

ГЛАВА 3

ПОЛИНОМИАЛЬНО РАЗРЕШИМЫЕ СЛУЧАИ ЗАДАЧИ РАЗМЕЩЕНИЯ СРЕДСТВ ОБСЛУЖИВАНИЯ

Задача размещения средств обслуживания, как уже неоднократно отмечалось, относится к числу труднорешаемых задач, для которых построение точных полиномиальных алгоритмов представляется делом весьма проблематичным. Одним из важных направлений исследования труднорешаемых задач является, как известно, рассмотрение частных случаев задач, что позволяет сузить множество исследуемых конкретных задач и, используя дополнительные свойства задач рассматриваемых подклассов, построить для их решения точные полиномиальные алгоритмы.

Частные случаи экстремальных задач выделяются, как правило, посредством наложения дополнительных условий на исходные данные. В случае задачи размещения производства такие условия накладываются на матрицу транспортных затрат или ее характеристическую матрицу. При этом следует подчеркнуть, что поскольку задача размещения производства рассматривается еще и как математическая модель принятия решений по размещению и использованию различных технических средств обслуживания, то важно, чтобы накладываемые на матрицу транспортных затрат дополнительные условия были бы понятны и реальны с практической точки зрения.

В настоящей главе рассматриваются два вида условий на матрицу транспортных затрат, обладающих полезными качествами, использование которых позволяет построить эффективные алгоритмы решения задачи размещения средств обслуживания. Эти условия приводят к матрицам, обладающим гриди свойством, и матрицам, обладающим свойством связности. Обзор некоторых результатов, полученных при рассмотрении матриц с указанными свойствами, содержится в монографии [18] и работе [36], в которой, в частности, указано на эквивалентность различных определений гриди свойства.

Первыми и наиболее простыми представителями класса матриц, обладающих гриди свойством, являются так называемые правильные матрицы. Характеристическая матрица правильной матрицы устроена достаточно просто и в каждом столбце такой матрицы нулевые строки расположены подряд. На такие матрицы впервые внимание было обращено в работе [16], а в монографии [18] они были достаточно подробно изучены. В качестве наиболее простого алгоритма решения задачи размещения с правильной матрицей в [30, 29, 18] рассмотрен алгоритм динамического программирования. Этот алгоритм применяется к так называемой задаче о ближайшем соседе, к которой сводится задача размещения средств обслуживания с правильной матрицей транспортных затрат. Другой подход к построению алгоритма для задачи размещения с правильной матрицей изложен в [16, 22] и связан с представлением такой задачи в виде задачи минимизации полиномов от булевых переменных, которые также названы правильными. Особенность правильных полиномов такова, что для решения задачи их минимизации можно построить эффективный алгоритм на базе метода псевдобулева программирования. Такое становится возможным потому, что удается в явном и достаточно простом виде построить функцию, осуществляющую сведение задачи минимизации исходного правильного полинома к задаче минимизации

аналогичного полинома, но с числом переменных на единицу меньше. Следует отметить также, что этот же подход к построению эффективного алгоритма минимизации полиномов удастся реализовать [112] и в более общей ситуации, когда характеристическая матрица у полинома правильная, а коэффициенты, в отличие от правильного полинома, могут иметь произвольные знаки.

Интересным обобщением правильных матриц являются так называемые почти правильные матрицы. Отличие состоит в том, что в случае почти правильной матрицы в каждом столбце характеристической матрицы подряд расположены либо нулевые, либо единичные строки столбца. Другими словами, если характеристическая матрица правильной матрицы реализуется как семейство подцепей некоторой цепи, то характеристическая матрица почти правильной — как семейство подцепей некоторого цикла. Задача минимизации почти правильных полиномов не является существенно более сложной по сравнению со случаем правильных полиномов и сводится к задаче минимизации таких полиномов. Достаточно полное исследование задачи минимизации почти правильных полиномов приведено в работе [9].

Другое обобщение правильных матриц рассмотрено в [4]. Характеристические матрицы для матриц этого класса представляются как характеристические матрицы семейства ориентированных цепей некоторого ориентированного дерева. Задача минимизации полиномов с характеристическими матрицами, обладающими указанным свойством, не сводится к задаче минимизации правильного полинома, однако идеи, заложенные в построение упомянутого выше алгоритма, могут быть перенесены на случай и таких полиномов.

Существенным расширением класса правильных матриц являются так называемые вполне уравновешенные матрицы. Широкую известность такие матрицы получили благодаря работе [116'], в которой введены такие важные понятия как гриди матрица и гриди матрица в стандартной форме, доказана теорема о возможности приведения произвольной $(0,1)$ -матрицы к дважды лексически упорядоченной матрице и тем самым поставлен знак равенства между вполне уравновешенными и гриди матрицами. В этой работе предложен также алгоритм решения задачи размещения средств обслуживания, представленный в виде задачи о покрытии множествами с матрицей ограничений, являющейся гриди матрицей. Рассматриваются релаксированная задача о покрытии множествами и двойственная к ней, известная под названием задачи об упаковке. Показано, что такая пара двойственных задач обладает тем замечательным свойством, что, во-первых, задача об упаковке может быть решена нетрудоемким алгоритмом, названным гриди алгоритмом, и, во-вторых, по решению задачи об упаковке достаточно просто строится оптимальное решение прямой задачи. Это оптимальное решение является целочисленным и, следовательно, в результате работы гриди алгоритма получается оптимальное решение релаксированной задачи о покрытии, которое одновременно является оптимальным решением исходной задачи о покрытии множествами.

Отмечая важность работы [116'] для исследования вполне уравновешенных матриц, следует, тем не менее, заметить, что впервые гриди алгоритм для указанной пары двойственных задач был построен В.А. Трубиным в работе [64]. В этой работе не использовались понятия вполне уравновешенной и гриди матрицы, а рассматривалось эквива-

лентное определение гриди свойства через так называемые Q -матрицы. Было показано, что если матрица ограничений задачи об упаковке является Q -матрицей, то алгоритмом, практически ничем не отличающимся от упомянутого выше гриди алгоритма из [116'], можно получить оптимальное решение данной задачи, которое порождает оптимальное решение задачи размещения средств обслуживания.

Строение вполне уравновешенных матриц с использованием различной терминологии изучалось в ряде работ (см, например, [74', 93', 125]), из которых следует, что всякая такая матрица может рассматриваться, как характеристическая матрица семейства поддеревьев некоторого дерева. С учетом этого обстоятельства и того факта, что вполне уравновешенные матрицы привлекли к себе внимание при изучении задачи размещения на древовидных сетях, основные примеры порождения таких матриц связаны с деревьями. В [97'] введено понятие поддеревьев соседства и показано, что характеристическая матрица семейства поддеревьев соседства является вполне уравновешенной. Следует отметить, что практически одновременно и независимо В.А. Трубин в работе [62], а позднее в монографии [51'] дал два примера семейств поддеревьев, порождающих Q -матрицы. Одним из этих примеров являются поддеревья соседства. Пример вполне уравновешенной матрицы, не являющейся характеристической матрицей семейства поддеревьев соседства, приведен в работе [82']. Как обобщение класса вполне уравновешенных матриц, порождаемых семействами поддеревьев соседства, можно рассматривать предложенный в [151'] класс вполне уравновешенных матриц, образованный матрицами смежности двух семейств поддеревьев соседства.

Хотя вполне уравновешенные матрицы образуют существенно более широкий класс, чем правильные матрицы, они сохраняют основное полезное свойство правильных матриц, позволяющее при решении задачи минимизации правильного полинома эффективно использовать метод псевдобулевого программирования. Первые замечания о возможности распространения алгоритма минимизации правильных полиномов на случай вполне уравновешенных полиномов содержатся в уже упомянутом обзоре [36]. Подробное описание алгоритма, основанного на сведении задачи минимизации исходного вполне уравновешенного полинома к задаче минимизации аналогичного полинома, но с числом переменных на единицу меньше, дано в [23]. Более того, по аналогии со случаем правильного полинома в [23] показано, что некоторая модификация данного алгоритма может быть использована для решения задачи минимизации полиномов аналогичных вполне уравновешенным, но с коэффициентами произвольных знаков.

Свойство связности матрицы транспортных затрат — еще один важный класс свойств, использование которых позволяет эффективно решать задачу размещения. При этом, если гриди свойство накладывает дополнительные требования на пары столбцов характеристической матрицы, то свойство связности требует выполнения некоторых дополнительных условий для всякой пары строк матрицы транспортных затрат или ее характеристической матрицы. И хотя с некоторой точки зрения эти требования на матрицу можно считать достаточно жесткими, тем не менее, матрицы с такими свойствами отражают реальные величины затрат на удовлетворение спроса потребителей при их размещении, например, вдоль некоторой магистрали.

Впервые понятие связности для матрицы транспортных затрат было введено Э.Х. Гимади в его работе [24']. В этой работе и работе [24] был развит подход к исследованию задачи размещения с матрицей, обладающей свойством связности, с использованием динамического программирования [10] и, в частности, задачи о ближайшем соседе [10]. Значительное внимание исследованию свойства связности матриц и построению эффективных алгоритмов, использующих это свойство, сделано в работах [29, 30], в том числе в монографии [18].

Обобщением свойства связности является свойство p -связности, $p = 1, 2, 3, \dots$, введенное в работе [12]. Это свойство при $p = 1$ совпадает со свойством связности, а при $p = 2$ позволяет существенно расширить класс матриц транспортных затрат, с которыми задача размещения является полиномиально разрешимой. Эффективные алгоритмы решения задачи размещения в случае p -связных матриц при $p \leq 2$ были построены в [12, 6] с использованием представления задачи размещения в виде задачи минимизации полиномов от булевых переменных. Соответствующие полиномы обладают тем свойством, что их характеристические матрицы являются p -связными. К сожалению, при $p = 2$ свойство p -связности исчерпывает свои возможности для построения эффективных алгоритмов. Задача размещения с 3-связной матрицей транспортных затрат становится труднорешаемой [4].

Обобщающим понятием связности является связность матрицы относительно графов. Первое определение матрицы, связной относительно ациклической сети, дано Э.Х. Гимади в [26] и использовано для построения на основе динамического программирования эффективного алгоритма решения задачи размещения с матрицей транспортных затрат, связной относительно дерева. Основные результаты в исследовании свойств матриц, связных относительно графов, получены А.А. Агеевым в работе [1]. Им показано, что матрица, связная относительно внешнепланарного графа, приводима к матрице со свойством 2-связности и, следовательно, задача размещения с матрицей транспортных затрат, связной относительно внешнепланарного графа, полиномиально разрешима. Показано также, что такое естественное расширение класса матриц, связных относительно внешнепланарных графов, до класса матриц, связных относительно планарных графов, делает задачу труднорешаемой. Таким образом, хотя введение понятия связности относительно графов не привело к открытию новых свойств матрицы транспортных затрат, позволяющих строить эффективные алгоритмы для задачи размещения, тем не менее это понятие позволило дать наглядную геометрическую интерпретацию свойства 2-связности и получить достаточное условие приведения матрицы к 2-связной.

Особенно полезно понятие связности относительно графов для установления возможности эффективного разрешения задачи размещения на сети. Поскольку матрица, связная относительно внешнепланарного графа, приводима к 2-связной матрице, то задача размещения на сети в виде внешнепланарного графа и, в частности, в виде дерева будет эффективно разрешимой. Важными и интересными являются результаты об эффективной разрешимости задачи размещения на сети, не являющейся внешнепланарным графом. Здесь следует отметить работы [5,], показывающие, что полиномиально разрешимой является задача размещения на последовательно-параллельных сетях.

Наличие у матрицы транспортных затрат гриди свойства или свойства связности зависит соответственно от порядка строк и от порядка столбцов матрицы. При одном порядке матрица может не обладать полезными свойствами, а при другом — такие свойства могут иметь место. В этой связи возникает вопрос о построении эффективных алгоритмов распознавания полезных свойств, то есть алгоритмов приведения (путем перестановки строк и столбцов) сходной матрицы к матрице с требуемым свойством. Такие алгоритмы должны либо строить необходимую перестановку, либо устанавливать, что нужной перестановки не существует и матрица не приводима к матрице с нужным свойством. Алгоритмы приведения произвольной $(0,1)$ -матрицы к правильной и почти правильной построены в [9]. Алгоритм распознавания гриди свойства $(0,1)$ -матрицы легко может быть построен на основе результатов, полученных в [116']. Эффективный алгоритм, приводящий произвольную $(0,1)$ -матрицу к 1-связной, либо устанавливающий, что это не возможно, предложен в [20]. В случае свойства 2-связности вопрос о построении аналогичного алгоритма остается открытым.

Настоящая глава состоит из семи параграфов. В первом рассматриваются правильные и почти правильные матрицы транспортных затрат и исследуется задача размещения с такими матрицами. Второй параграф посвящен описанию алгоритмов распознавания правильных и почти правильных $(0,1)$ -матриц. В § 3 рассматривается класс вполне уравновешенных $(0,1)$ -матриц и исследуется задача о покрытии множествами и задача минимизации полиномов от $(0,1)$ -переменных, соответствующие задаче размещения в случае, когда матрица ограничений задачи о покрытии и характеристическая матрица полинома являются вполне уравновешенными. § 4 посвящен исследованию задачи размещения в случае, когда матрица транспортных затрат является p -связной, а в § 5 приводится алгоритм распознавания свойства 1-связности у произвольной $(0,1)$ -матрицы. В § 6 приводятся основные результаты, касающиеся связности относительно графов. В заключительном § 7 рассматриваются эффективно разрешимые случаи задачи размещения на сети.

1 Правильные и почти правильные матрицы и полиномы

Начиная с этого параграфа, рассмотрим условия, налагаемые на матрицу транспортных затрат C , использование которых позволяет построить для решения задачи FL эффективные алгоритмы. Первыми такими матрицами, обладающими полезными свойствами, являются так называемые правильные и почти правильные матрицы. Эти матрицы имеют достаточно простое строение, но вместе с тем отражают часто встречающиеся на практике свойства транспортных затрат. Действительно, предположим, что потребители и места возможного открытия предприятий расположены вдоль некоторой магистрали. Тогда для каждого потребителя в любую сторону по магистрали от места его размещения транспортные затраты будут тем больше, чем дальше по магистрали расположено предприятие. Это свойство транспортных затрат фактически и отражают правильные матрицы.

Аналогичное свойство описывают и почти правильные матрицы для случая не прямой, а замкнутой магистрали.

1.1. Свойства квазивыпуклости и квазивогнутости

Вектор-столбец (c_i) ($i \in I$) назовем *квазивыпуклым*, если для любых $i, k, l \in I$, $I < k < l$, выполняется неравенство

$$c_k \leq \max \{c_i, c_l\}$$

и назовем *квазивогнутым*, если выполняется неравенство

$$c_k \geq \min \{c_i, c_l\}.$$

Матрицу $C = (c_{ij})$ ($i \in I, j \in J$) назовем *правильной*, если все ее столбцы являются квазивыпуклыми, и назовем *почти правильной*, если ее столбцы либо квазивыпуклые, либо квазивогнутые.

Понятно, что свойство матрицы быть правильной (почти правильной) зависит от нумерации строк матрицы. При одной нумерации матрица будет обладать этим свойством, а при другой — нет. Будем говорить, что матрица C *приводима* перестановкой строк к правильной (почти правильной) матрице, если существует перестановка строк, при которой полученная матрица C' становится правильной (почти правильной). Вопрос об отыскании такой перестановки представляет безусловный интерес и к его обсуждению ниже мы еще вернемся.

Всякий $(0,1)$ -вектор является, очевидно, квазивыпуклым, если не содержит подматрицу $\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$. Отсюда становится ясно, как устроена максимальная по числу столбцов пра-

вильная $(0,1)$ -матрица с числом строк m . Она содержит ровно $m - l + 1$ различных столбцов с числом нулей, равным l , $1 \leq l \leq m-1$. Поэтому максимальное число столбцов характеристической матрицы H правильной матрицы C будет равно $\frac{1}{2}(m-1)(m+2)$. Понятно,

что если $(0,1)$ -вектор (h_i) ($i \in I$) является квазивогнутым, то $(0,1)$ -вектор $(1-h_i)$ ($i \in I$) будет квазивыпуклым и наоборот. Поэтому максимальное число столбцов характеристической матрицы H для матрицы C , состоящей из квазивогнутых столбцов, будет также равно $\frac{1}{2}(m-1)(m+2)$, а максимальное число различных столбцов характеристической матрицы H для почти правильной матрицы C с числом строк m равняется $m(m-1)$.

Строение $(0,1)$ -матрицы можно охарактеризовать, рассматривая ее как характеристическую матрицу семейства подграфов некоторого графа. Пусть имеется граф с множеством вершин I . Рассмотрим подграф этого графа, индуцированный подмножеством вершин $I' \subset I$. Вектор-столбец (h_i) ($i \in I$), где

$$h_i = \begin{cases} 0, & \text{если } i \in I', \\ 1 & \text{иначе,} \end{cases}$$

назовем *характеристическим вектором* данного подграфа.

Рассмотрим некоторое семейство подграфов данного графа, каждый из которых индуцирован соответствующим подмножеством вершин. Матрицу, столбцы которой — характеристические векторы подграфов, назовем *характеристической матрицей семейства подграфов*. В этих терминах правильные и почти правильные $(0,1)$ -матрицы описываются очень просто. Правильная матрица есть характеристическая матрица семейства подцепей некоторой цепи, а почти правильная — семейства цепей некоторого цикла.

1.2. Задача о ближайшем соседе. Алгоритм динамического программирования

Наиболее простой алгоритм решения задачи FL в случае правильной матрицы C получается, если воспользоваться идеями метода рекурсивных соотношений динамического программирования [10', 22]. При этом будет удобнее строить такой алгоритм не непосредственно для задачи FL , а для некоторой вспомогательной задачи, называемой задачей о ближайшем соседе [10], к которой сводится задача FL с правильной матрицей C . Кроме того, эта задача будет полезна и в дальнейшем.

Пусть имеется целочисленный сегмент $Z = \{0, 1, \dots, n\}$ и неотрицательная функция $f(x, y)$, заданная на парах (x, y) таких, что $0 \leq x < y \leq n$. Задачей о ближайшем соседе будем называть следующую задачу

$$\begin{aligned} \min_{N, \{x_p\}} \sum_{i=1}^N f(x_{i-1}, x_i); \\ 0 = x_0 < x_1 < \dots < x_N = n; \\ x_i \in Z, \quad i = 1, \dots, N. \end{aligned}$$

В этой задаче нужно отыскать такую возрастающую последовательность $\{x_0, x_1, \dots, x_N\}$ элементов множества Z , чтобы сумма значений функции $f(x, y)$ от каждых двух соседних элементов последовательности была наименьшей.

Алгоритм решения задачи о ближайшем соседе строится на основе метода динамического программирования, суть которого состоит в том, чтобы не использовать рассматриваемую задачу изолированно, а попытаться включить ее в некоторую последовательность однотипных задач и составить соотношения (называемые рекуррентными), связывающие оптимальные значения целевых функций задач последовательности.

Рассмотрим последовательность задач $\{PL(y), y = 0, 1, \dots, n\}$, где каждая задача $PL(y)$ отличается от исходной задачи о ближайшем соседе только тем, что вместо целочисленного сегмента Z рассматривается часть этого сегмента $Z(y) = \{0, 1, \dots, y\}$. Обозначим через $S(y)$ оптимальное значение целевой функции задачи $PL(y)$. При этом по определению считаем, что $S(0) = 0$. Имеет место следующая лемма, устанавливающая справедливость соотношений, позволяющих отыскивать оптимальные значения целевых функций каждой задачи последовательности по оптимальному значению целевых функций предыдущих задач.

Лемма 3.1.1. Справедливы следующие равенства:

$$S(0) = 0;$$

$$S(y) = \min_{0 \leq x < y} \{S(x) + f(x, y)\}, \quad y = 1, \dots, n.$$

Для доказательства второго равенства при произвольном y , $0 \leq y \leq n$, заметим прежде всего, что для любого x , $0 \leq x < y$, имеет место неравенство

$$S(y) \leq S(x) + f(x, y).$$

Действительно, пусть $\{x_0, x_1, \dots, x_N\}$ — оптимальное решение задачи $П(x)$. Тогда последовательность $\{x_0, x_1, \dots, x_N, y\}$ будет, очевидно, допустимым решением задачи $П(y)$ со значением целевой функции $S(x) + f(x, y)$.

Покажем, кроме того, что существует элемент x , $0 \leq x < y$, для которого выполняется равенство

$$S(y) = S(x) + f(x, y).$$

Действительно, пусть $\{x_0^*, x_1^*, \dots, x_N^*\}$ — оптимальное решение задачи $П(y)$. Тогда, очевидно, последовательно $\{x_0^*, x_1^*, \dots, x_{N-1}^*\}$ будет оптимальным решением задачи $П(x_{N-1}^*)$. Тогда можем написать

$$S(y) = \sum_{i=1}^{N-1} f(x_{i-1}^*, x_i^*) + f(x_{N-1}^*, x_N^*) = S(x_{N-1}^*) + f(x_{N-1}^*, y).$$

Это завершает доказательство.

Следствие. При любом y , $1 \leq y \leq n$, если для некоторого x , $0 \leq x < y$, выполняется равенство $S(y) = S(x) + f(x, y)$, то существует оптимальное решение $\{x_0^*, x_1^*, \dots, x_N^*\}$ задачи $П(y)$, такое, что $x_{N-1}^* = x$.

Основанный на приведенных выше рекуррентных соотношениях алгоритм решения задачи о ближайшем соседе состоит из двух этапов. Первый этап включает n шагов. На k -м, $k = 1, \dots, n$, шаге вычисляется величина $S(k)$ по формуле

$$S(k) = \min_{0 \leq i < k} \{S(i) + f(i, k)\}.$$

Второй этап состоит из предварительного шага и конечного числа однотипных основных шагов. На предварительном шаге полагается $\bar{y}_0 = n$. На k -м, $k = 1, 2, \dots$ основном шаге рассматривается элемент y_{k-1} , найденный на предыдущем шаге, и отыскивается элемент y_k , $0 \leq y_k < y_{k-1}$, такой, что

$$S(y_{k-1}) = S(y_k) + f(y_k, y_{k-1}).$$

Если $y_k > 0$, то начинается следующий шаг. Если $y_k = 0$, то алгоритм заканчивает работу и $\{y_k, y_{k-1}, \dots, y_1\}$ — оптимальное решение задачи.

Легко видеть, что для реализации первого этапа алгоритма требуется $O(n^2)$ действий, а для реализации второго — $O(n)$. Поэтому трудоемкость рассмотренного алгоритма оценивается величиной $O(n^2)$.

Следующая теорема показывает, что если матрица транспортных затрат правильная, то для решения задачи FL может быть эффективно использован метод динамического программирования.

Теорема 3.1.1. Если матрица $C = (c_{ij})$ ($i \in I, j \in J$) является правильной, то задача FL с матрицей транспортных затрат C сводится к задаче о ближайшем соседе.

Убедимся прежде в справедливости следующего выражения.

Лемма 3.1.2. Если вектор–столбец (c_i) ($i \in I$) является квазивыпуклым, то

$$\min_{i \in I} c_i = c_1 - \sum_{i=2}^m \max\{0; c_{i-1} - c_i\}.$$

Доказательство. Пусть $c_{i_0} = \min_{i \in I} c_i$. В силу квазивыпуклости вектор–столбца (c_i)

имеем

$$c_1 \geq c_2 \geq \dots \geq c_{i_0} \leq c_{i_0+1} \leq \dots \leq c_m.$$

Но тогда можем написать

$$c_{i_0} = c_1 - \sum_{i=2}^{i_0} (c_{i-1} - c_i) = c_1 - \sum_{i=2}^{i_0} \max\{0; c_{i-1} - c_i\} = c_1 - \sum_{i=2}^m \max\{0; c_{i-1} - c_i\}.$$

Лемма доказана.

Для доказательства теоремы вместо задачи FL будет удобней рассматривать задачу $MINF_0$ и доказать ее сведение к задаче о ближайшем соседе. Поэтому поставим в соответствие исходной задаче $MINF_0$ задачу о ближайшем соседе с множеством $Z = \{0, 1, \dots, m, m+1\}$ и функцией $f(x, y)$ вида

$$f(x, y) = \begin{cases} c_y + \sum_{j \in J} c_{yj}, & \text{если } 0 = x < y \leq m; \\ c_y + \sum_{j \in J} \max\{0; c_{xj} - c_{yj}\}, & \text{если } 0 < x < y \leq m; \\ 0, & \text{если } 0 < x < y = m+1; \\ \infty, & \text{если } 0 = x < y = m. \end{cases}$$

Допустимое решение $X = \{i_1, \dots, i_P\}$ исходной задачи $MINF_0$ и допустимое решение $\{i_0, i_1, \dots, i_P, i_{P+1}\}$ рассматриваемой задачи о ближайшем соседе назовем соответствующими решениями. Покажем, что для ответствующих решений значения целевых функций исследуемых задач совпадает. В самом деле, учитывая предыдущую лемму 3.1.2, можем написать

$$\begin{aligned} \sum_{i \in X} f_i + \sum_{j \in J} \min_{i \in X} c_{ij} &= \sum_{p=1}^P f_{i_p} + \sum_{j \in J} \min_{1 \leq p \leq P} c_{i_p j} = \\ &= \sum_{p=1}^P f_{i_p} + \sum_{j \in J} \left\{ c_{i_1 j} - \sum_{p=2}^P \max\{0; c_{i_{p-1} j} - c_{i_p j}\} \right\} = \end{aligned}$$

$$f_{i_1} + \sum_{j \in J} c_{i_1 j} + \sum_{p=2}^P \left\{ f_{i_p} - \sum_{j \in J} \max\{0; c_{i_{p-1} j} - c_{i_p j}\} \right\} =$$

$$f(0, i_1) + \sum_{p=2}^P f(i_{p-1}, i_p) + f(i_P, m+1) = \sum_{p=1}^{P+1} f(i_{p-1}, i_p).$$

Поскольку по оптимальному решению каждой из задач очевидным образом строится соответствующее ему решение другой задачи и при этом значения целевых функций на соответствующих решениях равны, то критерий оптимальности (лемма 1.1.1) выполняется. Следовательно, построенное по оптимальному решению задачи о ближайшем соседе решение задачи $MINF_0$ будет оптимальным. Теорема доказана.

Нетрудно получить оценку трудоемкости алгоритма решения задачи FL с правильной матрицей C , использующего в качестве подалгоритма алгоритм решения соответствующей задачи о ближайшем соседе. Трудоемкость процедуры построения исходных данных задачи о ближайшем соседе оценивается величиной $O(m^2 n)$. Эта оценка перекрывает оценку трудоемкости вычислений по рекуррентным соотношениям и является оценкой трудоемкости алгоритма решения задачи FL в целом.

1.3. Правильные полиномы с неотрицательными коэффициентами

Полином $P_0(y_1, \dots, y_m)$ с неотрицательными коэффициентами назовем правильным, если его характеристическая матрица — правильная. Правильный полином может быть записан следующим образом:

$$P_0(y_1, \dots, y_m) = \sum_{i=1}^m \sum_{k=i}^m b_{ik} y_i \cdot \dots \cdot y_k,$$

где $b_{ik} \geq 0$ при $i \neq k$.

Отметим, что приведенный вид полинома $P_0(y_1, \dots, y_m)$ предполагает некоторый специальный порядок столбцов характеристической матрицы. При таком порядке первые номера имеют столбцы, у которых в первой строке нули, вторые места занимают столбцы, которые имеют нули во второй строке, следующие места — столбцы, имеющие нули в третьей строке и т.д. Кроме того, столбцы каждой такой группы упорядочены еще в порядке лексикографического убывания. Правильную $(0,1)$ -матрицу с указанным порядком столбцов будем называть *матрицей в стандартной форме*.

Правильные полиномы обладают тем свойством, что для решения задачи их минимизации можно построить эффективный алгоритм на базе псевдобулева программирования. Это становится возможным, поскольку удастся в явном виде построить функцию, осуществляющую редукцию исходного полинома к правильному полиному с числом переменных на единицу меньше и такого, что оптимальные значения переменных задачи минимизации этого полинома являются оптимальными значениями соответствующих переменных задачи минимизации исходного полинома.

Имеет место следующая

Теорема 3.1.2. Задача $MINP_0$ правильного полинома сводится к задаче $MINP_0$ правильного полинома, с числом переменных на единицу меньше, чем у исходного полинома.

Доказательство. В соответствии с методом псевдобулевого программирования для представления полинома $P_0(y_1, \dots, y_m)$ в виде

$$P_0(y_1, \dots, y_m) = y_1 \cdot \sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k + \sum_{i=2}^m \sum_{k=i}^m b_{ik} y_i \cdot \dots \cdot y_k,$$

построим функцию $y_1(y_2, \dots, y_m)$ такую, что

$$y_1(y_2, \dots, y_m) = \begin{cases} 0, & \text{если } \sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k > 0, \\ 1, & \text{если } \sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k < 0. \end{cases}$$

Для этого рассмотрим функцию

$$\rho(l) = \sum_{k=1}^l b_{1k}, \quad 1 \leq l \leq m,$$

и определим такой наименьший номер l_1 , $1 \leq l_1 \leq m$, что $\rho(l_1) > 0$. Если $\rho(m) \leq 0$, то считаем, что $l_1 = m + 1$.

Положим

$$y_1(y_2, \dots, y_m) = \begin{cases} 0, & \text{если } l_1 = 1, \\ 1 - y_2 \cdot \dots \cdot y_{l_1}, & \text{если } 1 < l_1 \leq m, \\ 1, & \text{если } l_1 = m + 1, \end{cases}$$

и покажем, что данная функция обладает указанным выше свойством.

Для этого рассмотрим первоначально произвольный неединичный $(0,1)$ -вектор (y_2, \dots, y_m) и пусть i_0 , $1 < i_0 \leq m$, — наименьший номер такой, что $y_{i_0} = 0$. Поскольку

$$\sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k = \sum_{k=1}^{i_0-1} b_{1k} = \rho(i_0-1),$$

то исследуем зависимость между знаком величины $\rho(i_0-1)$ и значением функции $y_1(y_2, \dots, y_m)$. Если $\rho(i_0-1) > 0$, то $l_1 \leq i_0 - 1$ и, следовательно, $y_1(y_2, \dots, y_m) = 0$, если $l_1 = 1$, и $y_1(y_2 \cdot \dots \cdot y_m) = 1 - y_2 \cdot \dots \cdot y_{l_1} = 0$, то $l_1 > 1$. Пусть $\rho(i_0-1) < 0$. Тогда $l_1 \geq i_0$ и $y_1(y_2, \dots, y_m) = 1$.

К аналогичным результатам приходим, когда (y_2, \dots, y_m) — единичный вектор. В этом случае рассмотрим знак величины $\rho(m)$. Если $\rho(m) > 0$, то $l_1 \leq m$ и $y_1(y_2, \dots, y_m) = 0$. Если же $\rho(m) < 0$, то $l_1 = m + 1$ и, следовательно, $y_1(y_2, \dots, y_m) = 1$.

Таким образом, получаем полином

$$P'_0(y_2, \dots, y_m) = y_1(y_2, \dots, y_m) \sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k + \sum_{i=2}^m \sum_{k=i}^m b_{ik} y_i \cdot \dots \cdot y_k,$$

который по построению обладает тем свойством, что если (y_2^*, \dots, y_m^*) — оптимальное решение задачи $MINP_0$ для этого полинома, то $(y_1^*, y_2^*, \dots, y_m^*)$, где $y_1^* = y_1(y_2^*, \dots, y_m^*)$ является оптимальным решением задачи $MINP_0$ исходного полинома $P_0(y_1, \dots, y_m)$.

Покажем, что полученный полином $P'_0(y_2, \dots, y_m)$ — правильный полином с неотрицательными коэффициентами. Для этого рассмотрим полином

$$y_1(y_2, \dots, y_m) \sum_{k=1}^m b_{1k} y_2 \cdot \dots \cdot y_k$$

и заметим, что это правильный полином с неотрицательными коэффициентами. Если $y_1(y_2, \dots, y_m) = 0$ или $y_1(y_2, \dots, y_m) = 1$, то доказывать нечего. Если же $y_1(y_2, \dots, y_m) = 1 - y_2 \cdot \dots \cdot y_{l_1}$, то можем написать

$$\sum_{k=1}^m b_{1k} (1 - y_2 \cdot \dots \cdot y_{l_1}) y_2 \cdot \dots \cdot y_k = \sum_{k=1}^{l_1-1} b_{1k} y_2 \cdot \dots \cdot y_k \cdot \left(\sum_{k=1}^{l_1-1} b_{1k} \right) y_2 \cdot \dots \cdot y_{l_1}.$$

Отсюда с учетом того, что $\rho(l_1-1) \leq 0$, получаем требуемое. Теорема доказана.

Из доказательства теоремы следует, что задача $MINP_0$ для исходного полинома

$$P_0(y_1, \dots, y_m) = \sum_{i=1}^m \sum_{k=i}^m b_{ik} y_i \cdot \dots \cdot y_k$$

сводится к задаче $MINP_0$ для полинома

$$P'_0(y_2, \dots, y_m) = \sum_{i=2}^m \sum_{k=i}^m b'_{ik} y_i \cdot \dots \cdot y_k,$$

коэффициенты которого вычисляются следующим образом:

если $l_1 = 1$, то

$$b'_{ik} = b_{ik}, \quad 2 \leq i \leq k \leq m;$$

если $1 < l_1 \leq m$, то

$$b'_{2k} = b_{1k} + b_{2k}, \quad 2 \leq k \leq l_1-1,$$

$$b'_{2l_1} = - \sum_{k=1}^{l_1-1} b_{1k} + b_{2l_1},$$

$$b'_{2k} = b_{2k}, \quad l_1 < k \leq m,$$

$$b'_{ik} = b_{ik}, \quad 3 \leq i \leq k \leq m;$$

если $l_1 = m + 1$, то

$$b'_{2k} = b_{1k} + b_{2k}, \quad 2 \leq k \leq m,$$

$$b'_{ik} = b_{ik}, \quad 3 \leq i \leq k \leq m.$$

Из доказательства следует также, что задача $MINP_0$ для исходного полинома сводится к задаче минимизации полинома от одной переменной, которая решается очевидным образом. Следовательно, базируясь на доказанной теореме и используя приведенные выше формулы для расчета коэффициентов нового полинома с числом переменных на

единицу меньше, можно построить эффективный алгоритм минимизации правильного полинома.

Алгоритм состоит из двух этапов, на первом из которых строятся формулы для вычисления оптимальных значений переменных, а на втором вычисляются сами эти значения.

Первый этап включает m шагов. На s -м шаге, $s = 1, 2, \dots, m-1$, рассматривается полином

$$P_0^{(s)}(y_s, \dots, y_m) = y_s \sum_{k=s}^m b_{sk}^{(s)} y_{s+1} \cdot \dots \cdot y_k + \sum_{i=s+1}^m \sum_{k=i}^m b_{ik}^{(s)} y_i \cdot \dots \cdot y_k$$

и отыскивается номер l_s , $s \leq l_s \leq m+1$. После этого в зависимости от величины номера l_s , строятся новые коэффициенты $b_{ik}^{(s+1)}$, $s+1 \leq i \leq k \leq m$, и начинается следующий шаг.

На m -м шаге рассматривается полином

$$P_0^{(m)}(y_m) = y_m b_{mm}^{(m)}$$

и отыскивается номер l_m , равный либо m , либо $m+1$. На этом первый этап алгоритма заканчивается и начинается второй.

Второй этап также включает m шагов. На s -м шаге, $s = m, m-1, \dots, 1$, выполняются следующие действия.

На m -м шаге полагается

$$y_m^* = \begin{cases} 0, & \text{если } l_m = m, \\ 1, & \text{если } l_m = m+1, \end{cases}$$

и начинается $m-1$ -й шаг.

На s -м шаге, $s \leq m-1$, имеются вычисленные на предыдущих шагах компоненты y_{s+1}^*, \dots, y_m^* оптимального решения, с использованием которых вычисляется значение y_s^* по формуле

$$y_s^* = \begin{cases} 0, & \text{если } l_s = s, \\ 1 - y_{s+1}^* \cdot \dots \cdot y_{l_s}^*, & \text{если } s < l_s \leq m, \\ 1, & \text{если } l_s = m+1. \end{cases}$$

После этого, если $s > 1$ начинается $s-1$ -й шаг, иначе алгоритм заканчивает работу.

Несложно получить оценку временной сложности данного алгоритма. Трудоемкость каждого шага первого этапа оценивается, очевидно, величиной $O(m)$. Аналогично, трудоемкость каждого шага второго этапа оценивается величиной $O(m)$. Поэтому в целом оценка трудоемкости рассмотренного алгоритма решения задачи $MINP_0$ для правильного полинома имеет вид $O(m^2)$. Заметим, кроме того, что трудоемкость процедуры построения по правильной матрице $C=(c_{ij})$ ($i \in I, j \in J$) коэффициентов правильного полинома $P_0(y_1, \dots, y_m)$ оценивается величиной $O(m \cdot n)$. Поэтому оценкой временной сложности алгоритма решения задачи FL с использованием процедуры минимизации правильного полинома будет величина $O(m \cdot n + m^2)$.

1.4. Почти правильные полиномы с неотрицательными коэффициентами

Полином $q_0(y_1, \dots, y_m)$ с неотрицательными коэффициентами назовем *почти правильным*, если его характеристическая матрица является почти правильной. Почти правильный полином, по аналогии с правильным, при соответствующем упорядочении столбцов характеристической матрицы может быть записан следующим образом

$$q_0(y_1, \dots, y_m) = \sum_{i=1}^m \sum_{k=i}^m b_{ik} y_i \cdot \dots \cdot y_k + \sum_{i=1}^{m-2} \sum_{k=i+2}^m d_{ik} y_1 \cdot \dots \cdot y_i y_k \cdot \dots \cdot y_m,$$

где $b_{ik} \geq 0$ при $i \neq k$ и $d_{ik} \geq 0$.

Почти правильный полином обладает тем свойством, что оптимальное решение задачи минимизации такого полинома может быть получено с использованием рассмотренного выше алгоритма минимизации правильного полинома. Действительно, непосредственного из вида почти правильного полинома вытекает следующая

Лемма 3.1.3. Если $q_0(y_1, \dots, y_m)$ — почти правильный полином, то при любом k , $1 \leq k \leq m-1$, полином

$$q'_0(y_{k+1}, \dots, y_m) = q_0(1, \dots, 1, 0, y_{k+1}, \dots, y_m)$$

будет правильным.

Отсюда получаем, что оптимальное решение задачи $MINP_0$ для почти правильного полинома может быть найдено в результате решения $m-1$ задач $MINP_0$ для соответствующих правильных полиномов.

Трудоёмкость такого алгоритма, очевидно, оценивается величиной $O(m^3)$, а трудоёмкость алгоритма решения задачи FL с почти правильной матрицей $C=(c_{ij})$ ($i \in I, j \in J$) — величиной $O(m \cdot n + m^3)$.

1.5. Некоторые обобщения правильных и почти правильных полиномов

Ранее было отмечено, что правильные и почти правильные $(0,1)$ -матрицы можно рассматривать как характеристические матрицы семейств подцепей соответственно некоторой цепи и некоторого цикла. Можно указать графы более общие, чем цепи и циклы, характеристические матрицы семейств некоторых подграфов которых обладают свойствами, аналогичными свойствам правильных и почти правильных матриц. К таким графам относятся, в частности, так называемые корневые деревья.

Корневым деревом называют ориентированный граф без контуров с выделенной вершиной, называемой *корневой*, и обладающей тем свойством, что из любой вершины существует единственная ориентированная цепь (путь) в корневую вершину.

Рассмотрим $(0,1)$ -матрицу H , являющуюся характеристической для произвольного семейства P_1, \dots, P_k ориентированных цепей корневого дерева T с множеством вершин $I = \{1, \dots, m\}$ и полином, порожденный данной $(0,1)$ -матрицей. Понятно, что указанный класс матриц содержит в себе правильные $(0,1)$ -матрицы, которые являются характеристическими для семейств подцепей некоторой цепи. Заметим далее, что полиномы с таки-

ми характеристическими матрицами обладают тем же полезным свойством, что и правильные полиномы, использование которого позволяет строить эффективные алгоритмы их минимизации. Действительно, пусть T' — подграф корневого дерева T , порожденный вершинами рассматриваемого семейства ориентированных цепей. Пусть i — произвольная висячая вершина любой компоненты связности графа T' . Понятно, что множества вершин путей, проходящих через вершину i , будут сравнимы между собой по включению. Следовательно, полином, состоящий из членов исходного полинома, содержащих переменную y_i , будет иметь то же строение, что и полином, включающий члены правильного полинома, содержащие переменную y_1 . Поэтому задача минимизации рассматриваемого полинома, так же как и в случае правильного полинома сводится к задаче минимизации того же класса, но от меньшего числа переменных.

Более строго наличие полезного свойства у характеристической матрицы H семейства ориентированных цепей корневого дерева T доказывает следующее утверждение.

Лемма 3.1.4. Семейство P_1, \dots, P_k ориентированных цепей корневого дерева T обладает тем свойством, что либо найдутся вершина дерева T , принадлежащая только одной из цепей, либо в рассматриваемом семействе найдется две цепи, одна из которых является частью другой.

Доказательство. Пусть i — вершина дерева T , принадлежащая одной из цепей семейства и максимально удаленная от корневой вершины дерева T . Предположим, что вершина i принадлежит, по крайней мере, двум цепям рассматриваемого семейства. Но поскольку все вершины этих двух цепей лежат на пути из вершины i в корневую вершину, то одна из цепей будет подцепью другой. Лемма доказана.

Из доказанного следует, что если H — характеристическая матрица семейства ориентированных цепей корневого дерева T , то матрица \bar{H} будет так называемой Q -матрицей. Обсуждению таких $(0,1)$ -матриц далее будет уделено значительное внимание. Сейчас отметим лишь, что если H — характеристическая матрица полинома, а \bar{H} является Q -матрицей, то для решения задачи минимизации рассматриваемого полинома можно построить эффективный алгоритм, по существу ничем не отличающийся от алгоритма минимизации правильного полинома. Оценка трудоемкости такого алгоритма имеет вид $O(m^2)$.

По аналогии с расширением класса правильных полиномов до почти правильных можно обобщить и класс полиномов, порождаемых характеристическими матрицами семейств цепей корневого дерева, и при этом получить класс полиномов, задача минимизации которых сводится к задаче минимизации исходных полиномов.

Для этого рассмотрим граф G с множеством вершин $I = \{1, \dots, m\}$, представляющий собой один цикл C , каждая вершина которого есть корневая вершина корневого дерева. Пусть вершины графа G занумерованы таким образом, что цикл C имеет вершины с номерами $m' + 1, \dots, m$. Справедливо следующее утверждение, аналогичное лемме 3.1.3.

Лемма 3.1.5. Если $q_1(y_1, \dots, y_m)$ — полином с характеристической матрицей, являющейся характеристической матрицей семейства ориентированных цепей графа G , то при любом k , $m' + 1 \leq k \leq m - 1$, задача минимизации полинома

$$q'(y_1, \dots, y_{m'}, y_{k+1}, \dots, y_m) = q(y_1, \dots, y_{m'}, 1, \dots, 1, y_{k+1}, \dots, y_m)$$

сводится к задаче минимизации полиномов с характеристической матрицей, являющейся характеристической матрицей семейства цепей корневого дерева.

Оценка трудоемкости алгоритма минимизации полиномов, порождаемых характеристическими матрицами семейств ориентированных цепей графа G , равняется $O(m^3)$.

2 Алгоритмы распознавания правильных и почти правильных матриц

Относительно простое строение правильных и почти правильных матриц не создает больших сложностей для построения соответствующих алгоритмов приведения. Одна из первых попыток распознавания правильных матриц содержится в [1], где предлагаются некоторые необходимые и достаточные условия существования приводящей перестановки, однако сама эта перестановка в явном виде не строится. Достаточно полное исследование вопроса о приведении исходной матрицы к правильной и почти правильной дается в работе [9]

2.1. Алгоритм приведения (0,1)-матрицы к правильной

Рассмотрим (0,1)-матрицу $A = (a_{ij})$ ($i \in I, j \in J$) и будем считать, что она не имеет нулевых столбцов и единичных строк. Для всякого $j \in J$ рассмотрим множества $X_j^0 = \{i \in I \mid a_{ij} = 0\}$ и $X_j^1 = \{i \in I \mid a_{ij} = 1\}$, которые называются *индикаторными множествами* j -го столбца матрицы A . Понятно, что если матрица A — правильная, то для всякого $j \in J$ индикаторное множество X_j^0 совпадает с множеством $\{i, i+1, \dots, k\}$, где $1 \leq i \leq k \leq m$, а если A — почти правильная, то для всякого $j \in J$ хотя бы одно из индикаторных множеств X_j^0, X_j^1 совпадает с множеством $\{i, i+1, \dots, k\}$, где $1 \leq i \leq k \leq m$.

Алгоритм построения перестановки $\{i_1, \dots, i_m\}$, приводящей рассматриваемую матрицу A к правильной, либо показывающий, что такое приведение невозможно, будем представлять как процесс последовательного применения к исходной матрице, а затем, возможно, к некоторым ее подматрицам нижеприводимой процедуры.

Процедура приведения матрицы $A = (a_{ij})$ ($i \in I, j \in J$) к правильной состоит из двух этапов, каждый из которых, в свою очередь, включает конечное число однотипных шагов. При этом на каждом шаге имеется множество J_0 просмотренных (приведенных) столбцов и некоторое разбиение $\{Z_1, \dots, Z_P\}$ множества $I_0 = \bigcup_{j \in J_0} X_j^0$. Это разбиение обладает тем

свойством, что если $\{i_1, \dots, i_m\}$ — приводящая перестановка, то

$$Z_p = \{i_{s_{p-1}+1}, \dots, i_{s_p}\}, \quad p = 1, \dots, P,$$

где $0 \leq s_0 < s_1 < \dots < s_P \leq m$. Другими словами, данное разбиение задает множество всех перестановок, приводящих к квазивыпуклому виду столбцы из множества J_0 . Допустимые перестановки могут отличаться друг от друга только порядком элементов внутри мно-

жеств Z_p , $p = 1, \dots, P$. Однако никакая перестановка элементов из разных множеств Z_p не является допустимой, поскольку нарушает свойство квазивыпуклости у некоторых столбцов из множества J_0 .

Первый этап процедуры состоит из предварительного шага и конечного числа однотипных основных шагов. На предварительном шаге отыскивается элемент $j \in J$ такой, что

$$j_0 = \arg \max_{j \in J} |X_j^0|.$$

Пусть на очередном основном шаге имеется множество $J_0 \neq J$ приведенных столбцов и разбиение $\{z_1, \dots, z_P\}$ множества $I_0 = \bigcup_{j \in J_0} X_j^0$, обладающее указанным выше свойством.

На первом шаге имеем $J_0 = \{j_0\}$ и $Z_1 = X_{j_0}^0$. Шаг начинается с поиска произвольного элемента $j \notin J_0$, для которого $I_0 \cap X_j^0 \neq \emptyset$ и $X_j^0 \setminus I_0 \neq \emptyset$. Если такого элемента j не существует, то первый этап завершается и начинается второй. В противном случае для множества I_j отыскиваются номера p_1 и p_2 , которые есть соответственно наименьший и наибольший элемент p , $1 \leq p \leq P$, для которого $X_j^0 \cap Z_p \neq \emptyset$. Далее для каждого p , $p_1 < p < p_2$, проверяется условие $Z_p \subset X_j^0$. Если для некоторого p , $p_1 < p < p_2$, указанное включение не выполняется, то алгоритм заканчивает работу и приводящей перестановки не существует. В противном случае проверяются условия $Z_1 \subset X_j^0$ и $Z_P \subset X_j^0$. Если ни одно из этих условий не выполняется, то алгоритм заканчивает работу с отрицательным исходом. Пусть $Z_1 \subset X_j^0$. Тогда строится разбиение

$$\{X_j^0 \setminus I_0, Z_1, \dots, Z_{p_2-1}, Z_{p_2} \cap X_j^0, Z_{p_2} \setminus X_j^0, Z_{p_2+1}, \dots, Z_P\},$$

из которого вычеркивается элемент $Z_{p_2} \setminus X_j^0$, если это пустое множество. Пусть теперь $Z_1 \not\subset X_j^0$, но $Z_P \subset X_j^0$. Тогда строится разбиение

$$\{Z_1, \dots, Z_{p_1-1}, Z_{p_1} \setminus X_j^0, Z_{p_1} \cap X_j^0, Z_{p_1+1}, \dots, Z_P, X_j^0 \setminus I_0\},$$

из которого вычеркивается элемент $Z_{p_1} \setminus X_j^0$, если это пустое множество. Далее в обоих случаях полагается $J_0 \leftarrow J_0 \cup \{j\}$. Если $J_0 = J$, то процедура приведения заканчивает работу с положительным исходом. Если же $J_0 \neq J$, то начинается следующий шаг с новым разбиением, построенным на данном шаге. Заметим, что в силу построения новое разбиение удовлетворяет сформулированному выше необходимому условию.

Отметим, что полученное в результате первого этапа подмножество J_0 можно считать таким, что $\bigcup_{j \in J_0} X_j^0 = I$. Действительно, если $\bigcup_{j \in J_0} X_j^0 = I$ и $I_0 \neq I$, то исходная матрица

$A = (a_{ij})$ ($i \in I, j \in J$) будет приводится к правильной тогда и только тогда, когда таковыми

будут матрицы $A_1 = (a_{ij}) (i \in I_0, j \in J_0)$ и $A_2 = (a_{ij}) (i \in I \setminus I_0, j \in J \setminus J_0)$. Поэтому рассматриваемая процедура приведения может последовательно и независимо применяться к исследованию матриц A_1 и A_2 .

Второй этап аналогичен первому и включает конечное число однотипных шагов.

Пусть на очередном основном шаге имеется множество приведенных столбцов $J_0 \neq J$ и разбиение $\{Z_1, \dots, Z_P\}$ множества I . На первом шаге рассматриваются множество J_0 и разбиение Z_1, \dots, Z_P , полученные в результате первого этапа. Шаг начинается с поиска произвольного элемента $j \notin J_0$ такое, что $X_j^0 \not\subset Z_p$ при любом $p, 1 \leq p \leq P$. Если элемента j с этим свойством не существует, то второй этап и вместе с ним вся процедура приведения завершается с положительным исходом. В противном случае, также как и на первом этапе, для множества X_j^0 отыскиваются номера p_1 и p_2 , являющиеся соответственно наименьшим и наибольшим номером $p, 1 \leq p \leq P$, для которого $X_j^0 \cap Z_p \neq \emptyset$. Далее для каждого $p, p_1 < p < p_2$, проверяется условие $Z_p \subset X_j^0$. Если для некоторого p указанное включение не выполняется, то процедура заканчивает работу и приводящей перестановки не существует. В противном случае, строится разбиение

$$\{Z_1, \dots, Z_{p_1-1}, Z_{p_1} \setminus X_j^0, Z_{p_1} \cap X_j^0, Z_{p_1+1}, \dots, Z_{p_2-1}, Z_{p_2} \cap X_j^0, Z_{p_2} \setminus X_j^0, Z_{p_2+1}, \dots, Z_P\},$$

из которого удаляются множества $Z_{p_1} \setminus X_j^0$ и $Z_{p_2} \setminus X_j^0$ в случае, если они оказываются пустыми. Далее полагается $J_0 \leftarrow J_0 \cup \{j\}$ и, если $J_0 \neq J$, начинается следующий шаг. Если $J_0 = J$, то второй этап и в целом процедура приведения заканчивают работу с положительным исходом.

Пусть по завершению процедуры приведения матрицы $A = (a_{ij}) (i \in I, j \in J)$ получены множество J и разбиение Z_1, \dots, Z_P . Если $J_0 = J$, то алгоритм приведения матрицы A к правильной считаем законченным. Пусть $J_0 \neq J$. Заметим, что в этом случае для всякого $j \notin J_0$ имеем $X_j^0 \subset Z_p$ для некоторого $p, 1 \leq p \leq P$. Положим

$$J_p = \{j \notin J_0 \mid X_j^0 \subset Z_p\}, \quad p = 1, \dots, P.$$

Тогда исходная матрица $A = (a_{ij}) (i \in I, j \in J)$ будет приводимой к правильной матрице тогда и только тогда, когда таковыми будут матрицы $A_0 = (a_{ij}) (i \in I, j \in J_0)$ и $A_p = (a_{ij}) (i \in I, j \in J_p), p = 1, \dots, P$. В силу этого, рассмотренная процедура может последовательно и независимо применяться к матрицам A_p . В этом случае алгоритм приведения матрицы A к правильной завершит свою работу с положительным исходом, если с таким исходом заканчивают свою работу алгоритмы приведения матриц A_p для каждого $p=1, \dots, P$. Соответственно алгоритм заканчивает работу с отрицательным исходом, если с таким исходом закончит работу алгоритм приведения матрицы A_p для некоторого $p, 1 \leq p \leq P$.

Оценим трудоемкость процедуры приведения и алгоритма приведения в целом. Заметим, прежде всего, что трудоемкость каждого шага на обоих этапах процедуры приведения оценивается величиной $O(m \cdot n)$. Заметим также, что в результате каждого шага

либо некоторый столбец исходной матрицы приводится к квазивыпуклому, либо завершается этап процедуры. Поэтому общее число шагов всех процедур приведения, используемых при исследовании исходной матрицы, оценивается величиной $O(n)$. Таким образом, получаем оценку трудоемкости процедуры приведения и в целом алгоритма приведения $(0,1)$ -матрицы к правильной равную $O(mn^2)$.

2.2. Алгоритм приведения $(0,1)$ -матрицы к почти правильной

Исходной $(0,1)$ -матрице $A = (a_{ij})$ ($i \in I, j \in J$) поставим в соответствие $(0,1)$ -матрицу $\tilde{A} = (\tilde{a}_{ij})$ ($i \in I, j \in J$), элементы которой определяются следующим образом:

$$\tilde{a}_{ij} = \begin{cases} a_{ij}, & \text{если } a_{1j} = 1, \\ 1 - a_{ij}, & \text{если } a_{1j} = 0. \end{cases}$$

Следующая лемма позволяет выгодно переформулировать задачу о приведении $(0,1)$ -матрицы к почти правильной.

Лемма 3.2.1. Произвольная $(0,1)$ -матрица A приводима к почти правильной тогда и только тогда, когда матрица \tilde{A} приводима к правильной.

Доказательство. Пусть исходная матрица $A = (a_{ij})$ ($i \in I, j \in J$) приводима к почти правильной матрице $A' = (a'_{ikj})$ ($k = 1, \dots, m; j \in J$). Для приводящей перестановки $\{i_1, \dots, i_m\}$ можно считать, что $i_1 = 1$. Действительно, если $i_k = 1, k \neq 1$, то можно построить перестановку $\{i_k, i_{k+1}, \dots, i_m, i_1, \dots, i_{k-1}\}$, которая также будет приводящей. Рассмотрим матрицу $\tilde{A}' = (\tilde{a}'_{ikj})$ ($k = 1, \dots, m; j \in J$), элементы которой определяются следующим образом

$$\tilde{a}'_{ikj} = \begin{cases} a'_{ikj}, & \text{если } a_{1j} = 1, \\ 1 - a'_{ikj}, & \text{если } a_{1j} = 0. \end{cases}$$

Матрица \tilde{A}' является, очевидно, правильной и приводимой к матрице \tilde{A} .

Обратно. Пусть матрица $\tilde{A} = (\tilde{a}_{ij})$ ($i \in I, j \in J$) приводима к правильной матрице $\tilde{A}' = (\tilde{a}'_{ikj})$ ($k = 1, \dots, m; j \in J$). Поскольку $\tilde{a}_{1j} = 1$ для всякого $j \in J$, то можно считать, что для приводящей перестановки $\{i_1, \dots, i_m\}$ имеем $i_1 = 1$. Построим по матрице \tilde{A}' матрицу $A' = (a'_{ikj})$ ($k = 1, \dots, m; j \in J$), элементы которой определяются следующим образом:

$$a'_{ikj} = \begin{cases} a'_{ikj}, & \text{если } a_{1j} = 1, \\ 1 - a'_{ikj}, & \text{если } a_{1j} = 0. \end{cases} \quad !$$

Матрица A' является, очевидно, почти правильной и приводимой к матрице A .

Из доказанного следует, что задача распознавания почти правильной матрицы сводится к задаче распознавания правильной матрицы.

Пусть $A = (a_{ij})$ ($i \in I, j \in J$) — исходная $(0,1)$ -матрица

Алгоритм приведения матрицы A к почти правильной состоит из следующих этапов. Первоначально по матрице A строится матрица \tilde{A} , из которой удаляется единичная

первая строка. Далее к полученной матрице $\tilde{A} = (\tilde{a}_{ij})$ ($i=2, \dots, m; j \in J$) применяется рассмотренный выше алгоритм приведения к правильной матрице. Если приводящей перестановки не существует, то исходная матрица A не приводима к почти правильной матрице и алгоритм заканчивает работу с отрицательным исходом. Если $\{i_2, \dots, i_m\}$ — перестановка, приводящая матрицу \tilde{A} к правильной, то искомой перестановкой, приводящей исходную матрицу A к правильной будет перестановка $\{1, i_2, \dots, i_m\}$ и алгоритм заканчивает работу с положительным исходом.

Оценка трудоемкости данного алгоритма совпадает с оценкой для алгоритма приведения матрицы к правильной и равняется $O(mn^2)$.

3 Вполне уравновешенные матрицы и полиномы

Важным эффективно разрешимым подклассом задачи FL являются задачи с матрицами транспортных затрат, имеющие характеристические матрицы $H = (h_{ij})$ такие, что матрицы $\bar{H} = (1 - h_{ij})$ — вполне уравновешенные. Матрицы, обладающие этим свойством, включают правильные матрицы и рассмотренные выше $(0,1)$ -матрицы, порожденные семействами цепей корневого дерева.

Прогресс в решении задачи FL в случае, когда матрица \bar{H} является вполне уравновешенной, связан с наличием у таких матриц некоторого свойства, называемого гриди свойством. Ниже дается несколько эквивалентных определений этого свойства, каждое из которых оказывается полезным либо при выяснении строения вполне уравновешенных матриц, либо при построении алгоритмов решения задачи FL .

Гриди свойство матрицы \bar{H} позволяет построить эффективные алгоритмы решения задачи FL . Наиболее известным таким алгоритмом является алгоритм из [116'], где задача FL представлена в виде задачи SC , у которой матрица ограничений обладает гриди свойством. В этом случае задача, двойственная к релаксированной задаче SC , известная как задача об упаковке, может быть решена нетрудоемким алгоритмом, названным гриди алгоритмом. Одновременно может быть построено целочисленное оптимальное решение релаксированной задачи SC , которое является оптимальным решением исходной задачи SC . Следует отметить, что данный алгоритм может быть обоснован, и не прибегая к представлению задачи FL в виде задачи SC . Дело в том, что задача, двойственная к релаксированной задаче FL с матрицей транспортных затрат, представленной в канонической форме, является все той же задачей об упаковке, для которой необходимо построить соответствующее тупиковое решение и подобрать блокирующее множество, порождающее оптимальное решение задачи FL . Именно такая идея, хотя и в несколько завуалированном виде, была реализована в уже упоминавшейся работе [64], в которой фактически впервые был описан гриди алгоритм решения задачи FL .

Гриди свойство матрицы \bar{H} позволяет также построить эффективный алгоритм для задачи FL , используя другую эквивалентную формулировку этой задачи — задачу минимизации полиномов от $(0,1)$ -переменных. Строение характеристических матриц та-

ких полиномов, названных вполне уравновешенными, таково, что позволяет перенести идею построения эффективного алгоритма псевдобулева программирования, реализованную для правильных полиномов, на более общий случай вполне уравновешенных полиномов. Получаемый при этом алгоритм также можно отнести к разряду градиентных алгоритмов, поскольку, как известно из предыдущего, он представляет собой процедуру последовательного вычисления значений переменных, которые в последующем уже не изменяются. Оценки трудоемкости всех указанных гриди алгоритмов решения задачи FL с матрицей транспортных затрат размера $m \times n$ — одинаковые и равняются величине $O(m^2n)$.

3.1. Вполне уравновешенные и гриди матрицы

Квадратная $(0,1)$ -матрица A размером $k \times k$, $k \geq 3$, не имеющая одинаковых строк и столбцов, называется *циклической*, если в каждой строке и в каждом столбце содержится ровно две единицы.

Отметим, что если циклическая матрица A размером $k \times k$ не имеет циклической подматрицы, то она есть матрица инцидентий цикла из k вершин.

Вполне уравновешенной матрицей называется $(0,1)$ -матрица, не содержащая циклической подматрицы.

Полезность вполне уравновешенных матриц обуславливается наличием у них так называемого *гриди свойства*. Этому свойству будет дано несколько эквивалентных определений, первое из которых связано с понятием гриди матрицы.

Гриди матрицей называется $(0,1)$ -матрица, не содержащая подматриц вида

$$G_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \text{ и } G_2 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Понятно, что правильная матрица является гриди матрицей. Понятно также, что свойство матрицы быть гриди матрицей, как и в случае правильной матрицы, зависит от нумерации строк. При одной нумерации это свойство может не выполняться, а при другой иметь место. Поэтому, как и ранее для правильных матриц, возникает вопрос о приведении произвольной $(0,1)$ -матрицы к гриди матрице и поиске приводящей перестановки строк.

Подматрицу $(0,1)$ -матрицы $A = (a_{ij})$ ($i \in I, j \in J$) назовем i -усеченной, $1 \leq i \leq m$, и будем обозначать A_i , если эта подматрица получена из матрицы A удалением первых $i - 1$ строк. Использование i -усеченных подматриц помогает прояснить строение гриди матриц. Несложно понять, что произвольная $(0,1)$ -матрица A будет гриди матрицей тогда и только тогда, когда для любого i , $1 \leq i \leq m-1$, столбцы подматрицы A_i , имеющие единицу в i -ой строке, будут сравнимы по включению индикаторных множеств **единичных строк этих столбцов**.

Гриди матрицей в стандартной форме называется $(0,1)$ -матрица, не содержащая подматрицы вида

$$G_0 = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}.$$

Связь между гриди матрицами и гриди матрицами в стандартной форме устанавливает следующая

Лемма 3.3.1. Матрица A является гриди матрицей тогда и только тогда, когда перестановкой столбцов может быть приведена к гриди матрице в стандартной форме.

Доказательство. Пусть матрица A перестановкой столбцов приведена к гриди матрице в стандартной форме. Полученная матрица не содержит подматрицу G_0 , но тогда исходная матрица A не могла содержать подматриц G_1 и G_2 и, следовательно, A — гриди матрица. Обратно, пусть A — гриди матрица. Переставим столбцы этой матрицы в соответствии с лексикографическим порядком по последней несовпадающей компоненте вектора–столбца. Полученная в результате матрица A' будет гриди матрицей в стандартной форме. Действительно, если A' содержит подматрицу G_0 , то A' содержит и подматрицу G_2 . Но тогда исходная матрица A содержит либо подматрицу G_1 , либо подматрицу G_2 , что невозможно. Лемма доказана.

Из доказанного следует, что гриди матрица в стандартной форме отличается от гриди матрицы тем, что у всякой i -усеченной подматрицы A_i матрицы A в стандартной гриди форме столбцы, содержащие единицу в i -м столбце, не просто сравнимы между собой по включению индикаторных множеств единичных строк, но и упорядочены по возрастанию числа элементов в этих множествах.

Таким образом, наличие у $(0,1)$ -матрицы A гриди свойства будем связывать с возможностью ее приведения перестановкой строк к гриди матрице или перестановкой строк и столбцов к гриди матрице в стандартной форме.

Убедимся, что таким свойством обладают и вполне уравновешенные матрицы. Для этого используем понятие дважды лексически упорядоченной матрицы.

Дважды лексически упорядоченной называется $(0,1)$ -матрица $A = (a_{ij})$ ($i \in I, j \in J$), не имеющая одинаковых строк и столбцов, если

1. при $i < k$ имеем $a_{ij} = 0, a_{kj} = 1$, где j — наибольший номер столбца, для которого $a_{ij} \neq a_{kj}$.
2. при $j < l$ имеем $a_{ij} = 0, a_{il} = 1$, где i — наибольший номер строки, для которой $a_{ij} \neq a_{il}$.

Другими словами, дважды лексически упорядоченная матрица — это такая матрица, в которой лексически упорядочены одновременно и строки и столбцы, и при этом упорядочение производится не по первому, а по последнему несовпадающему элементу строк и столбцов.

Из ниже приводимой леммы следует, что дважды лексически упорядочить можно любую $(0,1)$ -матрицу.

Лемма 3.3.2 ([116']). Всякая $(0,1)$ -матрица, размера $m \times n$, не имеющая одинаковых строк и столбцов, приводима перестановкой строк и столбцов к дважды лексически упорядоченной матрице. Соответствующий алгоритм приведения имеет оценку трудоемкости $O(m^2n)$.

Следующая лемма показывает, какими свойствами обладает дважды лексически упорядоченная матрица, полученная из вполне уравновешенной.

Лемма 3.3.3 ([116']). Дважды лексически упорядоченная матрица, полученная из вполне уравновешенной матрицы, является гриди матрицей в стандартной форме.

Доказательство. Предположим противное и пусть дважды лексически упорядоченная матрица $A' = (a'_{ij}) (i \in I, j \in J)$, полученная из вполне уравновешенной матрицы A , содержит подматрицу

$$A'_2 = G_0 = \begin{matrix} & j_1 & j_2 \\ \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} & i_1 \\ & & i_2 \end{matrix}$$

с номерами строк и столбцов соответственно i_1, i_2 и j_1, j_2 . Пусть строка с номером i_3 — последняя строка, по которой различаются j_1 -й и j_2 -й столбцы, а столбец с номером j_3 — последний столбец, по которому различаются i_1 -я и i_2 -я строки. Поскольку A' — дважды лексически упорядоченная матрица, то $a'_{i_3 j_1} = 0, a'_{i_3 j_2} = 1$ и $a'_{i_1 j_3} = 0, a'_{i_2 j_3} = 1$. Таким образом, получаем матрицу

$$A'_3 = \begin{matrix} & j_1 & j_2 & j_3 \\ \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} & i_1 \\ & & & i_2, \\ & & & i_3 \end{matrix}$$

в которой $a'_{i_3 j_3} = 0$, поскольку в противном случае A'_3 была бы циклической матрицей.

Пусть далее строка с номером i_4 — последняя строка, по которой различаются j_2 -й и j_3 -й столбцы, а столбец с номером j_4 — последний столбец, по которому различаются i_2 -я и i_3 -я строки. Поскольку A' — дважды лексически упорядоченная матрица, то $a'_{i_4 j_2} = 0, a'_{i_4 j_3} = 1$ и $a'_{i_2 j_4} = 0, a'_{i_3 j_4} = 1$. В результате получаем матрицу

$$A'_4 = \begin{matrix} & j_1 & j_2 & j_3 & j_4 \\ \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} & i_1 \\ & & & & i_2 \\ & & & & i_3, \\ & & & & i_4 \end{matrix}$$

в которой $a'_{i_4 j_1} = 0$, так как по построению строка с номером i_3 — последняя строка, различающая j_1 -й и j_2 -й столбцы, $a'_{i_1 j_4} = 0$ по аналогичным причинам, а $a'_{i_4 j_4} = 0$, поскольку в противном случае матрица A'_4 была бы циклической.

Понятно, что рассмотренный процесс построения матриц $A'_k, k = 1, 2, 3, 4, \dots$ будет продолжаться до бесконечности, что противоречит конечности матрицы A' . Поэтому матрица A' не может содержать подматрицу G_0 . Лемма доказана.

Таким образом, вполне уравновешенная матрица обладает гриди свойством и перестановкой строк и столбцов приводима к гриди матрице в стандартной форме и переста-

новой строкой — к гриди матрице. Однако наличие гриди свойства у вполне уравновешенной матрицы — это не только необходимое свойство вполне уравновешенной матрицы, но и, как следует из приводимого ниже утверждения, также и достаточное.

Лемма 3.3.4. Если $(0,1)$ -матрица A приводима перестановкой строк к гриди матрице, то A есть вполне уравновешенная матрица.

Доказательство. Пусть матрица A' , полученная из исходной матрицы A , является гриди матрицей. Тогда она будет вполне уравновешенной матрицей, поскольку в противном случае матрица A' содержала бы циклическую подматрицу и, следовательно, одну из подматриц G_1 или G_2 .

Установленная эквивалентность гриди свойства и свойства матрицы быть вполне уравновешенной не исчерпывает перечень полезных определений гриди свойства. Еще одно такое определение связано с понятием Q -матрицы.

Q -матрицей называется $(0,1)$ -матрица A , если для любой ее подматрицы A' выполняется одно из двух:

1. A' содержит строку с одной единицей;
2. A' содержит два сравнимых столбца.

Лемма 3.3.5. Произвольная $(0,1)$ -матрица A есть Q -матрица тогда и только тогда, когда она является вполне уравновешенной.

Доказательство. Если A есть Q -матрица, то A не может содержать циклическую подматрицу, поскольку такая матрица не обладает ни одним из свойств, указанных в определении Q -матрицы.

Обратно. Пусть A есть вполне уравновешенная матрица и пусть A' — произвольная ее подматрица, которая, очевидно, является вполне уравновешенной. Тогда с точностью до перестановки строк матрицу A' можно считать гриди матрицей. Но тогда, если в первой строке матрицы A' содержится более одной единицы, то два столбца с единицами в первой строке сравнимы между собой. Лемма доказана.

Суммируя сказанное выше, получаем

Теорема 3.3.1. Для $(0,1)$ -матрицы A следующие условия эквивалентны:

1. Матрица A является вполне уравновешенной;
2. Матрица A приводима перестановкой строк к гриди матрице;
3. Матрица A приводима перестановкой строк и столбцов к гриди матрице в стандартной форме;
4. Матрица A является Q -матрицей.

С учетом эквивалентности указанных свойств $(0,1)$ -матрицы и, используя леммы 3.3.2 и 3.3.3, можно построить следующий алгоритм распознавания наличия гриди свойства у произвольной $(0,1)$ -матрицы A .

Алгоритм распознавания гриди свойства включает в себя два этапа. На первом исследуемая матрица A приводится к дважды лексически упорядоченной матрице A' . На втором этапе матрица A' проверяется на наличие подматрицы G_0 . Если такая подматрица обнаруживается, то алгоритм заканчивает работу и исходная матрица A не обладает гриди свойством. Если после проверки всех сочетаний пар столбцов и пар строк матрицы A'

подматрицы G_0 обнаружить не удастся, то алгоритм заканчивает работу и исходная матрица A обладает гриди свойством.

Поскольку второй этап алгоритма имеет оценку трудоемкости $O(m^2n^2)$, а трудоемкость первого этапа оценивается величиной $O(m^2n)$, то в целом трудоемкость алгоритма распознавания гриди свойства имеет оценку равную $O(m^2n^2)$.

3.2. Строение вполне уравновешенных матриц

Рассмотренные в теореме 3.3.1 эквивалентные определения вполне уравновешенных матриц дают некоторое представление об их строении. В частности, вполне уравновешенная матрица перестановкой строк приводима к гриди матрице, у которой любая i -усеченная подматрица состоит из столбцов таких, что любые два столбца, имеющие единицу в первой строке, сравнимы по включению индикаторных множеств единичных строк этих столбцов. Укажем на некоторые следствия из этого свойства, дающие дополнительную информацию о строении вполне уравновешенных матриц.

Лемма 3.3.6. Всякая вполне уравновешенная матрица с числом строк равным m не может иметь более $m - l + 1$ различных столбцов с числом единиц, равным l , $1 \leq l \leq m$, и может быть дополнена до минимальной по числу столбцов вполне уравновешенной матрицы, содержащей ровно $m - l + 1$ различных столбцов с числом единиц, равным l , $1 \leq l \leq m$.

Доказательство. Пусть исходная вполне уравновешенная матрица $A=(a_{ij})$ ($i \in I, j \in J$) является гриди матрицей. Заметим, что столбец с числом единиц, равным l , может иметь первую единицу только в строке с номером $i \leq m-l+1$. Кроме того, никакие два таких столбца не могут иметь первую единицу в одной, например, i -й строке, поскольку в i -усеченной подматрице соответствующие столбцы должны быть сравнимы. Отсюда следует первое утверждение леммы.

Для доказательства второго утверждения рассмотрим следующую процедуру пополнения исходной матрицы A до максимальной.

Очевидно, что исходная гриди матрица A может быть пополнена до матрицы, содержащей все столбцы с одной единицей и столбец, состоящий из единиц, и при этом остаться гриди матрицей. Предположим, что исходная матрица A содержит все столбцы с числом единиц равным $l-1$, $l \geq 2$, и не содержит столбец с числом единиц, равным l , и с первой единицей в i -й строке. Построим такой столбец и покажем, что после его присоединения к матрице A полученная матрица A' останется гриди матрицей.

Для этого рассмотрим столбец матрицы A (пусть это будет s -й столбец), который имеет $l-1$ единиц и первую единицу в i -й строке. Такой столбец, очевидно, существует и содержит хотя бы один нулевой элемент. Пусть i_0 , $i_0 \geq i$, — наибольший номер такой, что $a_{i_0s} = 1$, $a_{i_0+1s} = 0$. Рассмотрим i_0 -усеченную подматрицу A_{i_0} и выберем столбец этой матрицы (пусть это будет t -й столбец) с наименьшим индикаторным множеством X_t^1 таким, $X_s^1 \subset X_t^1$. Отметим, что столбец с необходимым свойством существует, поскольку

исходная матрица содержит столбец, состоящий из единиц. Рассмотрим s -й и t -й столбцы матрицы A_{i_0} и пусть k — наибольший номер строки, на которой различаются элементы s -го и t -го столбцов, то есть такой наибольший номер k , что $a_{ks} = 0$ и $a_{kt} = 1$. Построим столбец матрицы A , который отличается от s -го столбца тем, что в k -й строке имеет единицу. Припишем этому столбцу 0-й номер, присоединим его к матрице A и покажем, что полученная матрица A' является гриди матрицей.

Предположим противное и пусть матрица A' содержит подматрицу G_1 или G_2 . Возможны следующие четыре вида таких подматриц:

$$\begin{array}{ccc} \begin{array}{cc} 0 & j \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{array}{c} k \\ i_1 \\ i_2 \end{array} \end{array} & \begin{array}{cc} 0 & j \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} k \\ i_1 \\ i_2 \end{array} \end{array} & \begin{array}{cc} 0 & j \\ \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} & \begin{array}{c} i_1 \\ k \\ i_2 \end{array} \end{array} & \begin{array}{cc} 0 & j \\ \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} & \begin{array}{c} i_1 \\ i_2 \\ k \end{array} \end{array}$$

имеющих k -ю строку и 0-й столбец.

Покажем, что ни одна из указанных матриц не может быть подматрицей матрицы A . Действительно, поскольку $k > i_0$, то существование первой подматрицы противоречит выбору номера i_0 , как наибольшего номера такого, что $a_{i_0s} = 1$ и $a_{i_0+1s} = 0$. Чтобы исключить возможность существования второй подматрицы, дополним ее t -м столбцом и рассмотрим следующую подматрицу:

$$\begin{array}{ccc} 0 & j & t \\ \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} k \\ i_1 \\ i_2 \end{array} \end{array}$$

В силу выбора номера k , как максимального элемента, для которого $a_{ks} \neq a_{kt}$, в рассматриваемой матрице имеем $a_{kt} = 1$, $a_{i_1t} = 0$, $a_{i_2t} = 1$. Но тогда исходная матрица A содержит подматрицу G_2 и, следовательно, не является гриди матрицей.

Для исследования вопроса существования подматриц двух последних видов дополним их i_0 -й строкой и t -м столбцом. При этом сразу отметим, что в силу выбора элемента i_0 , как наибольшего, для которого $a_{i_0s} = 1$, $a_{i_0+1s} = 0$, имеем $i_0 \geq i_1$. Так что возможны лишь следующие варианты расширенных подматриц:

$$\begin{array}{ccc} \begin{array}{ccc} 0 & j & t \\ \begin{bmatrix} 1 & 1 & * \\ 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & * \end{bmatrix} & \begin{array}{c} i_1 \\ i_0 \\ k' \\ i_2 \end{array} \end{array} & \begin{array}{ccc} 0 & j & t \\ \begin{bmatrix} 1 & 1 & * \\ 1 & 1 & 1 \\ 0 & 1 & * \\ 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} i_1 \\ i_0 \\ i_2' \\ k \end{array} \end{array} & \begin{array}{ccc} 0 & j & t \\ \begin{bmatrix} 1 & 1 & * \\ 0 & 1 & * \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix} & \begin{array}{c} i_1 \\ i_2 \\ i_0' \\ k \end{array} \end{array}$$

где $*$ — произвольный элемент из множества $\{0, 1\}$.

Отметим сразу, что в приведенных матрицах $a_{i_0t} = 1$, поскольку в противном случае матрица A содержит подматрицу G_1 или G_2 и не является гриди матрицей. Отметим также, что первая матрица может быть отброшена, поскольку содержит подматрицу G_2 . Из вида двух других матриц вытекает, что если рассмотреть матрицу A_{i_0} , то получим $X_s^1 \subset X_j^1$. Отсюда, с учетом того, что $a_{kj} = 0$ и $a_{kt} = 1$ приходим к противоречию с выбором t -го столбца.

Таким образом, ни одна из возможных подматриц вида G_1 или G_2 не может быть подматрицей матрицы A' и, следовательно, эта матрица является гриди матрицей. Лемма доказана.

Следующее утверждение указывает на тесную связь вполне уравновешенных матриц и деревьев. Это не удивительно, поскольку, как было замечено ранее, характеристическая матрица H семейства ориентированных цепей корневого дерева обладает тем свойством, что матрица \bar{H} является гриди матрицей.

Пусть A'_2 — подматрица максимальной вполне уравновешенной матрицы, состоящей из столбцов с числом единиц равным двум.

Лемма 3.3.7. Подматрица A'_2 является матрицей инцидентий некоторого дерева.

Доказательство. Если матрица A'_2 имеет m строк, то число ее столбцов равняется $m - 1$. Поэтому соответствующий граф имеет m вершин и $m - 1$ ребер. Кроме того, этот граф не имеет циклов, поскольку A'_2 не содержит циклическую подматрицу. Следовательно, рассматриваемый граф является деревом. Лемма доказана.

Лемма 3.3.8. Если A — вполне уравновешенная матрица, то матрица \bar{A} есть характеристическая матрица семейства поддеревьев некоторого дерева.

Доказательство. Дополним матрицу A до максимальной вполне уравновешенной матрицы A' и рассмотрим дерево, порождаемое, согласно предыдущей лемме, матрицей A'_2 . Отметим, что каждый столбец матрицы A'_2 соответствует некоторому ребру дерева и указывает на две вершины, которым это ребро инцидентно. Возьмем j -й столбец матрицы A , не принадлежащий подматрице A'_2 . Покажем, что индикаторное множество X_j^1 есть множество вершин некоторого поддерева рассматриваемого дерева. Предположим, что это не так и пусть существуют вершины $i, k \in X_j^1$ такие, что единственная цепь, соединяющая вершины i и k , содержит вершины, не принадлежащие множеству X_j^1 . Но тогда существуют вершины $i', k' \in X_j^1$ такие, что единственный путь, соединяющий вершины i' и k' , проходит через вершины, ни одна из которых не принадлежит множеству X_j^1 . Пусть I' — множество строк, соответствующих этим вершинам, а J' — множество столбцов матрицы A'_2 , соответствующих ребрам пути, соединяющего вершины i' и k' . Рассмотрим подматрицу матрицы A' , включающую строки из множества I' и столбцы из множества

$J' \cup \{j\}$. Эта подматрица, как легко видеть, есть циклическая матрица, что противоречит тому, что A' является вполне уравновешенной матрицей.

Таким образом, получаем, что у произвольного j -го столбца матрицы A множество X_j^1 соответствует множеству вершин некоторого поддерева. Это означает, что матрица \bar{A} есть характеристическая матрица семейства поддеревьев некоторого дерева. Лемма доказана.

Из доказанного получаем, что всякая вполне уравновешенная матрица реализуется как семейство поддеревьев некоторого дерева. Напомним, что правильные матрицы и почти правильные реализуются как семейства подцепей соответственно некоторой цепи и некоторого цикла. При этом характеристическая матрица любого семейства подцепей некоторой цепи или цикла является соответственно правильной или почти правильной матрицей. К сожалению, для вполне уравновешенных матриц, в отличие от правильных и почти правильных, обратное утверждение не верно и не для всякого семейства поддеревьев характеристическая матрица превращается во вполне уравновешенную. Чтобы получить вполне уравновешенную матрицу \bar{H} , семейство поддеревьев должно обладать дополнительным свойством. Это свойство легче всего сформулировать, опираясь на определение Q -матрицы. Для всякого подсемейства рассматриваемого семейства поддеревьев должна либо найтись вершина, принадлежащая только одному поддереву, либо должны найтись два поддерева, одно из которых является поддеревом другого.

В качестве примера семейства поддеревьев, порождающего вполне уравновешенную матрицу, рассмотрим семейство так называемых деревьев соседства.

Пусть $T = (I, E)$ — взвешенное дерево с множеством вершин $I = \{1, \dots, m\}$ и с неотрицательными весами ребер $d_e, e \in E$. Расстояние $d(i, j)$ между вершинами $i, j \in I$ определим как сумму весов ребер, составляющих единственный путь из вершины i в вершину j . Пусть

r_0 — некоторая неотрицательная величина. Поддеревом соседства с вершиной — центром $j \in I$ и радиусом r_0 назовем поддерево $T(j, r_0) = (I', E')$ с множеством вершин $I' = \{i \in I \mid d(i, j) \leq r_0\}$ и соответствующим множеством ребер. Таким образом, каждой вершине $j \in I$ ставится в соответствие не более m различных поддеревьев, каждое из которых включает в себя некоторое количество вершин, расстояние от которых до вершины j не превосходит величины радиуса. Поддерево с вершиной — центром j может состоять, в частности, из одной вершины j .

Справедливо следующее утверждение, из которого вытекает, что семейство деревьев соседства порождает вполне уравновешенную матрицу.

Лемма 3.3.9 [162]. Всякое семейство T_1, \dots, T_k поддеревьев соседства дерева T обладает тем свойством, что либо найдется вершина дерева T , принадлежащая только одному из поддеревьев семейства, либо в рассматриваемом семействе найдется два дерева, одно из которых является поддеревом другого.

Доказательство. Обозначим через I_1, \dots, I_k множества вершин деревьев T_1, \dots, T_k и рассмотрим подграф T' дерева T с множеством вершин $I' = I_1 \cup \dots \cup I_k$. Можно считать,

что подграф T' является деревом, поскольку в противном случае рассмотрим любую связную компоненту подграфа T' , которая будет деревом. Предположим, что для дерева T' первое свойство, указанное в формулировке леммы, не выполняется и, следовательно, каждая вершина дерева T' принадлежит, по крайней мере, двум множествам из семейства I_1, \dots, I_k . Рассмотрим в дереве T' цепь наибольшей длины и пусть i и i' — соответственно начальная и конечная вершина этой цепи. Вершина i является концевой вершиной дерева T' и принадлежит не менее двум подмножествам из семейства I_1, \dots, I_k . Пусть для определенности $i \in I_1$ и $i \in I_2$ и пусть j_1 — центр дерева T_1 , а j_2 — центр дерева T_2 . Предположим, что для деревьев T_1, T_2 не выполняется второй вариант альтернативы и найдутся вершины $i_1 \in I_1$ и $i_2 \in I_2$ такие, что $i_1 \notin I_2$ и $i_2 \notin I_1$. Из вершины i_1 и вершины i_2 существуют единственные цепи в вершину i .

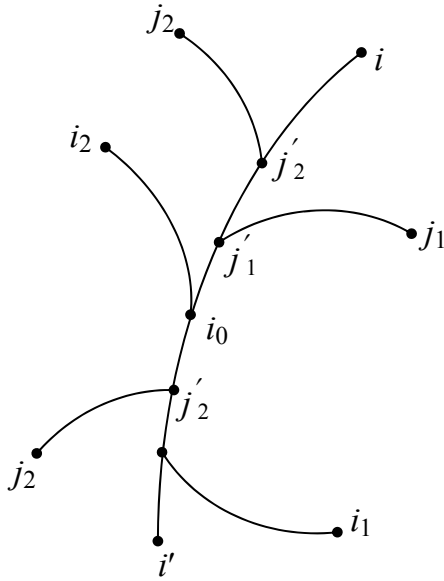


Рис. 3.3.1

Пусть i'_1 — вершина, принадлежащая одновременно цепи из i_1 в i и цепи из i' в i , и при этом максимально удаленная от вершины i . Аналогично, пусть i'_2 — вершина, принадлежащая цепи из i_2 в i и цепи из i' в i и максимально удаленная от вершины i . Будем считать для определенности, что $d(i'_1, i) \geq d(i'_2, i)$ и

обозначим вершину i'_2 через i_0 (рис. 3.3.1). Рассмотрим цепь из j_1 в i и пусть j'_1 — вершина этой цепи, принадлежащая также цепи из i' в i и максимально удаленная от вершины i . Поскольку цепь из i' в i — цепь максимальной длины, то $d(i_0, i) \geq d(i_0, i_2)$. Отсюда, с учетом того, что $i_2 \notin I_1$, получаем $d(i_0, i) \geq d(j'_1, i)$ и, следовательно $d(i_0, i_2) \geq d(i_0, i_1)$. Отсюда, с учетом неравенства $d(i_0, i) \geq d(i_0, i_2)$, получаем $d(i_0, i) > d(i_0, i_1)$. Рассмотрим цепь из j_2 в i и пусть j'_2 — вершина этой цепи, принадлежащая цепи из i' в i и максимально удаленная от вершины i . Возможны два случая: $d(i_0, i) \geq d(j'_2, i)$ и $d(i_0, i) < d(j'_2, i)$. В первом случае, поскольку $d(i_0, i_2) > d(i_0, i_1)$, получаем $i_1 \in I_2$. Во втором случае, в силу того, что $d(i_0, i_1) < d(i_0, i)$, так же получаем $i_1 \in I_2$. Значит, для деревьев T_1 и T_2 выполняется второй вариант альтернативы. Лемма доказана.

Из доказанного следует, что если H характеристическая матрица семейства поддеревьев соседства T_1, \dots, T_k , то матрица \bar{H} является Q -матрицей.

Поскольку вполне уравновешенные матрицы реализуются как характеристические матрицы семейства поддеревьев, а поддеревья соседства порождают вполне уравновешенные матрицы, то может возникнуть вопрос о покрытии поддеревьями соседства все-

возможных порождений вполне уравновешенных матриц. В [82'] приведена вполне уравновешенная матрица A_6

$$A_6 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

такая, что матрица \bar{A}_6 не может быть характеристической матрицей никакого семейства поддеревьев соседства. Легко видеть, что матрица \bar{A}_6 есть характеристическая матрица

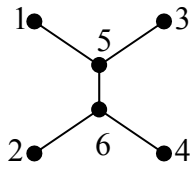


Рис. 3.3.2.

семейства поддеревьев следующего дерева (рис. 3.3.2). Соответствующие 6 поддеревьев имеют следующие множества вершин: $I_1 = \{1,5\}$, $I_2 = \{3,5\}$, $I_3 = \{2,6\}$, $I_4 = \{4,6\}$, $I_5 = \{1,2,5,6\}$, $I_6 = \{3,4,5,6\}$. При этом никакие веса, приписываемые дугам рассматриваемого дерева, не смогут превратить поддеревья с множествами вершин I_5 и I_6 в поддеревья соседства.

3.3. Гриди алгоритм

Приводимый ниже алгоритм применяется к задаче FL , представленной в виде задачи SC . Заменяя условие целочисленности переменных задачи SC , соответствующей исходной задаче FL , на условие их неотрицательности, превратим задачу SC в следующую задачу линейного программирования:

$$\begin{aligned} \min_{(x_i), (w_s)} & \sum_{i \in I} f_i x_i + \sum_{s=1}^S b_s w_s; \\ & \sum_{i \in I} a_{is} x_i + w_s \geq 1, \quad s = 1, \dots, S; \\ & x_i, w_s \geq 0, \quad i \in I, \quad s = 1, \dots, S. \end{aligned}$$

Рассмотрим задачу, двойственную к данной, называемую задачей об упаковке и имеющую следующий вид:

$$\begin{aligned} \max_{(u_s)} & \sum_{s=1}^S u_s; \\ & \sum_{s=1}^S a_{is} u_s \leq f_i, \quad i \in I; \\ & 0 \leq u_s \leq b_s, \quad s = 1, \dots, S. \end{aligned}$$

Считая матрицу $A = (a_{is})$ ($i \in I, s = 1, \dots, S$) гриди матрицей в стандартной форме, приведем алгоритм решения рассмотренной пары двойственных задач, называемый гриди алгоритмом.

Гриды алгоритм состоит из двух этапов. На первом этапе последовательно вычисляются значения переменных u_s , $s = 1, \dots, S$, двойственной задачи и одновременно строятся множества $I_0 \subset I$ и $S_0 \subset \{1, \dots, S\}$. По этим множествам на втором этапе определяется множество I^* , с использованием которого строится решение (x_i) , (w_s) прямой задачи.

Первый этап включает S однотипных шагов. Пусть к очередному s -му, $s = 1, \dots, S$, шагу вычислены значения переменных u_1, \dots, u_{s-1} и определены множества I_0 и S_0 . На первом шаге $I_0 = \emptyset$ и $S_0 = \emptyset$. Шаг начинается с вычисления величины u_s по формуле

$$u_s = \min \left\{ b_s; \min_{i \in I | a_{is}=1} \left\{ f_i - \sum_{t=1}^{s-1} a_{it} u_t \right\} \right\}.$$

Если $u_s > 0$ и

$$u_s = \min_{i \in I | a_{is}=1} \left\{ f_i - \sum_{t=1}^{s-1} a_{it} u_t \right\},$$

то выбирается наибольший элемент i_0 , для которого выполняется последнее равенство, и полагается $I_0 \leftarrow I_0 \cup \{i_0\}$, $S_0 \leftarrow S_0 \cup \{s\}$. При этом будем говорить, что переменная u_s насыщает i_0 -е ограничение, а индекс переменной u_s обозначать через $s(i)$. После этого, если $s < S$, начинается следующий шаг, а если $s = S$, то осуществляется переход ко второму этапу.

Второй этап включает в себя конечное число однотипных шагов, на каждом из которых изменяется множество I_0 , полученное на первом этапе, и строится множество I^* . Пусть к очередному шагу имеются множества $I_0 \neq \emptyset$ и I^* . На первом шаге I_0 — множество, полученное на первом этапе, а $I^* = \emptyset$. Шаг начинается с выбора наибольшего элемента i_0 множества I_0 . Если $\sum_{i \in I^*} a_{is(i_0)} = 0$, то полагается $I^* \leftarrow I^* \cup \{i_0\}$. В любом случае полагается $I_0 \leftarrow I_0 \setminus \{i\}$ и, если $I_0 \neq \emptyset$, начинается следующий шаг. Если же $I_0 = \emptyset$, то полагается

$$x_i = \begin{cases} 1, & \text{если } i \in I^*, \\ 0 & \text{иначе;} \end{cases}$$

$$w_s = \begin{cases} 1, & \text{если } \sum_{i \in I^*} a_{is} = 0, \\ 0 & \text{иначе;} \end{cases}$$

и после этого второй этап и в целом алгоритм считаются законченными.

Построенные решения (u_s) и (x_i) , (w_s) рассматриваемой пары двойственных задач, очевидно, будут допустимыми решениями этих задач. Для доказательства их оптимальности убедимся предварительно в справедливости следующего вспомогательного утверждения.

Пусть

$$S_1 = \{s \notin S_0 \mid u_s = 0\},$$

$$S_2 = \{s \notin S_0 \mid u_s > 0\}.$$

Лемма 3.3.10. Для построенного гриди алгоритмом множества I^* справедливы соотношения

$$\sum_{i \in I^*} a_{is} \begin{cases} = 1 & \text{для } s \in S_0, \\ \geq 1 & \text{для } s \in S_1, \\ \leq 1 & \text{для } s \in S_2. \end{cases}$$

Доказательство. Пусть $s \in S_1$. Поскольку $u_s = 0$, то существует $i_0 \in I_0$ такой, что $a_{i_0 s} = 1$ и $s(i_0) < S$. Если $i_0 \in I^*$, то требуемое показано. Если $i_0 \notin I^*$, то найдется элемент $k \in I^*$, $k > i_0$, такой, что $a_{ks(i_0)} = 1$. Но тогда имеем $a_{ks} = 1$, поскольку в противном случае матрица A содержит подматрицу G_0 . Следовательно, $\sum_{i \in I^*} a_{is} \geq 1$, если $s \in S_1$.

Пусть теперь $s \in S_2$. Предположим, что найдутся $i, k \in I^*$, $i < k$, такие, что $a_{is} = 1$ и $a_{ik} = 1$. Поскольку $i \in I_0$, то $s(i) > s$ и, следовательно, $a_{is(i)} = 1$. Если предположить, что $a_{ks(i)} = 1$, то придем к противоречию с тем, что $i \in I^*$. Если же $a_{ks(i)} = 0$, то матрица A содержит подматрицу G_0 , что также невозможно. Таким образом $\sum_{i \in I^*} a_{is} \leq 1$, когда $s \in S_2$.

Пусть, наконец, $s \in S_0$ и пусть переменная u_s насыщает i_0 -е ограничение. Заметим, прежде всего, что $\sum_{i \in I^*, i > i_0} a_{is} \leq 1$. Действительно, если найдутся $i, k \in I^*$, $i_0 < i < k$, такие, что

$a_{is} = 1$ и $a_{ik} = 1$, то придем к противоречию точно также, как и при рассмотрении случая $s \in S_2$. Отсюда с учетом правила построения множества I^* получаем, что $\sum_{i \in I^*, i \geq i_0} a_{is} = 1$. Заме-

тим далее, что $\sum_{i \in I^*, i < i_0} a_{is} = 0$. Действительно, пусть $i \in I^*$, $i < i_0$ и пусть $a_{is} = 1$. Поскольку

$i \in I_0$, то $s(i) > s$ и $a_{is(i)} = 1$. Если предположить, что $a_{i_0 s(i)} = 1$, то придем к противоречию с тем, что $s(i) \in S_0$. Если же $a_{i_0 s(i)} = 0$, то матрица A содержит подматрицу G_0 , что также невозможно. Таким образом, $\sum_{i \in I^*} a_{is} = 1$, если $s \in S_0$. Лемма доказана.

Теорема 3.3.2. Построенные гриди алгоритмом решения (u_s) и (x_i) , (w_s) рассматриваемой пары двойственных задач являются оптимальными решениями этих задач.

Доказательство. Покажем, что для рассматриваемых решений выполняются соотношения дополняющей нежесткости, имеющие вид:

$$\begin{aligned} x_i \left(\sum_{s=1}^S a_{is} u_s - f_i \right) &= 0, \quad i \in I; \\ u_s \left(1 - \sum_{i \in I} a_{is} x_i - w_s \right) &= 0, \quad s = 1, \dots, S; \end{aligned}$$

$$w_s(b_s - u_s) = 0, \quad s = 1, \dots, S.$$

Первое равенство, очевидно, выполняется, поскольку, если $i \notin I^*$, то $x_i = 0$, а если $i \in I^*$, то $\sum_{s=1}^S a_{is} u_s - f_i = 0$. Второе равенство, очевидно верно, для $s \in S_1$ и выполняется для $s \notin S_1$ в силу предыдущей леммы и определения значений переменных w_s , $s = 1, \dots, S$. Наконец, третье равенство выполняется для $s \in S_0 \cup S_1$, поскольку в силу предыдущей леммы $w_s = 0$, и справедливо для $s \in S_2$, поскольку $u_s = b_s$. Теорема доказана.

Таким образом, в результате работы гриди алгоритма получается оптимальное решение (x_i) , (w_s) релаксированной задачи SC . Это решение — целочисленное и, следовательно, является оптимальным решением исходной задачи SC , а значит и задачи FL .

Нетрудно оценить трудоемкость представленного алгоритма решения задачи FL . Поскольку трудоемкость первого этапа оценивается величиной $O(mS)$, а второго — величиной $O(m)$, то алгоритм в целом имеет оценку трудоемкости $O(mS)$. Если переписать эту оценку с использованием параметров размерности задачи FL , то получаем, что рассмотренный гриди алгоритм решения задачи FL имеет оценку трудоемкости $O(m^2 n)$.

В заключение отметим, что описанный выше гриди алгоритм решения задачи FL может быть построен без привлечения вспомогательной задачи SC , а в результате рассмотрения непосредственно задачи FL . Только при этом матрица транспортных затрат C должна быть представлена в канонической форме и рассматриваться должна не задача FL с матрицей C , а эквивалентная ей задача FL с матрицей $\tilde{C} = (b_s \cdot h_{is})$ ($i \in I, s = 1, \dots, S$).

Для задачи FL с матрицей транспортных затрат \tilde{C} рассмотрим задачу, эквивалентную задаче DFL и отличающуюся от нее тем, что в ней фигурируют только тупиковые решения задачи DFL . Такая задача, как известно, имеет следующий вид:

$$\begin{aligned} & \max_{(u_s)} \sum_{s=1}^S u_s; \\ & \sum_{s=1}^S (1 - h_{is}) u_s \leq f_s, \quad s = 1, \dots, S; \\ & 0 \leq u_s \leq b_s, \quad s = 1, \dots, S, \end{aligned}$$

то есть является задачей об упаковке.

Представленный выше алгоритм решения этой задачи можно теперь рассматривать как алгоритм построения тупикового решения задачи DFL и определения блокирующего множества I^* . То, что множество I^* , построенное в результате работы гриди алгоритма, является блокирующим, то есть таким, что $I^* \cap I_s \neq \emptyset$ для всякого $s = 1, \dots, S$, вытекает из построения тупикового решения задачи DFL и построения самого множества I^* . Заметим, кроме того, что дефект множества I^* равен нулю, поскольку, как следует из доказанной выше леммы 3.3.10, неравенство $|I^* \cap I_s^+| \leq 1$ выполняется для всякого $s = 1, \dots, S$. Тогда из

леммы 2.2.1 вытекает, что множество I^* является оптимальным решением задачи $MINF_0$ с матрицей \tilde{C} .

Таким образом, в случае, когда у матрицы транспортных затрат C характеристическая матрица H такова, что матрица \bar{H} является вполне уравновешенной, рассмотренный выше гриди алгоритм дает тупиковое решение (u_s) задачи DFL и блокирующее множество I^* , которое порождает оптимальное решение задачи FL .

3.4. Алгоритм минимизации вполне уравновешенного полинома с неотрицательными коэффициентами

Полином $P_0(y_1, \dots, y_m)$ с неотрицательными коэффициентами и характеристической матрицей H такой, что матрица \bar{H} является вполне уравновешенной, назовем *вполне уравновешенным*.

Для задачи минимизации вполне уравновешенного полинома, так же как для задачи минимизации правильного полинома, удастся построить эффективный алгоритм на базе псевдобулевого программирования. Это оказывается возможным поскольку, по аналогии со случаем правильных полиномов, удастся в явном виде построить функцию, осуществляющую редукцию исходного вполне уравновешенного полинома к аналогичному полиному с числом переменных на единицу меньше и таким, что оптимальные значения переменных задачи минимизации построенного полинома являются оптимальными значениями тех же переменных задачи минимизации исходного полинома.

Указанное свойство вполне уравновешенного полинома устанавливается следующей теоремой, обобщающей теорему 3.1.2.

Теорема 3.3.3. Задача минимизации вполне уравновешенного полинома с неотрицательными коэффициентами сводится к задаче минимизации аналогичного полинома, но с числом переменных на единицу меньше.

Доказательство. Рассмотрим полином $P_0(y_1, \dots, y_m)$ с неотрицательными коэффициентами, имеющий характеристическую матрицу $H=(h_{is})$, $(i \in I, s=1, \dots, S)$, и запишем его следующим образом:

$$P_0(y_1, \dots, y_m) = \sum_{i \in I} f_i (1 - y_i) + \sum_{s=1}^S b_s \prod_{i|h_{is}=0} y_i.$$

Если матрица \bar{H} является гриди матрицей, то все столбцы этой матрицы, имеющие единицу в первой строке, сравнимы по включению множеств **единичных строк данных столбцов**. Поэтому рассмотрим перестановку (i_1, \dots, i_m) , $i_1=1$, строк матрицы \bar{H} по невозрастанию числа единиц в строках указанных столбцов и представим полином $P_0(y_1, \dots, y_m)$ в следующем виде:

$$P_0(y_1, \dots, y_m) = y_1 (-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k}) + \eta(y_2, \dots, y_m),$$

где каждый коэффициент a_k , $1 \leq k \leq m$, равняется либо соответствующему коэффициенту b_s , $1 \leq s \leq S$, либо равен нулю, а полином $r_1(y_2, \dots, y_m)$ включает в себя все члены полинома $P_0(y_1, \dots, y_m)$, не содержащие переменной y_1 .

В соответствии с основной идеей метода псевдоболева программирования построим функцию $y_1(y_2, \dots, y_m)$ от $(0,1)$ -переменных y_2, \dots, y_m такую, что

$$y_1(y_2, \dots, y_m) = \begin{cases} 0, & \text{если } -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} > 0, \\ 1, & \text{если } -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} < 0. \end{cases}$$

Для этого рассмотрим функцию

$$\rho(l) = -f_1 + \sum_{k=1}^l a_k, \quad 1 \leq l \leq m,$$

и определим такой наименьший номер l_1 , $1 \leq l_1 \leq m$, что $\rho(l_1) > 0$. Если $\rho(m) \leq 0$, то считаем, что $l_1 = m+1$.

Положим

$$y_1(y_2, \dots, y_m) = \begin{cases} 0, & \text{если } l_1 = 1, \\ 1 - y_{i_2} \cdot \dots \cdot y_{i_{l_1}}, & \text{если } 1 < l_1 \leq m, \\ 1, & \text{если } l_1 = m+1, \end{cases}$$

и покажем, что эта функция обладает требуемым свойством.

Рассмотрим произвольный неединичный $(0,1)$ -вектор (y_2, \dots, y_m) и пусть q , $2 \leq q \leq m$, — наименьший номер такой, что $y_{i_q} = 0$. Поскольку

$$-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} = -f_1 + \sum_{k=1}^{q-1} a_k = \rho(q-1),$$

то исследуем зависимость между знаком величины $\rho(q-1)$ и значением функции $y_1(y_2, \dots, y_m)$.

Если $\rho(q-1) > 0$, то $l_1 \leq q-1$. Когда $l_1 = 1$, имеем по определению $y_1(y_2, \dots, y_m) = 0$, а когда $l_1 > 1$, получаем $y_1(y_2, \dots, y_m) = 1 - y_{i_2} \cdot \dots \cdot y_{i_{l_1}} = 0$. Пусть $\rho(q-1) < 0$. Тогда $l_1 \geq q$ и $y_1(y_2, \dots, y_m) = 1$. Таким образом, в случае неединичного вектора (y_2, \dots, y_m) функция $y_1(y_2, \dots, y_m)$ обладает нужным свойством.

Рассмотрим единичный вектор (y_2, \dots, y_m) . В этом случае исследуем зависимость между знаком величины $\rho(m)$ и значением функции $y_1(y_2, \dots, y_m)$. Если $\rho(m) > 0$, то $l_1 \leq m$ и $y_1(y_2, \dots, y_m) = 0$. Если же $\rho(m) < 0$, то $l_1 = m+1$ и, следовательно, $y_1(y_2, \dots, y_m) = 1$.

Таким образом, получаем полином

$$P_2(y_2, \dots, y_m) = y_1(y_2, \dots, y_m)(-f_1 + \sum_{k=2}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k}) + r_1(y_2, \dots, y_m),$$

который по построению обладает тем свойством, что если (y_2^*, \dots, y_m^*) — оптимальное решение задачи минимизации этого полинома, то $(0, 1)$ -вектор $(y_1(y_2^*, \dots, y_m^*), y_2^*, \dots, y_m^*)$ является оптимальным решением задачи минимизации исходного полинома $P_0(y_1, \dots, y_m)$.

Покажем, что $P'_0(y_2, \dots, y_m)$ есть вполне уравновешенный полином с неотрицательными коэффициентами. Для этого рассмотрим полином

$$y_1(y_2, \dots, y_m) = (-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k}).$$

Если $y_1(y_2, \dots, y_m) = 1 - y_{i_1} \cdot \dots \cdot y_{i_{l_1}}$, то можно написать

$$\begin{aligned} (1 - y_{i_1} \cdot \dots \cdot y_{i_{l_1}}) &= (-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k}) = \\ &= -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} + \left(f_1 - \sum_{k=1}^{l_1-1} a_k \right) y_{i_2} \cdot \dots \cdot y_{i_k} - \sum_{k=l_1+1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} = \\ &= -f_1 + \sum_{k=1}^{l_1-1} a_k y_{i_2} \cdot \dots \cdot y_{i_k} + \left(f_1 - \sum_{k=1}^{l_1-1} a_k \right) y_{i_2} \cdot \dots \cdot y_{i_{l_1}}. \end{aligned}$$

Отсюда следует, что характеристическая матрица H' полинома $P'_0(y_2, \dots, y_m)$ получается из матрицы H вычеркиванием первой строки и некоторых столбцов, **имеющих ноль в первой строке**. Таким образом \bar{H}' — подматрица матрицы \bar{H} и, следовательно, является вполне уравновешенной. Кроме того, так как $\rho(l_1 - 1) \leq 0$, то коэффициенты полинома $P'_0(y_2, \dots, y_m)$ неотрицательны. К такому же результату приходим, когда $y_1(y_2, \dots, y_m) = 0$ или $y_1(y_2, \dots, y_m) = 1$. Теорема доказана.

Из приведенного доказательства ясно, как устроен предлагаемый алгоритм минимизации вполне уравновешенного полинома или решения задачи FL с матрицей транспортных затрат C , имеющей характеристическую матрицу H такую, что матрица \bar{H} является вполне уравновешенной. Приводимое ниже более подробное описание алгоритма предполагает, что по исходной паре матриц (F_0, C) , где C — матрица размера $m \times n$ построена характеристическая матрица $H = (h_{is})$ ($i \in I, s = 1, \dots, S$) такая, что матрица \bar{H} есть гриди матрица в стандартной форме, и вычислены веса $b_s, s = 1, \dots, S$, столбцов характеристической матрицы. В ходе работы алгоритма по исходному полиному $P_0(y_1, \dots, y_m)$ последовательно строятся полиномы $P_k(y_k, \dots, y_m), k = 2, \dots, m$, с характеристическими матрицами $H_k = (h_{is})$ ($i = k, \dots, m, s = 1, \dots, S$) и коэффициентами $b_s, s = 1, \dots, S$, которые изменяются при переходе от одного полинома к другому и отличаются от исходных, в частности тем, что среди них могут быть нулевые элементы.

Алгоритм минимизации вполне уравновешенного полинома состоит из двух этапов. В результате первого этапа для каждой переменной $y_i, i \in I$, строятся формулы для вычисления оптимального значения этой переменной по оптимальным значениям переменных с

большими номерами. На втором этапе с использованием указанных формул последовательно восстанавливаются оптимальные значения переменных y_i , $i = m, m-1, \dots, 1$.

Первый этап состоит из m однотипных шагов.

На первом шаге рассматриваются исходные коэффициенты b_s , $s = 1, \dots, S$. Шаг начинается с поиска наименьшего номера $s(1)$, $1 \leq s(1) \leq S$, такого, что

$$-f_1 + \sum_{s=1}^{s(1)} b_s(1 - h_{1s}) > 0.$$

Если такого номера нет, то считается $s(1) = S+1$. Далее полагается $b_{s(1)} = \sum_{s=1}^{s(1)-1} b_s(1 - h_{1s})$ и

$b_s = 0$ для $s > s(1)$ таких, что $h_{1s} = 0$. После этого начинается второй шаг. На очередном k -м, $k = 2, \dots, m-1$, шаге рассматриваются коэффициенты b_s , $s = 1, \dots, S$, частично измененные на предыдущих шагах. Шаг начинается с процедуры «приведения подобных членов» полинома $P_k(y_k, \dots, y_m)$. Для этого для всякого s , $s = 1, \dots, S-1$, такого, что $h_{ks} = 0$, в случае, когда $h_{ks+1} = 0$ и $\sum_{i=k}^m h_{is} = \sum_{i=k}^m h_{is+1}$, полагается $b_{s+1} \leftarrow b_s + b_{s+1}$, $b_s = 0$. Далее отыскивается наименьший номер $s(k)$, $1 \leq s(k) \leq S$, такой, что

$$-f_k + \sum_{s=1}^{s(k)} b_s(1 - h_{ks}) > 0.$$

Если такого номера нет, то полагается $s(k) = S+1$. Далее полагается $b_{s(k)} = \sum_{s=1}^{s(k)-1} b_s(1 - h_{ks})$ и

$b_s = 0$ для $s > s(k)$ таких, что $h_{ks} = 0$. После этого начинается следующий шаг.

На m -м заключительном шаге рассматриваются коэффициенты b_s , $s = 1, \dots, S$, вычисленные на предыдущих шагах. Если $-f_m + \sum_{s=1}^S b_s(1 - h_{ms}) > 0$, то полагается $s(m) = S$, в противном случае считается, что $s(m) = S+1$. После этого первый этап завершается и начинается второй.

Второй этап алгоритма также состоит из m шагов. На первом шаге определяется оптимальное значение y_m^* переменной y_m следующим образом:

$$y_m^* = \begin{cases} 1, & \text{если } s(m) = S+1, \\ 0, & \text{если } s(m) = S. \end{cases}$$

После этого начинается второй шаг.

На очередном $(m-k+1)$ -м, $k = m, m-1, \dots, 1$, шаге имеются оптимальные значения y_{k+1}^*, \dots, y_m^* соответствующих переменных, вычисленных на предыдущих шагах, и определяется оптимальное значение y_k^* переменной y_k . Полагается $y_k^* = 1$, если $s(k) = S+1$, и

$$y_k^* = 1 - \prod_{i>k|h_{is}(k)=0} y_i^*$$

в противном случае. После этого, если $k < m$, начинается следующий шаг и, если $k = m$, работа алгоритма заканчивается.

Трудоемкость каждого шага первого этапа алгоритма, как несложно увидеть, оценивается величиной $O(S)$, поэтому оценкой трудоемкости этапа служит величина $O(m \cdot S)$. Трудоемкость каждого шага второго этапа можно оценить величиной $O(m)$, и, следовательно, трудоемкость второго этапа оценивается величиной $O(m^2)$, а трудоемкость алгоритма в целом — величиной $O(mS + m^2)$. Отсюда, с учетом того, что $S \leq m \cdot n$, где $m \times n$ — размер матрицы транспортных затрат C задачи FL , получаем, что алгоритм решения задачи FL с матрицей C такой, что \bar{H} — вполне уравновешенная матрица, имеет оценку трудоемкости $O(m^2 n)$.

3.5. Алгоритм минимизации вполне уравновешенного полинома с коэффициентами произвольного знака

Полином $P(y_1, \dots, y_m)$, полученный из вполне уравновешенного полинома $P_0(y_1, \dots, y_m)$ с неотрицательными коэффициентами в результате замены данных коэффициентов на производные коэффициенты разных знаков, будем называть *вполне уравновешенным полиномом с коэффициентами произвольного знака* или просто *вполне уравновешенным полиномом*.

Доказанная выше теорема 3.3.3 остается справедливой и в случае вполне уравновешенного полинома $P(y_1, \dots, y_m)$. Тем самым и для более общего случая полиномов открываются возможности для построения эффективного алгоритма минимизации.

Теорема 3.4.4. Задача минимизации вполне уравновешенного полинома сводится к задаче минимизации аналогичного полинома, но с числом переменных на единицу меньше.

Доказательство отличается от доказательства теоремы 3.3.3 только в части, связанной с более сложным видом функции $y_1(y_2, \dots, y_m)$. Для ее построения, как и ранее представим исходный полином $P(y_1, \dots, y_m)$ в виде

$$P(y_1, \dots, y_m) = y_1(-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k}) + r_1(y_2, \dots, y_m),$$

где каждый коэффициент a_k , $1 \leq k \leq m$, равняется либо соответствующему коэффициенту b_s , $1 \leq s \leq S$, либо равен нулю, а полином $r_1(y_2, \dots, y_m)$ включает в себя все члены полинома $P(y_1, \dots, y_m)$, не содержащие переменной y_1 . Далее рассмотрим вспомогательную функцию

$$\rho(l) = -f_1 + \sum_{k=1}^l a_k, \quad 1 \leq l \leq m,$$

которая в отличие от предыдущей функции не является неубывающей, а может несколько раз менять знак. Обозначим через l_1, l_2, \dots, l_T , ($0 = l_0 < l_1 < \dots < l_T < l_{T+1} = m+1$), $T \geq 0$, точ-

ки перемены знака функции $\rho(l)$, то есть такие точки, что при каждом $t = 1, \dots, T$ имеет место одна из следующих возможностей:

$$\begin{aligned} \rho(l) \leq 0, \text{ если } l_{t-1} < l < l_t; \rho(l_t) > 0; \rho(l) \geq 0, \text{ если } l_t < l < l_{t+1}; \\ \rho(l) \geq 0, \text{ если } l_{t-1} < l < l_t; \rho(l_t) < 0; \rho(l) \leq 0, \text{ если } l_t < l < l_{t+1}. \end{aligned}$$

В случае, когда $T = 0$, то есть функция $\rho(l)$ принимает значения только одного знака, будем считать, что $\rho(l_1) > 0$, когда $\rho(l) \leq 0, l = 1, \dots, m$, и $\rho(l_1) < 0$, когда $\rho(l) \geq 0, l = 1, \dots, m$.

Положим

$$y_1(y_2, \dots, y_m) = \begin{cases} 1 + \sum_{t=1}^T (-1)^t y_{i_2} \cdot \dots \cdot y_{i_t}, & \text{если } \rho(l_1) > 0, \\ \sum_{t=1}^T (-1)^{t+1} y_{i_2} \cdot \dots \cdot y_{i_t}, & \text{если } \rho(l_1) < 0, \end{cases}$$

и покажем, что эта функция обладает необходимыми свойствами.

Рассмотрим произвольный неединичный вектор $(0,1)$ -вектор (y_2, \dots, y_m) , и пусть $q, 2 \leq q \leq m$, — наименьший номер такой, что $y_{i_q} = 0$. Пусть, кроме того, $t_0, 1 \leq t_0 \leq T+1$, — такой номер, что $l_{t_0-1} < q \leq l_{t_0}$. Поскольку для рассматриваемого вектора (y_2, \dots, y_m) имеем

$$-f_1 + \sum_{l=1}^m a_l y_{i_2} \cdot \dots \cdot y_{i_l} = \rho(q-1),$$

то исследуем зависимость между знаком величины $\rho(q-1)$ и значением функции $y_1(y_2, \dots, y_m)$.

Если $\rho(q-1) > 0$, то $\rho(l_{t_0}) < 0$. Рассмотрим два случая $\rho(l_1) > 0$ и $\rho(l_1) < 0$. В первом случае t_0 — четно и, следовательно, $y_1(y_2, \dots, y_m) = 1 + \sum_{t=1}^{t_0-1} (-1)^t = 0$. Во втором случае

t_0 — нечетно и $y_1(y_2, \dots, y_m) = \sum_{t=1}^{t_0-1} (-1)^{t+1} = 0$. Таким образом, если $\rho(q-1) > 0$, то $y_1(y_2, \dots, y_m) = 0$.

Пусть $\rho(q-1) < 0$. Тогда с учетом того, что $\rho(l_{t_0}) > 0$, имеем t_0 — нечетно при $\rho(l_1) > 0$ и t_0 — четно при $\rho(l_1) < 0$. В любом из этих случаев получаем $y_1(y_2, \dots, y_m) = 1$.

К таким же результатам приходим, когда (y_2, \dots, y_m) — единичный вектор. В этом случае значение функции $y_2(y_1, \dots, y_m)$ необходимо сравнить с величиной $\rho(m)$.

Если $\rho(m) > 0$, то $\rho(l_T) > 0$. Тогда если $\rho(l_1) > 0$, то T — нечетно и, следовательно, $y_1(y_2, \dots, y_m) = 1 + \sum_{t=1}^T (-1)^t = 0$. Если же $\rho(l_1) < 0$, то T — четно и $y_1(y_2, \dots, y_m) =$

$$\sum_{t=1}^T (-1)^{t+1} = 0. \text{ Если } \rho(m) < 0, \text{ то независимо от знака } \rho(l_1) \text{ получаем } y_1(y_2, \dots, y_m) = 1.$$

Покажем, что в случае рассматриваемой функции $y_1(y_2, \dots, y_m)$ полином

$$P'(y_2, \dots, y_m) = y_1(y_2, \dots, y_m) \left(-f_1 + \sum_{k=2}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} \right) + r_1(y_2, \dots, y_m)$$

является вполне уравновешенным. Для этого достаточно показать, что таковым будет полином

$$y_1(y_2, \dots, y_m) \left(-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} \right)$$

при любом из двух возможных выражений для функции $y_1(y_2, \dots, y_m)$.

Если $y_1(y_2, \dots, y_m) = 1 + \sum_{t=1}^T y_{i_2} \cdot \dots \cdot y_{i_{l_t}}$, то можем написать

$$\begin{aligned} & \left(1 + \sum_{t=1}^T (-1)^t y_{i_2} \cdot \dots \cdot y_{i_{l_t}} \right) \left(-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} \right) = \\ & = -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} + \sum_{t=1}^T (-1)^t y_{i_2} \cdot \dots \cdot y_{i_{l_t}} \left(-f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} \right) = \\ & = -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} + \sum_{t=1}^T (-1)^t \left[\left(-f_1 + \sum_{k=1}^m a_k \right) y_{i_2} \cdot \dots \cdot y_{i_{l_t}} + \sum_{k=l_t+1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} \right] = \\ & = -f_1 + \sum_{k=1}^m a_k y_{i_2} \cdot \dots \cdot y_{i_k} + \sum_{t=1}^T \left\{ \left[a_{l_t} \sum_{\tau=1}^{t-1} (-1)^\tau + (-1)^t \left(-f_1 + \sum_{k=1}^{l_t} a_k \right) \right] y_{i_2} \cdot \dots \cdot y_{i_{l_t}} + \right. \\ & \quad \left. + \sum_{k=l_t+1}^{l_{t+1}-1} a_k \sum_{\tau=1}^t (-1)^\tau y_{i_2} \cdot \dots \cdot y_{i_k} \right\}. \end{aligned}$$

Ясно, что рассматриваемый полином преобразуется в полином $\sum_{k=2}^m a'_k y_{i_2} \cdot \dots \cdot y_{i_k}$,

коэффициенты которого выражаются через коэффициенты a_k , $k = 1, \dots, m$, следующим образом:

$$a'_k = \begin{cases} -f_1 + \sum_{k=1}^{l_t} a_k, & \text{если } k = l_t \text{ и } t - \text{четное;} \\ f_1 - \sum_{k=1}^{l_t-1} a_k, & \text{если } k = l_t \text{ и } t - \text{нечетное;} \\ a_k, & \text{если } l_t < k < l_{t+1} \text{ и } t - \text{четное;} \\ 0, & \text{если } l_t < k < l_{t+1} \text{ и } t - \text{нечетное,} \end{cases}$$

$k = 2, \dots, m$.

Случай $y_1(y_2, \dots, y_m) = \sum_{t=1}^T (-1)^{t+1} y_{i_2} \cdot \dots \cdot y_{i_{l_t}}$ приводит к полиному $\sum_{k=2}^m a'_k y_{i_2} \cdot \dots \cdot y_{i_k}$

со следующими коэффициентами

$$a'_k = \begin{cases} f_1 - \sum_{k=1}^{l_t-1} a_k, & \text{если } k = l_t \text{ и } t - \text{четное;} \\ -f_1 + \sum_{k=1}^{l_t} a_k, & \text{если } k = l_t \text{ и } t - \text{нечетное;} \\ 0, & \text{если } l_t < k < l_{t+1} \text{ и } t - \text{четное;} \\ a_k, & \text{если } l_t < k < l_{t+1} \text{ и } t - \text{нечетное,} \end{cases}$$

$k=2, \dots, m$.

Из приведенных формул для вычисления коэффициентов a'_k , $k=1, \dots, m$, ясно, что если для некоторого k , $1 \leq k \leq m$, имеем $a_k = 0$, то $a'_k = 0$. Поэтому характеристическая матрица H' , связанная с полиномом $P'(y_2, \dots, y_m)$, так же как и в случае полинома с неотрицательными коэффициентами, получается из исходной характеристической матрицы H удалением первой строки и некоторых столбцов, имеющих ноль в первой строке. Теорема доказана.

Из доказательства ясно, что незначительная модификация предложенного выше алгоритма минимизации вполне уравновешенного полинома с неотрицательными коэффициентами позволяет расширить область его применения до минимизации вполне уравновешенного полинома с коэффициентами произвольного знака.

Модифицированный алгоритм состоит из двух этапов, первый из которых включает m однотипных шагов. На k -м, $k=1, \dots, m-1$, шаге рассматриваются коэффициенты b_s , $s=1, \dots, S$, вычисленные по исходным коэффициентам на предыдущих шагах. Шаг начинается с процедуры «приведения подобных членов». Для этого для всякого s , $s=1, \dots, S$, тако- го, что $h_{ks} = 0$, если $h_{ks+1} = 0$ и $\sum_{i=k}^m h_{is} = \sum_{i=k}^m h_{is+1}$, то полагается $b_{s+1} \leftarrow b_s + b_{s+1}$, $b_s = 0$. Далее определяются номера $s_1(k), \dots, s_{T(k)}(k)$, $T(k) \geq 0$, задающие точки перемены знака функции

$$\rho_k(t) = -f_k + \sum_{s=1}^t b_s(1 - h_{ks}).$$

При этом считается, что $s_0(k) = 1$, если $\rho_k(s_1(k)) > 0$ и $s_0(k) = -1$, если $\rho_k(s_1(k)) < 0$. После этого начинается процедура пересчета коэффициентов $b_s \neq 0$, $s=1, \dots, S$, для которых $h_{ks} = 0$. Для этого используются приведенные выше формулы для расчета коэффициентов a'_i , $i=k+1, \dots, m$. Затем начинается следующий шаг.

На m -ом заключительном шаге первого этапа рассматриваются коэффициенты b_s , $s=1, \dots, S$, вычисленные на предыдущих шагах, и полагается $s_0(m) = 1$, если $\rho_m(s) \leq 0$, и $s_0(m) = -1$ иначе. После этого начинается второй этап.

Второй этап алгоритма также состоит из m шагов. На первом шаге определяется оптимальное значение y_m^* переменной y_m следующим образом:

$$y_m^* = \begin{cases} 1, & \text{если } s_0(m) = 1; \\ 0, & \text{если } s_0(m) = -1. \end{cases}$$

На очередном $(m-k+1)$ -м, $k=m, m-1, \dots, 1$, шаге имеются оптимальные значения y_{k+1}^*, \dots, y_m^* , соответствующих переменных, вычисленные на предыдущих шагах, и определяется оптимальное значение y_k^* переменной y_k следующим образом:

$$y_k^* = \begin{cases} 1 + \sum_{t=1}^{T(k)} (-1)^t \prod_{i>k|h_{is_t}(k)=0} y_i^*, & \text{если } s_0(k) = 1, \\ \sum_{t=1}^{T(k)} (-1)^{t+1} \prod_{i>k|h_{is_t}(k)=0} y_i^*, & \text{если } s_0(k) = 0. \end{cases}$$

После этого, если $k \leq m-1$, начинается следующий шаг, в противном случае второй этап и в целом алгоритм заканчивают работу.

Указанные выше изменения, внесенные в первоначальный вариант алгоритма минимизации вполне уравновешенного полинома, не ухудшают оценки временной сложности. Поэтому трудоемкость модифицированного алгоритма оценивается прежней величиной $O(m^2+mS)$.

4 Связные матрицы и полиномы

Другим, не менее важным, чем класс вполне уравновешенных матриц, является класс матриц с так называемым свойством связности. Эти матрицы так же как и вполне уравновешенные включают в себя правильные матрицы, а задача FL с матрицей транспортных затрат, обладающей свойством связности, является полиномиально разрешимой. При построении соответствующих алгоритмов используется основное свойство связной матрицы транспортных затрат, гарантирующее существование для всякого открываемого предприятия связной (непрерывной) области применения, то есть множества потребителей, обслуживаемых данным предприятием.

Естественным обобщением свойства связности является так называемое свойство p -связности, $p = 1, 2, 3, \dots$ которое в случае $p = 1$, совпадает со свойством связности. В случае матрицы транспортных затрат, обладающей свойством 2-связности задача FL также является эффективно разрешимой. К сожалению, на этом свойство p -связности исчерпывает свои возможности для построения эффективных алгоритмов. Задача FL с матрицей транспортных затрат со свойством 3-связности является NP -трудной.

4.1. Матрицы со свойством связности и p -связности

Матрицу $C = (c_{ij})$ ($i \in I, j \in J$) называют матрицей, обладающей свойством *связности* (связной матрицей), если для произвольной пары строк $i, k \in I$ разность $c_{ij} - c_{kj}$ меняет знак при монотонном изменении $j \in J$ не более одного раза.

Аналогично матрицу $C = (c_{ij})$ ($i \in I, j \in J$) называют матрицей, обладающей свойством p -связности (p -связной матрицей), где $p = 1, 2, 3, \dots$, если разность $c_{ij} - c_{kj}$ меняет знак при монотонном изменении $j \in J$ не более p раз.

Понятно, что свойство связности и 1-связности совпадают. Для матрицы C наличие этого свойства означает, что для произвольной пары $i, k \in I$ существует номер j_0 , $1 \leq j_0 \leq n$, такой, что разность $c_{ij} - c_{kj}$ неотрицательна (неположительна) для любого $j \leq j_0$ и неположительна (неотрицательна) для всякого $j > j_0$. Аналогично, в случае 2-связности существуют номера j_0 и j_1 , $1 \leq j_0 \leq j_1 \leq n$, такие, что разность $c_{ij} - c_{kj}$ имеет один знак для всякого $j \leq j_0$, противоположный знак для всякого j , $j_0 < j \leq j_1$, и вновь первый знак для всякого $j > j_1$.

Несложно заметить, что свойство матрицы быть p -связной зависит от нумерации столбцов. Если при одной нумерации это свойство имеет место, то при другой может не выполняться. Поэтому исходная матрица C , не обладающая свойством p -связности, может стать таковой в результате некоторой перестановки столбцов. В этом случае будем говорить, что матрица C *приводима* перестановкой столбцов к матрице со свойством p -связности. В этой связи представляет безусловный интерес вопрос об отыскании перестановки столбцов, при которой полученная матрица будет обладать нужным свойством.

Для матрицы C рассмотрим ее характеристическую матрицу H и каноническую форму \tilde{C} . Легко понять, что если матрица C приводима к матрице со свойством p -связности, то матрицы H и \tilde{C} также будут приводимы к p -связной матрице. Обратное утверждение неверно. Несложно привести пример матрицы C , которая ни при какой перестановке столбцов не превращается в матрицу со свойством связности. Вместе с тем характеристическая матрица H или каноническая форма \tilde{C} могут быть приведены к связной матрице. Это связано с тем, что при рассмотрении канонической формы \tilde{C} переставлять местами можно уже не целиком столбцы матрицы C , а «части» столбцов матрицы C . Примером таких матриц C и \tilde{C} являются, например, следующие матрицы:

$$C = \begin{bmatrix} 4 & 3 & 3 \\ 3 & 2 & 4 \\ 2 & 1 & 1 \\ 1 & 4 & 2 \end{bmatrix} \quad \tilde{C} = \begin{bmatrix} 1 & 2 & 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & 1 & 0 & 2 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 1 & 1 \end{bmatrix}$$

Из сделанного замечания можно заключить, что не всегда следует пытаться искать перестановку столбцов исходной матрицы, приводящей ее к матрице со свойством p -связности. Такой перестановки может и не существовать. Гораздо больше шансов на успех будут иметь попытки построения перестановки, приводящей матрицу в канонической форме \tilde{C} к матрице со свойством p -связности.

С учетом сказанного можно утверждать, что правильные матрицы являются частным случаем связных матриц, хотя не сложно привести пример матрицы C

$$C = \begin{pmatrix} 1 & 5 & 4 \\ 2 & 1 & 3 \\ 3 & 2 & 2 \\ 4 & 3 & 1 \\ 5 & 4 & 5 \end{pmatrix}$$

такой, что каждый ее столбец является квазивыпуклым, но никакой перестановкой столбцов эту матрицу невозможно привести к связной. Вместе с тем характеристическая матрица H любой правильной матрицы C приводима перестановкой столбцов к связной матрице. Для этого достаточно правильную $(0,1)$ -матрицу H привести к стандартной форме. Несложно увидеть, что правильная матрица в стандартной форме обладает свойством связности. Для приведенного выше примера матрицы C стандартная форма характеристической матрицы H имеет вид

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Эта матрица, как легко увидеть, является связной.

4.2. Свойство связности. Применение динамического программирования

Так же как и в случае правильных матриц, свойство связности матрицы позволяет эффективно использовать для решения задачи FL динамическое программирование. Если матрица C транспортных затрат обладает свойством связности, то задача FL сводится к задаче о ближайшем соседе и, следовательно, для задачи FL может быть построен эффективный алгоритм динамического программирования.

Такое сведение становится возможным в силу того, что существует оптимальное решение задачи FL такое, что у каждого открытого предприятия области использования являются связными. Напомним, что если (z_i) (x_{ij}) — решение задачи FL , то областью использования предприятия $i \in I$ называется множество $J(I) = \{j \in J \mid x_{ij} = 1\}$. Непустое множество $J' \subset J$ называют *связным* или *неразрывным*, если оно совпадает с некоторым множеством $\{r, r+1, \dots, s\}$, где $1 \leq r \leq s \leq n$.

Доказательство существования оптимального решения со связными областями использования становится более простым, если считать строки матрицы транспортных затрат соответствующим образом упорядоченными.

Пусть матрица $C = (c_{ij})$ ($i \in I, j \in J$) обладает свойством связности, и пусть у матрицы C нет одинаковых строк. Покажем, что существует перестановка $\{i_1, i_2, \dots, i_m\}$ множества I

такая, что для любых $k, l, k < l$, разность $c_{ikj} - c_{ilj}$ либо неположительна для всякого $j \in J$, либо меняет знак с минуса на плюс при монотонно возрастающем изменении $j \in J$.

Действительно, определим на множестве I бинарное отношение следования \prec следующим образом: для любых $i, k \in I$ положим $i \prec k$, если первая ненулевая компонента вектора $(c_{ij} - c_{kj})$ ($j \in J$) — отрицательная. Понятно, что поскольку у матрицы C нет одинаковых строк, то для любых элементов $i, k \in I$ выполняется одно из двух: либо $i \prec k$, либо $k \prec i$. Заметим также, что данное отношение является транзитивным, то есть если $i \prec k$ и $k \prec l$, то $i \prec l$. В самом деле, пусть p и q — наименьшие элементы множества J , для которых соответственно $c_{ip} - c_{kp} < 0$ и $c_{kq} - c_{lq} < 0$. Пусть $j_0 = \min\{p, q\}$. Тогда $c_{ij_0} - c_{lj_0} < 0$ и $c_{ij} - c_{lj} = 0$ для $j < j_0$.

Таким образом, множество I можно линейно упорядочить отношением \prec и перестановка $\{i_1, i_2, \dots, i_m\}$, полученная в результате такого упорядочения, будет искомой.

Из сказанного следует, что если матрица $C = (c_{ij})$ обладает свойством связности, то можно считать, что при $i < k$ разность $c_{ij} - c_{kj}$ либо неположительна, либо меняет знак с минуса на плюс.

Лемма 3.4.1. Если матрица $C = (c_{ij})$ обладает свойством связности, то существует оптимальное решение задачи FL такое, что область применения каждого открытого предприятия будет связной.

Доказательство. Рассмотрим произвольное оптимальное решение $(z_i^*)(x_{ij})$, и пусть $I_0 = \{i \in I \mid z_i^* = 1\}$ совпадает с множеством $\{i_1, \dots, i_p\}$. Положим $c_j = \min_{i \in I_0} c_{ij}, j \in J$. Построим по решению $(z_i^*)(x_{ij}^*)$ оптимальное решение с нужным свойством. Для этого определим номера r_0, r_1, \dots, r_p ($r_0 \leq r_1 \leq \dots \leq r_p$) следующим образом. Положим $r_0 = 0$ и пусть уже найдены номера r_0, r_1, \dots, r_{p-1} . Тогда в качестве r_p возьмем наибольший номер s , для которого $c_j = c_{i_p j}$ при всех $j, r_{p-1} < j \leq s$. Если такого номера не существует, то $r_p = r_{p-1}$. Покажем, что $r_p = n$. В самом деле, пусть $r_p < n$ и пусть p — наибольший номер, для которого $c_{r_p+1} = c_{i_p r_p+1}$. По построению $r_p < r_p$. Тогда для некоторого номера $q, q > p$, имеем, с одной стороны, $c_{i_p r_p+1} > c_{i_q r_p+1}$ и, с другой стороны, $c_{i_p r_p+1} > c_{i_q r_p+1}$. Но это противоречит свойству связности матрицы C .

Поскольку $(z_i^*)(x_{ij}^*)$ — оптимальное решение, то для всякого $p, 1 \leq p \leq P$, существует точка $j_p \in J$ такая, что $c_{i_p j_p} < c_{i_q j_p}$ для любого $q \neq p$. В силу построения последовательности точек r_0, r_1, \dots, r_p для всякого $p, 1 \leq p \leq P$, имеем $r_{p-1} < j_p \leq r_p$. Отсюда следует, что $0 = r_0 < r_1 < \dots < r_p = n$ и тем самым точки r_0, r_1, \dots, r_p образуют разбиение множества J на P связных подмножеств.

Для всякого $p = 1, \dots, P$ положим $x_{i_p j} = 1$, если $r_{p-1} < j \leq r_p$, и $x_{i_p j} = 0$ в противном случае. Тогда в силу выбора точек r_0, r_1, \dots, r_P получаем оптимальное решение $(z_i^*)(x_{ij})$ со связными областями применения. Лемма доказана.

Следующая теорема показывает, что если существует оптимальное решение задачи FL со связными областями применения, то эта задача сводится к задаче о ближайшем соседе. С учетом доказанной леммы сказанное означает, что задача FL с матрицей C , обладающей свойством связности, может быть эффективно решена методом динамического программирования.

Теорема 3.4.1. Если существует оптимальное решение задачи FL со связными областями применения, то задача FL сводится к задаче о ближайшем соседе.

Доказательство. Рассмотрим задачу о ближайшем соседе следующего вида. Пусть $Z = \{0, 1, \dots, n\}$ и пусть

$$f(x, y) = \min_{i \in I} \left\{ f_i + \sum_{j=x+1}^y c_{ij} \right\}.$$

Для оптимальности решения $\{x_0, x_1, \dots, x_P\}$, $0 = x_0 < x_1 < \dots < x_P = n$, этой задачи рассмотрим номера i_1, \dots, i_P такие, что

$$i_p = \arg \min_{i \in I} \left\{ f_i + \sum_{j=x_{p-1}^*+1}^{x_p^*} c_{ij} \right\}, \quad p = 1, \dots, P.$$

Поставим в соответствие решению $\{x_0, x_1, \dots, x_P\}$ допустимое решение $(z_i)(x_{ij})$ задачи FL , положив

$$z_i = \begin{cases} 1, & \text{если } i = i_p \text{ для некоторого } p = 1, \dots, P; \\ 0 & \text{в противном случае,} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{если } i = i_p \text{ и } x_{p-1}^* < j \leq x_{i_p}^* \text{ для некоторого } p = 1, \dots, P; \\ 0 & \text{в противном случае.} \end{cases}$$

Покажем, воспользовавшись критерием оптимальности (лемма 1.1.1), что построенное решение $(z_i)(x_{ij})$ является оптимальным решением исходной задачи FL . Для этого заметим, прежде всего, что

$$\sum_{p=1}^P f(x_{p-1}^*, x_p^*) = \sum_{p=1}^P \left(f_{i_p} + \sum_{j=x_{p-1}^*+1}^{x_p^*} c_{i_p j} \right) \geq \sum_{i \in I} \left\{ f_i z_i + \sum_{j \in J} c_{ij} \right\}.$$

Кроме того, покажем, что если $(z_i^*)(x_{ij}^*)$ — оптимальное решение задачи FL со связными областями использования, то найдется решение рассматриваемой задачи о ближайшем соседе, для которого выполняется обратное неравенство. Действительно, пусть множество $I_0 = \{i \in I \mid z_i^* = 1\}$ совпадает с множеством $\{i_1, \dots, i_P\}$ и пусть точки x_0, x_1, \dots, x_P ,

$0 = x_0 < x_1 < \dots < x_p = n$ такие, что $J(i_p) = \{x_{p-1}+1, \dots, x_p\}$. Тогда для решения $\{x_0, x_1, \dots, x_p\}$ задачи о ближайшем соседе будем иметь

$$\sum_{i \in I} \left\{ z_i^* + \sum_{j \in J} c_{ij} x_{ij}^* \right\} = \sum_{p=1}^P \left\{ f_{i_p} + \sum_{j=x_{p-1}+1}^{x_p} c_{i_p j} \right\} \geq \sum_{p=1}^P \min_{i \in I} \left\{ f_i + \sum_{j=x_{p-1}+1}^{x_p} c_{ij} \right\} = \sum_{p=1}^P f(x_{p-1}, x_p).$$

Таким образом, критерий оптимальности выполняется и требуемая сводимость доказана.

Суммируя сказанное выше, получаем, что если матрица транспортных затрат C обладает свойством связности, то для решения задачи FL может быть построен алгоритм, использующий рекуррентные соотношения динамического программирования для задачи о ближайшем соседе.

Трудоемкость этого алгоритма складывается из трудоемкости процедуры построения самой задачи о ближайшем соседе, оцениваемой величиной $O(mn^2)$ и трудоемкости процедуры вычисления по рекуррентным соотношениям, равной величине $O(m^2)$. Поэтому для времени работы алгоритма в целом получаем оценку $O(mn^2 + m^2)$.

4.3. Свойство p -связности при $p \leq 2$

Если матрица транспортных затрат C является p -связной и $p \leq 2$, то для решения задачи FL могут быть предложены алгоритмы, построенные на основе соотношений, которые также можно назвать рекуррентными. Однако эти соотношения отличаются от рекуррентных соотношений динамического программирования для задачи о ближайшем соседе и приводят при $p = 1$ к алгоритму с лучшей оценкой трудоемкости, чем у рассмотренного выше.

Для построения таких соотношений рассмотрим задачу $MINF_0$ в виде задачи минимизации функции

$$f_0(z_1, \dots, z_m) = \sum_{i \in I} f_i z_i + \sum_{j \in J} \min_{i|z_i=1} c_{ij}$$

на множестве $(0,1)$ -векторов (z_1, \dots, z_m) . Рассмотрим также семейство задач $\{MINF_0(r,s), 0 \leq r \leq s \leq n\}$, где при любых r и s задача $MINF_0(r,s)$ состоит в минимизации функции

$$f_{rs}(z_1, \dots, z_m) = \sum_{i \in I} f_i z_i + \sum_{j=r+1}^s \min_{i|z_i=1} c_{ij}$$

на множестве $(0,1)$ -векторов (z_1, \dots, z_m) .

Легко видеть, что задача $MINF_0(r,s)$ отличается от исходной задачи $MINF_0$ тем, что в ней требуется удовлетворить спрос не всех потребителей из множества J , а только из промежутка от $r+1$ до s .

Оптимальное значение целевой функции задачи $MINF_0(r,s)$ обозначим через F_{rs} и при этом будем считать, что $F_{rr} = 0$ при любых $r = 0, 1, \dots, n$. Для всякого $i \in I$ оптимальное решение задачи $MINF_0(r,s)$ с дополнительным условием $z_i = 1$ будем называть i -оптимальным решением.

Если (z_1, \dots, z_m) — решение задачи $MINF_0(r,s)$, то с этим решением будем связывать следующие множества:

$$I_0 = \{i \in I \mid z_i = 1\},$$

$$I_j = \{i \in I_0 \mid c_{ij} = \min_{k \in I_0} c_{kj}\}, \quad r < j \leq s.$$

Понятно, что если (z_1, \dots, z_m) — оптимальное решение задачи, то для всякого $i \in I_0$ найдется j , $r < j \leq s$, что $I_j = \{i\}$. Назовем i -оптимальное решение задачи $MINF_0(r, s)$ *существенным*, если $I_j = \{i\}$ для некоторого j , $r < j \leq s$.

Отметим некоторые простые свойства функций $f_{rs}(z_1, \dots, z_m)$.

Если $(z'_1, \dots, z'_m), (z''_1, \dots, z''_m), (z_1, \dots, z_m)$ — такие $(0, 1)$ -вектора, что $I'_0 = \{i \in I \mid z'_i = 1\}$, $I''_0 = \{i \in I \mid z''_i = 1\}$ и $I' \cup I'' = \{i \in I \mid z_i = 1\}$, то для любого j , $r < j \leq s$, выполняется неравенство

$$f_{rj}(z'_1, \dots, z'_m) + f_{js}(z''_1, \dots, z''_m) \geq f_{rs}(z_1, \dots, z_m).$$

Вектор (z_1, \dots, z_m) будем называть *объединением* векторов (z'_1, \dots, z'_m) и (z''_1, \dots, z''_m) .

Если (z_1, \dots, z_m) — i -оптимальное решение задачи $MINF_0(r, s)$ и $i \in I_s$, то (z_1, \dots, z_m) — i -оптимальное решение задачи $MINF_0(r, s-1)$.

Если (z_1, \dots, z_m) — i_0 -оптимальное решение задачи $MINF_0(r, s)$ и $I_j \setminus \{i_0\} \neq \emptyset$ для всякого j , $r < j \leq s$, то вектор (z'_1, \dots, z'_m) , отличающийся от (z_1, \dots, z_m) тем, что $z'_{i_0} = 0$, — оптимальное решение задачи $MINF_0(r, s)$.

Пусть (z_1, \dots, z_m) — некоторое решение задачи $MINF_0(r, s)$. Для любого j , $r < j \leq s$, введем на множестве I_0 следующее отношение порядка \preceq_j . Для любых $i, k \in I_0$, считаем что $i \preceq_j k$, если разность $c_{ij} - c_{kj}$ при изменении j от $r+1$ до s либо последний раз меняет знак с плюса на минус, либо является неположительной. Понятно, что если для некоторого j , $r < j \leq s$, имеем $c_{ij} \neq c_{kj}$, то выполняется одно из двух: либо $i \preceq_j k$, либо $k \preceq_j i$. Понятно также, что данное отношение транзитивно, то есть для любых $i, k, l \in I_0$ из $i \preceq_j k$ и $k \preceq_j l$ следует $i \preceq_j l$. Элемент $i_0 \in I_0$ такой, что $i_0 \preceq_j i$ для всякого $i \in I_0$, назовем наименьшим элементом множества I_0 по отношению \preceq_j . Понятно, что если i_0 — наименьший элемент по отношению \preceq_j , то $i_0 \in I_j$.

Пусть матрица $C = (c_{ij})$ ($i \in I, j \in J$) обладает свойством 1-связности. Для всякого s , $0 \leq s \leq n$, рассмотрим заданную на множестве I функцию $F_{0s}(i)$, определенную следующим образом:

$$F_{00}(i) = f_i,$$

$$F_{0s}(i) = c_{is} + \min_{j=0, \dots, s-1} \{F_{0j}(i) + F_{js-1}\}, \quad 0 < s \leq n.$$

Нашей целью будет доказательство теоремы, показывающей, что величину $F_{0s}(i)$ можно рассматривать как значение целевой функции задачи $MINF_0(0, s)$ на i -оптимальном решении. Тем самым будет показано, что с помощью приведенного соотношения можно последовательно вычислять величины F_{0s} .

Теорема 3.4.2. Для всякого $s = 1, \dots, n$ справедливо равенство

$$F_{0s} = \min_{i \in I} F_{0s}(i).$$

Доказательство теоремы вытекает из нижеследующих предложений.

Лемма 3.4.2. Для любого $s, 0 < s \leq n$, и всякого $i_0 \in I$ существует решение (z_1, \dots, z_m) , $z_{i_0} = 1$, такое, что

$$F_{0s}(z_1, \dots, z_m) \leq F_{0s}(i_0).$$

Доказательство проведем индукцией по s . Если $s = 1$, то

$$F_{01}(i_0) = c_{i_01} + F_{00}(i_0) + F_{00} = c_{i_01} + f_{i_0} = f_{01}(z_1, \dots, z_m),$$

где (z_1, \dots, z_m) — вектор, у которого $z_{i_0} = 1$, а остальные компоненты нулевые.

Предположим, что утверждение справедливо при $s = t-1$ и докажем его при $s = t$. Если

$$F_{0s}(i_0) = c_{is} + F_{00}(i_0) + F_{0s-1},$$

то рассмотрим оптимальное решение (z_1^*, \dots, z_m^*) задачи $MINF_0(0, s-1)$ и вектор (z_1, \dots, z_m) , отличающийся от (z_1^*, \dots, z_m^*) лишь тем, что $z_{i_0} = 1$. Тогда можем написать

$$F_{0s}(i_0) = c_{i_0s} + f_{i_0} + f_{0s-1}(z_1^*, \dots, z_m^*) \geq f_{0s}(z_1, \dots, z_m).$$

Если

$$F_{0s}(i_0) = c_{is} + F_{0s-1}(i_0) + F_{s-1s-1},$$

то по предположению индукции можем написать

$$F_{0s}(i_0) \geq f_{0s-1}(z_1, \dots, z_m) + c_{i_0s} \geq f_{0s}(z_1, \dots, z_m).$$

Лемма доказана.

Лемма 3.4.3. Пусть (z_1, \dots, z_m) — i_0 -оптимальное решение задачи $MINF_0(0, s)$, $0 < s \leq n$, и пусть для $j = 0$ или $j = s-1$ выполняется равенство

$$f_{0s}(z_1, \dots, z_m) = c_{i_0s} + F_{0j}(i_0) + F_{js-1}.$$

Тогда $f_{0s}(z_1, \dots, z_m) = F_{0s}(i_0)$.

Доказательство. Предположим противное и пусть $f_{0s}(z_1, \dots, z_m) > F_{0s}(i_0)$, но тогда по предыдущей лемме приходим к противоречию с тем, что (z_1, \dots, z_m) — i_0 -оптимальное решение.

Лемма 3.4.4. Пусть (z_1, \dots, z_m) — существенное i_0 -оптимальное решение задачи $MINF_0(0, s)$, $0 < s \leq n$, и пусть i_0 — наименьший элемент множества I_0 по отношению \preceq_s . Тогда $f_{0s}(z_1, \dots, z_m) = F_{0s}(i_0)$.

Доказательство проведем индукцией по s . Если $s = 1$, то

$$f_{01}(z_1, \dots, z_m) = f_{i_0} + c_{i_01} = F_{01}(i_0).$$

Предположим, что утверждение справедливо при $s = t-1$ и покажем его справедливость при $s = t$. Для заданного i_0 -оптимального решения (z_1, \dots, z_m) исследуем два случая: $I_j \setminus \{i_0\} \neq \emptyset$ для всякого $j \leq t-1$ и $I_j = \{i_0\}$ для некоторого $j \leq t-1$. В первом случае рас-

смотрим решение (z'_1, \dots, z'_m) , отличающееся от исходного i_0 -оптимального решения (z_1, \dots, z_m) лишь тем, что $z'_{i_0} = 0$. Для этого решения, которое является оптимальным решением задачи $MINF_0(0, t-1)$, учитывая то, что i_0 — наименьший элемент множества I_0 по отношению \preceq_t , и принимая во внимание предыдущую лемму, можем написать

$$f_{0t}(z_1, \dots, z_m) = f_{0t-1}(z'_1, \dots, z'_m) + f_{i_0} + c_{i_0t} = c_{i_0t} + F_{00}(i_0) + F_{0t-1} = F_{0t}(i_0).$$

Пусть теперь $I_{j_0} = \{i_0\}$ для некоторого $j_0 \leq t-1$. Тогда решение (z_1, \dots, z_m) будет существенным i_0 -оптимальным решением задачи $MINF_0(0, t-1)$. Покажем, что i_0 — наименьший элемент множества I_0 по отношению \preceq_{t-1} . Предположим, что это не так и пусть таковым является некоторый элемент $k \in I_0$. Рассмотрим разность $c_{i_0j} - c_{kj}$ и заметим, что в точке j_0 эта разность отрицательная, далее до точки $t-1$ она меняет знак на плюс и в точке t опять отрицательная. Это противоречит свойству 1-связности матрицы C . Следовательно, i_0 — наименьший элемент множества I_0 по отношению \preceq_{t-1} . Тогда по предположению индукции и с учетом предыдущей леммы можем написать

$$f_{0t}(z_1, \dots, z_m) = f_{0t-1}(z_1, \dots, z_m) + c_{i_0t} = c_{i_0t} + F_{0t-1}(i_0) + F_{t-1t-1} = F_{0t}(i_0).$$

Лемма доказана.

Для доказательства теоремы 3.4.2 остается заметить, во-первых, что в силу леммы 3.4.2 для всякого $i \in I$ имеем $F_{0s} \leq F_{0s}(i)$ и, во-вторых, что в силу леммы 3.4.4 существует $i_0 \in I$, для которого $F_{0s} = F_{0s}(i_0)$. Действительно, пусть (z_1^*, \dots, z_m^*) — оптимальное решение задачи $MINF_0(0, s)$. Тогда для всякого $i \in I_0$ данное решение будет, очевидно, i -оптимальным. Определим элемент $i_0 \in I$, являющийся наименьшим по отношению \preceq_s . Тогда по лемме 3.4.4 получаем $F_{0t} = F_{0t}(i_0)$.

Доказанная теорема 3.4.2 составляет основу алгоритма решения задачи FL в случае, когда матрица транспортных затрат C является 1-связной.

Алгоритм состоит из двух этапов.

Первый этап включает n шагов. На s -м, $s = 1, \dots, n$, шаге для всякого $i \in I$ вычисляется величина $F_{0s}(i)$ по формуле

$$F_{0s}(i) = c_{is} + f_i + \min_{j=0, s-1} \{F_{0j}(i) - f_i + F_{js-1}\}$$

и далее определяется величина

$$F_{0s} = \min_{i \in I} F_{0s}(i)$$

и номер $i(0, s)$ такой, что

$$i(0, s) = \arg \min_{i \in I} F_{0s}(i).$$

Второй этап состоит из предварительного шага и конечного числа однотипных основных шагов. На предварительном шаге строится начальное нулевое решение (z_1^*, \dots, z_m^*) .

На первом основном шаге рассматривается пара $(0, n)$ и элемент $i_1 = i(0, n)$. Шаг состоит в отыскании номера j_1 , $0 \leq j_1 \leq n-1$, такого, что

$$F_{0n}(i_1) = f_{i_1} + F_{0j_1} + \sum_{j=j_1+1}^n c_{i_1j}.$$

После этого полагается $z_{i_1}^* = 1$. Если $j_1 = 0$, то второй этап и алгоритм в целом заканчивает работу, в противном случае строится пара $(0, j_1)$, рассматриваемая на втором шаге, и начинается второй шаг, после чего третий и т.д.

Легко видеть, что для реализации каждого шага первого этапа требуется $O(m)$ действий, поэтому трудоемкость первого этапа оценивается величиной $O(mn)$. Число шагов второго этапа не превосходит m , поскольку на каждом шаге одна из компонент начального нулевого решения меняет значение. При этом для реализации всех этих шагов требуется $O(n)$ действий. Таким образом, в целом рассматриваемый алгоритм решения задачи FL имеет трудоемкость, оцениваемую величиной $O(mn)$, что лучше, чем у рассмотренного выше алгоритма динамического программирования.

Предположим теперь, что матрица $C = (c_{ij})$ ($i \in I, j \in J$) обладает свойством 2-связности. По аналогии с рассмотренными выше функциями $F_{0s}(i)$, $0 \leq s \leq n$, определим заданные на множестве I функции $F_{rs}(i)$, $0 \leq r \leq s \leq n$, следующим образом:

$$F_{rr}(i) = f_i, \quad 0 \leq r \leq n;$$

$$F_{rs}(i) = c_{is} + \min_{r \leq j < s} \{F_{rj}(i) + F_{js-1}\}, \quad 0 \leq r < s \leq n.$$

Докажем теорему, аналогичную теореме 5.1., показывающую, что с помощью приведенных соотношений можно последовательно вычислять величины F_{rs} .

Теорема 3.4.3. Для любых r, s , $0 \leq r \leq s \leq n$, справедливо равенство

$$F_{rs} = \min_{i \in I} F_{rs}(i).$$

Доказательство этого утверждения строится по той же схеме, что и доказательство теоремы 3.4.2, и вытекает из предложений, аналогичных рассмотренным выше леммам 3.4.2 и 3.4.4.

Лемма 3.4.5. При любых r, s , $0 \leq r < s \leq n$, для всякого $i_0 \in I$ существует решение (z_1, \dots, z_m) , где $z_{i_0} = 1$, для которого

$$f_{rs}(z_1, \dots, z_m) \leq F_{rs}(i_0).$$

Доказательство проведем индукцией по s . Если $s = r + 1$, то

$$F_{rr+1}(i_0) = c_{i_0r+1} + F_{rr}(i_0) + F_{rr} = f_{i_0} + c_{i_0r+1} = f_{rr+1}(z_1, \dots, z_m),$$

где (z_1, \dots, z_m) — решение, у которого $z_{i_0} = 1$, а остальные компоненты нулевые.

Предположим, что утверждение справедливо при $s \leq t - 1$ и докажем его при $s = t$. Пусть

$$F_{rt}(i_0) = F_{rr}(i_0) + F_{rt-1} + c_{i_0t} = f_{i_0} + F_{rt-1} + c_{i_0t}.$$

Тогда искомым решением будет вектор (z_1, \dots, z_m) , который отличается от оптимального решения задачи $MINF_0(r, t-1)$ лишь тем, что $z_{i_0} = 1$.

Пусть теперь

$$F_{rt}(i_0) = F_{rj}(i_0) + F_{jt-1} + c_{i_0t}$$

для некоторого j , $r < j \leq t-1$. По предположению индукции существует решение (z_1, \dots, z_m) , $z_{i_0} = 1$, для которого $F_{rj}(i_0) \geq f_{rj}(z_1, \dots, z_m)$. Рассмотрим также оптимальное решение (z_1^*, \dots, z_m^*) задачи $MINF_0(r, j)$ и решение (z'_1, \dots, z'_m) такое, что $z'_i = \max\{z_i, z_i^*\}$, $i \in I$. Тогда можем написать

$$F_{rt}(i_0) \geq f_{rj}(z_1, \dots, z_m) + f_{jt-1}(z_1^*, \dots, z_m^*) + c_{i_0t} \geq f_{rt}(z'_1, \dots, z'_m),$$

что завершает доказательство леммы.

Лемма 3.4.6. Пусть (z_1, \dots, z_m) — i_0 -оптимальное решение задачи $MINF_0(r, s)$, $0 \leq r < s \leq n$, и пусть для некоторого j , $r < j < s$, выполняется равенство

$$f_{rs}(z_1, \dots, z_m) = c_{i_0s} + F_{rj}(i_0) + F_{js-1}.$$

Тогда

$$f_{rs}(z_1, \dots, z_m) = F_{rs}(i_0).$$

Доказательство данного утверждения точно так же как и леммы 3.4.3 вытекает из предыдущей леммы, поскольку предположение о невыполнении указанного равенства приводит к противоречию с этой леммой.

Лемма 3.4.7. Пусть (z_1, \dots, z_m) — существенное i_0 -оптимальное решение задачи $MINF_0(r, s)$, $0 \leq r < s \leq n$, и пусть i_0 — наименьший элемент множества I_0 по отношению \preceq_s . Тогда $f_{rs}(z_1, \dots, z_m) = F_{rs}(i_0)$.

Доказательство проведем индукцией по s . Если $s = r + 1$, то равенство, очевидно, выполняется. Предположим, что утверждение верно при $s \leq t - 1$ и покажем его справедливость при $s = t$. Для заданного i_0 -оптимального решения (z_1, \dots, z_m) исследуем два случая: $I_j \setminus \{i_0\} \neq \emptyset$ для всякого $j \leq t - 1$ и $I_j = \{i_0\}$ для некоторого $j \leq t - 1$. В первом случае рассмотрим решение (z'_1, \dots, z'_m) , отличающееся от исходного i_0 -оптимального решения (z_1, \dots, z_m) только тем, что $z'_{i_0} = 0$. Для этого решения, которое является оптимальным решением задачи $MINF_0(r, t - 1)$, учитывая то, что i_0 — наименьший элемент множества I_0 по отношению \preceq_t и используя предыдущую лемму, можем написать

$$f_{rt}(z_1, \dots, z_m) = f_{rt_1}(z'_m, \dots, z'_m) + f_{i_0} + c_{i_0t} = F_{rr}(i_0) + F_{rt-1} + c_{i_0t} = F_{rt}(i_0).$$

Предположим теперь, что $I_j = \{i_0\}$ для некоторого $j \leq t - 1$. Пусть $j(i_0)$ — наибольший элемент j , удовлетворяющий этому свойству. Обозначим через j_0 наибольший элемент j , $r < j \leq t - 1$, для которого i_0 — наименьший элемент множества I_0 по отношению \preceq_j . Очевидно, $j(i_0) \leq j_0$. Рассмотрим множество

$$I'_0 = \{i \in I_0 \mid I_j = \{i\}, j_0 < j \leq t - 1\}.$$

Нашей целью будет сейчас доказательство следующих соотношений:

$$I_j \cap I'_0 \neq \emptyset \text{ при } j_0 < j \leq t - 1;$$

$$I_j \setminus I'_0 \neq \emptyset \text{ при } r < j \leq j_0.$$

Начнем с первого неравенства. Предположим, что $I_{j'} \cap I'_0 \neq \emptyset$ для некоторого $j', j_0 < j' \leq t-1$, и пусть i — наименьший элемент множества I_0 по отношению $\preceq_{j'}$. Рассмотрим разность $c_{i_0j} - c_{ij}$. Поскольку $i \notin I'_0$, то эта разность имеет разные знаки на промежутке от r до j_0 , поскольку $i_0 \preceq_{j_0} i$, то рассматриваемая разность последний раз меняет знак с плюса на минус. Кроме того, в силу отношения $i \preceq_{j'} i_0$, рассматриваемая разность становится положительной для некоторого $j, j_0 < j \leq j'$. Наконец, из условия $i_0 \preceq_t i$ вытекает, что разность $c_{i_0j} - c_{ij}$ принимает отрицательное значение для некоторого $j, j' < j \leq t$. Таким образом, получаем, что рассматриваемая разность $c_{i_0j} - c_{ij}$ при монотонном изменении j от $r+1$ до t меняет знак, по крайней мере, три раза, что противоречит свойству 2-связности матрицы C . Следовательно, первое неравенство доказано. Из него, в частности, вытекает, что если $I'_0 = \emptyset$, то $j_0 = t-1$.

Убедимся в справедливости второго соотношения. Поскольку $i_0 \notin I'_0$ и $i_0 \in I_{j_0}$, то $I_{j_0} \setminus I'_0 \neq \emptyset$. Предположим, что $I_{j'} \setminus I'_0 = \emptyset$ для некоторого $j', r < j' < j_0$. Пусть $i \in I_{j'}$. Рассмотрим разность $c_{i_0j} - c_{ij}$. Поскольку $i_0 \notin I_{j'}$, то данная разность имеет знак плюс при $j = j'$. Тогда из соотношения $i_0 \preceq_{j_0} i$ следует, что рассматриваемая разность меняет знак с плюса на минус при изменении j от j' до j_0 . Но поскольку $i \in I'_0$ и $i_0 \preceq_t i$, то получаем, что разность $c_{i_0j} - c_{ij}$ меняет знак с плюса на минус при изменении j от j_0+1 до t . Таким образом, разность $c_{i_0j} - c_{ij}$ меняет знак, по крайней мере, три раза при изменении j от $r+1$ до t , что противоречит свойству 2-связности матрицы C . Тем самым второе неравенство также доказано.

Рассмотрим векторы (z'_1, \dots, z'_m) и (z''_1, \dots, z''_m) такие, что $\{i \in I \mid z'_i = 1\} = I_0 \setminus I'_0$ и $\{i \in I \mid z''_i = 1\} = I'_0$. В силу доказанных соотношений имеет место равенство

$$f_{rt}(z_1, \dots, z_m) = f_{rj_0}(z'_1, \dots, z'_m) + f_{j_0t-1}(z''_1, \dots, z''_m) + c_{i_0t}.$$

Отсюда, с учетом того, что (z_1, \dots, z_m) — i_0 -оптимальное решение, получаем, что (z'_1, \dots, z'_m) — существенное i_0 -оптимальное решение задачи $MINF_0(r, j_0)$, а (z''_1, \dots, z''_m) — оптимальное решение задачи $MINF_0(j_0, t-1)$. Следовательно, по предположению индукции и в силу леммы 5.5, получаем

$$f_{rt}(z_1, \dots, z_m) = F_{rj_0}(i_0) + F_{j_0t-1} + c_{i_0t} = F_{rt}(i_0).$$

Лемма доказана.

С использованием лемм 3.4.5 и 3.4.7 доказательство теоремы 3.4.3 проводится точно так же как и теоремы 3.4.2. С одной стороны, по лемме 3.4.5 для всякого $i \in I$ имеем $F_{rt}(i_0) \leq F_{rt}$. С другой стороны, по лемме 3.4.7 существует $i_0 \in I$, такое, что $F_{rt}(i_0) = F_{rt}$.

Алгоритм решения задачи FL с матрицей C , обладающей свойством 2-связности, основанный на представленных выше рекуррентных соотношениях, состоит из двух этапов.

Первый этап включает n шагов и при этом каждый s -й шаг, $s=1, 2, \dots, n$, включает в себя s однотипных подшагов. На r -м, $r=1, 2, \dots, s-1$, подшаге s -го шага для всякого $i \in I$ вычисляется величина $F_{rs}(i)$ по формуле

$$F_{rs}(i) = \min_{r \leq j < s} \{F_{rj}(i) + F_{js-1}\} + c_{is}$$

и далее определяются величина F_{rs} по формуле

$$F_{rs} = \min_{i \in I} F_{rs}(i)$$

и номер $i(r, s)$ такой, что

$$i(r, s) = \arg \min_{i \in I} F_{rs}(i).$$

Второй этап состоит из предварительного шага и конечного числа однотипных основных шагов, каждый из которых, в свою очередь, включает некоторое число однотипных подшагов. На предварительном шаге строится нулевое начальное решение (z_1^*, \dots, z_m^*) и формируется начальный список пар (r, s) , $0 \leq r < s \leq n$, исследуемых на основных шагах. Начальный список включает только одну пару $(0, n)$.

На первом шаге рассматриваются пара $(0, n)$ и элемент $i_1 = i(0, n)$ и производится некоторое число подшагов.

На первом подшаге рассматривается пара $(0, n)$ и определяется наибольший номер j_1 , $0 \leq j_1 < n$, такой, что

$$F_{0n}(i_1) = F_{0j_1}(i_1) + F_{j_1n-1} + c_{i_1n}.$$

Если $j_1 < n-1$, то пара $(j_1, n-1)$ заносится в список пар, исследуемых на основных шагах. Если $j_1=0$, то подшаги данного шага завершаются и начинаются заключительные действия первого шага. Если $j_1 > 0$, то начинается второй подшаг.

На втором подшаге рассматривается пара $(0, j_1)$ и определяется наибольший номер j_2 , $0 \leq j_2 < j_1$, такой, что

$$F_{0j_1}(i_1) = F_{0j_2}(i_1) + F_{j_2j_1-1} + c_{i_1j_1}.$$

Если $j_2 < j_1-1$, то пара (j_2, j_1-1) заносится в список пар, исследуемых на основных шагах. Если $j_2=0$, то подшаги данного шага завершаются и начинаются заключительные действия первого шага. Если $j_2 > 0$, то переходим к третьему подшагу и т.д.

Если подшаги первого шага завершены, то на заключительных действиях первого шага пара $(0, n)$ исключается из списка исследуемых пар, полагается $z_{i_1}^* = 1$ и начинается второй шаг.

Очередной k -й, $k > 1$, шаг начинается с просмотра списка пар, исследуемых на основных шагах. Если этот список пар пуст, то алгоритм заканчивает работу. В противном случае выбирается произвольная пара (r, s) , полагается $i_k = i(r, s)$ и производится некоторое число подшагов.

На первом подшаге рассматривается пара (r, s) и определяется наибольший номер j_1 такой, что

$$F_{rs}(i_k) = F_{rj_1}(i_k) + F_{j_1s-1} + c_{i_k s}.$$

Если $j_1 < s-1$, то пара $(j_1, s-1)$ заносится в список пар, исследуемых на основных шагах. Если $j_1=r$, то подшаги данного шага завершаются и начинаются заключительные действия на k -м шаге. Если $j_1 > r$, то начинается второй подшаг с парой (r, j_1) и т.д.

После выполнения всех подшагов k -го шага на заключительных действиях этого шага пара (r, s) исключается из списка пар, исследуемых на основных шагах, полагается $z_{i_k}^* = 1$ и начинается следующий шаг.

Несложно оценить трудоемкость обоих этапов рассмотренного алгоритма. Поскольку каждый шаг первого этапа содержит не более n подшагов, то общее число подшагов первого этапа оценивается величиной n^2 . На каждом подшаге значение соответствующей функции вычисляется в m точках, что требует $O(m \cdot n)$ действий. Следовательно, трудоемкость первого этапа оценивается величиной $O(m \cdot n^3)$.

Число основных шагов второго этапа оценивается величиной m , поскольку на каждом шаге одна из компонент исходного нулевого вектора изменяет свое значение. Трудоемкость же процедуры реализации всех подшагов каждого шага оценивается величиной $O(n^2)$. Поэтому оценка трудоемкости второго этапа равняется $O(m \cdot n^2)$, а алгоритма в целом — $O(m \cdot n^3)$.

4.4. Свойство 3-связности

Свойство p -связности матрицы C при $p = 1$ и $p = 2$, как следует из сказанного выше, позволяет построить для задачи FL с матрицей C в качестве матрицы транспортных затрат эффективные алгоритмы. Однако при $p > 2$ это свойство не дает новых возможностей для расширения класса эффективно разрешимых задач FL . Более того, задача FL с матрицей транспортных затрат C , обладающей свойством 3-связности, является NP -трудной. Справедливость этого утверждения вытекает из того, что NP -трудная задача вершинного покрытия кубического графа [30'] сводится к задаче FL с матрицей C , обладающей свойством 3-связности.

Если $A = (a_{ij})$ ($i \in I, j \in J$) — матрица инцидентий кубического графа, то задача вершинного покрытия кубического графа записывается следующим образом:

$$\begin{aligned} \min_{(x_i)} \sum_{i \in I} f_i x_i; \\ \sum_{i \in I} a_{ij} x_i \geq 1, \quad j \in J; \\ x_i \in \{0, 1\}, \quad i \in I; \end{aligned}$$

Матрица ограничений $A=(a_{ij})$ этой задачи обладает свойствами

$$\sum_{i \in I} a_{ij} = 2, \quad j \in J;$$

$$\sum_{j \in J} a_{ij} = 3, \quad i \in I;$$

Отсюда ясно, что матрица A обладает свойством 5-связности.

Рассматриваемая задача SC , как известно, сводится к задаче FL с парой (F_0, C) , где $C = (c_{ij})$ ($i \in I, j \in J$) — матрица с элементами $c_{ij} = (1 - a_{ij})$. Отсюда ясно, что если матрицу A перестановкой столбцов можно привести к матрице со свойством 3-связности, то к матрице с тем же свойством можно привести и матрицу C .

Понятно также, что перестановка столбцов матрицы A определяется нумерацией ребер кубического графа, поэтому рассмотрим нумерацию ребер кубического графа, приводящую матрицу A к матрице со свойством 3-связности.

Поскольку кубический граф имеет четное число вершин, то, произвольным образом разбив множество вершин на пары, соединим каждую полученную пару дополнительным ребром. Построенный граф (мультиграф) имеет четную степень каждой вершины и, следовательно, является эйлеровым, то есть графом, для которого существует замкнутая цепь, содержащая все ребра. Используем эту замкнутую цепь для нумерации ребер эйлерова графа. Для этого рассмотрим произвольную вершину графа и припишем номер 1 дополнительному ребру, инцидентному этой вершине, и далее последовательно занумеруем ребра в порядке их следования в цепи.

Если теперь переставить столбцы матрицы A по возрастанию номеров ребер исходного кубического графа, то полученная матрица будет 3-связной. Действительно, цикл в эйлеровом графе проходит через каждую вершину ровно один раз по инцидентным данной вершине ребрам исходного кубического графа. Поэтому в каждой строке матрицы A две из трех единиц стоят в соседних столбцах. Следовательно, для любой пары строк разность не будет менять знак более трех раз.

6 Матрицы, связанные относительно графа

Обобщающим понятием связности матрицы является связность относительно бинарного отношения на множестве столбцов или, другими словами, относительно графа. Приведем основные результаты, полученные с использованием этого понятия, следуя, в основном, работе [1]. Но прежде напомним необходимые для изложения этих результатов сведения и утверждения из теории графов, заимствованные из известных монографий по теории графов [38', 66].

6.1. Некоторые сведения из теории графов

Неориентированный граф $G = (J, E)$ с множеством вершин J и множеством ребер E называют *планарным*, если его можно уложить (нарисовать) на плоскости так, чтобы ни какие два его ребра не пересекались. Планарный граф, уложенный на плоскости указанным образом, разбивает плоскость на связные области, названные *гранями*. Планарный граф называется *внепланарным*, если его можно уложить на плоскости так, чтобы ни какие два его ребра не пересекались, а все его вершины принадлежали одной грани.

Граф называется *связным*, если любая пара его вершин может быть соединена цепью. Максимальный связный подграф графа называется *компонентой* графа. *Точкой сочленения* графа называется такая его вершина, удаление которой увеличивает число компонент графа. *Неразделимым* графом называется связный граф, не имеющий точек сочленения. *Блок* графа — это его максимальный неразделимый подграф. Простой цикл, содержащий все вершины графа называется *гамильтоновым*. *Хордой* гамильтонова цикла называется ребро графа, не принадлежащее этому циклу.

Основные факты из теории внешнепланарных графов сформулируем в виде следующих лемм.

Лемма 3.6.1 [66]. Граф внешнепланарен тогда и только тогда, когда каждый его блок внешнепланарен.

Лемма 3.6.2 [147']. Неразделимый внешнепланарный граф имеет единственный гамильтонов цикл.

Лемма 3.6.3 [1]. Неразделимый граф внешнепланарен тогда и только тогда, когда он представляет собой гамильтонов цикл с попарно непересекающимися ребрами.

Из приведенных утверждений вытекает, что каждый блок связного внешнепланарного графа есть либо ребро, либо простой цикл с попарно непересекающимися хордами, а следующая лемма утверждает, что всякий связный внешнепланарный граф может быть достроен до простого цикла с непересекающимися хордами.

Лемма 3.6.4. Всякий связный внешнепланарный граф $G = (J, E)$ может быть дополнен ребрами до неразделимого внешнепланарного графа $G' = (J, E')$, $E \subset E'$.

Доказательство. Предположим первоначально, что исходный граф G состоит из двух блоков G_1 и G_2 , а j_0 — точка сочленения графа. Если подграфы G_1 и G_2 представляют собой соответственно ребро (j_1, j_0) и ребро (j_0, j_2) , то, добавив в исходный граф G ребро (j_1, j_2) , получим граф G' являющимся циклом $\{j_0, j_1, j_2, j_0\}$. Если G_1 — цикл $\{j_0, j_1, \dots, j_p, j_0\}$ с непересекающимися хордами, а G_2 — ребро (j_0, j_{p+1}) , то добавив в граф G ребро (j_1, j_{p+1}) , получим граф G' , являющийся циклом $\{j_0, j_1, \dots, j_p, j_{p+1}, j_0\}$ с непересекающимися хордами. Наконец, если G_1 и G_2 являются соответственно циклами $\{j_0, j_1, \dots, j_p, j_0\}$ и $\{j_0, j_{p+1}, \dots, j_q, j_0\}$ с непересекающимися хордами, то, пополнив граф G ребром (j_p, j_{p+1}) , получим граф G' , представляющий собой цикл $\{j_0, j_1, \dots, j_p, j_{p+1}, \dots, j_q, j_0\}$ с непересекающимися хордами. Таким образом, в любом случае получаем граф G' , который по лемме 3.6.3 будет внешнепланарным.

Если исходный граф G состоит более чем из двух блоков, то последовательным применением описанной выше операции добавления ребра получим требуемый граф G' . Лемма доказана.

Отметим, что если разбиение исходного графа на блоки задано и для каждого блока известен его гамильтонов цикл, то трудоемкость рассмотренной в ходе доказательства леммы 3.6.4 процедуры построения неразделимого внешнепланарного графа G' пропорциональна числу блоков. Следовательно, оценка трудоемкости этой процедуры есть величина $O(|E|)$.

6.2. Связность относительно графов

Пусть $G = (J, E)$ — неориентированный связный граф с множеством вершин J и множеством ребер E .

Матрица $C = (c_{ij})$ ($i \in I, j \in J$) называется *связной относительно графа* $G = (J, E)$, если для любых $i, k \in I$ существует разбиение (S, R) множества J такое, что подграфы G_1 и G_2 , порожденные соответственно множествами вершин S и R , являются связными и, кроме того, $c_{ij} \leq c_{kj}$ для всех $j \in S$ и $c_{kj} \leq c_{ij}$ для всех $j \in R$. Будем говорить, что матрица C *связна относительно класса графов*, если в этом классе найдется граф, относительно которого данная матрица C будет связной.

Понятно, что любая матрица C является связной относительно соответствующего полного графа, так что имеет смысл рассматривать связность матриц относительно графов, в которых некоторые пары вершин не соединяются ребрами.

Отметим также, что понятие связности относительно графа позволяет по иному взглянуть на свойство p -связности при $p \leq 2$ и дать эквивалентные определения этим свойствам. Легко понять, что матрица C является связной относительно цепи тогда и только тогда, когда обладает свойством 1-связности. Нетрудно увидеть также, что матрица C связна относительно цикла тогда и только тогда, когда удовлетворяет свойству 2-связности. Действительно, если матрица $C = (c_{ij})$ ($i \in I, j \in J$) является связной относительно цикла с множеством вершин $J = \{1, \dots, n\}$, то для любых $i, k \in I$ существует разбиение $S = \{p, p+1, \dots, q-1\}$, $R = \{q, \dots, n, 1, \dots, p-1\}$, где $1 \leq p \leq q \leq n$, множества J такое, что разность $c_{ij} - c_{kj}$ сохраняет знак на каждом из множеств S и R . Это означает, что указанная разность при монотонном изменении j от 1 до n меняет знак не более двух раз. Обратно, если матрица C обладает свойством 2-связности, то для любых $i, k \in I$ существуют $p, q \in I$, $p \leq q$, такие, что разность $c_{ij} - c_{kj}$ сохраняет знак на каждом из множеств $\{1, \dots, p-1\}$, $\{p-1, \dots, q-1\}$ и $\{q, \dots, n\}$. Это означает, что множество вершин цикла $J = \{1, \dots, n\}$ может быть разбито на два подмножества $S = \{p, \dots, q-1\}$, $P = \{q, \dots, p-1\}$ таких, что, во-первых, они являются множествами вершин цепей и, во-вторых, разность $c_{i_1 j} - c_{i_2 j}$ на каждом из этих множеств сохраняет знак.

Из сказанного следует, что задача FL в случае, когда матрица транспортных затрат C обладает свойством связности относительно цепей или циклов является полиномиально разрешимой. В связи с этим возникает задача о построении достаточно широкого класса графов, включающего цепи и циклы и такого, что если матрица C связна относительно этого класса, то задача FL является полиномиально разрешимой. Кроме того, возникает задача поиска относительно узкого класса графов такого, что связность матрицы относительно этого класса не является достаточным условием для построения эффективного алгоритма решения задачи FL и, более того, задача FL с матрицами транспортных затрат, связными относительно этого класса является труднорешаемой. Такими классами графов, как следует из дальнейшего, являются соответственно классы внешнепланарных и планарных графов.

6.3. Связность относительно внешнепланарных и планарных графов.

На первый из поставленных выше вопросов ответ дает следующая

Теорема 3.6.1. Матрица C размера $m \times n$, связанная относительно внешнепланарного графа G , является связной относительно некоторого цикла с n вершинами, который может быть найден по графу G полиномиальным алгоритмом с оценкой трудоемкости $O(n)$.

Доказательство теоремы опирается на следующую лемму.

Лемма 3.6.5. Матрица, связанная относительно неразделимого внешнепланарного графа, является связной относительно его гамильтонова цикла.

Доказательство. Пусть матрица $C = (c_{ij})$ ($i \in I, j \in J$) является связной относительно неразделимого внешнепланарного графа G со множеством вершин $J = \{1, \dots, n\}$. Пусть $\{1, 2, \dots, n, 1\}$ — гамильтонов цикл графа G . По условию для произвольных $i, k \in I$ найдется разбиение (R, S) множества вершин J такое, что подграфы G_1 и G_2 , порожденные множествами R и S , связны и, кроме того, $c_{ij} \leq c_{kj}$, если $j \in R$, $c_{kj} \leq c_{ij}$, если $j \in S$. Очевидно, что требуемое будет доказано, если будет показано, что одно из множеств R, S совпадает с множеством $\{p, \dots, q-1\}$, а второе — с множеством $\{q, \dots, n, 1, \dots, p-1\}$ для некоторых $p, q, 1 \leq p \leq q \leq n$.

Предположим противное. Тогда найдутся $r', r'' \in R$ и $s', s'' \in S$, для которых либо $r' < s' < r'' < s''$, либо $s' < r' < s'' < r''$. Будем считать, что имеет место первый случай (второй случай рассматривается аналогично). Поскольку подграф G_1 , порожденный множеством R , связан, то существует простая цепь (r_0, r_1, \dots, r_L) , соединяющая r' и r'' такая, что $r_l \in R, l = 0, \dots, L$. Так как $r_0 \in J \setminus \{s', \dots, s''\}$ и $r_L \in \{s', \dots, s''\}$, то найдется ребро (r_l, r_{l+1}) такое, что $r_l \in J \setminus \{s', \dots, s''\}$ и $r_{l+1} \in \{s', \dots, s''\}$. Это означает, что имеет место одно из двух: либо $r_l < s' < r_{l+1} < s''$, либо $s' < r_{l+1} < s'' < r_l$. Пусть имеет место первый случай (второй случай рассматривается аналогично). Поскольку подграф G_2 , порожденный множеством S , связан, то существует простая цепь (s_0, s_1, \dots, s_T) , соединяющая s_0 и s_T такая, что $s_t \in S, t = 0, \dots, T$. Поскольку $r_l < s' < r_{l+1} < s''$, то $s_0 \in \{r_l, \dots, r_{l+1}\}$ и $s_T \in J \setminus \{r_l, \dots, r_{l+1}\}$. Следовательно, в данной цепи найдется ребро (s_t, s_{t+1}) такое, что $s_t \in \{r_l, \dots, r_{l+1}\}$ и $s_{t+1} \in J \setminus \{r_l, \dots, r_{l+1}\}$. Но это означает, что ребра $(r_l, r_{l+1}), (s_t, s_{t+1})$ являются пересекающимися хордами гамильтонова цикла $\{1, 2, \dots, n, 1\}$, что согласно лемме 3.6.3 противоречит внешнепланарности графа G и тем самым доказывает лемму.

Используя доказанную лемму легко показать справедливость первой части теоремы 3.6.1. Действительно, согласно лемме 3.6.4 всякий внешнепланарный граф G может быть дополнен ребрами до неразделимого внешнепланарного графа G' . Если матрица C связана относительно внешнепланарного графа G , то она, очевидно, будет связана относительно графа G' и согласно лемме 3.6.5 она также будет связной относительно гамильтонова цикла графа G' .

Для доказательства второго утверждения теоремы 3.6.1 рассмотрим алгоритм построения соответствующего гамильтонова цикла для произвольного связного внешнепла-

нарного графа G . Прежде всего, с использованием известного алгоритма выделения блоков графа (см., например, [8']), имеющего оценку трудоемкости $O(n)$, находятся блоки и точки сочленения исходного графа G . Для построения гамильтонова цикла в каждом приведенном блоке может быть использован алгоритм распознавания неразделимых внешнепланарных графов, описанный в [147']. Этот алгоритм в ходе своей работы помечает все хорды гамильтонова цикла, оставляя непомеченными его ребра. Трудоемкость процедуры построения гамильтоновых циклов также оценивается величиной $O(n)$. После этого с помощью процедуры, описанной при доказательстве леммы 3.6.4, граф G дополняется ребрами до неразделимого внешнепланарного графа G' . Одновременно строится также и искомый гамильтонов цикл графа G' . Трудоемкость рассмотренного алгоритма в целом оценивается величиной $O(n)$. Теорема 3.6.1 доказана.

Доказанная теорема показывает, что задача FL с матрицей транспортных затрат C , связной относительно внешнепланарного графа, сводится к задаче FL с матрицей C , связной относительно цикла, и тем самым сводится к задаче FL с матрицей C , эффективно приводимой к матрице, обладающей свойством 2-связности. Следовательно, задача FL с матрицей C , связной относительно внешнепланарного графа является полиномиально разрешимой. Таким образом, внешнепланарные графы можно рассматривать как класс графов, включающих цепи и циклы, но при этом сохраняющих полезное свойство цепей и циклов, позволяющее сводить задачу FL с матрицей C , связной относительно графа, к задаче FL с матрицей C , обладающей свойством 2-связности.

Следующая теорема утверждает, что класс внешнепланарных графов — максимальный класс, обладающий этим свойством.

Теорема 3.6.2. Если связный граф не является внешнепланарным, то существует матрица C , связная относительно G , но не связная относительно цикла.

Для доказательства теоремы установим предварительно некоторые свойства матрицы инцидентий графа $G = (J, E)$. Рассмотрим транспортную матрицу $A = (a_{ij})$ ($i \in I, j \in J$) матрицы инцидентий графа G и покажем справедливость двух лемм, устанавливающих эквивалентность внешнепланарности графа G и связности его матрицы A относительно цикла.

Лемма 3.6.6. Матрица A связна относительно связного графа G .

Доказательство. Для произвольных $i, k \in E$ рассмотрим множества $R_1 = \{j \in J \mid a_{ij} > a_{kj}\}$ и $S_1 = \{j \in J \mid a_{kj} > a_{ij}\}$. Поскольку каждая строка матрицы A содержит ровно две единицы, то каждое из множеств R_1, S_1 содержит не более двух вершин, а подграфы графа G , порожденные указанными множествами, являются либо ребрами, либо вершинами, если ребра i и k смежные. Пополним множество R_1 теми вершинами графа G , из которых существует цепь в одну из вершин множества R_1 , не содержащая вершин из множества S_1 . Соответственно пополним множество S_1 теми вершинами графа G , из которых не существует таких цепей. Полученные множества R и S по построению образуют разбиение множества J и порождают связные полграфы графа G . Кроме того, поскольку $a_{ij} = a_{kj}$ для $j \notin R_1 \cup S_1$, получаем, что $a_{ij} \geq a_{kj}$ для $j \in R$ и $a_{kj} \geq a_{ij}$ для $j \in S$. Это означает, что матрица A связна относительно графа G . Лемма доказана.

Лемма 3.6.7. Связный граф G внешнепланарен тогда и только тогда, когда матрица A связна относительно цикла.

Доказательство. Если граф G внешнепланарен, то по предыдущей лемме матрица A связна относительно графа G , а по теореме 3.6.1 матрица A связна относительно цикла.

Обратно. Пусть матрица A связна относительно цикла $\{1, \dots, n, 1\}$ и, следовательно, является 2-связной. Разместим вершины графа G на окружности в порядке их следования в рассмотренном цикле. Заметим далее, что поскольку матрица A является 2-связной, то любые два несмежные ребра (j', l') и (j'', l'') графа G не могут пересекаться. Дополним граф G недостающими ребрами, образующими цикл $\{1, \dots, n, 1\}$. Получим граф G' , представляющий собой гамильтонов цикл с непересекающимися хордами. По лемме 3.6.3 этот граф G' и, следовательно, исходный граф G являются внешнепланарными. Лемма доказана.

Доказательство теоремы 3.6.2 вытекает из доказанных лемм. Действительно, пусть граф G не является внешнепланарным. Тогда матрица A , будучи по лемме 3.6.6 связной относительно графа G , не является связной относительно цикла, поскольку в противном случае в силу леммы 3.6.7 граф G был бы внешнепланарным. Теорема доказана.

Доказанная теорема дает ответ на первый из сформулированных ранее вопросов о максимальном классе графов, связность матрицы относительно которого означает связность матрицы относительно цикла и, следовательно, ее 2-связность. Таким максимальным классом является класс внешнепланарных графов. Попытка расширения этого класса за счет графов, не являющихся внешнепланарными, как следует из теоремы, приводит к тому, что матрица может быть связной относительно расширенного класса, но не будет связной относительно цикла и, следовательно, не будет обладать тем полезным свойством, использование которого позволяло строить эффективный алгоритм решения задачи FL .

Вторым вопросом был вопрос о минимальном расширении класса внешнепланарных графов, при которых связность матрицы относительно этого класса не только перестает быть полезным свойством для построения эффективных алгоритмов, но делает задачу FL с матрицей, связной относительно этого класса, трудоемкой, то есть эквивалентной задаче FL в общем случае. Ответ на этот вопрос дает следующая

Теорема 3.6.3. Задача FL с матрицей транспортных затрат C , связной относительно планарных графов, является NP -трудной.

Доказательство. Сведем к рассматриваемой задаче NP -трудную задачу о вершинном покрытии связного кубического планарного графа [30']. Пусть $G=(I, J)$ — связный кубический планарный граф с множеством вершин I и множеством ребер J . Пусть $A=(a_{ij})$ ($i \in I, j \in J$) — матрица инцидентностей графа G .

Задача VC для данного графа G согласно теореме 1.4.1 сводится к задаче FL с матрицами $F = (f_i)$ ($i \in I$) и $C = (c_{ij})$ ($i \in I, j \in J$) вида $f_i = 1, i \in I$; $c_{ij} = ** (1 - a_{ij})$ $i \in I, j \in J$. Для завершения доказательства остается показать, что матрица C будет связна относительно некоего планарного графа. В качестве такого графа используем реберный граф $\tilde{G} = (J, E)$ графа G , который является связным и планарным.

Для произвольных $i, k \in I$ рассмотрим множества $R_1 = \{j \in J \mid c_{ij} < c_{kj}\}$ и $S_1 = \{j \in J \mid c_{kj} < c_{ij}\}$. Поскольку каждая строка матрицы C содержит ровно три нуля, то каждое из множеств R_1 и S_1 содержит не более трех вершин, а подграфы графа \tilde{G} , порожденные указанными множествами, являются связными и представляют собой либо треугольник, либо ребро. Пополним множество R_1 теми вершинами графа \tilde{G} , из которых существует цепь в одну из вершин множества R_1 , не содержащая вершин из множества S_1 . Соответственно пополним множество S_1 теми вершинами графа \tilde{G} , из которых не существует таких цепей. Полученные множества R и S по построению образуют разбиение множества J и порождают связные подграфы графа \tilde{G} . Кроме того, поскольку $c_{ij} = c_{kj}$ для $j \notin R_1 \cup S_1$ получаем, что $c_{ij} \leq c_{kj}$ для $j \in R$ и $c_{kj} \leq c_{ij}$ для $j \in S$. Это означает, что матрица C связна относительно графа \tilde{G} . Теорема доказана.

Из сказанного получаем, что если матрица C является связной относительно внешнепланарных графов, то она приводима посредством эффективной процедуры к матрице со свойством 2-связности. Следовательно, задача FL с матрицей транспортных затрат C размера $m \times n$, связной относительно внешнепланарных графов, может быть эффективно решена алгоритмом с оценкой трудоемкости $O(mn^3)$. Естественное расширение класса матриц, связных относительно внешнепланарных графов, до матриц, связных относительно планарных графов, приводит к матрицам уже не обладающим полезными свойствами, позволяющими строить эффективные алгоритмы решения задачи FL .

Таким образом, получаем, что понятие связности относительно графа не приводит к расширению класса матриц транспортных затрат, с которыми задача FL является полиномиально разрешимой. Вместе с тем это понятие дает полезное эквивалентное определение свойству 2-связности и позволяет сформулировать важное условие приводимости исходной матрицы к 2-связной. Последнее особенно полезно при рассмотрении задач размещения на сети.

7 Эффективно разрешимые случаи задачи размещения на сети

7.1. Задача размещения на сети

Рассмотрим задачу NFL на сети $G = (J, E)$ с множеством вершин J и ребер E . Напомним, что элементы матрицы транспортных затрат $C = (c_{ij})$ ($i \in J, j \in J$) этой задачи имеют вид $c_{ij} = p_j \cdot d_{ij}$, где d_{ij} — наименьшая длина цепи из вершины i в вершину j .

Отметим, что свойство связности матрицы относительно графов, позволяющее строить эффективные алгоритмы решения задачи FL , может быть использовано для построения эффективных алгоритмов решения задачи NFL на таких графах. Основанием тому является следующая лемма.

Лемма 3.7.1. Матрица C задачи NFL на сети G является связной относительно графа G .

Доказательство. Рассмотрим матрицу $C = (c_{ij})$ ($i \in J, j \in J$) и для произвольных $i, k \in J$ устроим разбиение (R, S) множества J следующим образом: $R = \{j \in J \mid d_{ij} \leq d_{ik}\}$, $S = \{j \in J \mid d_{ij} > d_{ik}\}$. Покажем, что графы G_1 и G_2 , порождаемые множествами R и S , являются связными. Для этого заметим, что для любого $j \in R$ цепь наименьшей длины из i в j содержит только вершины из множества R . Действительно, если в этой цепи имеется вершина $l \in S$, то $d_{il} > d_{kl}$ и, следовательно, $d_{ij} > d_{kl}$, а это противоречит условию $j \in R$. Аналогично заметим, что для любого $j \in S$ цепь наименьшей длины из k и j содержит только вершины из множества S . Действительно, пусть для вершины l этой сети имеем $l \in R$. Тогда $d_{il} \leq d_{kl}$ и, следовательно, $d_{ij} \leq d_{kj}$, что противоречит условию $j \in S$. Таким образом, графы G_1 и G_2 являются связными и лемма доказана.

Из доказанного с учетом того, что задача FL с матрицей C , связной относительно внешнепланарного графа, сводится к задаче FL с 2-связной матрицей, получаем, что задача NFL на сети в виде внешнепланарного графа сводится к задаче FL с 2-связной матрицей. Следовательно, задача NFL на внешнепланарном графе с n вершинами может быть решена алгоритмом с оценкой трудоемкости $O(n^4)$. Отсюда, в частности, получаем, что задача NFL на сети в виде дерева также может быть решена алгоритмом с той же оценкой трудоемкости. Отметим, что эта оценка хуже, чем оценка $O(n^3)$, упоминавшихся ранее градиентных алгоритмов из [116', 125] для задачи NFL на дереве. В действительности же специфика дерева такова, что позволяет строить существенно менее трудоемкие алгоритмы. В [26] для задачи FL с матрицей C , связной относительно дерева, построен алгоритм с оценкой трудоемкости $O(n^2)$. Это означает, что задача NFL на сети в виде дерева может быть эффективно решена алгоритмом с оценкой трудоемкости $O(n^2)$. Ниже для этой задачи предлагается алгоритм той же трудоемкости, построенный по аналогии с алгоритмом решения задачи FL с матрицей C , обладающей свойством p -связности, $p \leq 2$.

7.2. Алгоритм для задачи размещения на сети в виде дерева

Для построения такого алгоритма исходную задачу NFL на дереве $G = (J, E)$ с матрицей транспортных затрат $C = (c_{ij})$ ($i \in J, j \in J$), где $c_{ij} = p_j \cdot d_{ij}$, представим как задачу минимизации функции

$$g(z_1, \dots, z_n) = \sum_{i \in J} f_i z_i + \sum_{j \in J} \min_{i \mid z_i = 1} c_{ij}$$

на множестве $(0,1)$ -векторов. Кроме того, введем на множестве вершин J рассматриваемого дерева G отношение частичного порядка, которое необходимо для построения упорядоченного семейства подзадач исследуемой задачи и составления соответствующих рекуррентных соотношений.

Чтобы задать такой частичный порядок выберем произвольную вершину $j_0 \in J$ и ориентируем ребра графа G так, чтобы существовала ориентированная цепь из вершины j_0 в любую вершину $j \in J$. Тем самым получим ориентированное дерево $\bar{G} = (J, \bar{E})$ с корневой вершиной j_0 . Через \bar{G}_j обозначим поддерево дерева \bar{G} , для которого вершина $j \in J$ яв-

ляется корневой, а через J_j — множество вершин этого поддерева. Про вершины множества J_j будем говорить, что они *следуют* за вершиной j . Через L_j обозначим вершины поддерева \bar{G}_j , которые являются конечными вершинами дуг с начальной вершиной j . Про вершины этого множества будем говорить, что они *непосредственно следуют* за вершиной j .

Рассмотрим теперь помимо исходной задачи NFL семейство задач $\{NFL(j), j \in J\}$, где при любом $j \in J$ задача $NFL(j)$ состоит в минимизации функции

$$g_j(z_1, \dots, z_n) = \sum_{i \in J} f_i z_i + \sum_{l \in J_j} \min_{i|z_i=1} c_{il}$$

на множестве $(0,1)$ -векторов (z_1, \dots, z_n) . Понятно, что задача $NFL(j)$ отличается от исходной задачи NFL тем, что выбранные средства должны удовлетворять спрос не во всех вершинах J графа G , а только в части вершин J_j .

Оптимальное значение целевой функции задачи $NFL(j)$ обозначим через F_j . Для всякого $i \in J$ оптимальное решение задачи $NFL(j)$ с дополнительным условием $z_i=1$ назовем *i -оптимальным* решением.

Если (z_1, \dots, z_n) — решение задачи $NFL(j)$, то с этим решением будем связывать следующие множества

$$I_0 = \{i \in J \mid z_i = 1\},$$

$$I_l = \left\{ i \in I_0 \mid c_{il} = \min_{k \in I_0} c_{ik} \right\}, \quad l \in J_j.$$

Назовем *i -оптимальное решение существенным*, если $I_l = \{i\}$ для некоторого $l \in J_j$.

Для всякого $j \in J$ рассмотрим заданную на множестве J функцию $F_j(i)$, определенную следующим образом:

$$F_j(i) = c_{ij} + f_i + \sum_{l \in L_j} \min\{F_l(i) - f_i; F_l\}.$$

Содержательно величину $F_j(i)$ можно рассматривать как значение целевой функции задачи $NFL(j)$ на i -оптимальном решении. Нашей целью будет доказательство равенства аналогичного равенствам, полученным для задачи FL в случае матриц C , обладающих свойством p -связности, $p \leq 2$.

Теорема 3.7.1. Для всякого $j \in J$ справедливо равенство

$$F_j = \min_{i \in J} F_j(i).$$

Для того, чтобы доказать приведенное равенство убедимся в справедливости следующих утверждений.

Лемма 3.7.2. Для любого $j \in J$ и всякого $i_0 \in J$ существует решение (z_1, \dots, z_n) , $z_{i_0} = 1$, такое, что

$$g_j(z_1, \dots, z_n) \leq F_j(i_0).$$

Доказательство проведем индукцией по числу элементов в множестве J_j . Если j — концевая вершина, то

$$F_j(i_0) = c_{i_0j} + f_{i_0} = g_j(z_1, \dots, z_n),$$

где (z_1, \dots, z_n) — такой вектор, у которого $z_{i_0} = 1$ и $z_i = 0$ при $i \neq i_0$.

Предположим, что утверждение справедливо для вершин j таких, что $|J_j| < S$ и покажем его справедливость для вершины j , для которой $|J_j| = S$. Пусть

$$F_j(i_0) = c_{i_0j} + f_{i_0} + \sum_{l \in L_j} \min\{F_l(i_0) - f_{i_0}; F_l\} = c_{i_0j} + f_{i_0} + \sum_{l \in L_j^1} (F_l(i_0) - f_{i_0}) + \sum_{l \in L_j^0} F_l.$$

По предположению индукции для всякого $l \in L_j^1$ существует вектор (z_{1l}, \dots, z_{nl}) , $z_{i_0l} = 1$, такой, что $F_l(i_0) \geq g_l(z_{1l}, \dots, z_{nl})$. Для всякого $l \in L_j^0$ обозначим через $(z_{1l}^*, \dots, z_{nl}^*)$ такой вектор, что $F_l = g_l(z_{1l}^*, \dots, z_{nl}^*)$. Пусть вектор (z_1, \dots, z_n) есть объединение рассмотренных векторов. Тогда можем написать

$$F_j(i_0) \geq c_{i_0j} + f_{i_0} + \sum_{l \in L_j^1} (g_l(z_{1l}, \dots, z_{nl}) - f_{i_0}) + \sum_{l \in L_j^0} g_l(z_{1l}^*, \dots, z_{nl}^*) \geq g_j(z_1, \dots, z_n).$$

Лемма доказана.

Лемма 3.7.3. Пусть (z_1, \dots, z_n) — i_0 -оптимальное решение задачи $NFL(j)$ и пусть существует разбиение (L_j^1, L_j^0) множества L_j такое, что

$$g_j(z_1, \dots, z_n) = c_{i_0j} + f_{i_0} + \sum_{l \in L_j^1} (F_l(i_0) - f_{i_0}) + \sum_{l \in L_j^0} F_l.$$

Тогда $g_j(z_1, \dots, z_n) = F_j(i_0)$.

Доказательство. Предположим противное и пусть $g_j(z_1, \dots, z_n) > F_j(i_0)$. По предыдущей лемме существует решение (z'_1, \dots, z'_n) , $z'_{i_0} = 1$, такое, что $F_j(i_0) \geq g_j(z'_1, \dots, z'_n)$. Но это противоречит i_0 -оптимальности решения (z_1, \dots, z_n) . Лемма доказана.

Лемма 3.7.4. Пусть (z_1, \dots, z_n) — существенное i_0 -оптимальное решение задачи $NFL(j)$, $j \in J$, и пусть $i_0 \in I_j$. Тогда $f_j(z_1, \dots, z_n) = F_j(i_0)$.

Доказательство проведем индукцией по числу элементов множества J_j или по числу потомков вершины j в графе \bar{G} . Если j — корневая вершина, для которой $L_j = \emptyset$, то можем написать

$$f_j(z_1, \dots, z_n) = c_{i_0j} + f_{i_0} = F_j(i_0).$$

Предположим, что утверждение верно для вершин j таких, что $|J_j| < S$ и покажем его справедливость для вершины j , у которой $|J_j| = S$. Для этого рассмотрим множества

$$I_0 = \{i \in J \mid z_i = 1\};$$

$$I_{0l} = \{i \in I_0 \mid \{i\} = I_j, j \in J_l\}, l \in L_j.$$

и покажем, что выполняются следующие два соотношения:

$$(I_{0l} \setminus \{i_0\}) \cap (I_{0t} \setminus \{i_0\}) = \emptyset \text{ для любых } l, t \in L_{j_0},$$

$$(I_{0l} \cup \{i_0\}) \cap I_t \neq \emptyset \text{ для любых } l \in L_j, t \in J_l.$$

Начнем с первого. Пусть для $i \neq i_0$ имеем $\{i\} = I_{j_1}$ и $\{i\} = I_{j_2}$, где $j_1 \in J_l, j_2 \in J_t$ и $l, t \in L_j, l \neq t$. Поскольку $i_0 \in I_j$, то $d_{ij} \geq d_{i_0j}$. Тогда либо $d_{i_0j_1} \leq d_{ij_1}$, либо $d_{i_0j_2} \leq d_{ij_2}$, что противоречит условию $I_{j_1} = I_{j_2} = \{i\}$. Убедимся также в справедливости второго соотношения. Пусть для некоторых $l \in L_j$ и $t \in J_l$ имеем $I_{0l} \cap I_t = \emptyset$. Пусть тогда $i \in I_t, i \notin I_{0l}$. Поскольку $i \notin J_l$ и $d_{ij} \geq d_{i_0j}$, то $d_{it} \geq d_{i_0t}$. Следовательно, $i_0 \in I_t$, что доказывает требуемое.

Для всякого $l \in L_j$ по исходному решению (z_1, \dots, z_n) построим вектор (z_{1l}, \dots, z_{nl}) следующим образом:

$$z_{ij} = \begin{cases} 0, & \text{если } i \notin I_{0l}, \\ 1, & \text{если } i \in I_{0l}. \end{cases}$$

Кроме того, положим $z_{i_0l} = 1$, если $I_{0l} \cap I_t = \emptyset$ для некоторого $j \in J_l$.

Положим

$$\begin{aligned} L_j^1 &= \{l \in L_j \mid z_{i_0l} = 1\}, \\ L_j^0 &= \{l \in L_j \mid z_{i_0l} = 0\}. \end{aligned}$$

Тогда с учетом того, что множества $I_{0l} \setminus \{i_0\}$, $l \in L$, не пересекаются, можем написать

$$f_j(z_1, \dots, z_n) = c_{i_0j} + f_{i_0} + \sum_{l \in L_j^1} (f_l(z_{1l}, \dots, z_{nl}) - f_{i_0}) + \sum_{l \in L_j^0} f_l(z_{1l}, \dots, z_{nl}).$$

Поскольку (z_1, \dots, z_n) — существенное i_0 -оптимальное решение задачи $NFL(j)$, то решение (z_{1l}, \dots, z_{nl}) при $l \in L_j^1$ будет существенным i_0 -оптимальным решением задачи $NFL(l)$, а при $l \in L_j^0$ — оптимальным решением задачи $NFL(l)$. Кроме того, для $l \in L_j^1$ имеем $i_0 \in I_l$. Поэтому с учетом предположения индукции справедливо равенство

$$f_j(z_1, \dots, z_n) = c_{i_0j} + f_{i_0} + \sum_{l \in L_j^1} (F_l(i_0) - f_{i_0}) + \sum_{l \in L_j^0} F_l.$$

Отсюда, принимая во внимание предыдущую лемму, получаем

$$f_j(z_1, \dots, z_n) = F_j(i_0).$$

Лемма доказана.

Для доказательства теоремы 3.7.1. остается заметить, во-первых, что в силу леммы 3.7.2, для всякого $i \in I$ имеем $F_j \leq F_j(i)$ и, во-вторых, что существует $i_0 \in I$, для которого $F_j = F_j(i_0)$. Действительно, пусть (z_1^*, \dots, z_n^*) — существенное оптимальное решение задачи $NFL(j)$. Рассмотрим элемент $i_0 \in I_0$ такой, что $i_0 \in J_j$. Тогда (z_1, \dots, z_n) — существенное i_0 -оптимальное решение задачи $NFL(j)$ и по лемме 3.7.4 получаем $F_j = F_j(i_0)$.

На основе доказанной теоремы 3.7.1 может быть построен малотрудоемкий алгоритм решения задачи NFL на сети в виде дерева G . При этом предполагается, что вершины дерева G занумерованы в порядке убывания рангов вершин в корневом дереве \bar{G} так, что номер 1 имеет одна из корневых вершин, а номер n — корневая вершина j_0 . Считается

также, что дерево \bar{G} задано списком вершин с указанием для каждой вершины ее непосредственных потомков.

Алгоритм состоит из двух этапов. Первый этап включает n шагов. На j -м шаге, $j = 1, \dots, n$, для всякого $i \in J$ вычисляется величина $F_j(i)$ по формуле

$$F_j(i) = f_i + \sum_{l \in L_j} \min\{F_l(i) - f_i; F_l\}.$$

Далее определяется величина

$$F_j = \min_{i \in J} F_j(i)$$

и элемент $i(j)$ такой, что

$$i(j) = \arg \min_{i \in J} F_j(i).$$

После этого, если $j < n$, начинается следующий шаг в противном случае — второй этап.

Второй этап состоит из предварительного шага и n основных шагов.

На предварительном шаге полагается $z_i^* = 0, i \in J$.

На очередном k -м, $k = 1, \dots, n$, основном шаге рассматривается вершина с номером $j = n - k + 1$. Полагается $z_{i(j)}^* = 1$. Далее для всякого $l \in L_j$ вычисляется величина

$$\min\{F_l(i(j)) - f_{i(j)}; F_l\}$$

и определяется разбиение (L_j^1, L_j^0) множества L_j такое, что

После этого для всякого $l \in L_j^1$ полагается $i(l) = i(j)$ и, если $j > 1$, начинается следующий шаг. В противном случае алгоритм заканчивает работу.

Оценим трудоемкость этого алгоритма. Предварительный этап, связанный с вычислением рангов вершин и их перенумерацией в порядке невозрастания рангов, в случае дерева имеет трудоемкость $O(n)$. Суммарная трудоемкость вычисления на всех шагах первого этапа величин $F_j(i)$ равняется $O(n^2)$, а трудоемкость определения величины F_j на соответствующем шаге оценивается величиной $O(n)$. Поэтому оценка трудоемкости первого этапа есть величина $O(n^2)$. Та же оценка справедлива и для второго этапа, поскольку для вычисления необходимых разбиений множеств L_j на всех шагах второго этапа алгоритма требуется порядка n^2 действий. Таким образом, трудоемкость представленного алгоритма решения задачи NFL на сети в виде дерева равняется $O(n^2)$.