$\sum_{(i,t)\in C_j} x_{it} \geq z_j, \quad j \in J; \tag{2.35}$$

$$\sum_{i\in I} x_{it} \leq P_t, \quad t \in T; \tag{2.36}$$

$$x_{it}, z_j \in \{0,1\}, \quad i \in I, j \in J, t \in T, \tag{2.37}$$

где $\{C_j\}, j \in J,$ – произвольная система подмножеств множества $I \times T$.

Если заменить ограничения (2.37) на ограничения

$$0 \leq x_{it} \leq 1, \quad 0 \leq z_j \leq 1, \quad i \in I, j \in J, t \in T, \tag{2.38}$$

то получим линейную релаксацию задачи (2.34)–(2.37).

Приближенный алгоритм решения задачи (2.34)–(2.37) состоит из двух этапов. На первом этапе решается релаксированная задача одним из известных полиномиальных алгоритмов для решения задач линейного программирования. Пусть $(x^*_{it}), (z^*_j)$ – оптимальное решение задачи (2.34)-(2.36),(2.38). Без ограничения общности будем считать, что $\sum_{i \in I} x^*_{it} = P_t$ для всех $t \in T$. На втором этапе для каждого $t \in T$ формируется множество S_t по следующему правилу.

Из множества I выбирается элемент ξ случайным образом с распределением $P(\xi = i) = x^*_{it}/P_t$, затем ξ добавлятся во множество S_t. Процедура повторяется независимо P_t раз.

Заметим, что если $S \subseteq I$, то $P(S \cap S_t = \emptyset) = (1 - \sum_{i \in S} x^*_i/P_t)^{P_t}$. Заметим также, что для каждого $t \in T$ множества S_t формируются независимо. Определяем решение $(x_{it}), (z_j)$ задачи (2.34)-(2.37) следующим образом:

$$x_{it} = \begin{cases} 1, & \text{если } i \in S_t, \\ 0, & \text{в противном случае,} \end{cases} \quad i \in I, t \in T$$

$z_j = \max_{(i,t) \in C_j} x_{it}$, $j \in J$.

Пусть $C_{jt} = \{i : (i,t) \in C_j\}$, тогда $\sum_{(i,t) \in C_j} x^*_{it} = \sum_{t \in T} \sum_{i \in C_{jt}} x^*_{it}$. Оценим математическое ожидание значения целевой функции на округленном решении, используя то, что алгоритм выбирает множества S_t независимо,

$$E = \sum_{j \in J} f_j P(z_j = 1) = \sum_{j \in J} f_j [1 - P(C_{j1} \cap S_1 = \emptyset, \ldots, C_{j\mathbf{T}} \cap S_{\mathbf{T}} = \emptyset)] =$$

$$\sum_{j \in J} f_j [1 - \prod_{t \in T} P(C_{jt} \cap S_t = \emptyset)] = \sum_{j \in J} f_j \left[1 - \prod_{t \in T} \left(1 - \frac{\sum_{i \in C_{jt}} x^*_{it}}{P_t} \right)^{P_t} \right].$$

Пусть $P = \sum_{t \in T} P_t$ и $a_t = \sum_{i \in C_{jt}} x^*_{it} / P_t$. Применяя известное неравенство о среднем арифметическом и среднем геометрическом, получаем

$$\prod_{t \in T} (1 - a_t)^{P_t} \leq \left(\frac{\sum_{t \in T} P_t (1 - a_t)}{P} \right)^P = \left(1 - \frac{\sum_{t \in T} \sum_{i \in C_{jt}} x^*_{it}}{P} \right)^P.$$

Следовательно,

$$\sum_{j \in J} f_j \left[1 - \prod_{t \in T} \left(1 - \frac{\sum_{i \in C_{jt}} x^*_{it}}{P_t} \right)^{P_t} \right] \geq \sum_{j \in J} f_j \left[1 - \left(1 - \frac{\sum_{(i,t) \in C_j} x^*_{it}}{P} \right)^P \right],$$

Так как решение $(x^*_{it}), (z^*_j)$ удовлетворяют ограничениям (2.35), то

$$\sum_{j \in J} f_j \left[1 - \left(1 - \frac{\sum_{(i,t) \in C_j} x^*_{it}}{P} \right)^P \right] \geq \sum_{j \in J} f_j \left[1 - \left(1 - \frac{z^*_j}{P} \right)^P \right].$$

Поскольку функция $f(z) = 1 - (1 - z/p)^p$ является вогнутой и $f(0) = 1, f(1) = 1 - (1 - 1/p)^p$, имеем

$$\sum_{j \in J} f_j \left[1 - \left(1 - \frac{z^*_j}{P} \right)^P \right] \geq \sum_{j \in J} f_j \left[1 - \left(1 - \frac{1}{P} \right)^P \right] z^*_j \geq (1 - e^{-1}) \sum_{j \in J} f_j z^*_j.$$

Таким образом, получаем оценку математического ожидания целевой функции на округленном решении через оптимальное значение линейной релаксации. В следующем параграфе, используя метод условных вероятностей [37], будет получен приближенный алгоритм для решения задачи (2.34)–(2.37) с оценкой относительной погрешности $1 - e^{-1}$. Таким образом, мы доказали следующее утверждение.

Теорема 7 *Для динамической задачи о p-медиане существует полиномиальный приближенный вероятностный алгоритм с оценкой точности* $(1 - e^{-1})$.

2.6 Дерандомизация алгоритма

Напомним, что вероятностный приближенный алгоритм состоит из $\sum_{t \in T} P_t$ независимых шагов. С номером шага n свяжем пару (t, k), где $n = k + \sum_{\tau=1}^{t-1} P_\tau$. Метод условных ожиданий [37] гарантирует нахождение такого допустимого решения, на котором значение целевой функции не меньше математического ожидания значения целевой функции на решении, получаемом вероятностным алгоритмом. Пусть $N = (N_{11}, N_{12}, \ldots, N_{1P_1}, N_{21}, \ldots, N_{\mathbf{T}P_\mathbf{T}})$ – вектор назначений, получаемый в конце рандомизированного алгоритма, т.е. $N_{tk} = (i, t)$, если на шаге (t, k) выбирается i-й элемент из множества I. Пусть $E(N)$ математическое ожидание целевой функции (2.30) на векторе N.

В методе условных вероятностей n-я переменная вектора N определяется на n-ой итерации. Пусть на n-ой итерации уже определены первые $n-1$ переменных $\bar{N}_1, \ldots, \bar{N}_{n-1}$ вектора N. Для каждого $i \in I$ вычисляем условные математические ожидания $E(N|\bar{N}_1, \ldots, \bar{N}_{n-1}, \bar{N}_n = (i, t))$. Используя известное равенство для условной вероятности, имеем

$$E(N|\bar{N}_1, \ldots, \bar{N}_{n-1}) = \sum_{i \in I} E(N|\bar{N}_1, \ldots, \bar{N}_{n-1}, N_n = (i, t)) P(N_n = (i, t)) \leq$$

$$\max_{i \in I}\{E(N|\bar{N}_1, \ldots, \bar{N}_{n-1}, N_n = (i,t))\}.$$

Полагаем $\bar{N}_n = (\tilde{i}, t)$, где $\tilde{i}$ индекс, на котором достигается максимум. Повторяя эту процедуру $P = \sum_{\tau=1}^{\mathbf{T}} P_\tau$ раз, мы получим решение, на котором значение целевой функции не превосходит $E(N)$.

Покажем как вычислять условное ожидание за полиномиальное время. Используя линейность метематического ожидания, достаточно показать как вычислять $P(z_j = 1|\bar{N}_1, \ldots, \bar{N}_n)$. Пусть $S = \{\bar{N}_i | 1 \le i \le n\}$, т.е. S – это множество зафиксированных индексов из $I \times T$. Пусть $n = k + \sum_{\tau=1}^{t-1} P_\tau$ для некоторых k и t, $1 \le k \le P_t$, $1 \le t \le \mathbf{T}$. Условная вероятность вычисляется по формуле:

$$P(z_j = 1|\bar{N}_1, \ldots, \bar{N}_n) =$$

$$\begin{cases} 1, \text{ если } S \cap C_j \neq \emptyset, \\ 1 - \left[\prod_{\tau=t+1}^{\mathbf{T}} \left(1 - \frac{\sum_{i \in C_{j\tau}} x^*_{i\tau}}{P_\tau}\right)^{P_\tau}\right] \left(1 - \frac{\sum_{i \in C_{jt}} x^*_{it}}{P_t}\right)^{P_t - k}, \text{ иначе.} \end{cases}$$

Очевидно, что если $S \cap C_j \neq \emptyset$, то $P(z_j = 1|\bar{N}_1, \ldots, \bar{N}_n) = 1$. Если $S \cap C_j = \emptyset$, то требуется посчитать вероятность того, что из множества C_j будет выбран элемент на оставшихся $P - n$ шагах. Величину $P(z_j = 1|\bar{N}_1, \ldots, \bar{N}_n)$ можно вычислить за $O(|I|\mathbf{T})$ действий. Следовательно, общая трудоемкость алгоритма не превосходит величины $T = O(T_{LP} + |I|P|J| \cdot |I|\mathbf{T}) = O(T_{LP} + |I|^3\mathbf{T}^2|J|)$, где T_{LP} – временная сложность решения линейной релаксации задачи (2.30)–(2.33).

Глава 3

Новая техника округления для задач с ограничением мощность

Введение

В настоящей главе рассматривается частный случай задачи о p-медиане на максимум, так называемая задача о максимальном покрытии p множествами, а также ряд задач, близких к ней по структуре и свойствам.

Для решения этих задач предлагается новый способ округления решений линейных релаксаций, названный методом "перекачки". Округление линейных релаксаций – один из наиболее распространенных способов построения приближенных решений. При этом качество получаемых оценок точности, естественно, зависит от используемой техники округления. Одной из таких продуктивных техник является метод вероятностного округления. Однако при его использовании возникает дополнительная проблема дерандомизации алгоритма.

Предложенная процедура перекачки позволяет построить для задачи о максимальном покрытии p множествами приближенный алгоритм с оценкой точности $(1-(1-1/k)^k)$, где k максимальный размер рассматриваемых в задаче подмножеств. Построен также алгоритм решения задачи о максимальном разрезе с заданными мощностями элементов разбиения с оценкой точности равной $1/2$. Это первый алгоритм с константной оценкой точности для данной версии задачи о максимальном разрезе. Показано, что оба предложенных алгоритма могут быть обобщены и использованы для решения более общих задач, соответственно, задачи о максимальном покрытии с ранцевым ограничением и задачи о максимальном k-разрезе с заданными мощностями элементов разбиения. При этом оценки точности алгоритмов не изменяются.

Чтобы пояснить основную идею предлагаемого метода округления, дадим описание общей схемы приближенного алгоритма, использующего процедуру перекачки, применительно к следующей задаче:

$$\max\ F(x_1, \ldots, x_n) \tag{3.1}$$

$$\sum_{i=1}^{n} x_i = p \tag{3.2}$$

$$x_i \in \{0, 1\}, \quad i = 1, \ldots, n. \tag{3.3}$$

Здесь p – целое неотрицательное число и $F(x_1, \ldots, x_n)$ – полиномиально вычислимая функция с действительными значениями, определенная на векторах $(x_1, \ldots, x_n)$, где $x_i \in [0, 1]$. Предположим, что существует функция $L(x_1, \ldots, x_n)$ такая, что $L(x_1, \ldots, x_n) = F(x_1, \ldots, x_n)$ на всех булевых векторах и что оптимальное решение задачи

$$\max\ L(x_1, \ldots, x_n)$$

$$\sum_{i=1}^{n} x_i = p$$
$$x_i \in [0,1], \quad i = 1,\dots,n.$$

может быть найдено за полиномиальное время, например, в результате сведения этой задачи к задаче линейного програмирования. Указанную задачу назовем оценочной. Предположим, что дополнительно выполняются два свойства:

(A) Существует константа $C > 0$ такая, что для любого $x \in [0,1]^n$ выполняется неравенство $F(x_1,\dots,x_n) \geq CL(x_1,\dots,x_n)$;

(B) функция $\varphi(\varepsilon, x, i, j) = F(x_1,\dots,x_i+\varepsilon,\dots,x_j-\varepsilon,\dots,x_n)$ является выпуклой по переменной $\varepsilon \in [-\min\{x_i, 1-x_j\}, \min\{1-x_i, x_j\}]$ для каждой пары индексов i и j и каждого $x \in [0,1]^n$.

Утверждается, что при вышеприведенных предположениях можно найти за полиномиальное время допустимое решение $\tilde{x} = (\tilde{x}_1,\dots,\tilde{x}_n)$ задачи (3.1)–(3.3), которое удовлетворяет неравенству $F(\tilde{x}) \geq CF(x^*)$, где x^* оптимальное решение задачи (3.1)–(3.3).

Пусть $x' = (x'_1,\dots,x'_n)$ – оптимальное решение оценочной задачи. Если вектор x' не булевый, то в силу равенства (3.2) существует по крайней мере две компоненты x'_i и x'_j со значениями строго между 0 и 1. Рассмотрим вектор $(x'_1,\dots,x'_i+\varepsilon,\dots,x'_j-\varepsilon,\dots,x'_n)$ со значением $\varepsilon = -\min\{x_i, 1-x_j\}$ или $\varepsilon = \min\{1-x_i, x_j\}$ в результате получим вектор $(x''_1,\dots,x''_n)$ с меньшим числом нецелочисленных компонент, для которого в силу свойства (B) выполняется неравенство $F(x'') \geq F(x')$. После повторения указанного шага "перекачки"не более n раз будет построено допустимое решение $\tilde{x}$ исходной задачи. Если x^* оптимальное решение задачи (3.1)–(3.3), то справедливы соотношения

$$F(\tilde{x}) \geq CL(x') \geq CL(x^*) = CF(x^*).$$

Таким образом получаем приближенное решение задачи (3.1)–(3.3) с оценкой точности C.

3.1 Задача о максимальном покрытии p множествами

В задаче о максимальном покрытии p множествами (MCP) дано семейство $\mathcal{F} = \{S_j : j = 1, \ldots, m\}$ подмножеств множества $I = \{1, \ldots, n\}$ с соответствующими весами w_j и положительное целое число p. Требуется найти подмножество $X \subset I$, $|X| = p$, которое максимизирует общий вес множеств в $\mathcal{F}$, имеющих непустое пересечение с X. Полиномиальная разрешимость MCP влечет полиномиальную разрешимость задачи о покрытии и поэтому она NP-трудна. В некотором смысле MCP может трактоваться как обратная задача о покрытии и так же имеет огромное число приложений (смотри к примеру [88]). Широко известен факт, что простой жадный алгоритм решает MCP с оценкой точности $(1 - e^{-1})$ (Cornuejols, Fisher, Nemhauser [61]). Feige [68] доказал, что не существует полиномиального приближенного алгоритма с лучшей оценкой точности при условии P≠NP. Cornuejols, Nemhauser и Wolsey [63] доказали, что жадный алгоритм почти всегда находит оптимальное решение MCP в случае двухэлементных множеств. Мы докажем, что MCP может быть решена за полиномиальное время с оценкой точности $1 - (1 - 1/k)^k$, где $k = \max\{|S_j| : j = 1, \ldots, m\}$. Это лучше, чем оценка точности жадного алгоритма в случае ограниченного k, и совпадает с ней, когда k неограничено. Наш результат (в некотором смысле) является аналогом известного результата Hochbaum [86] о том, что для задачи о покрытии может быть построен приближенный алгоритм с оценкой точности r, где r максимальное число множеств, содержащих элемент.

Заметим, что MCP эквивалентна максимизации функции

$$F(x) = \sum_{j=1}^{m} w_j (1 - \prod_{i \in S_j} (1 - x_i))$$

над всеми булевыми векторами x, удовлетворяющими мощностному ограничению $\sum_{i=1}^{n} x_i = p$.

Формулировка оценочной задачи выглядит следующим образом, максимизировать

$$L(x) = \sum_{j=1}^{m} \min w_j \left\{ 1, \sum_{i \in S_j} x_i \right\}$$

при мощностном ограничении $\sum_{i=1}^{n} x_i = p$. Очевидно, что $L(x) = F(x)$ на всех векторах $x \in [0,1]^n$. Кроме того последняя задача, аналогично задаче MAX SAT (см. [77]), может быть сформулирована как следующая задача линейного программирования:

$$\max \sum_{j=1}^{m} w_j z_j \tag{3.4}$$

$$\sum_{i \in S_j} x_i \geq z_j, \quad j = 1, \ldots, m, \tag{3.5}$$

$$\sum_{i=1}^{n} x_i = p, \tag{3.6}$$

$$0 \leq x_i \leq 1, \quad i = 1, \ldots, n, \tag{3.7}$$

$$0 \leq z_i \leq 1, \quad i = 1, \ldots, m. \tag{3.8}$$

Докажем, что функции F и L удовлетворяют свойствам (A) и (B). Свойство (A) выполняется с константой $C = (1 - (1 - 1/k)^k)$, где $k = \max\{|S_j| : j = 1, \ldots, m\}$; доказательство основывается на следующем неравенстве (в работе [77] доказывается более общее неравенство):

$$1 - \prod_{i=1}^{k}(1 - y_i) \geq (1 - (1 - 1/k)^k) \min\left\{1, \sum_{i=1}^{k} y_i\right\}, \tag{3.9}$$

верного для всех $0 \leq y_i \leq 1$, $i = 1, \ldots, k$. Докажем неравенство (3.9). Из неравенства о среднем арифметическом и среднем геометрическом следует

$$1 - \prod_{i=1}^{k}(1 - y_i) \geq 1 - \left(1 - \frac{\sum_{i=1}^{k} y_i}{k}\right)^k \geq 1 - \left(1 - \frac{\min\left\{1, \sum_{i=1}^{k} y_i\right\}}{k}\right)^k$$

Используя выпуклость функции $g(z) = 1 - (1 - z/k)^k$ и факты $g(0) = 0, g(1) = 1 - (1 - 1/k)^k$ получаем

$$1 - \left(1 - \frac{\min\left\{1, \sum_{i=1}^{k} y_i\right\}}{k}\right)^k \geq (1 - (1 - 1/k)^k) \min\left\{1, \sum_{i=1}^{k} y_i\right\}.$$

Для проверки свойства (B) заметим, что $\varphi(\varepsilon, x, i, j)$ является выпуклой функцией, потому что она является квадратичным полиномом от переменной ε, имеющим неотрицательный главный коэффициент для каждой пары индексов i, j и каждого $x \in [0, 1]^n$. Следовательно, конкретизируя общую схему, описанную во введении, получим приближенный алгоритм с оценкой точности $(1-(1-1/k)^k)$ для задачи о максимальном покрытии p множествами.

3.2 Задача о максимальном разрезе с ограничением на мощность

Пусть G неориентированный граф с множеством вершин $V(G) = \{1,\dots,n\}$. Под разрезом графа G мы будем понимать множество ребер графа G, которые имеют точно один конец в $X \subseteq V(G)$. Обозначим через $\delta(X)$ разрез, определяемый множеством X. Пусть w_{ij} неотрицательные веса, назначенные на ребра $e = (i,j)$ графа G. В задаче о максимальном разрезе с заданными мощностями частей (MCGS) требуется найти разрез $\delta(X)$, имеющий максимальный общий вес ребер среди всех разрезов с $|X| = p$. Для специального случая, также известного как MAX BISECTION PROBLEM, приближенное решение, которое отличается не более, чем в два раза от оптимального, может быть найдено за полиномиальное время простым вероятностным алгоритмом. Однако, естественное обобщение алгоритма на случай с произвольным p не имеет константной оценки точности. Frieze и Jerrum [75], используя релаксацию полуопределенным програмированием (semidefinite programming), доказали, что для MAX BISECTION PROBLEM существует полиномиальный приближенный алгоритм с оценкой точности 0.65, но их техника не позволяет получать обобщения на случай произвольного

p. Хотелось бы отметить, что для минимизационной версии задачи о разбиении графа (или GRAPH PARTITION PROBLEM) неизвестно приближенного алгоритма с константной оценкой точности, а лучший известный алгоритм имеет оценку точности $O(\log^2 n)$ [105] (смотри также обзор по различным техникам для задач о разрезах [125]).

Заметим, что MCGS эквивалентна максимизации следующей квадратичной функции:

$$F(x) = \sum_{i<j} w_{ij}(x_i + x_j - 2x_i x_j)$$

над всеми булевыми векторами x с точно p единицами. Заметим, что функция F имеет свойство (B) по той же самой причине, что функция из предыдущего раздела.

Формулировка оценочной задачи выглядит следующим образом, максимизировать

$$L(x) = \sum_{i<j} w_{ij} \min\{x_i + x_j, 2 - x_i - x_j\}$$

при мощностном ограничении $\sum_{i=1}^{n} x_i = p$. Очевидно, что $L(x) = F(x)$ на всех векторах $x \in [0,1]^n$. Рассмотрим следующую линейную программу эквивалентную оценочной задаче:

$$\max \sum_{i<j} w_{ij} z_{ij}, \tag{3.10}$$

$$z_{ij} \le x_i + x_j, \quad i < j, \tag{3.11}$$

$$z_{ij} \le 2 - x_i - x_j, \quad i < j, \tag{3.12}$$

$$\sum_{i \in V(G)} x_i = p, \tag{3.13}$$

$$x_i, z_{ij} \in [0,1], \quad i \in V(G).\ i < j. \tag{3.14}$$

Обозначим через x оптимальное решение линейной программы (3.10)–(3.14). Мы утверждаем, что $F(x) \geq 1/2L(x)$. Действительно, требуется проверить, что

$$x_i + x_j - 2x_ix_j \geq 1/2 \min\{x_i + x_j, 2 - x_i - x_j\} \tag{3.15}$$

для всех пар индексов i, j с $i < j$. Фиксируем некоторую пару $i < j$ и предположим вначале, что $a = x_i + x_j \leq 1$. Тогда неравенство (3.15) следует из факта, что минимум в правой части (3.15) равен a и $a - 2t(a - t) \geq 1/2a$ для всех t. Случай, когда $a = x_i + x_j > 1$, рассматривается аналогично. Следовательно, мы доказали свойство (B) с $C = 1/2$ и получили приближенный алгоритм с оценкой точности 1/2 для MCGS.

3.3 Задача о максимальном к-разрезе с ограничениями на мощность

В этом разделе мы покажем, как техника "перекачки", описанная во введении, может обобщаться для приложения к естественному обобщению MCGS.

В задаче о максимальном k-разрезе с заданными мощностями частей (k-MCGS) задан полный неориентированный граф G с множеством вершин $V(G) = \{1, \ldots, n\}$, весами ребер w_{ij} и множеством неотрицательных k целых чисел $\{p_1, \ldots, p_k\}$, требуется найти разбиение

$\{P_1, \ldots, P_k\}$ множества вершин $V(G)$ с $|P_t| = p_t$ для каждого t, максимизирующее функцию $\sum_{r<s} \sum_{i \in P_r} \sum_{j \in P_s} w_{ij}$. Задача может быть переформулирована как следующая нелинейная задача математического програмирования:

$$\max \quad F(x) = \sum_{i<j} w_{ij}(1 - \sum_{t=1}^{k} x_{it}x_{jt}), \tag{3.16}$$

$$\sum_{i=1}^{n} x_{it} = p_t, \quad t = 1, \ldots k, \tag{3.17}$$

$$\sum_{t=1}^{k} x_{it} = 1, \quad i \in V(G), \tag{3.18}$$

$$x_{it} \in \{0, 1\}, \quad i \in V(G). \tag{3.19}$$

Легко проверить, что x^*_{it} является оптимальным решением задачи (3.16)–(3.19) тогда и только тогда, когда $\{P_1, \ldots, P_k\}$ с $P_t = \{i : x_{it} = 1\}$ для каждого t, является оптимальным разбиением в задаче k-MCGS.

Сформулируем оценочную задачу

$$L(x) = 1/2 \sum_{i<j} w_{ij} \sum_{t=1}^{k} \min\{x_{it} + x_{jt}, 2 - x_{it} - x_{jt}\},$$

$$\sum_{i=1}^{n} x_{it} = p_t, \quad t = 1, \ldots k,$$

$$\sum_{t=1}^{k} x_{it} = 1, \quad i \in V(G),$$

$$x_{it} \in [0, 1], \quad i \in V(G).$$

Рассмотрим следующую задачу линейного программирования, эквивалентную оценочной задаче и обобщающую программу (3.10)–(3.14):

$$\max \quad 1/2 \sum_{i<j} w_{ij} \sum_{t=1}^{k} z^t_{ij}, \tag{3.20}$$

$$z_{ij}^t \le x_{it} + x_{jt}, \quad i < j, \quad t = 1, \ldots, k, \tag{3.21}$$

$$z_{ij}^t \le 2 - x_{it} - x_{jt}, \quad i < j, \quad t = 1, \ldots, k, \tag{3.22}$$

$$\sum_{i=1}^{n} x_{it} = p_t, \quad t = 1, \ldots, k, \tag{3.23}$$

$$\sum_{t=1}^{k} x_{it} = 1, \quad i \in V(G), \tag{3.24}$$

$$x_{it}, z_{ij}^t \in \{0, 1\}, \quad i, j \in V(G), \quad t = 1, \ldots, k. \tag{3.25}$$

Мы утверждаем, что $F(x) \ge 1/2L(x)$ для всех допустимых $x = (x_{it})$. Фиксируем пару индексов i и j с $i < j$. Достаточно показать, что

$$1 - \sum_{t=1}^{k} x_{it}x_{jt} \ge 1/4 \sum_{t=1}^{k} \min\{x_{it} + x_{jt}, 2 - x_{it} - x_{jt}\}. \tag{3.26}$$

Действительно, в предыдущем разделе (см. (3.15)) доказано, что для каждого t

$$x_{it} + x_{jt} - 2x_{it}x_{jt} \ge 1/2 \min\{x_{it} + x_{jt}, 2 - x_{it} - x_{jt}\}. \tag{3.27}$$

Суммируя вместе (3.27) по t от 1 до k и принимая во внимание (3.24) получаем, что

$$2 - 2\sum_{t=1}^{k} x_{it}x_{jt} \ge 1/2 \sum_{t=1}^{k} \min\{x_{it} + x_{jt}, 2 - x_{it} - x_{jt}\},$$

что влечет (3.26).

Мы опишем сейчас шаг “перекачки”. Обозначим через x оптимальное решение линейной релаксации (3.20)–(3.25) (то есть мы заменили ограничение (3.25) на $x_{jt}, z_{ij}^t \ge 0$). Определим двудольный граф H с множествами вершин ($\{1, \ldots, n\}$ и $\{1, \ldots, k\}$) так, что $(j, t) \in E(H)$ тогда и только тогда x_{jt} дробная. Заметим, что (3.23) и (3.24) влекут, что каждая вершина графа H либо изолированная, либо имеет степень не меньше 2. Предположим, что x имеет дробные компоненты. Так как H двудольный, то H имеет цикл C четной длины. Пусть M_1 и M_2 являются паросочетаниями в H, и их объединение дает цикл C. Определим новое решение $x(\varepsilon)$ по следующему правилу: если (j,t) не ребро C, тогда $x_{jt}(\varepsilon)$ совпадает с x_{jt}, иначе, $x_{jt}(\varepsilon) = x_{jt} + \varepsilon$ if $(j,t) \in M_1$, и $x_{jt}(\varepsilon) = x_{jt} - \varepsilon$ if $(j,t) \in M_2$.

По определению $x(\varepsilon)$ является допустимым решением релаксации (3.20)–(3.25) для всех $\varepsilon \in [-\varepsilon_1, \varepsilon_2]$, где

$$\varepsilon_1 = \min\{\min_{jt\in M_1} x_{jt}, \min_{jt\in M_2}(1 - x_{jt})\}$$

и

$$\varepsilon_2 = \min\{\min_{jt\in M_1}(1 - x_{jt}), \min_{jt\in M_2} x_{jt}\}.$$

Кроме того, $F(x(\varepsilon))$ является квадратичным полиномом от переменной ε с неотрицательным главным коэффициентом и, следовательно,

$$F(x(\varepsilon^*)) \geq F(x) \geq 1/2L(x)$$

для некоторого $\varepsilon^* \in \{-\varepsilon_1, \varepsilon_2\}$. Полученное решение $x(\varepsilon^*)$является допустимым для линейной релаксации (3.20)–(3.25) и имеет меньшее число дробных компонентов.

Устанавливаем $x' = x(\varepsilon^*)$ и, если x' имеет дробные компоненты, повторяем с x' описанный выше шаг ‘перекачки” и так далее. После, самое большее, nk шагов, мы получим решение $\tilde{x}$, которое является допустимым для (3.16)–(3.19) и удовлетворяет

$$F(\tilde{x}) \geq 1/2L(x) \geq 1/2F^*,$$

где F^* – это оптимальное значение (3.16)–(3.19) (и первоначального примера задачи k-MCGS). Следовательно, мы получили полиномиальный 1/2-приближенный алгоритм для k-MCGS. Таким образом, верно следующее утверждение.

Теорема 8 *Существует полиномиальный приближенный алгоритм для задачи $k - MCGS$ с оценкой точности 1/2.*

3.4 Задача о максимальном покрытии с ранцевым ограничением

В настоящем параграфе мы покажем, что техника "перекачки" вместе с техникой частичного перебора, предложенной Khuller, Moss и Naor [102], может работать даже в случае ранцевого ограничения. Мы построим полиномиальный $1-(1-1/k)^k$-приближенный алгоритм для следующего обобщения задачи о максимальном покрытии p множествами, где k является размером максимального множества в индивидуальной задаче. Задача о максимальном покрытии с ранцевым ограничением (MCPK) определяется следующим образом: дано семейство $\mathcal{F} = \{S_j : j = 1, \ldots, m\}$ подмножеств множества $I = \{1, \ldots, n\}$ с соответствующими весами w_j, стоимостями c_j и положительным целым B. Требуется найти подмножество $X \subseteq I$ с $\sum_{j \in X} c_j \leq B$ так, чтобы максимизировать суммарный вес множеств в $\mathcal{F}$, имеющих непустые пересечения с X. Пусть $J = \{1, \ldots, m\}$. Обобщая (3.4)–(3.8), мы переписываем MCPK как следующую задачу целочисленного программирования:

$$\max \sum_{j=1}^{m} w_j z_j \tag{3.28}$$

при ограничениях

$$\sum_{i\in S_j} x_i \geq z_j, \quad j = 1, \ldots, m, \tag{3.29}$$

$$\sum_{i=1}^{n} c_i x_i \leq B, \tag{3.30}$$

$$x_i, z_j \in [0,1], \quad i \in I, j \in J, \tag{3.31}$$

$$x_i \in \{0,1\}, \quad i \in I. \tag{3.32}$$

Заметим, что мы можем исключить переменные z_j, переписывая (3.28)–(3.32) в виде следующей эквивалентной нелинейной программы:

$$\max \quad F(x) = \sum_{j=1}^{m} w_j (1 - \prod_{i\in S_j} (1 - x_i)) \tag{3.33}$$

при ограничениях (3.30)–(3.32).

Пусть $k = \max\{|S_j| : j \in 1, \ldots, m\}$. Обозначим через (x^*, z^*) оптимальное решение задачи (3.28)–(3.32) и через $X^* = \{i : x_i^* = 1\}$.

Предположим, что $|X^*| > 4$ (иначе мы можем найти оптимальное решение полным перебором всех множеств с мощностью не больше 4).

Наш алгоритм перебирает все подмножества $S \subset I$ такие, что $|S| = 4$ и решает линейные программы (3.28)–(3.31) со следующими дополнительными ограничениями:

$$x_i = 1, i \in S. \tag{3.34}$$

Зафиксируем множество $S \subset I$ такое, что $|S| = 4$. Пусть $J_S = \{j : S_j \cap S \neq \emptyset\}$.

Обозначим через $(x^*(S), z^*(S))$ оптимальное решение задачи целочисленного программирования (3.28)–(3.32), (3.34) и через $(x(S), z(S))$ оптимальное решение задачи линейного программирования (3.28)–(3.31), (3.34). Мы утверждаем, что

$$\sum_{j \in J} w_j z_j(S) \geq \sum_{j \in J_S} w_j z_j(S) + [1 - (1 - 1/k)^k] \sum_{j \in J \setminus J_S} w_j z_j. \qquad (3.35)$$

Доказательство следует из следующего неравенства:

$$\begin{aligned} \sum_{j \in J} w_j z_j(S) &\geq \sum_{j \in J} w_j z_j^*(S) \\ &\geq \sum_{j \in J} w_j (1 - \prod_{i \in S_j} (1 - x_i(S))) \\ &\geq \sum_{j \in J_S} w_j z_j(S) + [1 - (1 - 1/k)^k] \sum_{j \in J \setminus J_S} w_j z_j. \end{aligned}$$

Если все компоненты $x(S) = (x_i(S))$ являются целочисленными, то мы получим допустимое решение (3.28)–(3.32). Поэтому будем предполагать, что $x(S)$ имеет дробные компоненты. Мы опишем сейчас процедуру 'перекачки', которая трансформирует $x(S)$ в другое допустимое решение $\tilde{x}(S)$ задачи (3.28)–(3.31), (3.34) со свойством, при котором, самое большее, одна компонента $\tilde{x}(S)$ является дробной. Пусть i и l – такие индексы, что $0 < x_i(S) < 1$ и $0 < x_l(S) < 1$. Положим $x_k(S, \varepsilon) = x_k(S)$, если $k \neq i, l$; $x_k(S, \varepsilon) = x_k(S) + \varepsilon$, если $k = i$ и $x_k(S, \varepsilon) = x_k(S) - \varepsilon c_i / c_l$, если $k = l$.

По построению $x(S, \varepsilon)$ является допустимым для задачи (3.28)–(3.31), (3.34) для всех

$$\varepsilon \in [-\min\{x_i, (1-x_l)\frac{c_l}{c_i}\}, \min\{1-x_i, x_l\frac{c_l}{c_i}\}].$$

Заметим, что $F(x(S,\varepsilon))$ является квадратичным полиномом от переменной ε с неотрицательным главным коэффициентом и, поэтому $F(x(S,\varepsilon^*)) \geq F(x(S))$, где ε^* — это концевая точка интервала. Положим $x'(S) = x(S,\varepsilon)$ и заметим, что $x'(S)$ имеет меньшее число дробных компонент, чем $x(S)$. Если $x'(S)$ имеет, по меньшей мере, две дробных компоненты, мы повторяем описанный шаг 'перекачки'. После, самое большее, n шагов процедура выдает допустимое решение $\tilde{x}(S)$, имеющее, самое большее, одну дробную компоненту и $F(\tilde{x}(S)) \geq F(x(S))$.

Обозначим через t_1 индекс единственной дробной компоненты в решении $\tilde{x}(S)$. Положим $x^1_i(S) = \tilde{x}_i(S)$ для всех $i \neq t_1$ и $x^1_{t_1}(S) = 0$. Заметим, что $x^1(S)$ является допустимым для (3.28)–(3.32), (3.34).

Повторяя описанную выше процедуру к линейной программе, (3.28)–(3.31), (3.34) с дополнительным ограничением

$$x_{t_1} = 0. \qquad (c1)$$

Обозначим через $\tilde{x}^2(S)$ допустимое решение этой программы, с самое большее, одной дробной компонентой. Если вектор $\tilde{x}^2(S)$ булевый, то останавливаем процесс и полагаем $x^2(S) = \tilde{x}^2(S)$. Иначе, если t_2 является индексом дробной компоненты, полагаем $x^2_i(S) = \tilde{x}^2_i(S)$ для всех $i \neq t_2$ и $x^2_{t_2}(S) = 0$ и добавляем ограничение

$$x_{t_2} = 0 \qquad (c_2)$$

к линейной программе (3.28)–(3.31), (3.34), (c_1). Повторяем этот процесс, пока не получим набор целочисленных векторов $x^1(S), \ldots, x^q(S)$ $(q \leq n)$, каждый из которых является допустимым решением задачи (3.28)–(3.32). Обозначим через $y(S)$ вектор из этого набора, дающий максимальное значение целевой функции (3.28). Алгоритм выводит вектор x, дающий максимальное значение целевой функции (3.28) среди всех векторов $y(S)$, где S является 4-элементным подмножеством I.

Проанализируем оценку точности описанного алгоритма. Пусть $r = |X^*|$. Мы можем предполагать, что множество I упорядоченно таким способом, что $X^* = \{1, \ldots, r\}$ и что для каждого $i \in X^*$ элемент i покрывает множества в $\mathcal{F}$, непокрытые элементами $1, \ldots, i-1$, максимального общего веса среди всех $j \in X^* \setminus \{1, \ldots, i-1\}$.

Рассмотрим случай, когда $S = \{1, 2, 3, 4\}$. Вспомним, что алгоритм находит множество $x^1(S), \ldots, x^q(S)$ целочисленных допустимых решений задач, полученных из первоначальной задачи последовательным добавлением ограничения $x_{t_i} = 0$ для некотрых индексов $t_1, t_2, \ldots, t_q$. Предположим вначале, что $\{t_1, t_2, \ldots, t_q\} \cap X^* = \emptyset$. Тогда для каждого $s = 1, \ldots, q$, (x^*, z^*) является допустимым решением задачи линейного программирования (3.28)–(3.31), (3.34), (c_1)–(c_s). Используя аргумент аналогичный аргументу из главы 2 (условия A и B и допустимость решения на каждом шаге перекачки), получаем, что $x^q(S)$ является $(1 - (1 - 1/k)^k)$-приближенным решением задачи (3.28)–(3.32).

Предположим, что $t_s \in X^*$ для некоторого s. Для простоты рассмотрим случай, когда $s = 1$. Тогда мы утверждаем, что $F(x^1(S)) \geq (1-(1-1/k)^k)F^*$, где F^* –это оптимальное значение (3.28)–(3.32).

Действительно, функция F может рассматриваться как функция множества F(X), определенная на всех подмножествах $X \subseteq I$. Хорошо известно, что $F(X)$ является субмодулярной, т.е. обладает следующим свойством

$$F(X \cup \{i\}) - F(X) \geq F(X \cup Y \cup \{i\}) - F(X \cup Y)$$

для всех $X, Y \subseteq I$ и $i \in I$. Мы сейчас выведем из этого факта то, что если $Y \supseteq \{1,2,3,4\}$, тогда

$$1/4F(\{1,2,3,4\}) \geq F(Y \cup \{t\}) - F(Y). \tag{3.36}$$

Действительно,

$$\begin{aligned} 1/4F(\{1,2,3,4\}) &= 1/4(F(\{1,2,3,4\}) - F(\{1,2,3\}) + \\ &\quad F(\{1,2,3\}) - F(\{1,2\}) + \\ &\quad F(\{1,2\}) - F(\{1\}) + \\ &\quad F(\{1\}) - F(\emptyset)) \\ &\geq 4/4(F(Y \cup \{t\}) - F(Y)). \end{aligned}$$

Полагаем $Y = \{i : \tilde{x}(S) = 1\}$. По (3.35) и (3.36) мы имеем

$$\begin{aligned} F(x^1(S)) &= F(Y) \\ &= F(Y \cup \{t\}) - (F(Y \cup \{t\}) - F(Y)) \\ &\geq \sum_{j \in J_S} w_j z_j(S) + (1-(1-1/k)^k) \sum_{j \in J \setminus J_S} w_j z_j(S) - 1/4F(\{1,2,3,4\}) \end{aligned}$$

$$
\begin{aligned}
&= 3/4 \sum_{j\in J_S} w_j z_j(S) + (1-(1-1/k)^k) \sum_{j\in J\setminus J_S} w_j z_j(S)\\
&\geq (1-(1-1/k)^k) \sum_{j\in J} w_j z_j(S)\\
&\geq (1-(1-1/k)^k) F^*.
\end{aligned}
$$

Общий случай с произвольным $s \in \{1, \ldots, q\}$ рассматривается аналогично. Таким образом, верно следующее утверждение.

Теорема 9 *Существует полиномиальный приближенный алгоритм для задачи MCPK с оценкой точности* $1-(1-1/k)^k$.

Глава 4

Задача выполнимости на максимум с ограничением на мощность

Введение

Естественным обобщением задачи о максимальном покрытии p множествами является задача выполнимости на максимум с ограничением на мощность.

В задаче выполнимости на максимум (MAX SAT) задано конечное множество C булевых дизъюнкций. Каждая дизъюнкция состоит из литералов, где литерал либо переменная x, взятая из множества переменных $\{x_1, \ldots, x_n\}$, либо ее отрицание $\bar{x}$. Для каждой дизъюнкции $C_j \in C$ задан ее вес $w_j \geq 0$. Оптимальным решением задачи MAX SAT является такое назначение переменным $x_1, \ldots, x_n$ значений 'ИСТИНА' или 'ЛОЖЬ', которое максимизирует вес выполненных дизъюнкций.

Дизъюнкция считается выполненной, если содержит хотя бы один истинный литерал.

Задача MAX SAT является одной из центральных задач в комбинаторной оптимизации (см. например [95]). Первый нетривиальный приближенный алгоритм для ее решения был предложен Yannakakis [133] и имеет оценку точности 3/4. К настоящему времени наилучший приближенный алгоритм для задачи MAX SAT имеет оценку точности примерно равную 0.77 [47]. В [84] показано, что MAX E3SAT (частный случай задачи MAX SAT, в котором каждая дизъюнкция имеет ровно три литерала), не имеет приближенного алгоритма с оценкой точности лучше чем 7/8 при условии, что $P \neq NP$. Алгоритм с такой оценкой построен в [97] для задачи MAX 3SAT (каждая дизъюнкция имеет не более трех литералов). Таким образом, имеется разрыв между положительными и отрицательными результатами, устанавливающими оценки точности алгоритмов для задачи MAX SAT. Имеется, хотя и значительно меньший, разрыв и для частного случая этой задачи – MAX 2SAT (каждая дизъюнкция имеет не более двух литералов).

В [84] показано, что для этой задачи не существует алгоритма с оценко лучше чем 21/22 при условии, что $P \neq NP$, а в [69] построен алгоритм с оценкой 0,931.

В настоящей главе рассматривается задача MAX SAT с ограничением на число переменных, которые могут принимать значение 'ИСТИНА'. Их число не должно превышать p. Такую задачу будем обозначать CC-MAX SAT. Известно, что для частного случая этой задачи, в которой дизъюнкции не содержат отрицаний, не существует приближенного алгоритма с оценкой точности лучше, чем $1 - e^{-1}$ при условии, что $P \neq NP$ [68]. Для другого частного случая этой задачи построена аппроксимационная схема [100]. В настоящей главе для задачи CC-MAX SAT предлагается приближенный алгоритм с оценкой точности равной $1 - e^{-1}$, которая с учетом сказанного выше является наилучшей возможной. Алгоритм построен с использованием техники вероятностного округления решения линейной релаксации. Заметим, что для задачи выполнимости без ограничения на мощность лучшие известные результаты (см. [69, 78, 97]) получены при использовании других релаксаций.

4.1 Линейная релаксация и приближенный алгоритм

Рассмотрим следующую задачу целочисленного программирования:

$$\max \sum_{C_j \in C} w_j z_j, \tag{4.1}$$

$$\sum_{i \in I_j^+} y_i + \sum_{i \in I_j^-} (1 - y_i) \geq z_j, \quad C_j \in C, \tag{4.2}$$

$$\sum_{i=1}^{n} y_i \leq P, \tag{4.3}$$

$$0 \leq z_j \leq 1, \quad C_j \in C, \tag{4.4}$$

$$y_i \in \{0, 1\}, \quad i = 1, \ldots, n, \tag{4.5}$$

где I_j^+ (соответственно, I_j^-) обозначает множество переменных в дизъюнкции C_j , которые входят в нее без отрицаний (соответственно, с отрицаниями). Полагая $y_i = 1$, если x_i имеет значение 'ИСТИНА' и $y_i = 0$, если x_i имеет значение 'ЛОЖЬ', $z_j = 1$, если дизъюнкция C_j выполняется и $z_j = 0$, если дизъюнкция C_j не выполняется, мы получаем, что любое допустимое решение программы (4.1)-(4.5) соответствует допустимому решению задачи CC-MAX SAT. Схожая задача целочисленного программирования использовалась Goemans и Williamson [77] для построения приближенных алгоритмов для задачи MAX SAT.

Пусть $M \geq 1$ некоторая константа, которая будет определена в следующем разделе. Рассмотрим задачу (4.1)-(4.5) с дополнительным ограничением $\sum_{i=1}^n y_i \leq M$. Мы можем найти оптимальное решение (y_1, z_1) этой задачи за полиномиальное время полным перебором. Рассмотрим задачу (4.1)-(4.5) с другим дополнительным ограничением $\sum_{i=1}^n y_i \geq M$. Пусть (y_2, z_2) – это α-приближенное решение этой задачи. Очевидно, что лучшее из этих двух решений является α-приближенным решением задачи CC-MAX SAT. Следовательно, без ограничения общности можно рассматривать задачу (4.1)-(4.5) с ограничением $\sum_{i=1}^n y_i \geq M$.

Для $t = M, \ldots, P$ рассмотрим линейные программы LP_t, полученные заменой ограничений $y_i \in \{0, 1\}$ на ограничения $0 \leq y_i \leq 1$ и заменой (4.3) на ограничение

$$\sum_{i=1}^{n} y_i = t. \tag{4.6}$$

Обозначим через F_t^* значение целевой функции на оптимальном решении LP_t. Через k обозначим индекс такой, что $F_k^* = \max_{M \le t \le n} F_t^*$.

Так как любое допустимое решение задачи (4.1)-(4.5) с ограничением $\sum_{i=1}^{n} y_i \ge M$ является допустимым решением LP_t для некоторого t, мы получаем, что F_k^* является верхней оценкой для этой задачи. Сейчас мы опишем вероятностный приближенный алгоритм для задачи CC-MAX SAT.

Описание алгоритма

1. Решим линейные программы LP_t для всех $t = M, \dots, P$. Пусть (y^*, z^*) — это оптимальное решение LP_k.
2. Вторая часть алгоритма состоит из k независимых шагов. На каждом шаге алгоритм выбирает индекс i из множества $\{1, \dots, n\}$ случайным образом с вероятностью $P_i = \frac{y_i^*}{k}$. Обозначим через S множество выбранных индексов. Заметим, что $P \ge k \ge |S|$. Полагаем $x_i = 1$, если $i \in S$ и $x_i = 0$ в противном случае.

Окончательно алгоритм состоит из двух шагов. Первый шаг – это решение линейных программ LP_t для всех $t = M, \dots, P$. Мы можем сделать это, используя любой из известных полиномиальных алгоритмов для решения задач линейного программирования (например [32, 98]). Второй шаг – это дерандомизация вероятностной части описанного алгоритма. Мы покажем в разделе 4.4, что дерандомизация может быть сделана за полиномиальное время. В следующем разделе мы оценим математической ожидание значения целевой функции на округленном решении.

4.2 Анализ алгоритма

4.2.1 Технические леммы

В настоящем подразделе мы докажем некоторые технические леммы.

Лемма 5 *Вероятность того, что произойдет хотя бы одно из событий $A_1, \ldots, A_n$ определяется по формуле*

$$Pr(A_1 \cup \ldots \cup A_n) = \sum_{1 \le i \le n} Pr(A_i) - \sum_{1 \le i_1 < i_2 \le n} Pr(A_{i_1} \cap A_{i_2}) + \ldots$$
$$+(-1)^{t-1} \sum_{1 \le i_1 < \ldots < i_t \le n} Pr(A_{i_1} \cap \ldots \cap A_{i_t}) + \ldots$$

Доказательство. Смотри в [71], гл.1, раздел 4.

Лемма 6 *Вероятность того, что произойдет хотя бы одно из событий $B, A_1, \ldots, A_n$ определяется по формуле*

$$Pr(B \cup A_1 \cup \ldots \cup A_n) = Pr(B) + \sum_{1 \le i \le n} Pr(\bar{B} \cap A_i) -$$
$$- \sum_{1 \le i_1 < i_2 \le n} Pr(\bar{B} \cap A_{i_1} \cap A_{i_2}) + \ldots$$
$$+(-1)^{t-1} \sum_{1 \le i_1 < \ldots < i_t \le n} Pr(\bar{B} \cap A_{i_1} \cap \ldots \cap A_{i_t}) + \ldots$$

Доказательство. Утверждение следует из леммы 5 и фактов

$$Pr(B \cup A_1 \cup \ldots \cup A_n) = Pr(B) + Pr(\bar{B} \cap (A_1 \cup \ldots \cup A_n)) =$$

$$Pr(B) + Pr((\bar{B} \cap A_1) \cup \ldots \cup (\bar{B} \cap A_n)).$$

Лемма 7 *Неравенства*

$$1-e^{-y}\le e^{-1+y}, 1-e^{-4/k}e^{-y}\le e^{-1+y}-f(k),$$

$$1-e^{-4/k}e^{-y}-e^{-4/k}e^{-x}+e^{-x-y}\le e^{-2+x+y}$$

выполняются для всех $y,x\in[0,1]$, $k\ge M$, *где* M *достаточно большая константа, не зависящая от* x,y *и* $\lim_{k\to+\infty}f(k)=0$.

Доказательство. Пусть $x=e^y$, тогда первое неравенство эквивалентно $x^2-ex+e\ge 0$. Так как $e^2-4e<0$, мы получаем нужное утверждение. Используя аналогичный аргумент, мы можем доказать второе неравенство k. Докажем третье неравенство

$$1-e^{-4/k}e^{-y}-e^{-4/k}e^{-x}+e^{-x-y}=$$

$$=(1-e^{-4/k}e^{-x})(1-e^{-4/k}e^{-y})+f_1(k)\le$$

где $f_1(k)=e^{-x}e^{-y}(1-e^{-8/k})$. Пусть $f(k)=e\cdot f_1(k)$, тогда мы продолжаем, используя второе неравенство,

$$\le(e^{-1+x}-f(k))e^{-1+y}+f_1(k)\le e^{-2+x+y}.$$

В следующих леммах мы используем широко известные неравенства

$$e^{-1}\ge(1-1/k)^k\ge e^{-1-1/k}$$

for $k\ge 2ue^{-a}\ge(1-a/k)^k\ge e^{-a-a^2/k}$ for $k\ge 2a\ge 0$.

Лемма 8 *Неравенство*

$$f(x,y,z)=\left(1-\frac{x}{k}\right)^k+\left(1-\frac{y}{k}\right)^k+$$

$$\left(1-\frac{z}{k}\right)^k-\left(1-\frac{x+y}{k}\right)^k-\left(1-\frac{y+z}{k}\right)^k-$$

$$-\left(1-\frac{z+x}{k}\right)^k+\left(1-\frac{x+y+z}{k}\right)^k>1-e^{-1}.$$

выполняется для всех $x,y,z\in[0,1]$ и $k\geq M$, где M достаточно большая константа, не зависящая от x,y,z.

Доказательство. Заметим, что следующие неравенства выполняются для всех $x\in[0,1]$

$$e^{-x}-(1-x/k)^k\leq e^{-x}(1-e^{-x^2/k})\leq 1-e^{-1/k}.$$

Используя аналогичный аргумент, мы получаем

$$\lim_{k\to+\infty}\left\{e^{-x}+e^{-y}+e^{-z}+e^{-x-y-z}-\left(1-\frac{x}{k}\right)^k-\right.$$

$$\left.-\left(1-\frac{y}{k}\right)^k-\left(1-\frac{z}{k}\right)^k-\left(1-\frac{x+y+z}{k}\right)^k\right\}=0$$

и, следовательно, для больших k мы имеем

$$f(x,y,z)\geq e^{-x}+e^{-y}+e^{-z}-$$

$$-e^{-x-y}-e^{-x-z}-e^{-y-z}+e^{-x-y-z}-o(1)=$$

$$=1-(1-e^{-x})(1-e^{-y})(1-e^{-z})-o(1)\geq$$

$$1-(1-e^{-1})^3-o(1)>0.74>1-e^{-1}.$$

4.2.2 Оценка математического ожидания

Обозначим через S множество индексов, выбранных вероятностным алгоритмом, через $f(S)$ обозначим значение целевой функции на решении, определяемом по множеству S, и через $E(f(S))$ обозначим математическое ожидание $f(S)$. Докажем основное утверждение настоящей главы.

Теорема 10 *Имеют место неравенства*

$$F_k^* \geq E(f(S)) \geq (1 - e^{-1})F_k^*$$

Доказательство. Используя линейность математического ожидания, получаем:

$$E(f(S)) = \sum_{C_j \in C} w_j Pr(z_j = 1).$$

Пусть $X^+ = \sum_{i \in I_j^+} y_i^*$. Рассмотрим четыре случая.

Случай 1 Предположим, что $|I_j^-| = \emptyset$. Так как шаги алгоритма независимы и $X^+ \geq z_j^*$, мы получаем:

$$Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset) = 1 - \left(1 - \frac{X^+}{k}\right)^k \geq$$

$$1 - \left(1 - \frac{z_j^*}{k}\right)^k \geq \left(1 - \left(1 - \frac{1}{k}\right)^k\right) z_j^*.$$

Последнее неравенство следует из вогнутости функции $g(z) = 1 - (1 - z/k)^k$ и того, что $g(0) = 0, g(1) = 1 - (1 - 1/k)^k$.

Случай 2 Предположим, что $|I_j^-| = 1$. Пусть $I_j^- = \{t\}$ и $a = y_t^*$. Если $X^+ > 1$, то используя аргумент аналогичный аргументу из предыдущего случая, получаем $Pr(z_j = 1) \geq Pr(S \cap I_j^+ \neq \emptyset) \geq 1 - e^{-1}$. Предположим, что $X^+ \leq 1$, тогда, используя леммы 6,7 и неравенство $X^+ + (1-a) \geq z_j^*$, имеем:

$$Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset \text{ или } t \notin S) =$$

$$Pr(S \cap I_j^+ \neq \emptyset) + Pr(S \cap I_j^+ = \emptyset \text{ и } t \notin S) =$$

$$= 1 - \left(1 - \frac{X^+}{k}\right)^k + \left(1 - \frac{X^+ + a}{k}\right)^k \geq$$

$$\geq 1 - e^{-X^+} + e^{-X^+ - a - (X^+ + a)^2/k} \geq$$

$$\geq 1 - e^{-X^+} + e^{-4/k} e^{-X^+ - a} \geq$$

$$1 - e^{-X^+} e^{-1+a} \geq 1 - e^{-z_j^*} \geq (1 - e^{-1}) z_j^*$$

Случай 3 Предположим, что $|I_j^-| = 2$. Пусть $I_j^- = \{i_1, i_2\}, a = y_{i_1}^*$ и $b = y_{i_2}^*$. Без ограничения общности предположим, что $X^+ \leq 1$, тогда используя леммы 6, 7 и неравенство $X^+ + (1-a) + (1-b) \geq z_j^*$, мы получаем:

$$Pr(z_j = 1) = Pr(S \cap I_j^+ \neq \emptyset \text{ или } i_1 \notin S \text{ или } i_2 \notin S) =$$

$$= Pr(S \cap I_j^+ \neq \emptyset) + Pr(S \cap I_j^+ = \emptyset \text{ и } i_1 \notin S) +$$

$$+ Pr(S \cap I_j^+ = \emptyset \text{ и } i_1 \notin S) - Pr(S \cap I_j^+ = \emptyset \text{ и } i_1 \notin S \text{ и } i_2 \notin S) =$$

$$= 1 - \left(1 - \frac{X^+}{k}\right)^k + \left(1 - \frac{X^+ + a}{k}\right)^k +$$

$$\left(1 - \frac{X^+ + b}{k}\right)^k - \left(1 - \frac{X^+ + a + b}{k}\right)^k \geq$$

$$\geq 1 - e^{-X^+} + e^{-4/k} e^{-X^+ - a} + e^{-4/k} e^{-X^+ - b} - e^{-X^+ - a - b} \geq$$

$$\geq 1 - e^{-X^+} e^{-2+a+b} \geq 1 - e^{-z_j^*} \geq (1 - e^{-1}) z_j^*$$

Случай 4 Предположим, что $|I_j^-| \geq 3$. Пусть i_1, i_2, i_3 — это произвольные индексы из множества I_j^-, тогда, используя леммы 5, 8, мы получаем:

$$Pr(z_j = 1) \geq Pr(i_1 \notin S \text{ или } i_2 \notin S \text{ или } i_3 \notin S) =$$

$$= Pr(i_1 \notin S) + Pr(i_2 \notin S) + Pr(i_3 \notin S) -$$

$$- Pr(i_1 \notin S \text{ и } i_2 \notin S) - Pr(i_2 \notin S \text{ и } i_3 \notin S) -$$

$$- Pr(i_1 \notin S \text{ и } i_3 \notin S) + Pr(i_1 \notin S \text{ и } i_2 \notin S \text{ и } i_3 \notin S) =$$

$$= \left(1 - \frac{y_{i_1}^*}{k}\right)^k + \left(1 - \frac{y_{i_2}^*}{k}\right)^k + \left(1 - \frac{y_{i_3}^*}{k}\right)^k -$$

$$- \left(1 - \frac{y_{i_1}^* + y_{i_2}^*}{k}\right)^k - \left(1 - \frac{y_{i_2}^* + y_{i_3}^*}{k}\right)^k - \left(1 - \frac{y_{i_3}^* + y_{i_1}^*}{k}\right)^k +$$

$$+ \left(1 - \frac{y_{i_1}^* + y_{i_2}^* + y_{i_3}^*}{k}\right)^k > 1 - e^{-1}.$$

4.3 Дерандомизация

В настоящем разделе мы покажем, как детерменировано находить приближенное решение за полиномиальное время. Это будет сделано с использованием метода условных вероятностей [37]. Прямое применение метода условных вероятностей не дает полиномиального алгоритма, так как если задача CC-MAX SAT содержит дизъюнкции с не константным числом отрицаний, мы не можем вычислять (используя лемму 2) условные вероятности за полиномиальное время.

Пусть IP_1 является индивидуальной задачей CC-MAX SAT, заданной множеством дизъюнкций $C = \{C_j : j = 1, \dots, m\}$ и множеством переменных $\{x_1, \dots, x_n\}$. Обозначим через F_k^* значение целевой функции на оптимальном решении релаксации LP_k для IP_1. Определим пример IP_2 задачи CC-MAX SAT заменой каждой дизъюнкции такой, что $|I_j^-| \geq 3$ на дизъюнкцию $C_j' = \bar{x}_{i_1} \vee \bar{x}_{i_2} \vee \bar{x}_{i_3}$, где $i_1, i_2, i_3 \in I_j^-$.

Мы применяем вероятностный алгоритм с вероятностями, определенными решением релаксации LP_k. Пусть S является решением, полученным вероятностным алгоритмом, $f_1(S)$ - значением целевой функции задачи IP_1 на решении S и $f_2(S)$ - значением целевой функции задачи IP_2 на решении S, тогда, используя факт, что $Pr(z_j = 1) > 1 - e^{-1}$ для всех дизъюнкций C_j с $|I_j^-| \geq 3$, получаем $E(f_2(S)) \geq (1 - e^{-1})F_k^*$. Мы можем дерандомизировать алгоритм следующей процедурой

Описание дерандомизации

Дерандомизация состоит из k шагов. На s-ом шаге выбираем индекс i_s, который максимизирует условную вероятность, т.е.

$$E(f_2(S)|i_1 \in S, \dots, i_{s-1} \in S, i_s \in S) =$$

$$\max_{j \in \{1,\dots,n\}} E(f_2(S)|i_1 \in S, \dots, i_{s-1} \in S, j \in S).$$

Так как

$$\max_{j \in \{1,\dots,n\}} E(f_2(S)|i_1 \in S, \dots, i_{s-1} \in S, j \in S) \geq$$
$$E(f_2(S)|i_1 \in S, \dots, i_{s-1} \in S)$$

к концу дерандомизации получим решение $\tilde{S}$ такое, что $f_2(\tilde{S}) \geq E(f_2(S)) \geq (1-e^{-1})F_k^*$. Так как $F_k^* \geq f_1(\tilde{S}) \geq f_2(\tilde{S})$, это решение является $(1-e^{-1})$-приближенным решением для IP_1. Мы можем вычислить условные математические ожидания за полиномиальное время, используя их линейность, лемму 2 и тот факт, что $|I_j^-| \leq 3$ в задаче IP_2.

Глава 5

Квадратичная задача о назначениях

Введение

Квадратичная задача о назначениях (КЗН), записанная как задача целочисленного программирования, имеет вид

$$\min \sum_{i \in I} \sum_{p \in J} \left(\sum_{j \in I} w_{ij} \sum_{q \in J} d_{pq} y_{jq} \right) y_{ip}, \tag{5.1}$$

$$\sum_{p \in J} y_{ip} = 1, \quad i \in I, \tag{5.2}$$

$$\sum_{i \in I} y_{ip} \leq 1, \quad p \in J, \tag{5.3}$$

$$y_{ip} \in \{0.1\}, \quad i \in I, p \in J. \tag{5.4}$$

На содержательном уровне эту задачу можно проинтерпретировать следующим образом. Имеется множество $I = \{1, \ldots, n\}$ возможных пунктов размещения предприятий из множества $J = \{1, \ldots, m\}$ ($m \leq n$). Известны матрица (d_{pq}) размера $n \times n$ расстояний между пунктами и матрица (w_{ij}) размера $m \times m$ удельных (на единицу расстояния) стоимостей грузопотока между предприятиями. Необходимо так разместить предприятия, чтобы суммарная стоимость транспортных грузопотоков была наименьшей.

В [124] показано, что для КЗН не существует полиномиального приближенного алгоритма с константной оценкой точности при условии, что $P \neq NP$. В [118] этот результат усилен и доказано, что даже для частного случая КЗН с матрицей (d_{pq}), которая задается расстояниями на прямой (следовательно, удовлетворяет неравенству треугольника), и с матрицей (w_{ij}), элементы которой принимают только два значения, не существует полиномиального приближенного алгоритма с константной оценкой точности при условии, что $P \neq NP$. В [76] с использованием техники из [41, 42] построена аппроксимационная схема для КЗН, когда матрица (d_{pq}) – матрица евклидовых расстояний, а элементы матрицы (w_{ij}) больше единицы, но меньше некоторой константы.

В упомянутой выше работе [118] поставлен вопрос: существует ли приближенный алгоритм с константной оценкой точности для КЗН, если матрица (d_{pq}) удовлетворяет неравенству треугольника, а все элементы матрицы (w_{ij}) равны единице. В настоящей главе дается положительный ответ на этот вопрос. Для указанного частного случая КЗН предлагается приближенный алгоритм с оценкой точности равной 2 и оценкой временной сложности $O(mn^2)$.

5.1 Алгоритм и его анализ

Запишем рассматриваемую задачу в эквивалентной форме. Пусть $G = (V, E)$ – полный ориентированный граф без петель. Пусть для всех вершин $x, y \in V$ определено расстояние $d(x, y)$. Мы предполагаем, что $d(x, y)$ удовлетворяет неравенству треугольника и $d(x, x) = 0$. Тогда задача может быть записана следующим образом:

$$\min_{S \subseteq V, |S|=m} \sum_{x,y \in S} d(x, y). \tag{5.5}$$

Обозначим через $W(x, S)$ сумму длин ребер в графе $(x \cup S, E(x \cup S))$, где $E(x \cup S) = \{(x, y), (y, x) | y \in S\}$. Эту величину будем также называть весом графа $(x \cup S, E(x \cup S))$. Результатом работы алгоритма является множество $t \cup T$ такое, что

$$W(t, T) = \min_{x \in V, |S|=m-1} W(x, S).$$

Другими словами, алгоритм находит звезду минимального веса на m вершинах. Звезда минимального веса с центром в данной вершине x и с $m - 1$ лучом может быть найдена не более, чем за $O(mn)$ шагов. Лучами звезды минимального веса с центром в x являются $m - 1$ вершина с минимальным значением функции $d(v, x) + d(x, v)$. Перебирая все вершины $x \in V$, мы можем найти приближенное решение $t \cup T$ не более, чем за $O(mn^2)$ шагов.

Теорема 11 *Значение целевой функции на полученном приближенном решении отличается не более, чем в два раза от значения целевой функции на оптимальном решении S^*.*

Доказательство. Обозначим через $\tilde{F}$ значение целевой функции на полученном приближенном решении и через F^* значение целевой функции на оптимальном решении. Вначале получим нижнюю оценку для F^*

$$F^* = \frac{1}{2}\sum_{x \in S^*} W(x, S^* \setminus x) \geq \frac{1}{2} m W(t, T).$$

Затем получим верхнюю оценку для $\tilde{F}$

$$\tilde{F} = \sum_{x,y \in t \cup T} d(x,y) \leq \sum_{x,y \in T} d(x,t) + d(t,y) =$$
$$(m-1)\sum_{x \in T} d(x,t) + d(t,x) = (m-1)W(t,T).$$

Следовательно,

$$\frac{\tilde{F}}{F^*} \leq 2\left(1 - \frac{1}{m}\right).$$

Построим пример задачи, который показывает, что доказанная оценка неулучшаема. Пусть $V = K_1 \cup K_2$, где $|K_1| = |K_2| = m$. Функция d определяется следующим образом: $d(x,y) = 1+\varepsilon$ для некоторого $\varepsilon > 0$, если $x, y \in K_1$, $d(x,y) = d(y,x) = 2$ если $x \in K_1, y \in K_2$. Множество K_2 имеет специальную вершину z такую, что $d(z,x) = d(x,z) = 1$ если $x \in K_2$ и $d(x,y) = 2$ если $x, y \in K_2 \setminus \{z\}$. Алгоритм находит множество K_2, в то время как оптимальным решением является множество K_1. Следовательно,

$$\frac{\tilde{F}}{F^*} = \frac{2(n-1) + 2(n-1)(n-2)}{n(n-1)(1+\varepsilon)} < 2\left(1 - \frac{1}{n}\right).$$

Так как ε может быть произвольным положительным числом, то отношение $\frac{\tilde{F}}{F^*}$ может быть сколь угодно близким к выражению в правой части неравенства.

Глава 6

Задача о p-центре

Введение

Рассмотрим задачу о p-центре в следующей форме. В полном неориентированном взвешенном графе с множеством вершин V требуется найти подмножество вершин $S \subseteq V$, $|S| \le p$, которое минимизирует расстояние от S до подмножества $V \setminus S$. Расстояние от S до $V \setminus S$ определяется как величина $\max_{i \in V \setminus S} \min_{j \in S} d_{ij}$, поэтому задача о p-центре записывается следующим образом:

$$\min_{S \subseteq V} \{ \max_{i \in V \setminus S} \min_{j \in S} d_{ij} : |S| \le p \}.$$

В дальнейшем будем предполагать, что матрица весов ребер (d_{ij}) удовлетворяет неравенству треугольника, т. е. для всех i, j, k выполняется неравенство $d_{ij} + d_{jk} \ge d_{ik}$. В [93] доказано, что для задачи о p-центре не существует приближенного алгоритма с оценкой точности меньше 2 при условии, что $P \ne NP$. Hochbaum и Shmoys [91] построили первый приближенный алгоритм для задачи о p-центре с оценкой точности равной 2. Затем они же [92] обобщили свою технику на более широкий класс задач об узком месте. В [51, 53, 89, 90, 103, 104] изучались различные обобщения задач о p-центре, для многих из них были построены приближенные алгоритмы с наилучшими возможными оценками точности. Вместе с тем ряд очень интересных вопросов остается открытым для самой задачи о p-центре. Так в [67] показано, что для задачи о p-центре с матрицей (d_{ij}) евклидовых расстояний не существует приближенного алгоритма с оценкой точности меньше 1.81 при условии, что $P \neq NP$. Вместе с тем наилучший приближенный алгоритм для такой задачи имеет оценку точности равную 2.

В настоящей главе рассматривается обобщение задачи о p-центре, в котором для каждой вершины $i \in V$ известен ее вес $c_i \geq 0$ и ограничение на мощность множества S заменяется ограничением на суммарный вес вершин, входящих в S. Задача записывается следующим образом

$$\min_{S \subseteq V} \{ \max_{i \in V \setminus S} \min_{j \in S} d_{ij} | \sum_{i \in S} c_i \leq C \}. \tag{6.1}$$

Для данной задачи в [92] предложен приближенный алгоритм с оценкой точности равной 3 и оценкой трудоемкости $O(n^2 \log_2 n)$, где $n = |V|$. Ниже для этой задачи с использованием алгоритма из [65] предлагается алгоритм с такой же оценкой точности, но с лучшей оценкой трудоемкости $O(n^2)$.

6.1 Описание и анализ алгоритма

Введем следующие обозначения:

$r(S)$ – значение целевой функции на множестве $S \subseteq V$,

$H(R,i) = \{j | d_{ij} \leq R\}$ – окрестность вершины i радиуса R,

$v(R,i)$ – вершина минимального веса в окрестности $H(R,i)$,

$T(R,S) = \{v(R,i) | i \in S\}$ – набор вершин минимального веса, сопоставленных вершинам из множества S,

S^* – оптимальное решение задачи (6.1),

R^* – значение целевой функции на оптимальном решении, т. е. $R^* = r(S^*)$.

Алгоритм решения задачи (6.1) состоит из двух этапов. На первом этапе формируются два множества S и S_l. На каждом шаге первого этапа во множество S добавляется наиболее удаленная вершина. Если при этом изменяется расстояние от S до $V \setminus S$, то множество S_l изменяется. На втором этапе происходит улучшение уже полученного решения.

Алгоритм

Первый этап

1-й шаг. Выбирается произвольная вершина $i_1 \in V$ такая, что $c_{i_1} \leq C$, и полагается $S = \{i_1\}, S_l = \{i_1\}$. Для всех вершин $j \in V \setminus S$ находится $r_j = d_{i_1 j}$ (расстояние от j-й вершины до i_1-й вершины) и полагается $D_1 = \max_{j \in V \setminus S} r_j, p = 1, l = 1$.

k-й шаг. Находится вершина $i_k \in V \setminus S$ такая, что $r_{i_k} = D_{k-1}$ (т. е. самая удаленная от множества S вершина). Полагается $S = S \cup \{i_k\}$. Для всех $j \in V \setminus S$ находится $r_j = \min\{r_j, d_{ji_k}\}$ (расстояние от j-й вершины до множества S) и полагается $D_k = \max_{j \in V \setminus S} r_j, p = k$.

а) Если $D_k = D_{k-1}$, то алгоритм выполняет $(k+1)$-й шаг.

б) Если $D_k < D_{k-1}$ и $\sum_{i \in T(D_k/2,S)} c_i \leq C$, то полагается $l = k, S_l = T(D_k/2, S)$ и алгоритм выполняет $(k+1)$-й шаг.

в) Если $D_k < D_{k-1}$ и $\sum_{i \in T(D_k/2,S)} c_i > C$, то выполняется второй этап.

Второй этап

Пусть R – минимальное число из r_j таких, что $D_p/2 < r_j < D_l/2$ и $\sum_{i \in T(r_j,S)} c_i \leq C$. Если такого R не существует, то полагается $\tilde{S} = S_l$. В противном случае полагается $\tilde{S} = T(R, S)$.

Заметим, что в конце работы первого этапа выполняется неравенство $D_p > D_l$. Докажем вспомогательную лемму для оценки трудоемкости алгоритма.

Лемма 9 *Пусть дано множество $S \subseteq V$ такое, что $\min_{i,j \in S} d_{ij} > D$. Тогда множество $T(D/2, S)$ можно вычислить за $O(n)$ действий.*

Доказательство. Так как $H(D/2, i) \cap H(D/2, j) = \emptyset$ при $j \neq i$ и $j, i \in S$, то множества $H(D/2, i)$ при $i \in S$ вычисляются за один просмотр множества V, т. е. за $O(n)$ действий. Множество $T(D/2, S)$ вычисляется не более, чем за $O(\sum_{i \in S} |H(D/2, i)|)$ действий, что не превосходит величины $O(n)$. Лемма доказана.

Число шагов на первом этапе не превосходит n. Используя лемму 9, получаем, что все вычисления, производимые на k-м шаге, выполняются не более, чем за $O(n)$ действий. Следовательно, общая трудоемкость первого этапа не превосходит величины $O(n^2)$. На втором этапе для каждого r_j такого, что $D_p/2 < r_j < D_l/2$, вычисляется $T(r_j, S)$. Так как число различных r_j не превосходит n, то трудоемкость алгоритма не превосходит величины $O(n^2)$.

6.2 Оценка относительной погрешности

Лемма 10 *Если для множества S выполняется условие $\min_{i,j\in S} d_{ij} > 2R^*$, то множество $T(R^*, S)$ является допустимым решением задачи (1).*

Доказательство. Для любого $i \in S$ найдется $k(i)\in S^*$ такой, что $d_{ik(i)} \leq R^*$. Причем для всех $j \in S \setminus \{i\}$ выполняется неравенство $d_{jk(i)} > R^*$, так как в противном случае из неравенства треугольника следует, что $d_{ij} \leq d_{ik(i)} + d_{k(i)j} \leq 2R^*$. Это противоречит условию леммы. Следовательно, $k(i) \neq k(j)$ для всех $i, j \in S, i \neq j$, и выполняются неравенства

$$\sum_{i\in T(R^*,S)} c_i = \sum_{i\in S} c_{v(R^*,i)} \leq \sum_{i\in S} c_{k(i)} \leq \sum_{i\in S^*} c_i \leq C.$$

Лемма доказана.

Теорема 12 *Алгоритм находит такое множество $\tilde{S}$, что $r(\tilde{S})/r(S^*) \leq 3$.*

Доказательство. Расмотрим три случая.

1. Пусть $D_l/2 \leq R^*$. Тогда если $\tilde{S} = S_l$, то из неравенства треугольника следует, что $r(\tilde{S}) \leq D_l/2 + D_l \leq 3R^*$. Если $\tilde{S} = T(R, S)$, где $D_p/2 < R < D_l/2$, то $r(\tilde{S}) \leq R + D_p < D_l/2 + D_l \leq 3R^*$.

2. Пусть $R^* \leq D_p/2$. Так как $\min_{i,j \in S} d_{ij} = D_l > D_p \geq 2R^*$, то по лемме 10 имеем

$$\sum_{i \in T(D_p/2, S)} c_i \leq \sum_{i \in T(R^*, S)} c_i \leq C.$$

Это противоречит условию перехода с первого этапа на второй.

3. Пусть $D_p/2 < R^* < D_l/2$. Так как по лемме 10 множество $T(R^*, S)$ является допустимым решением задачи (6.1), то число R, выбираемое на втором этапе алгоритма существует и удовлетворяет условиям: $D_p/2 < R \leq \max_{i \in S} d_{ik(i)} \leq R^*$ и $\sum_{i \in T(R,S)} c_i \leq C$. Из неравенства треугольника следует, что $r(\tilde{S}) = r(T(R, S)) \leq R^* + D_p < 3R^*$.
Теорема доказана.

Заключение

Сформулируем основные результаты проведенных исследований.

1. Для простейшей задачи размещения на максимум построен полиномиальный приближенный алгоритм с оценкой точности равной $2(\sqrt{2}-1)$. Доказано, что для этой задачи не существует полиномиальной аппроксимационной схемы при условии, что $P \neq NP$.

2. Изучены различные обобщения задачи о p-медиане на максимум. Для задачи о р-медиане на максимум с дополнительными ресурсными ограничениями получена константная оценка точности жадного алгоритма. Для динамической задачи о p-медиане на максимум построен приближенный алгоритм с оценкой точности, совпадающей с неулучшаемой оценкой точности жадного алгоритма для задачи о р-медиане.

3. Предложен новый способ округления решений линейных релаксаций целочисленных задач с ограничением на мощность. С использованием этой техники построен приближенный алгоритм для задачи о максимальном покрытии p множествами с оценкой точности лучше известных. Аналогичный результат получен для задачи о максимальном разрезе с ограничением на мощность. Изучены возможности применения этой техники в случае более общих задач.

4. Для задачи выполнимости на максимум с ограничением на мощность построен приближенный алгоритм с неулучшаемой оценкой точности.

5. Для обобщенной задачи о p-центре предложен приближенный алгоритм с известной оценкой точности, но с лучшей оценкой трудоемкости.

Список литературы

[1] А. А. Агеев, О сложности задач минимизации полиномов от булевых переменных, Управляемые системы **23** (1983), стр.3–11.

[2] А. А. Агеев, Полиномиальный алгоритм решения задачи размещения на последовательно-параллельной сети, Управляемые системы **30** (1990), стр.3–16.

[3] А. А. Агеев, В. Л. Береснев, Алгоритмы минимизации для некоторых классов полиномов от булевых переменных, Модели и методы оптимизации, Труды ИМ СО РАН **10** (1988), стр.5–17.

[4] И. Г. Белинская, Об одном классе полиномов от булевых переменных, Управляемые системы **21** (1981), стр.6–12.

[5] В. Л. Береснев, Об одном классе задач оптимизации параметров однородной технической системы, Управляемые системы **9** (1971), стр.65–74.

[6] В. Л. Береснев, Об одной задаче математической теории стандартизации, Управляемые системы **11** (1973), стр.43–54.

[7] В. Л. Береснев, Об одной задаче математической теории стандартизации, Управляемые системы **13** (1974), стр.3–9.

[8] В. Л. Береснев, Алгоритм неявного перебора для задачи типа размещения и стандартизации, Управляемые системы **12** (1974), стр.24–34.

[9] В. Л. Береснев, Алгоритмы минимизации полиномов от булевых переменных, Проблемы кибернетики **36** (1979), стр.225–246.

[10] В. Л. Береснев, Эффективный алгоритм для задачи размещения производства с вполне уравновешенной матрицей, Дискретный анализ и исследование операций **5(1)** серия 1 (1998), стр.20–31.

[11] В. Л. Береснев, Э. Х. Гимади, В. Т. Дементьев, Экстремальные задачи стандартизации, Наука, 1978.

[12] М. М. Беркович, Задачи стандартизации и некоторые методы их решения, Экономика и математические методы **5(2)** (1969), стр.285–289.

[13] Г. В. Генс, Е. В. Левнер, Приближенные алгоритмы для некоторых универсальных задач теории расписаний, Известия АН СССР, Техническая кибернетика **6** (1978), стр.38–43.

[14] Э. Х. Гимадутдинов, О свойствах решений одной задачи оптимального размещения точек на отрезке, Управляемые системы **2** (1969), стр.77–91.

[15] Э. Х. Гимади, Выбор оптимальных шкал в одном классе задач типа размещения, унификации и стандартизации, Управляемые системы **6** (1970), стр.57–70.

[16] Э. Х. Гимади, Эффективный алгоритм решения задачи размещения с областями обслуживания связными относительно ациклической сети, Управлямые системы **23** (1983), стр.12–23.

[17] Э. Х. Гимади, Задача размещения на цепи с центрально-связными областями обслуживания, Управлямые системы **25** (1984), стр.38–47.

[18] Э. Х. Гимади, Обоснование априорных оценок качества приближенного решения задачи стандартизации, Управляемые системы **27** (1987), стр.3–11.

[19] Э. Х. Гимади, О некоторых математических моделях и методах планирования крупномасшабных проектов, Модели и Методы Оптимизации, Труды ИМ СО АН СССР **10** (1988), стр.89–115.

[20] Э. Х. Гимади, Задача стандартизации с данными произвольного знака и связными, квазивыпуклыми и почти квазивыпуклыми матрицами, Управляемые системы **30** (1990), стр.12–23.

[21] Э. Х. Гимади, Эффективные алгоритмы для решения многоэтапной задачи размещения на цепи, Дискретный анализ и исследование операций **2(4)** (1995), стр.13–31.

[22] Э. Х. Гимади, Н. И. Глебов, В. Т. Дементьев, Об одном методе построения нижней оценки и приближенного решения с апостериорной оценкой точности для задачи стандартизации, Управляемые системы **13** (1974), стр.26–31.

[23] Э. Х. Гимади, Н. И. Глебов, В. А. Перепелица, Алгоритмы с оценками для задач дискретной оптимизации, Проблемы кибернетики **31** (1975), стр.35–42.

[24] Э. Х. Гимади, В. Т. Дементьев, О методах решения некоторых задач оптимизации параметрических рядов, Стандарты и качество **12** (1971), стр.10–12.

[25] Э. Х. Гимади, В. Т. Дементьев, Некоторые задачи выбора оптимальных параметрических рядов и методы их решения (задачи стандартизации), Проблемы кибернетики **27** (1973), стр.19–32.

[26] В. Т. Дементьев, Задача выбора типажа оборудования, Методы управления большими системами, (1970), стр.60–62, Иркутск: СЭИ СО АН СССР.

[27] В. П. Гришухин, Полиномиальность в простейшей задаче размещения, препринт ЦЭМИ.

[28] М. Гэри, Д. Джонсон, Вычислительные машины и труднорешаемые задачи, Москва, Мир, 1979.

[29] В. А. Перепелица, Э. Х. Гимади, К задаче нахождения минимального гамильтонова контура на графе со взвешенными дугами, Дискретный анализ **15** (1969), стр.57–65.

[30] В. А. Трубин, Эффективный алгоритм размещения на сети в форме дерева, Доклады АН СССР **231(3)** (1976), стр.547–550.

[31] В. А. Трубин, Два класса задач размещения на древовидных сетях, Кибернетика **4** (1983), стр.84–87.

[32] Л. Г. Хачиян, Полиномиальный алгоритм в линейном програмировании, Доклады АН СССР **т.244** (1979), стр.1093–1096.

[33] В. П. Черенин, В. Р. Хачатуров, Решение методом последовательных расчетов одного класса задач о размещении производства, Экономика и математические методы **2** (1965), стр.279–290.

[34] A. A. Ageev, A Criterion of Polynomial-Time Solvability for the Network Location Problem, Integer Programming and Combinatorial Optimization (1992), Carnegie-Mellon University, pp.237–245.

[35] A. A. Ageev, Complexity of the Network Median Problem on Planar Grids, Siberian Advances in Mathematics **5** (1995), pp.1–9.

[36] S. Ahn, C. Cooper, G. Cornuejols and A. M. Frieze, Probabilistic Analysis of a Relaxation for the k-Median Problem, Mathematics of Operation Research **13** (1988), pp.1–31.

[37] N. Alon and J. N. Spencer, The Probabilistic Method, John Wiley and Sons, 1992.

[38] S. Arora, Probabilistic Checking of Proofs and the Hardness of Approximation Problems, PhD thesis, U.C. Berkeley, 1994.

[39] S. Arora, Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems, In Proceedings of the 37th IEEE FOCS (1996), pp.2–11.

[40] S. Arora, Nearly Linear Time Approximation Schemes for Euclidean TSP and Other Geometric Problems, In Proceedings of the 38th IEEE FOCS (1997), pp.554–563.

[41] S. Arora, A. M. Frieze and H. Kaplan, A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangement Problems, In Proceedings of the 37th IEEE FOCS (1996),

[42] S. Arora, D. Karger and M. Karpinski, Polynomial Time Approximation Schemes for Dense Instances of NP-Hard Problems, In Proceedings of the 27th Annual ACM Symposium on Theory of Computing (1995), pp.21–30.

[43] S. Arora, M. Grigni, D. Karger, P. Klein and A. Woloszyn, Polynomial Time Approximation Scheme for Weighted Planar Graph TSP, In Proceedings of the SODA97.

[44] S. Arora, P. Raghavan and S. Rao, Approximation Schemes for Euclidean k-medians and related problems, In Proceedings of the 30th STOC (1998).

[45] S. Arora and C. Lund, Hardness of Approximation, in the book "Approximation Algorithms for NP-hard Problems", PWS Publishing (1996).

[46] S. Arora, C. Lund, R. Motwani, M. Sudan and M. Szegedy, Proof Verification and Intractability of Approximation Problems, In Proceedings of the 33rd IEEE Symp. on Foundations of Computer Science (1992), pp.13–22.

[47] T. Asano, Approximation Algorithms for MAX SAT: Yannakakis vs. Goemans-Williamson, In Proceedings of the 5nd Israel Symposium on Theory and Computing Systems (1997), pp.182–189.

[48] Y. Bartal, Probabilistic Approximation of Metric Spaces and Its Algorithmc Applications, In Proceedings of the 37th IEE FOCS (1996), pp.184–193.

[49] Y. Bartal, On Approximating Arbitrary Metric by Tree Metric, In Proceedings of the 30th STOC (1998).

[50] B. S. Baker, Approximation algorithms for NP-complete problems on planar graphs, Jouranl of ACM **41(1)**,1994, Preliminary version in Proceedings of the 24th Annual Symposium on Foundations of Computer Science (1983), pp.265–273.

[51] J. Bar-Ilan, G. Kortsarz and D. Peleg, How to Allocate Network Centers, Journal of Algorithms **15** (1993), pp.385–415.

[52] R. Bar-Yehuda, One for the Price of Two: a Unified Approach for Approximating Covering Problems, In Proceedings of APPROX98.

[53] R. Bhatia, S. Guha, S. Khuller and Y. J. Sussman, Facility Location with Dynamic Distance Function, In Proceeding of 6th Scandinavian Workshop on Algorithmic Theory (SWAT) (1998).

[54] R. E. Burkard, E. Cela, P. M. Pardalos and L.S. Pitsoulis, Quadratic Assignment Problem, technical report of Graz University of Technology, SFB-Report 126, (1998).

[55] R. E. Burkard, E. Cela, V. Demidenko, N. Metelski and G. J. Woeginger, Perpespectives of Easy and Hard Cases of Quadratic Assignment Problem, technical report of Graz University of Technology, SFB-Report 104, (1997).

[56] R. Chandrasekaran and A. Tamir, A Polynomially Bounded Algorithms for Locating p-Centers on a Tree, Mathematical Programming **22** (1982), pp.304–315.

[57] M. Charikar, C. Chekuri, A. Goel and S. Guha, Rounding via Trees: Determenistic Approximation Algorithms for Group Steiner Trees and k-Median, In Proceedings of 30th STOC.

[58] V. Chvatal, A Greedy Heuristic for the Set Covering Problem, Mathematics of Operation Research **4** (1979), pp.233–235.

[59] F. Chudak and D. Shmoys, Improved Approximation Algorithm for Uncapacitated Facility Location, unpublished manuscript.

[60] M. Conforti and G. Cornuejols, Submodular Set Function, Matroids and the Greedy Algorithm: Worst-Case Bounds and Some Generalizations of the Rado-Edmonds Theorem, Discrete Applied Mathematics **7** (1984), pp.251–274.

[61] G. Cornuejols, M. L. Fisher and G. L. Nemhauser, Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms, Management Science **22** (1977), pp.789–810.

[62] G. Cornuejols, G. L. Nemhauser and L. A. Wolsey The Uncapacitated Facility Location Problem, In book edited by P. B. Mirchandani and R. L. Francis, Wiley&Sons, 1990.

[63] G. Cornuejols, G. L. Nemhauser and L. A. Wolsey Worst Case and Probalistic Analysis of Algorithms for Location Problems, Operation Research **28** (1980), pp.847–858.

[64] M. E. Dyer, An $O(n)$ Algorithm for the Multiple Choice Knapsack Problem, Mathematical Programming **29** (1984), pp.57–63.

[65] M. E. Dyer and A. M. Frieze, A Simple Heuristic for the p-Center Problem, Operation Research Letters **3** (1985), pp.285–288.

[66] D. Erlenkotter, A Dual-Based Approach for Uncapacitated Facility Location, Operation Research **26** (1978), pp.992–1009.

[67] T. Feder and D. H. Green, Optimal Algorithms for Approximate Clustering, In Proceedings of 20th ACM Symp. on Theory of Computing (1988), pp.434–444.

[68] U. Feige, A Threshold of $\ln n$ for Approximating Set Cover, Journal of ACM (to appear).

[69] U. Feige and M. Goemans, Approximating the Value of Two-prover Proof Systems, with Applications to MAX 2-SAT and MAX-DICUT. In Proceedings of the 3nd Israel Symposium on Theory and Computing Systems (1995), pp.182–189.

[70] U. Feige, S. Goldwasser, L. Lovasz, S. Safra and M. Szegedy, Approximating Clique is Almost NP-Complete, In Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science (1991), pp.2–12.

[71] W. Feller, An Introduction to Probability Theory and Its Applications, John Wiley & Sons New York, 1968.

[72] M. L. Fisher, Worst-Case Analysis of Algorithms, Management Science **26** (1980), pp.1–18.

[73] M. L. Fisher and D. Hochbaum, Probabilistic Analysis of the Planar K-Median Problem, Mathematics of Operation Research **5** (1980), pp.27–34.

[74] M. L. Fisher , G. L. Nemhauser and L. A. Wolsey, An Analysis of Approximations for Maximizing Submodular Set Functions-II, Mathematical Programming Study **8** (1978), pp.73–88.

[75] A. Frieze and M. Jerrum, Improved Approximation Algorithms for MAX k-CUT and MAX BISECTION, Algorithmica **18** (1997), pp.67–81.

[76] A. Frieze and R. Kannan, The Regularity Lemma and Approximation Schemes for Dense Problems, In Proceedings of the 37th IEEE FOCS (1996), pp.12–30.

[77] M. Goemans and D. Williamson, New 3/4-Approximation Algorithms for the Maximum Satisfiability Problem, SIAM Journal on Discrete Mathematics **7** (1994), pp.656–666.

[78] M. Goemans and D. Williamson, Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming, Journal of ACM **42** (1995), pp.1115–1145.

[79] M. Goemans and D. Williamson, The Primal-dual Method for Approximation Algorithms and Its Application to Network Design Problems, in the book "Approximation algorithms for NP-hard problems", PWS Publishing, (1996).

[80] R. L. Graham, Bounds for Certain Multiprocessing Anomalies, Bell System Technical Journal **45** (1966), pp.1563–1581.

[81] M. Grigni, E. Koutsoupias and C. Papadimitrou, An Approximation Scheme for Planar Graph TSP, In Proceedings of 36th FOCS (1995), pp.640–645.

[82] S. Guha and S. Khuller, Greedy Strikes Back: Improved Facility Location Algorithms, In Proceedings of the 9th SODA (1998).

[83] G. Y. Handler, p-Center Problems, in the book "Discrete Location Theory", edited by P. B. Mirchandani and R. L. Francis (1990).

[84] J. Hastad, Some Optimal Inapproximability Results, In Proceedings of the 28 Annual ACM Symp. on Theory of Computing (1996), pp.1–10.

[85] D. S. Hochbaum, Heuristics for the Fixed Cost Median Problems, Mathematical Programing **22** (1982), pp.148-162.

[86] D. S. Hochbaum, Approximation Algorithms for the Set Covering and Vertex Cover Problems, SIAM Journal of Computing **11** (1982), pp.555–556.

[87] D. S. Hochbaum, editor, Approximation Algorithms for NP-Hard Problems, PWS Publishing (1996).

[88] D. S. Hochbaum, Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems, in the book "Approximation algorithms for NP-hard problems", edited by D. S. Hochbaum, PWS Publishing Company, 1996.

[89] D. S. Hochbaum and A. Pathria, Generalized p-Center Problems: Complexity Results and Approximation Algorithms, European Journal of Operational Research **100** (1997), pp.594–607.

[90] D. S. Hochbaum and A. Pathria, Locating Centers in Dynamically Changing Network and Related Problems, to appear in Location Science.

[91] D. S. Hochbaum and D. B. Shmoys, A Best Possible Heuristic for the k-center Problem, Mathematics of Operation Research **10** (1985), pp.180–184.

[92] D. S. Hochbaum and D. B. Shmoys, A Unified Approach to Approximation Algorithms for Bottleneck Problems, Journal of ACM, **33** (1986), pp.533–550.

[93] W. L. Hsu and G. L. Nemhauser, Easy and Hard Bottleneck Location Problems, Discrete Applied Mathematics **1** (1979), pp.209–216.

[94] D. S. Johnson, Approximation Algorithms for Combinatorial Problems, Journal Comput. System Sci. **9** (1974), pp.256–278.

[95] D. S. Johnson and M. S. Trick, editors, Cliques, Colorings and Satisfiability, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **26**.

[96] O. Kariv and S. L. Hakimi, An Algorithmic Approach to Network Location Problems. II. The p-Medians., SIAM Journal of Applied Mathematics **37** (1979), pp.539–560.

[97] H. Karloff and U. Zwick, A 7/8-Approximation Algorithm for MAX 3SAT?, In Proceedings of the 38th FOCS (1997), pp.406–415.

[98] N. Karmarkar, A New Polinomial-Time Algorithm for Linear Programming, Combinatorica **4** (1984), pp.373–395.

[99] S. Khanna, V. Kann, J. Lagergren and A. Panconesi, On the Hardness of Approximating MAX k-CUT and Its Dual, Chicago Journal of Theoretical Computer Science **2** (1997).

[100] S. Khanna, R. Motwani and M. Sudan, Towards Syntactic Characterization of PTAS, In Proceedings of the 28th Annual ACM Symp. on Theory of Computing (1996), pp.329–337.

[101] S. Khanna, M. Sudan and D. Williamson, A Complete Classification of the Approximability of Maximization problems Derived from Boolean Constraint Satisfaction, In Proceedings of the 29th STOC, (1997), pp.11–20.

[102] S. Khuller, A. Moss and J. Naor, The Budgeted Maximum Coverage Problem, submitted for publication.

[103] S. Khuller, R. Pless and Y. J. Sussman, Fault Tolerant K-center Problems, In Proceedings of the 3rd Italian Conference on Algorithms and Complexity (1997), LNCS 1203, pp.37–48.

[104] S. Khuller and Y. J. Sussman, Capacitated K-center Problems, In Proceedings of 4th Annual Europeap Symposium on Algorithms (ESA) (1996), LNCS 1136, pp.152–166.

[105] T. Leighton and S. Rao, An Approximate Max-flow Min-cut Theorem for Uniform Multicommodity Flow Problems with Applications to Approximation Algorithms, In Proceedings of the 24th Annual Symposium on Foundations of Computer Science (1988), pp.422–431.

[106] J. H, Lin and J. S. Vitter, ε-Approximation with Minimum Packing Constraint Violation, In Proceedings of the 24th STOC (1992), pp.771–782.

[107] J. H. Lin and J. S. Vitter, Approximation Algorithms for Geometric Median Problems, Information Processing Letters **44** (1992), pp.245–249.

[108] L. Lovasz, On the Shannon Capacity of a Graph, IEEE Trans. of Information Theory **25(1)** (1979), pp.1–7.

[109] C. Lund and M. Yannakakis, On the Hardness of Approximating Minimization Problems, Journal of the ACM **41** (1994), pp.960–981.

[110] P. B. Mirchandani, The p-Median Problem and Generalizations, in the book "Discrete Location Theory", edited by P. B. Mirchandani and R. L. Francis (1990).

[111] R. Motwani , J. Naor and P. Raghavan , Randomized Algorithms in Combinatorial Optimization, in the book "Approximation algorithms for NP-hard problems", PWS Publishing Company, 1996.

[112] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press (1995).

[113] G. L. Nemhauser and L. A. Wolsey, Integer and Combinatorial Optimization, John Wiley & Sons, 1988.

[114] G.L.Nemhauser and L.A.Wolsey, Best Algorithms for Approximating the Maximum of a Submodular Function, Mathematics of Operation Research **3** (1978), pp.177–188.

[115] G. L. Nemhauser, L. A. Wolsey and M. L. Fisher, An Analysis of Approximations for Maximizing Submodular Set Functions-I, Mathematical Programming **14** (1978), pp.265–294.

[116] C. Papadimitrou, Worst-Case and Probabilistic Analysis of a Geometric Location Problem, SIAM Journal of Computing 10 (1981), pp.542–557.

[117] C. Papadimitrou and M. Yannakakis, Optimization, Approximation and Complexity Classes, Journal of Computer and System Sciences **43** (1991), pp.425–440.

[118] M. Queyranne, Performance Ratio of Polynomial Heuristics for Triangle Inequality Quadratic Assignment Problems, Operation Research Letters **4** (1986), pp. 231–234.

[119]

P.Raghavan, Probabilistic Construction of Determenistic Algorithms:

Approximating Packing Integer Programs, Journal of Computer and System Sciences **37** (1988), pp.130–143.

[120] P. Raghavan and C. Thompson, Randomized Rounding: a Technique for Provably Good Algorithms and Algorithmic Proofs, Combinatorica **7** (1987), pp.365–374.

[121] R. Raz, A Parallel Repetition Theorem, Journal of ACM (to appear).

[122] A. H. G. Rinnoy Kan, An Introduction to the Analysis of Aproximation Algorithms, Discrete Applied Mathematics **14** (1986), pp.171–186.

[123] S. Sahni, General Technique for Combinatorial Approximation, Operation Research **25** (1977), pp.920–936.

[124] S Sahni and T. Gonzalez, P-Complete Approximation Problems, Journal of ACM **23** (1976), pp.555–565.

[125] D. Shmoys, Cut Problems and Their Application to Divide-and-Conquer, in the "Approximation Algorithms for NP-hard problems", PWS Publishing (1996).

[126] D. Shmoys, Computing Near-Optimal Solutions to Combinatorial Optimization Problems, In W. Cook, L. Lovasz, P. Seymour, editors, Advances in Combinatorial Optimization, AMS, Providence RI (1995), pp.355–397.

[127] D. Shmoys, E. Tardos and K. Aardal, Approximation Algorithms for Facility Location Problems, In Proceedings of the 29 Annual ACM Symp. on Theory of Computing (1997).

[128] P. Slavik, A Tight Analysis of the Greedy Algorithm for Set Cover, In Proceedings of the 28th Annual ACM Symposium on Theory of Computing (1996), pp.435–439.

[129] B. C. Tansel, R. L. Francis and T. J. Lowe, Duality: Covering and Constraining p-Center Problems on Trees, in the book "Discrete Location Theory", edited by P. B. Mirchandani and R. L. Francis (1990).

[130] S. Vishwanathan, An $O(\log^* n)$ Approximation Algorithm for the Assymetric p-Center Problem, In Proceedings of the 7th SODA (1996). pp.1–5.

[131] L. A. Wolsey, An Analysis of the Greedy Algorithm for the Submodular Set Covering Problems, Mathematics of Operation Research **7** (1982), pp.417–425.

[132] L. A. Wolsey, Maximizing Real-valued Submodular Functions: Primal and Dual Heuristics for Location Problems, Mathematics of Operation Research **7** (1982), pp.410–425.

[133] M.Yannakakis, On the Approximation of Maximum Satisfiability, Journal of Algorithms **17** (1994), pp.475–502.

[134] E.Zemel, An $O(n)$ Algorithm for the Linear Multiple Choice Knapsack Problem and Related Problems, Information Processing Letters **18** (1984), pp.123–128.

Список публикаций автора по теме диссертации

1. М. И. Свириденко, О точности решений жадными алгоритмами задач размещения на максимум, Дискретный анализ и Исследование операций **4(3)** серия 1 (1997), стр.35–48.

2. М. И. Свириденко, Приближенный алгоритм для решения динамической задачи о p-медиане на максимум, Дискретный анализ и Исследование операций **4(2)** серия 2 (1997), стр.55–62.

3. А. А. Агеев и М. И. Свириденко, Приближенный алгоритм с априорной оценкой точности для простейшей задачи размещения на максимум, Тезисы конференции "Приблемы оптимизации и экономические приложения", Омск 1997.

4. М. И. Свириденко, Приближенный алгоритм для решения задачи о p-центре с неравенством треугольника, Дискретный анализ и Исследование операций **5(1)** серия 1 (1998), стр.60–63.

5. A. A. Ageev and M. I. Sviridenko, An Approximation Algorithm for the Uncapacitated Facility Location Problem, в тезисах 16-го международного симпозиума по математическому программированию, Лозанна

1997.

6. A. A. Ageev and M. I. Sviridenko, An Approximation Algorithm for the Uncapacitated Facility Location Problem, в тезисах симпозиума по исследованию операций, Йенна 1997.

7. A.A. Ageev and M. I. Sviridenko, An Approximation Algorithms for Some Combinatorial Problems with Cardinality Constraints, В трудах 11-й Байкальской школы-семинара Методы оптимизации и их приложения **т.1** (1998), стр.107–110.

8. M. I. Sviridenko, Best Possible Approximation Algorithm for MAX SAT with Cardinality Constraint, In Proceedings APPROX98, Lecture Notes in Computer Science **v.1444**, pp.193–199.