ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. Н.И. ЛОБАЧЕВСКОГО

На правах рукописи

Афраймович Лев Григорьевич

ПОТОКОВЫЕ МЕТОДЫ РЕШЕНИЯ МНОГОИНДЕКСНЫХ ЗАДАЧ ТРАНСПОРТНОГО ТИПА

Специальность: 01.01.09

Дискретная математика и математическая кибернетика

Диссертация
на соискание ученой степени
доктора физико-математических наук

Научный консультант:
доктор технических наук,
профессор Прилуцкий М.Х.

Нижний Новгород
2014

Содержание

Введение

Существует широкий класс прикладных задач, формализуемых в виде многоиндексных задач (целочисленного) линейного программирования транспортного типа. Примерами таких задач являются задачи распределения ресурсов в иерархических системах: задача объемно-календарного планирования, задача формирования портфеля заказов, задача переработки газового конденсата, задача распределения мощностей каналов передачи данных, транспортная задача с промежуточными пунктами и др. [27, 28, 29, 32, 65, 83, 84, 85]. Многоиндексные задачи транспортного типа возникают также в области статистики и в смежной области защиты статистических данных [138, 140, 142, 168], в задаче целочисленного сбалансирования многоиндексных матриц [94, 95, 101]. Многоиндексные задачи о назначениях (подкласс многоиндексных транспортных задач целочисленного линейного программирования) возникают в теории расписаний при планировании изготовления скоропортящейся продукции, при планировании прохождения практики студентами, при планировании учебы клинических ординаторов по отделениям, при составлении расписания занятий, при планировании спортивных матчей и др. [118, 123, 132, 145, 147, 154, 164]; в области технического анализа данных при сопровождении объектов в многосенсорных системах [175, 176, 184]; в военной области при назначении военной техники на цели [116, 162]. Известны результаты, посвященные исследованию вопросов сводимости задач линейного и целочисленного линейного программирования к многоиндексным транспортным задачам [136, 137], что также расширяет область применения методов решения многоиндексных задач.

Многоиндексные задачи линейного программирования транспортного типа относятся к классу задач линейного программирования, который согласно [108] является полиномиально разрешимым. Таким образом, для решения многоиндексных задач линейного программирования транспортного типа могут быть применены общие методы исследований задач линейного программирования – симплекс метод [133], алгоритм Кармаркара [158]. Выделим также следующие работы, посвященные алгоритмам решения задач линейного программирования, [39, 41, 47, 50, 61, 62, 81, 160, 166]. Существует ряд работ, посвященных непосредственно методам решения многоиндексных задач линейного программирования транспортного типа. Наиболее изученным является класс двухиндексных задач [43, 48]. В многоиндексном случае (число индексов не менее трех) наибольшее внимание уделено двум классам задач: многоиндексные аксиальные задачи и многоиндексные планарные задачи. Вопросы исследования совместности и построения алгоритмов решения данных задач обсуждаются в [49, 57, 67, 93, 121, 177, 186]. В общей постановке (система ограничений содержит произвольные подсуммы) класс многоиндексных задач рассматривается в [93]. Условия, при которых удается понизить размерность и (или) сократить количество индексов многоиндексных транспортных задач, обсуждаются в [42, 157]. Геометрические свойства множества допустимых решений многоиндексных транспортных задач обсуждаются в [57, 58, 68, 135]. Вопросы оценки числа нецелочисленных вершин многогранника многоиндексных транспортных задач рассматриваются в [66, 70–72]. В ряде работ исследуются многоиндексные задачи транспортного типа с нелинейными критериями [96, 97, 103, 124, 139, 183], в том числе с минимаксными [79, 80, 141, 152, 161, 185] и с квадратичными [73, 99, 125, 126, 130] критериями. Многокритериальные многоиндексные задачи транспортного типа обсуждаются в [55, 77, 78, 83, 84, 90–92]. Игровые постановки, связанные с многоиндексными транспортными задачами, рассматриваются в [75, 148, 178].

Особый интерес представляет решение многоиндексных задач целочисленного линейного программирования транспортного типа, относящихся к классу задач целочисленного линейного программирования [102]. В общей постановке класс целочисленных многоиндексных транспортных задач является *NP*-трудным уже в трехиндексном случае [52]. Более того, для задач данного класса не существует полиномиальных ε-приближенных алгоритмов, иначе *P*=*NP*, данный результат также справедлив уже в трехиндексном случае [131]. Класс целочисленных многоиндексных задач, как подкласс задач целочисленного линейного программирования, содержит ряд известных полиномиально разрешимых подклассов: задачи, матрица систем ограничений которых является абсолютно унимодулярной [51], задачи с фиксированным числом переменных [144, 163], задачи N-кратного целочисленного программирования [134]. Приведем далее примеры соответствующих полиномиально разрешимых классов целочисленных многоиндексных задач. Как хорошо известно, матрица системы ограничений двухиндексной транспортной задачи является абсолютно унимодулярной, и тем самым класс двухиндексных задач целочисленного линейного программирования транспортного типа разрешим за полиномиальное время [43]. Класс многоиндексных задач целочисленного линейного программирования с фиксированным количеством индексов, каждый из которых принимает фиксированное количество значений, является полиномильно разрешимым согласно [163]. Примером полиномиально разрешимого класса задач N-кратного целочисленного программирования является класс трехиндексных планарных задач, в котором два из трех индексов принимают фиксированное количество значений [134]. При отсутствии дополнительных ограничений на параметры для решения многоиндексных задач целочисленного линейного программирования транспортного типа применимы лишь экспоненциальные по верхней оценке вычислительной сложности общие методы целочисленного линейного программирования, например, метод динамического программирования, метод ветвей и границ, метод отсечения Гомори [100, 102, 105].

Среди целочисленных многоиндексных транспортных задач наиболее изученным является класс многоиндексных задач о назначениях. Обзор результатов, связанных с анализом вычислительной сложности и построением приближенных алгоритмов решения специальных подклассов многоиндексных задач о назначениях приведен в [126, 174, 182]. Особое внимание уделяется двум классам многоиндексных задач о назначениях: класс аксиальных задач о назначениях и класс планарных задач о назначениях. Многоиндексные аксиальные задачи о назначениях рассматриваются, например, в работах [45, 46, 119, 120, 127, 128, 131], планарные в [44, 56, 69, 98, 146, 167].

Одним из перспективных направлений при разработке эффективных алгоритмов исследования многоиндексных задач линейного программирования является нахождение подклассов задач, для решения которых применимы потоковые методы. Важное влияние на развитие данного направления оказывают активные исследования в области сетевой оптимизации [1, 38, 53, 54, 63, 76, 104, 106, 107, 115, 122, 143, 149, 151, 169, 172, 173, 180]. Существующие эффективные потоковые алгоритмы позволяют в случае сводимости задач линейного программирования к потоковым задачам построить алгоритмы их решения, обладающие более низкими оценками вычислительной сложности по сравнению с оценками общих методов решения задач линейного программирования. В ряде случаев сведение к потоковым задачам также позволяет предложить алгоритм решения исходной задачи, гарантирующий нахождение целочисленного решения, и тем самым позволяет выделять полиномиально разрешимые подклассы задач в NP-трудном классе задач целочисленного линейного программирования. Возможность сводимости задач линейного программирования к потоковым задачам исследовалась в [64, 74, 165, 153], важным различием которых являются применяемые концепции сводимости, в частности внимание уделяется распознаванию потоковой постановки в имеющейся задаче линейного программирования.

Рассматриваемая в диссертационной работе проблема сводимости многоиндексных транспортных задач линейного программирования к потоковым задачам является менее исследованной. Ранее было известно о сводимости двухиндексных задач к задаче поиска потока в сети [43]. Вопрос сводимости многоиндексных задач с произвольным числом индексов и произвольными многоиндексными подсуммами системы ограничений впервые рассматривался в работах [8, 9, 20, 22, 25, 28, 32], которые легли в основу диссертационной работы.

Из ученых существенный вклад в развитие рассматриваемого в диссертационной работе класса многоиндексных задач внесли Гимади Э.Х., Гольштейн Е.Г., Емеличев В.А., Канторович Л.В., Кириченко И.О., Кравцов М.К., Прилуцкий М.Х., Раскин Л.Г., Сергеев С.И., Сигал И.Х., Цурков В.И., Юдин Д.Б., Burkard R.E., Danzig G.L., De Loera J., Koopmans T.C., Onn S., Spieksma F.C.R. и др. Существенный вклад в развитие методов сетевой оптимизации, применяемых в работе при исследовании многоиндексных задач, внесли Диниц Е.А., Карзанов А.В., Новикова Н.М., Edmonds J., Ford L.R., Fulkerson D.R., Goldberg A.V., Karp R.M., Orlin J.B., Rao S., Skutella M., Tardos E., Tarjan R. E. и др.

Цель работы

Целью диссертационной работы является исследование вопросов сводимости многоиндексных задач транспортного типа к классу задач поиска потока в сети. Нахождение условий сводимости и выделение подклассов сводимых многоиндексных задач. Построение алгоритмов решения (целочисленных) многоиндексных задач, основанных на полученных результатах сводимости, анализ вычислительной сложности построенных алгоритмов.

Методы исследований

В работе используются методы сетевой оптимизации, теории линейного и целочисленного линейного программирования, теории графов, а также теория вычислительной сложности.

Достоверность результатов

Достоверность полученных в диссертационной работе результатов обусловлена строгостью математических доказательств.

Научная новизна

1. Формализованы схемы сведения задач линейного программирования к классу задач поиска потока в сети. В рамках построенных схем сведения исследованы вопросы сводимости многоиндексных задач транспортного типа к классу задач поиска потока в сети.

2. В рамках схемы сведения с сохранением соответствия ребер получен исчерпывающий ответ вопроса сводимости к классу задач поиска потока в сети:

– установлен критерий сводимости многоиндексных задач;
– на основе критерия сводимости выделен класс сводимых многоиндексных задач – класс многоиндексных задач с 2-вложенной структурой, возникающий в ряде прикладных задач;
– предложена конструктивная схема сведения класса многоиндексных задач с 2-вложенной структурой являющаяся оптимальной (сведение с асимптотически меньшими вычислительными затратами невозможно, любое увеличение вычислительных затрат на сведение не приводит к расширению класса сводимых многоиндексных задач);
– разработан подход к решению 2-вложенных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в сети; данный подход применен также при исследовании несовместных 2-вложенных многоиндексных систем и многокритериальных 2-вложенных многоиндексных задач с кусочно-постоянными критериями оптимальности.

3. В рамках схемы сведения с сохранениями соответствия ребер получен исчерпывающий ответ вопроса сводимости к классу задач поиска потока в древовидной сети:

– установлен критерий сводимости многоиндексных задач;
– на основе критерия сводимости выделен класс сводимых многоиндексных задач – класс многоиндексных задач с 1-вложенной структурой, возникающий в ряде прикладных задач;
– предложена конструктивная схема сведения класса многоиндексных задач с 1-вложенной структурой также являющаяся оптимальной;
– разработан подход к решению 1-вложенных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в сети; данный подход применен также при исследовании многокритериальных 1-вложенных многоиндексных задач с кусочно-постоянными критериями оптимальности.

4. Полученные результаты сводимости обобщаются при исследовании более широкого класса схем сводимости. В рамках схемы сведения с сохранением целочисленности установлен критерий сводимости, справедливый при выполнении известной гипотезы о неравенстве классов P и NP.
5. В рамках схемы сведения с сохранением соответствия циклов исследованы вопросы сводимости к классу задач поиска потока в сети:

– найдено достаточное условие сводимости многоиндексных задач;
– на основе условия сводимости выделен класс сводимых многоиндексных задач – класс многоиндексных задач с декомпозиционной структурой, возникающий в ряде прикладных задач;
– разработан подход к решению декомпозиционных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в сети; данный подход применен также при исследовании несовместных декомпозиционных многоиндексных систем, многокритериальных декомпозиционных многоиндексных задач с кусочно-постоянными критериями оптимальности;
– выделен новый полиномиально разрешимый подкласс в NP-трудном классе целочисленных многоиндексных задач;
– результаты сводимости применены при разработке приближенных алгоритмов решения ряда NP-трудных целочисленных многоиндексных задач с декомопзицонной структурой.

Теоретическая и практическая ценность работы

Полученные в диссертационной работе теоретические результаты относятся к теории многоиндексных задач транспортного типа и сетевой оптимизации.

В рамках исследованного класса многоиндексных задач ставятся различные прикладные задачи распределения ресурсов в производственных системах, транспортных системах, сетях передачи данных и так далее.

Предложенные в диссертационной работе методы исследования многоиндексных задач транспортного типа были использованы при разработке следующих программных систем: программное обеспечение «Заказ-О», программное обеспечение «Нагнетатель», программное обеспечение «Проектировщик-1», программное обеспечение «Проектировщик-2», заказчик ФГУП «РФЯЦ-ВНИИЭФ», г. Саров; «Модуль расчета оптимального плана производства для диалоговой системы объемно-календарного планирования производственных мощностей, функционирующей на предприятии», заказчик ОАО «ОКБМ Африкантов», г. Нижний Новгород; программная система ПО «Кристалл», заказчик ФГУП «ФНПЦ НИИИС им. Ю.Е. Седакова», г. Нижний Новгород, (см. приложения 1-7).

Проведенные исследования выполнены в рамках заданий Минобразования РФ номер госрегистрации 0120.0506816, тема НИР «Математическое моделирование и создание новых методов анализа динамических систем и систем оптимизации », номер госрегистрации 01.2.00 1 07703, тема НИР «Математическое моделирование и создание новых методов анализа динамических систем и систем оптимизации», номер госрегистрации 01201252499, тема НИР «Математическое моделирование и создание новых методов анализа эволюционных систем и систем оптимизации»; гранта Президента Российской Федерации для государственной поддержки молодых российских ученых - кандидатов наук, МК-3473.2010.1, тема «Быстрые алгоритмы решения задач глобальной оптимизации и их приложения»; ФЦП при финансовой поддержке Минобрнауки России, гос. соглашение № 14.B37.21.0878., тема «Высокоточные супервычисления и решение задач глобальной оптимизации на основе информационного подхода».

Результаты диссертационной работы используются в учебном процессе Нижегородского государственного университета им. Н.И. Лобачевского на факультете ВМК при преподавании курсов «Моделирование сложных систем», «Модели и методы эффективного использования распределенных вычислительных систем» и спецсеминара магистров кафедры ИАНИ [21, 26, 31] (см. приложение 8).

Апробация результатов

Полученные в диссертационной работе результаты обсуждались на Всероссийской конференции КоГраф (Н.Новгород, 2002 г.); Международных научно-практических семинарах «Высокопроизводительные параллельные вычисления на кластерных системах» (Н.Новгород, 2002, 2003 г.); Нижегородских сессиях молодых ученых (Саров, 2003 г., 2004 г., Нижний Новгород, 2004 г.); Научно-технической конференции ООО ТЕКОМ «Технические, программные и математические аспекты управления сложными распределенными системами» (Н.Новгород, 2003 г.); Юбилейной научно-технической конференции ВМК ННГУ и НИИ ПМК, «Математика и кибернетика 2003» (Н.Новгород, 2003 г.); VI международном конгрессе по математическому моделированию (Н.Новгород, 2004 г.); Конференциях «Технологии Microsoft в теории и практике программирования» (Москва, 2005 г., Н.Новгород, 2006 г., 2007 г., 2009 г.); Международной научной конференции, приуроченной к 200-летию со дня рождения Карла Густава Якоби (Калининград, 2005 г.); XIV, XV, XVI Международных конференциях «Проблемы теоретической кибернетики» (Пенза, 2005 г., Казань 2008 г., Нижний Новгород 2011 г.); V, VI Московских международных конференциях по исследованию операций (Москва 2007 г., 2010 г.); Международной научной конференции «Параллельные вычислительные технологии» (Нижний Новгород, 2009 г.); VIII Международной конференции «Дискретные модели в теории управляющих систем» (Москва, 2009 г.); X

Международном семинаре «Дискретная математика и ее приложения» (Москва, 2010 г.), Одесском семинаре по дискретной математике (Одесса, 2011 г.).

Результаты работы также обсуждались на научных семинарах отделения информатики университета г. Падерборна (Германия, г. Падерборн, 2005 г., 2007 г., руководитель семинара профессор Monien B.), научных семинарах Кафедры информатики и автоматизации научных исследований факультета Вычислительной математики и кибернетики Нижегородского государственного университета им. Н.И. Лобачевского (2006 г., 2011 г., 2013 г., руководитель семинара профессор Прилуцкий М.Х.), научном семинаре Кафедры исследования операций Математико-механического факультета Санкт-Петербургского государственного университета (2012 г., руководитель семинара профессор Романовский И.В.), научном семинаре центра исследования операций и бизнес статистики университета г. Левен (Бельгия, г. Левен, 2012 г., руководитель семинара профессор Spieksma F.C.R.), научном семинаре Вычислительного центра имени А.А. Дородницына Российской академии наук (2012 г., руководитель семинара академик Евтушенко Ю.Г.).

Основные результаты, полученные в ходе выполнения диссертационной работы, отражены в 41 научной работе [2–20, 22–25, 27–30, 32–37, 86–90, 110–112], в том числе в 11 статьях, опубликованных в ведущих рецензируемых научных журналах (Автоматика и телемеханика, Известия РАН. Теория и системы управления, Управление большими системами, Вестник Нижегородского университета им. Н.И. Лобачевского) из списка, рекомендованного ВАК [8, 9, 13, 20, 22, 25, 27, 28, 29, 30, 90], а также в 3 учебно-методических работах [21, 26, 31].

Все основные результаты, выносимые на защиту, получены автором самостоятельно.

Структура и содержание работы

Работа состоит из введения, пяти глав, заключения, списка литературы и приложения.

Во введении отражена актуальность класса многоиндексных задач транспортного типа, приведен обзор основных результатов в области многоиндексных транспортных задач, сформулированы цели исследования, показана научная новизна работы.

В главе 1 приводится обзор прикладных задач, относящихся к классу многоиндексных задач транспортного типа. Описываются содержательные постановки подобных прикладных задач, строятся их математические модели.

В главе 2 излагаются общие методы исследования задач транспортного типа: поиск потока в сети с двусторонними пропускными способностями; поиск потока в древовидной транспортной сети; поиск потока в несовместной транспортной сети; построение циклической декомпозиции потока; метод ортогональных проекций при решении транспортных систем; решение многокритериальных транспортных задач с кусочно-постоянными критериями оптимальности. Предложенные алгоритмы используются далее при разработке методов исследования многоиндексных задач транспортного типа, основанных на полученных в работе результатах сводимости.

В главе 3 описываются постановки исследуемых многоиндексных задач транспортного типа. Постановки задач приводится с использованием разработанной схемы формализации многоиндексных задач. Данная формализация используется далее в работе при описании результатов исследования многоиндексных задач. В главе описывается общая формализация схем сводимости, применяемая далее при классификации схем сведения многоиндексных задач к классу задач поиска потока в сети.

В главе 4 описываются результаты исследования сводимости с сохранением соответствия ребер класса многоиндексных задач к классу задач поиска потока минимальной стоимости в сети. Особенностью рассматриваемой концепции сводимости является существование соответствия между переменными исходной задачи и дугами вспомогательной сети. Предлагаемая схема сведения гарантирует, что произвольный оптимальный поток вспомогательной сети будет определять такое оптимальное решение исходной задачи, в котором переменным присваивается значение потока вдоль соответствующих дуг сети. В рамках рассматриваемой схемы сведения в данной главе устанавливается, что условие 2-вложенности множества подмножеств индексов, по которым происходит суммирование в системе ограничений задачи, является критерием сводимости. Предлагается конструктивная схема сведения класса многоиндексных задач с 2-вложенной структурой, являющаяся оптимальной (сведение с асимптотически меньшими вычислительными затратами невозможно, любое увеличение вычислительных затрат на сведение не приводит к расширению класса сводимых многоиндексных задач). В главе описывается подход к решению 2-вложенных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в сети. Данный подход применяется также при исследовании несовместных 2-вложенных многоиндексных систем и многокритериальных 2-вложенных многоиндексных задач с кусочно-постоянными критериями оптимальности. В главе 4 также получены аналогичные результаты при исследовании сводимости с сохранением соответствия ребер класса многоиндексных задач к классу задач поиска потока минимальной стоимости в древовидной сети. В рамках рассматриваемой схемы сведения в данной главе устанавливается, что условие 1-вложенности является критерием сводимости к классу задач поиска потока в древовидной сети. Предлагается конструктивная схема сведения класса многоиндексных задач с 1-вложенной структурой, также являющаяся оптимальной. В главе описывается подход к решению 1-вложенных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в древовидной сети. Данный подход применяется также при исследовании многокритериальных 1-вложенных многоиндексных задач с кусочно-постоянными критериями оптимальности. Полученные результаты сводимости обобщаются в главе 4 при исследовании более широкого класса схем сводимости. В рамках схемы сведения с сохранением целочисленности устанавливается критерий сводимости, справедливый при выполнении известной гипотезы о неравенстве классов P и NP.

В главе 5 описываются результаты исследования сводимости с сохранением соответствия циклов класса многоиндексных задач к классу задач поиска потока минимальной стоимости в сети. Особенностью рассматриваемой концепции сводимости является существование соответствия между переменными исходной задачи и простыми циклами вспомогательной сети. Предлагаемая схема сведения гарантирует, что произвольный оптимальный поток вспомогательной сети будет определять такое оптимальное решение исходной задачи, в котором переменным присваиваются значения потока вдоль соответствующих простых циклов. Величина потока вдоль простых циклов определяется через циклическую декомпозицию потока. В рамках рассматриваемой схемы сведения в данной главе устанавливается, что условие декомпозиционности многоиндексной задачи является достаточным условием сводимости. Предлагается конструктивная схема сведения класса многоиндексных задач с декомпозиционной структурой. В главе описывается подход к решению декомпозиционных (целочисленных) многоиндексных задач транспортного типа, основанный на сводимости к поиску потока в сети. Данный подход применяется также при исследовании несовместных декомпозиционных многоиндексных систем, многокритериальных декомпозиционных многоиндексных задач с кусочно-постоянными критериями оптимальности. Результаты сводимости применены также при разработке приближенных алгоритмов решения ряда NP-трудных целочисленных многоиндексных задач с декомопзицонной структурой.

В заключении подведены основные итоги проведенных в диссертационной работе исследований.

Глава 1. Многоиндексные задачи распределения ресурсов

Широкий класс прикладных задач распределения ресурсов относится к классу многоиндексных задач транспортного типа [24, 27–29, 32, 35–37, 65, 83, 84]. В главе приводятся содержательные постановки подобных прикладных задач и строятся их математические модели. Далее в главе 3 будет определена общая схема формализации многоиндексных задач и определено соответствие между рассмотренными прикладными задачами и классами многоиндексных задач транспортного типа. Полученные в главах 4 и 5 теоретические результаты сводимости многоиндексных задач к классу задач поиска потока в сети также будут проиллюстрированы на примере приведенных в данной главе прикладных задач.

1.1. Транспортная задача с промежуточными пунктами

Содержательная постановка

Имеются пункты производства, промежуточные пункты и пункты потребления однородного продукта. Заданы максимально возможные объемы производства продукта каждым пунктом производства, минимально допустимые объемы потребления продукции каждым пунктом потребления, ограничения на объемы перевозки продукта от каждого пункта производства до каждого промежуточного пункта, ограничения на объемы перевозки продукта от каждого промежуточного пункта до каждого пункта потребления.

Требуется найти план перевозок, обеспечивающий эффективное функционирование системы и удовлетворяющий ограничениям пунктов на возможные производимые, потребляемые и передаваемые объемы однородного продукта.

Исходные параметры

Пусть I – множество пунктов производства, J – множество промежуточных пунктов, K – множество пунктов потребления. Обозначим через A_i максимальный объем производства продукта пунктом производства i; B_k – минимально допустимый объем продукта, который необходимо доставить в пункт потребления k; C_{ij} - максимальное количество продукта, которое может быть доставлено из пункта производства i в пункт потребления k; D_{jk} - максимальный объем продукта, который может быть доставлен из промежуточного пункта j в пункт потребления k, $i \in I$, $j \in J$, $k \in K$.

Варьируемые параметры

Обозначим через x_{ijk} количество продукта, которое будет перевезено из пункта производства *i* через промежуточный пункт *j* в пункт потребления *k*, $i \in I$, $j \in J$, $k \in K$.

Ограничения математической модели

Общая математическая модель проблемы перевозки однородного продукта представляет собой следующую систему ограничений:

$$\sum_{j \in J} \sum_{k \in K} x_{ijk} \le A_i, \; i \in I, \tag{1.1}$$

(объем производства продукта пунктом производства не должен превышать максимально возможного объема);

$$\sum_{i \in I} \sum_{j \in J} x_{ijk} \ge B_k, \; k \in K, \tag{1.2}$$

(пункт потребления должен получить объем продукта не ниже минимально допустимого объема);

$$\sum_{k \in K} x_{ijk} \le C_{ij}, \; i \in I, \; j \in J, \tag{1.3}$$

(объем перевозки продукта от пункта производства до промежуточного пункта не должен превышать максимально допустимого объема);

$$\sum_{i \in I} x_{ijk} \le D_{jk}, \; j \in J, k \in K, \tag{1.4}$$

(объем перевозки продукта от промежуточного пункта до пункта потребления не должен превышать максимально допустимого объема);

$$x_{ijk} \ge 0, i \in I, \; j \in J, k \in K, \tag{1.5}$$

(естественные ограничения на переменные).

Критерий оптимальности

Критерии оптимальности, определяющие эффективность функционирования системы, могут зависеть от различных показателей искомого плана перевозок. Рассмотрим стоимостные показатели плана. Обозначим через a_i себестоимость производства единицы продукции пунктом производства *i*; b_k – цену реализации единицы продукции пункту потребления *k*; c_{ij} – стоимость доставки единицы продукции из пункта производства i в промежуточный пункт j; d_{jk} – стоимость доставки единицы продукции из промежуточного пункта j в пункт потребления k. Задача минимизации расхода заключается в определении такого плана перевозок x_{ijk}, $i \in I$, $j \in J$, $k \in K$, для которого выполняются ограничения (1.1)-(1.5), и принимает оптимальное значение критерий

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (a_i + c_{ij} + d_{jk}) x_{ijk} \to \min . \tag{1.6}$$

Задача максимизации дохода заключается в определении такого плана перевозок x_{ijk}, $i \in I$, $j \in J$, $k \in K$, для которого выполняются ограничения (1.1)-(1.5), и принимает оптимальное значение критерий

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} b_k x_{ijk} \to \max . \tag{1.7}$$

Задача максимизации прибыли заключается в определении такого плана перевозок x_{ijk}, $i \in I$, $j \in J$, $k \in K$, для которого выполняются ограничения (1.1)-(1.5), и принимает оптимальное значение критерий

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} (b_k - a_i - c_{ij} - d_{jk}) x_{ijk} \to \max . \tag{1.8}$$

В качестве показателей плана могу быть рассмотрены показатели, основанные на предпочтениях, определяемых контролируемыми элементами системы. Пусть контролируемыми элементами системы являются пункты потребления. С каждым из пунктов потребления k будем связывать функцию предпочтения $\phi_k : R \to R$, определяющую предпочтение потребителя k относительно объема потребляемого им продукта, $k \in K$. Заметим, что объем продуктов, потребленных элементом k, определяется как $\sum_{i \in I} \sum_{j \in J} x_{ijk}$, $k \in K$. На практике потребители затрудняются определить свои предпочтения относительно каждого из возможного объема потребленного ими продукта. Такие предпочтения обычно задаются путем ранжирования интервалов соответствующих объемов продукта. Формализация такого рода предпочтений связана с определением конечной вложенной последовательности интервалов соответствующих объемов продукта. Пусть каждый из потребителей задает p+1 вложенных интервалов $[B_{kl}^-, B_{kl}^+]$, $l = \overline{0, p}$, $k \in K$. Вложенность интервалов означает выполнение следующих условий $[B_{kl}^-, B_{kl}^+] \subseteq [B_{kl+1}^-, B_{kl+1}^+]$, $l = \overline{0, p-1}$, $k \in K$. Так интервал $[B_{k0}^-, B_{k0}^+]$ является наиболее предпочтительным для потребителя k, интервал $[B_{k1}^-, B_{k1}^+]$ менее предпочтителен и т.д.

Тогда функции предпочтения определяются как кусочно-постоянные функции следующего вида:

$$\phi_k(u)=\begin{cases}0, \text{если} \quad u\in[B_{k0}^-, B_{k0}^+]\\ l, \text{если} \quad \text{найдется} \quad l\in\{1,...,p\}, \text{что} \quad u\in[B_{kl}^-, B_{kl}^+], u\notin[B_{kl-1}^-, B_{kl-1}^+],\\ p+1, \text{если} \quad u\notin[B_{kp}^-, B_{kp}^+]\end{cases}$$

где $u\in R,\ \ k\in K.$ Тогда задача выбора наиболее предпочтительного плана перевозок ставится как следующая задача многокритериальной оптимизации. Необходимо определить план перевозок x_{ijk}, $i\in I,\ j\in J,\ k\in K$, для которого выполняются ограничения (1.1)-(1.5) и принимают оптимальные значения критерии, определяющие предпочтения пунктов потребления:

$$\phi_k(\sum_{i\in I}\sum_{j\in J}x_{ijk})\to\min,\ \ k\in K\ . \tag{1.9}$$

Если множество потребителей упорядочено с точки зрения их приоритета при определении эффективности функционирования всей системы в целом, то при решении задачи многокритериальной оптимизации (1.1)-(1.5),(1.9) может быть рассмотрена лексикографическая схема компромисса. В случае равнозначности потребителей в качестве схемы компромисса при решении задачи многокритериальной оптимизации (1.1)-(1.5),(1.9) может быть рассмотрена максиминная свертка.

1.2. Задача формирования портфеля заказов

Содержательная постановка

Имеется предприятие, состоящее из подразделений. Определены такты планирования (в качестве тактов могут рассматриваться сутки, месяца, кварталы и т.д.) работы предприятия. Каждое из подразделений характеризуется минимально допустимым и максимально возможным объемом работ, который может быть выполнен в подразделении за период планирования. Также подразделения характеризуется минимально допустимым и максимально возможным объемом работ, который может быть выполнен в подразделении в каждый из тактов планирования. Известны заказы, поступающие на предприятие. Заказы характеризуются минимально допустимым и максимально требуемым объемом работ. Для каждого из заказов задано раннее время начала и позднее время окончания работ.

Требуется определить план выполнения работ по заказам. В случае, если имеющиеся мощности предприятия не позволяют выполнить работы по всем заказам в заданные сроки, требуется определить план выполнения работ, удовлетворяющий «жестким» ограничениям и минимизирующий штрафы за нарушения «желательных» ограничений. Постановка задачи определяется выбором жестких и желательных ограничений.

Исходные параметры

Пусть I – множество подразделений предприятия, J – множество заказов, $T = \{1,2,..., p\}$ – множество номеров тактов планирования. Обозначим через A_i^- и A_i^+ минимально допустимый и максимально возможный объем работ, который подразделение i способно выполнить в течение всего периода планирования; B_{it}^- и B_{it}^+ – минимально допустимый и максимально возможный объем работ, который подразделение i способно выполнить в такт планирования t; C_j^- и C_j^+ – минимально возможный и максимально требуемый объем работ по заказу j; t_j^- и t_j^+ – минимально раннее время начала выполнения работ и максимально позднее время окончания выполнения работ по заказу j, $i \in I, j \in J, t \in T$.

Варьируемые параметры

Обозначим через x_{ijt} объем работ, выполненный подразделением i по заказу j в такт планирования t, $i \in I, j \in J, t \in T$.

Ограничения математической модели

Допустимым портфелем заказов будем называть набор значений x_{ijt}, $i \in I, j \in J, t \in T$, удовлетворяющий следующей системе ограничений:

$$A_i^- \le \sum_{j \in J} \sum_{t \in T} x_{ijt} \le A_i^+, \ i \in I, \tag{1.10}$$

(объем работ, выполненный подразделением в течение всего периода планирования, должен удовлетворять заданным двусторонним ограничениям);

$$B_{it}^- \le \sum_{j \in J} x_{ijt} \le B_{it}^+, \ i \in I, t \in T, \tag{1.11}$$

(объем работ, выполненный подразделением в течение такта планирования, должен удовлетворять заданным двусторонним ограничениям);

$$C_j^- \le \sum_{i\in I}\sum_{t\in T} x_{ijt} \le C_j^+, \;\; j\in J, \tag{1.12}$$

(объем выполненных работ по заказу должен удовлетворять заданным двусторонним ограничениям);

$$x_{ijt} = 0, t\in T \setminus \{t_j^-,...,t_j^+\}, i\in I, j\in J, \tag{1.13}$$

(работы по заказу могут выполняться лишь в течение заданного периода);

$$x_{ijt} \ge 0, \;\; i\in I, j\in J, t\in T, \tag{1.14}$$

(естественные ограничения на переменные).

Критерий оптимальности

Пусть система (1.10)-(1.14) несовместна. Несовместность может быть вызвана несогласованностью внутренних ограничений предприятия. Вопрос о несогласованности внутренних ограничений связан с исследованием совместности системы (1.10),(1.11),(1.14). В общем случае несовместность обусловлена тем, что ограниченные мощности предприятия не позволяют выполнить требуемые работы по поступившим заказам в заданные сроки.

Разобьем ограничения на «жесткие» и «желательные» и будем рассматривать задачу определения плана выполнения работ, удовлетворяющего жестким ограничениям и минимизирующего штрафы за нарушения желательных ограничений. Далее рассмотрим две задачи: задачу с возможными нарушениями требуемых объемов работ по заказам (соответствует случаю, когда ограничение (1.12) является желательным) и задачу с возможными нарушениями сроков выполнения работ по заказам (соответствует случаю, когда ограничение (1.13) является желательным).

Рассмотрим задачу формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам. Обозначим через c_j^- и c_j^+ штрафы за нарушение на единицу минимально возможного и максимально требуемого объем работ по заказу j; y_j^- и y_j^+ – искомые величины, на которые будут нарушены нижняя и верхняя границы объема работ по заказу j, $j\in J$. Введем дополнительные ограничения

$$C_j^- - y_j^- \le \sum_{i\in I}\sum_{t\in T} x_{ijt} \le C_j^+ + y_j^+,\ j \in J, \tag{1.15}$$

(объем выполненных работ по заказу с учетом возможных нарушений должен удовлетворять заданным двусторонним ограничениям);

$$y_j^-, y_j^+ \ge 0,\ j \in J, \tag{1.16}$$

(естественные ограничения на переменные). Тогда задача формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам заключается в определении величин x_{ijk}, $i\in I$, $j\in J$, $k\in K$ и y_j^-, y_j^+, $j \in J$, удовлетворяющих системе ограничений (1.10),(1.11),(1.13)–(1.16) и минимизирующих суммарный штраф за нарушение требуемых объемов работ

$$\sum_{j\in J}(c_j^- y_j^- + c_j^+ y_j^+) \to \min. \tag{1.17}$$

Рассмотрим задачу формирования портфеля заказов с возможными нарушениями сроков выполнения работ по заказам. Обозначим через d_j^- и d_j^+ штраф за нарушение начального и конечного сроков выполнения работ на один такт, на одну единицу объема. Тогда задача заключается в определении величин x_{ijk}, $i\in I$, $j\in J$, $k\in K$ и y_j^-, y_j^+, $j \in J$, удовлетворяющих системе ограничений (1.10)–(1.12),(1.14) и минимизирующих суммарный штраф за нарушение сроков выполнения работ

$$\sum_{i\in I}\sum_{j\in J}\sum_{t=1}^{t_j^- -1} d_j (t_j^- - t) x_{ijt} + \sum_{i\in I}\sum_{j\in J}\sum_{t=t_j^+ +1}^{p} d_j (t - t_j^+) x_{ijt} \to \min. \tag{1.18}$$

Замечание. Предложенная задача формирования портфеля заказов может быть использована руководством предприятий при заключении договоров с заказчиками на начальном этапе планирования, когда еще не определены детальные планы выполнения заказов, и планируемые работы оцениваются лишь в объемных показателях. На данном этапе возникает предложенная задача формирования портфеля заказов, которая заключается в согласовании (во времени) требуемых объемов работ по заказам с производительностью предприятия. Если предприятие (с учетом имеющихся обязательств) не может в заданные сроки выполнить требуемые объемы работ по заказам, ставится задача поиска компромиссного решения.

1.3. Объемно-календарное планирование переработки газового конденсата

Предприятие по переработки газового конденсата представляет собой производственную систему с непрерывным циклом изготовления основной продукции, в которой из сырья, под воздействием технологических режимов, изготавливаются продукты производства. Основными элементами рассматриваемой системы являются:

– емкости резервуарного парка,

– технологические установки,

– потребители продукции.

Емкости характеризуются максимальной вместимостью. Для технологических установок заданы их производительности – минимально и максимально возможные объемы производимых продуктов в такт времени. Для потребителей в каждый из тактов планируемого времени (тактами могут быть недели, месяцы, кварталы и т.д.) заданы минимально возможные и максимально требуемые объемы продуктов.

Рассматриваемая производственная система функционирует по следующей схеме. Изначально сырье (газовый конденсат) поступает в емкости резервуарного парка предприятия, откуда по трубопроводам поступает на технологические установки. В технологических установках под воздействием технологических режимов из поступающего сырья производятся продукты производства (бензин АИ-92, АИ-80, ДТ, пропан и др.). Далее готовая продукция отправляется потребителям. Требуется определить план производства и поставки продуктов потребителям с учетом ограничений производственной системы и требований потребителей, при котором система будет функционировать эффективно. Условия эффективности формализуются в виде критериев оптимальности, которые определяют различные экономические показатели плана.

Исходные параметры

Пусть I – множество емкостей под сырье, J – множество технологических установок, K – множество различных видов готовой продукции, выпускаемой предприятием, P – множество потребителей готовой продукции, T – множество тактов планирования. Обозначим через A_i максимальную вместимость емкости i; B^-_{jk} и B^+_{jk} - минимальная и максимальная производительность технологической установки j по продукции k; C^{-}_{ptk} и C^{+}_{ptk} – минимально возможный и максимально требуемый для потребителя p в такт t объемы продукта k, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$.

Варьируемые параметры

Обозначим через x_{ijkpt} объем сырья, который из емкости i поступит на установку j для изготовления продукта k, который будет отправлен потребителю p в такт t, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$.

Ограничения математической модели

Общая математическая модель объемно-календарного планирования процесса переработки газового конденсата представляет собой следующую систему ограничений:

$$\sum_{j\in J}\sum_{k\in K}\sum_{p\in P}\sum_{t\in T} x_{ijkpt} \le A_i,\ i \in I, \tag{1.19}$$

(объем сырья помещенные в емкость не должен превышать ее вместимости);

$$B^{-}_{jk} \le \sum_{i\in I}\sum_{p\in P} x_{ijkpt} \le B^{+}_{jk},\ j \in J, k \in K, t \in T, \tag{1.20}$$

(в каждый из тактов времени объем производимой продукции должен удовлетворять ограничениям на производительность установки);

$$C^{-}_{ptk} \le \sum_{i\in I}\sum_{j\in J} x_{ijkpt} \le C^{+}_{ptk},\ p \in P, t \in T, k \in K, \tag{1.21}$$

(объем каждого из видов продукции, получаемый потребителями в каждый из тактов времени, должен удовлетворять заданным двусторонним ограничениям);

$$x_{ijkpt} \ge 0,\ i \in I, j \in J, k \in K, p \in P, t \in T, \tag{1.22}$$

(естественные ограничения на переменные).

Критерий оптимальности

Условия эффективного функционирования системы связаны с различными экономическими показателями плана, которые определяются следующими параметрами: a_{kpt} – ожидаемый доход от реализации единицы продукции k потребителю p в такт t; b_{jk} – затраты технологической установки j на переработку единицы сырья в продукцию k; c_{ij} – затраты на перемещения единицы сырья из емкости i в технологическую установку j; d_t

– ожидаемая стоимость сырья в такт t; e_{kp} – затраты на доставку единицы готовой продукции k потребителю p, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$.

Тогда задача максимизации суммарного дохода предприятия от реализации продукции заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{k \in K} \sum_{p \in P} \sum_{t \in T} a_{kpt} \sum_{i \in I} \sum_{j \in J} x_{ijkpt} \to \max . \tag{1.23}$$

Задача минимизации суммарных затрат технологических установок на переработку сырья в готовую продукцию заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{j \in J} \sum_{k \in K} b_{jk} \sum_{i \in I} \sum_{p \in P} \sum_{t \in T} x_{ijkpt} \to \min . \tag{1.24}$$

Задача минимизации суммарных затрат на перемещение сырья из емкостей в технологические установки заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{i \in I} \sum_{j \in J} c_{ij} \sum_{k \in K} \sum_{p \in P} \sum_{t \in T} x_{ijkpt} \to \min . \tag{1.25}$$

Задача минимизации суммарных затрат на закупку предприятием сырья заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{t \in T} d_t \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{p \in P} x_{ijkpt} \to \min . \tag{1.26}$$

Задача минимизации суммарных затрат на доставку предприятием готовой продукции потребителям заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{k \in K} \sum_{p \in P} e_{kp} \sum_{i \in I} \sum_{j \in J} \sum_{t \in T} x_{ijkpt} \to \min . \tag{1.27}$$

Задача максимизации суммарной прибыли предприятия заключается в определении такого плана x_{ijkpt}, $i \in I$, $j \in J$, $k \in K$, $p \in P$, $t \in T$, для которого выполняются ограничения (1.19)-(1.22) и принимает оптимальное значение критерий

$$\sum_{i \in I} \sum_{j \in J} \sum_{k \in K} \sum_{p \in P} \sum_{t \in T} (a_{kpt} - b_{jk} - c_{ij} - d_t - e_{kp}) x_{ijkpt} \to \max. \tag{1.28}$$

Замечание. Предложенная постановка задачи планирования переработки газового конденсата возникает в нефтегазовой отрасли на этапе объемно-календарного планирования, когда требуется согласовать обобщенные мощности предприятия с требованиями потребителей. При этом производительности установок, определяющие обобщенные мощности предприятия, связаны с выбранными режимами их функционирования. Установка устроена таким образом, что при выбранном режиме ее работы известны интервальные границы по каждому из видов готовой продукции на выходе установки, что определяет комплексную природу работы таких установок. Переключение режимов установки является сложным технологическим процессом, требующим согласования и множества регламентных работ. Поэтому на этапе объемно-календарного планирования целесообразным оказывается решение задачи при заданных (стандартных) режимах установок.

1.4. Задача составления расписания занятий

Для каждого из преподавателей известны занятия, которые он может провести, и время, в которое он может провести занятия. Для каждой из аудиторий известно время, когда она доступна. Необходимо составить допустимое расписание проведения занятий учебного заведения, т.е. определить время проведения занятий, преподавателя и выделить аудиторию с учетом описанных ограничений.

Наряду с проблемой составления допустимого расписания будем рассматривать задачу построения оптимального расписания. Пусть для каждого из преподавателей заданы предпочтения относительно занятий и времени их проведения; для каждой аудитории известны предпочтения относительно времени, когда ее можно использовать. Необходимо составить расписание проведения занятий учебного заведения, максимизирующее суммарные предпочтения.

Исходные параметры

Пусть I – множество преподавателей, J – множество занятий, K – множество аудиторий, T – множество непересекающихся интервалов времени, в которые занятия могут проходить. Обозначим через A_i – множество занятий, которые могут быть проведены преподавателем i, $A_i \subseteq J$; B_i – множество интервалов времени, в которые преподаватель i может провести занятия, $B_i \subseteq T$; C_k – множество интервалов времени, в течение которых аудитория k доступна, $C_k \subseteq T$, $i \in I$, $j \in J$, $k \in K$. Предпочтения будем связывать с весовыми коэффициентами. Тогда дополнительно обозначим через a_{ij} – предпочтение преподавателя i относительно занятия j; b_{it} – предпочтения преподавателя i относительно интервала времени t; c_{kt} – предпочтение для аудитории k относительно интервала времени t, $i \in I$, $j \in J$, $k \in K$, $t \in T$.

Варьируемые параметры

Обозначим через x_{ijkt} переменную, принимающую значение равное 1, если преподаватель i проводит занятие j в аудитории k в интервал времени t, и принимающую значение равное 0, иначе, $i \in I$, $j \in J$, $k \in K$, $t \in T$.

Ограничения математической модели

Общая математическая модель проблемы составления допустимого расписания представляет собой следующую систему ограничений:

$$\sum_{j \in J} \sum_{k \in K} x_{ijkt} \le 1, \; i \in I, t \in T, \tag{1.29}$$

(преподаватель может провести не более одного занятия в каждый из интервалов времени);

$$\sum_{i \in I} \sum_{k \in K} \sum_{t \in T} x_{ijkt} = 1, \; j \in J, \tag{1.30}$$

(занятие должно быть проведено);

$$\sum_{i \in I} \sum_{j \in J} x_{ijkt} \le 1, \; k \in K, t \in T, \tag{1.31}$$

(в каждый интервал времени аудитория может быть выделена не более чем под одно занятие);

$$x_{ijkt} = 0, \; j \in J \setminus A_i, i \in I, k \in K, t \in T, \tag{1.32}$$

(преподаватель не проводит занятия, которые не может провести);

$$x_{ijkt} = 0,\ t \in T \setminus B_i, i \in I, j \in J, k \in K, \quad (1.33)$$

(преподаватель не проводит занятия, в интервалы времени, в которые не может провести);

$$x_{ijkt} = 0,\ t \in T \setminus C_k, i \in I, j \in J, k \in K, \quad (1.34)$$

(аудитория не может быть выделена в тот интервал времени, когда она недоступна);

$$x_{ijkt} \in \{0,1\},\ i \in I, j \in J, k \in K, t \in T, \quad (1.35)$$

(естественные ограничения на переменные).

Критерий оптимальности

Задача составления оптимального расписания, максимизирующего суммарные предпочтения заключается в определении расписания x_{ijkt}, $i\in I$, $j\in J$, $k\in K$, $t\in T$, для которого выполняются ограничения (1.29)–(1.35) и принимает оптимальное значение критерий

$$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{t\in T}(a_{ij} + b_{it} + c_{kt})x_{ijkt} \to \max. \quad (1.36)$$

Несложно увидеть, что проблема составления допустимого расписания (1.29)-(1.35) сводится к задаче составления оптимального расписания (1.29)-(1.31),(1.35),(1.36). Действительно, пусть

$$a_{ij} = \begin{cases} 0,\ j \in A_i \\ -1,\ j \in J \setminus A_i \end{cases},\ i \in I,$$

$$b_{it} = \begin{cases} 0, t \in B_i \\ -1, t \in T \setminus B_i \end{cases},\ i \in I,$$

$$c_{kt} = \begin{cases} 0, t \in C_k \\ -1, t \in T \setminus C_k \end{cases},\ k \in K.$$

Тогда система ограничений (1.29)-(1.35) совместна тогда и только тогда, когда оптимальное значение критерия в соответствующей задаче (1.29)-(1.31),(1.35),(1.36) равно нулю.

Замечание. Предложенная постановка задачи составления расписания занятий возникает в ряде приложений. Так, например, подобная задача возникает при проведении курсов по повышению квалификации, выездных семинаров и т.д., когда необходимо составить расписание (выбрать лекторов, назначить аудитории и время лекций). А далее участники семинара смогу записаться на интересующие их лекции.

Для составления расписания занятие в учебных заведениях предложенная постановка задачи составления расписания является слишком идеализированной, т.к. не учитывает целый ряд встречающихся на практике требований. Однако здесь возможны иные сценарии использования предложенной задачи:

– данная задача может быть решена на начальном этапе планирования для составления «грубого» расписания,
– предложенная задача может быть рассмотрена в качестве релаксации к более общей задаче составления расписания занятий в учебных заведениях, ее решение будет являться нижней (недостижимой) оценкой, используемой, например, в методе ветвей и границ.

Выводы

Рассмотрены примеры постановок прикладных задач распределения ресурсов, относящихся к классу многоиндексных задач транспортного типа:

– транспортная задача с промежуточными пунктами,

– задача формирования портфеля заказов,

– задача объемно-календарного планирования переработки газового конденсата,

– задача составления расписания занятий.

Далее в работе при описании общей схемы формализации многоиндексных задач транспортного типа будет определено место рассмотренных прикладных задач в классе многоиндексных задач транспортного типа. Приведенные прикладные многоиндексные задачи будут использованы также в диссертационной работе при иллюстрации полученных теоретических результатов.

Глава 2. Методы исследования транспортных задач

Глава посвящена изложению методов исследования ряда задач транспортного типа, рассмотренных в [13, 29, 30, 32, 83, 85]. Строятся алгоритмы решения рассматриваемых транспортных задач, проводится анализ их вычислительной сложности. Далее в работе, если не оговорено иного, при анализе сложности алгоритмов будем оценивать, как и в [82], количество вычислительных операций (арифметических операции, логических операции, элементарных операций с памятью) на машине с произвольным доступом к памяти. Полученные в данной главе результаты будут использованы далее при построении алгоритмов решения многоиндексных задач транспортного типа, основанных на сводимости к классу задач поиска потока в сети.

2.1. Поток в сети с двусторонними ограничениями

Многие работы посвящены методам решения классических потоковых задач – задачи поиска максимального потока и задачи поиска потока минимальной стоимости заданной величины в сети с односторонними пропускными способностями. Обзор и оценки вычислительной сложности разработанных методов решения данных задач приводятся в [115, 151, 156]. В данном разделе рассматриваются задачи поиска потока в сети с двусторонними пропускными способностями. Приводится схема, позволяющая сводить задачи поиска потока в сети с двусторонними пропускными способностями к классическим задачам поиска потока в сети (с односторонними пропускными способностями). Формальное описание и доказательство корректности данной схемы были описаны в [13, 18, 86], сама схема основана на идеи, предложенной в [106]. Также отметим, что алгоритмы решения потоковых задач с двусторонними пропускными способностями применялись ранее при исследовании задач распределения ресурсов в иерархических системах [3, 4, 6, 10, 11, 14, 16, 17, 19, 23, 33, 87, 88, 110].

Задан ориентированный граф без петель $G=(V_G,A_G)$, $A_G \subseteq V_G^{\,2}$. Здесь V_G и A_G – множество вершин и дуг графа G соответственно. Там, где это не вызывает неоднозначности, будем опускать нижний индекс G при записи множества вершин и множества дуг графа. Среди множества вершин графа выделены специальные вершины s, t, называемые истоком и стоком соответственно, $s \neq t$, $s,t \in V$. Обозначим через a_{ij}, b_{ij} и c_{ij} заданные значения нижней пропускной способности, верхней пропускной способности и стоимости дуги (i, j), соответственно, $(i, j) \in A$. Здесь $0 \le a_{ij} \le b_{ij}$, $(i, j) \in A$. Через x_{ij} обозначим поток вдоль дуги (i, j), $(i, j) \in A$; через v – величину потока.

Тогда задача поиска максимального потока заключается в нахождение значений x_{ij}, $(i, j) \in A$, и v, являющихся решением следующей задачи линейного программирования:

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = \begin{cases} v, i = s \\ 0, i \in V \setminus \{s,t\}, \\ -v, i = t \end{cases} \tag{2.1}$$

$$0 \le x_{ij} \le b_{ij}, (i, j) \in A, \tag{2.2}$$

$$v \to \max. \tag{2.3}$$

Далее задачу (2.1)-(2.3) будем обозначать через $z_{MF}(G; s; t; b_{ij}, (i, j) \in A)$ (*MF* соответствует англоязычному названию *max flow problem*).

Определение 2.1. Набор значений x_{ij}, $(i, j) \in A$, удовлетворяющих системе ограничений (2.1),(2.2) задачи $z_{MF}(G; s; t; b_{ij}, (i, j) \in A)$ при фиксированном значении параметра v, будем называть потоком величины v.

Определение 2.2. Пусть x_{ij}, $(i, j) \in A$, и v – решение задачи $z_{MF}(G; s; t; b_{ij}, (i, j) \in A)$. Тогда набор значений x_{ij}, $(i, j) \in A$, будем называть максимальным потоком; а значение v – величиной максимального потока.

При фиксированном значение параметра v задача поиска потока минимальной стоимости заданной величины заключается в нахождении значений x_{ij}, $(i, j) \in A$, являющихся решением задачи линейного программирования с системой ограничений (2.1),(2.2) и критерием

$$\sum_{(i,j)\in A} c_{ij} x_{ij} \to \min. \tag{2.4}$$

Задачу (2.1),(2.2),(2.4) будем обозначать через $z_{MCF}(G; s; t; v; b_{ij}, c_{ij}, (i, j) \in A)$ (*MCF* соответствует англоязычному названию *min cost flow problem*).

Проблема поиска допустимой циркуляции заключается в нахождении величин x_{ij}, $(i, j) \in A$, удовлетворяющих системы линейных неравенств

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, i \in V, \tag{2.5}$$

$$a_{ij} \le x_{ij} \le b_{ij}, (i,j) \in A. \tag{2.6}$$

Проблему (2.5),(2.6) будем обозначать через $z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$ (FC соответствует англоязычному названию *feasible circulation problem*). Задача поиска циркуляции минимальной стоимости ставится как задача линейного программирования (2.5),(2.6),(2.4) и обозначается через $z_{MCC}(G; a_{ij} b_{ij}, c_{ij}, (i,j) \in A)$ (MCC соответствует англоязычному названию *min cost circulation problem*).

Определение 2.3. Набор значений x_{ij}, $(i,j) \in A$, удовлетворяющих системе ограничений (2.5),(2.6) проблемы $z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$, будем называть допустимой циркуляцией.

Определение 2.4. Решение задачи $z_{MCC}(G; a_{ij} b_{ij}, c_{ij}, (i,j) \in A)$ будем называть циркуляцией минимальной стоимости.

Приведем схему, позволяющую сводить проблему поиска допустимой циркуляции (задачу поиска циркуляции минимальной стоимости) к задаче поиска максимального потока (задаче поиска потока минимальной стоимости заданной величины).

Определение 2.5. Пусть задана проблема поиска допустимой циркуляции $z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$, где $G = (V, A)$, тогда соответствующей задачей поиска максимального потока будем называть задачу $z_{MF}(G'; s; t; b'_{ij}, (i,j) \in A')$, определяемую следующим образом:

– $G' = (V', A')$,

– $V' = V \cup \{s, t\}$, где s, t - новые вершины,

– $A' = A \cup (\bigcup_{i \in V} \{(s,i), (i,t)\})$,

– $b'_{ij} = b_{ij} - a_{ij}$, $(i,j) \in A$,

– $b'_{si} = \sum_{(j,i)\in A} a_{ji}$, $b'_{it} = \sum_{(i,j)\in A} a_{ij}$, $i \in V$.

Теорема 2.1. Для существования допустимой циркуляции проблемы $z_{FC}(G;a_{ij}b_{ij},(i,j)\in A)$ необходимо и достаточно, чтобы величина максимального потока соответствующей задачи $z_{MF}(G';s;t;b'_{ij},(i,j)\in A')$ равнялась $\sum_{(i,j)\in A} a_{ij}$.

Доказательство. Рассмотрим проблему поиска допустимой циркуляции $Z_{FC}=Z_{FC}(G;a_{ij}b_{ij},(i,j)\in A)$ и соответствующую ей задачу поиска максимального потока $Z_{MF}=Z_{MF}(G';s;t;b'_{ij},(i,j)\in A')$.

Пусть x_{ij}, $(i,j)\in A$, допустимая циркуляция проблемы z_{FC}. Тогда покажем, что величина максимального потока в соответствующей задачи z_{MF} равна $\sum_{(i,j)\in A} a_{ij}$. Приведем конструктивное доказательство, основанное на построении максимального потока задачи Z_{MF}. Определим значения x'_{ij}, $(i,j)\in A'$, следующим образом:

– $x'_{ij}=x_{ij}-a_{ij}$, $(i,j)\in A$;

– $x'_{si}=\sum_{(j,i)\in A} a_{ji}$, $x'_{it}=\sum_{(i,j)\in A} a_{ij}$, $i\in V$.

Рассмотрим величины $\sum_{(i,j)\in A'} x'_{ij}-\sum_{(j,i)\in A'} x'_{ji}$, $i\in V'$. Рассмотрим следующие возможные случаи. Пусть $i=s$, тогда

$$\sum_{(i,j)\in A'} x'_{ij}-\sum_{(j,i)\in A'} x'_{ji}=\sum_{(i,j)\in A'} x'_{ij}=\sum_{i'\in V}\sum_{(j',i')\in A} a_{j'i'}=\sum_{(i',j')\in A} a_{i'j'}.$$

Пусть $i=t$, тогда

$$\sum_{(i,j)\in A'} x'_{ij}-\sum_{(j,i)\in A'} x'_{ji}=-\sum_{(j,i)\in A'} x'_{ji}=-\sum_{i'\in V}\sum_{(i',j')\in A} a_{i'j'}=-\sum_{(i',j')\in A} a_{i'j'}.$$

Пусть $i\in V$, тогда согласно (2.5)

$$\sum_{(i,j)\in A'} x'_{ij}-\sum_{(j,i)\in A'} x'_{ji}=x_{it}+\sum_{(i,j)\in A} x'_{ij}-x_{si}-\sum_{(j,i)\in A} x'_{ji}=$$

$$=\sum_{(i,j)\in A} a_{ij}+\sum_{(i,j)\in A}(x_{ij}-a_{ij})-\sum_{(j,i)\in A} a_{ji}-\sum_{(j,i)\in A}(x_{ji}-a_{ji})=\sum_{(i,j)\in A} x_{ij}-\sum_{(j,i)\in A} x_{ji}=0.$$

Отсюда x'_{ij}, $(i,j)\in A'$, удовлетворяет ограничению (2.1), при фиксированном значении параметра $v=\sum_{(i,j)\in A} a_{ij}$. Далее рассмотрим величины x'_{ij}, $(i,j)\in A'$:

$$x'_{si}=b'_{si},\ x'_{it}=b'_{it},\ i\in V;\ x'_{ij}=x_{ij}-a_{ij},\ (i,j)\in A,$$

тогда в соответствии с неравенством (2.6) выполняется следующее условие $0 \le x'_{ij} \le b_{ij} - a_{ij} = b'_{ij}$, $(i, j) \in A$. Таким образом, x'_{ij}, $(i, j) \in A'$, удовлетворяет ограничению (2.2). Следовательно, x'_{ij}, $(i, j) \in A'$, является потоком величины $\sum\limits_{(i,j)\in A} a_{ij}$. По теореме о максимальном потоке и минимальном разрезе [107] величина максимального потока не превосходит пропускной способности любого из разрезов сети. Рассмотрим разрез $(s, V' \setminus \{s\})$, его пропускная способность равна $\sum\limits_{(i,j)\in A} a_{ij}$, но так как величина потока x'_{ij}, $(i, j) \in A'$, также равна $\sum\limits_{(i,j)\in A} a_{ij}$, то построенный поток является максимальным.

Далее пусть x'_{ij}, $(i, j) \in A'$, максимальный поток величины $\sum\limits_{(i,j)\in A} a_{ij}$ в задаче z_{MF}. Тогда покажем, что существует допустимая циркуляция проблемы z_{FC}. Приведем здесь также конструктивное доказательство, основанное на построении допустимой циркуляции x_{ij}, $(i, j) \in A$. Определим значения x_{ij}, $(i, j) \in A$, следующим образом: $x_{ij} = x'_{ij} + a_{ij}$, $(i, j) \in A$. Рассмотрим величины $\sum\limits_{(i,j)\in A} x_{ij} - \sum\limits_{(j,i)\in A} x_{ji}$, $i \in V$. Заметим, что так как величина максимального потока x'_{ij}, $(i, j) \in A'$, равна $\sum\limits_{(i,j)\in A} a_{ij}$, то $x'_{si} = \sum\limits_{(j,i)\in A} a_{ji}$, $x'_{it} = \sum\limits_{(i,j)\in A} a_{ij}$, $i \in V$. Отсюда согласно (2.1)

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = \sum_{(i,j)\in A} (x'_{ij} + a_{ij}) - \sum_{(j,i)\in A} (x'_{ji} + a_{ji}) =$$

$$= \sum_{(i,j)\in A} x'_{ij} + x'_{it} - \sum_{(j,i)\in A} x'_{ji} - x'_{si} = \sum_{(i,j)\in A'} x'_{ij} - \sum_{(j,i)\in A'} x'_{ji} = 0,\ i \in V.$$

Таким образом, x_{ij}, $(i, j) \in A$, удовлетворяет ограничению (2.5) и является циркуляцией. Далее рассмотрим величины $x_{ij} = x'_{ij} + a_{ij}$, $(i, j) \in A$. Так как $b'_{ij} = b_{ij} - a_{ij}$, $(i, j) \in A$, то в соответствии с ограничением (2.2) справедливо следующее условие $a_{ij} \le x_{ij} = x'_{ij} + a_{ij} \le b'_{ij} + a_{ij} = b_{ij}$, $(i, j) \in A$. Таким образом, x_{ij}, $(i, j) \in A$, удовлетворяет ограничению (2.6). Следовательно, построенная циркуляция является допустимой циркуляцией проблемы z_{FC}. *Теорема доказана.*

На основании конструктивного доказательства теоремы 2.1 можно сформулировать следующее следствие.

Следствие 2.1. Если величина максимального потока задачи $z_{MF} = z_{MF}(G'; s; t; b'_{ij}, (i, j) \in A')$, соответствующей проблеме поиска допустимой циркуляции

$z_{FC} = z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$, равна $\sum_{(i,j)\in A} a_{ij}$ и x'_{ij}, $(i,j) \in A'$ – максимальный поток задачи Z_{MF}, то $x_{ij} = x'_{ij} + a_{ij}$, $(i,j) \in A$, является допустимой циркуляцией проблемы z_{FC}.

Алгоритм 2.1. Поиск допустимой циркуляции.

Вход. Проблема $z_{FC} = z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$.

Шаг 1. Построить задачу поиска максимального потока $z_{MF} = z_{MF}(G'; s; t; b'_{ij}, (i,j) \in A')$, соответствующую проблеме z_{FC}. Перейти на шаг 2.

Шаг 2. Найти максимальный поток x'_{ij}, $(i,j) \in A'$, в задаче z_{MF}. Перейти на шаг 3.

Шаг 3. Если величина максимального потока задачи z_{MF} равна $\sum_{(i,j)\in A} a_{ij}$, то перейти на шаг 4; иначе проблема z_{FC} несовместна и алгоритм завершен.

Шаг 4. Построить допустимую циркуляцию проблемы z_{FC} по следующему правилу $x_{ij} = x'_{ij} + a_{ij}$, $(i,j) \in A$, алгоритм завершен.

Согласно определению (2.5), количество вершин и дуг в графе G' задачи $z_{MF} = z_{MF}(G'; s; t; b'_{ij}, (i,j) \in A')$, соответствующей задаче $z_{FC} = z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$, равно $|V| + 2 = O(|V|)$ и $2|V| + |A| = O(|V| + |A|)$ соответственно. Применим для решения задачи поиска максимального потока z_{MF} алгоритм, требующий $\alpha(n,m)$ вычислительных операций, где n и m количество вершин и дуг графа G', соответственно. Отсюда:

Утверждение 2.1. Алгоритм 2.1 решения проблемы $z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$ требует $\alpha(O(|V|), O(|V| + |A|))$ вычислительных операций.

Определение 2.6. Пусть задана задача поиска циркуляции минимальной стоимости $z_{MCC}(G; a_{ij} b_{ij}, c_{ij}, (i,j) \in A)$, где $G = (V, A)$, тогда соответствующей задачей поиска потока минимальной стоимости заданной величины будем называть задачу $z_{MCF}(G'; s; t; v; b'_{ij}, c'_{ij}(i,j) \in A')$, определяемую следующим образом. Здесь граф $G' = (V', A')$, вершины s, t и пропускные способности b'_{ij}, $(i,j) \in A'$, зададим по аналогии с определением (2.5). Параметр v зададим равным $\sum_{(i,j)\in A} a_{ij}$. Стоимости дуг определим следующим образом: $c'_{ij} = c_{ij}$, $(i,j) \in A$; $c'_{ij} = 0$, $(i,j) \in A' \setminus A$.

Следствие 2.2. Если задача $z_{MCF} = z_{MCF}(G';s;t;v;b'_{ij},c'_{ij}(i,j) \in A')$, соответствующая задаче поиска циркуляции минимальной стоимости $z_{MCC} = z_{MCC}(G;a_{ij}b_{ij},c_{ij},(i,j) \in A)$ совместна и x'_{ij}, $(i,j) \in A'$ – поток минимальной стоимости заданной величины задачи z_{MCF}, то $x_{ij} = x'_{ij} + a_{ij}$, $(i,j) \in A$, является циркуляцией минимальной стоимости задачи z_{MCC}.

Алгоритм 2.2. Поиск циркуляции минимальной стоимости.

Вход. Задача $z_{MCC} = z_{MCC}(G;a_{ij}b_{ij},c_{ij},(i,j) \in A)$.

Шаг 1. Используя алгоритм 2.1, решить проблему $z_{FC} = z_{FC}(G;a_{ij}b_{ij},(i,j) \in A)$. Если проблема z_{FC} совместна, то перейти на шаг 2; иначе задача z_{MCC} несовместна и алгоритм завершен.

Шаг 2. Построить задачу поиска потока минимальной стоимости заданной величины $z_{MCF}(G';s;t;v;b'_{ij},c'_{ij}(i,j) \in A')$, соответствующую задаче z_{MCC}. Перейти на шаг 2.

Шаг 3. Найти поток минимальной стоимости заданной величины x'_{ij}, $(i,j) \in A'$, задачи z_{MCF}. Перейти на шаг 3.

Шаг 4. Построить циркуляцию минимальной стоимости задачи z_{MCC} по следующему правилу $x_{ij} = x'_{ij} + a_{ij}$, $(i,j) \in A$, алгоритм завершен.

Как уже было установлено, количество вершин и дуг в графе G' равно $O(|V|)$ и $O(|V|+|A|)$ соответственно. Применим для решения задачи поиска потока минимальной стоимости заданной величины z_{MCF} алгоритм, требующий $\beta(n,m)$ вычислительных операций, где n и m количество вершин и дуг графа G' соответственно. Отсюда:

Утверждение 2.2. Алгоритма 2.2 решения задачи поиска циркуляции минимальной стоимости $z_{MCC}(G;a_{ij}b_{ij},c_{ij},(i,j) \in A)$ требует $\alpha(O(|V|),O(|V|+|A|)) + \beta(O(|V|),O(|V|+|A|))$ вычислительных операций.

2.2. Поток в древовидной транспортной сети

В данном разделе будут рассмотрены специальные потоковые алгоритмы анализа древовидных сетевых моделей, предложенные независимо в работах [85, 150]. Вопросы исследования древовидных сетевых моделей рассматривались также в [112].

Задано корневое ориентированное дерево $G=(V,A)$, $V \subseteq A^2$. Среди множества вершин дерева выделена специальная вершина s – корень дерева, $s \in V$, и подмножество T – множество листьев, $T \subseteq V$. Обозначим через a_i, b_i и c_i заданные значения нижней пропускной способности, верхней пропускной способности и стоимости вершины i соответственно, $i \in V$. Здесь $0 \le a_i \le b_i$, $i \in V$. Через x_i обозначим поток через вершину i, $i \in V$.

Тогда задача поиска потока минимальной стоимости в древовидной сети заключается в нахождение значений x_i, $i \in V$, являющихся решением следующей задачи линейного программирования:

$$x_i - \sum_{(i,j)\in A} x_j = 0, i \in V \setminus T, \tag{2.7}$$

$$a_i \le x_i \le b_i, (i,j) \in A, \tag{2.8}$$

$$\sum_{i\in V} c_i x_i \to \min. \tag{2.9}$$

Далее задачу (2.7)-(2.9) будем обозначать через $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$ (*MCFT* соответствует англоязычному названию *min cost flow problem in tree-like network*). Проблему исследования совместности системы (2.8), (2.9) будем обозначать через $z_{FFT}(G; a_i, b_i, i \in V)$ (*FFT* соответствует англоязычному названию *feasible flow problem in tree-like network*).

Определение 2.7. Набор значений x_i, $i \in V$, удовлетворяющих системе ограничений (2.8), (2.9) проблемы $z_{FFT}(G; a_i, b_i, i \in V)$, будем называть допустимым потоком древовидной сети.

Определение 2.8. Набор значений x_i, $i \in V$, являющийся решением задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$, будем называть потоком минимальной стоимости древовидной сети.

Рассмотрим приведенные величины a_i^p, b_i^p, $i \in V$, которые определяются с помощью рекуррентных соотношений:

$$a_i^p = a_i,\ i \in T,$$

$$b_i^p = b_i,\ i \in T,$$

$$a_i^p = \max\left(a_i,\ \sum_{j \in Q(i)} a_j^p\right),\ i \in V \setminus T, \tag{2.10}$$

$$b_i^p = \min\left(b_i,\ \sum_{j \in Q(i)} b_j^p\right),\ i \in V \setminus T,$$

где $Q(i) = \{ j \mid (i, j) \in A\}$, $i \in V$. Метод исследования совместности системы (2.8), (2.9) основан на следующей теореме о приведенных границах [85].

Теорема 2.2. Система (2.8), (2.9) совместна тогда и только тогда, когда $a_i^p \le b_i^p$, $i \in V$.

Доказательство. Необходимость условий теоремы очевидна. Доказательство достаточности проведем конструктивно. Покажем, что при выполнении условий теоремы специальным образом построенный набор значений неизвестных x_i, $i \in V$, является допустимым решением системы (2.8), (2.9).

Так как по условию теоремы $a_s^p \le b_s^p$, то найдется значение $x_s \in [a_s^p, b_s^p]$. Далее пусть для некоторого $i \in V$ найдено значение переменной x_i, что $x_i \in [a_i^p, b_i^p]$. Из рекуррентных соотношений (2.10), в случае выполнения условий теоремы, следует, что $[a_i^p, b_i^p] \subseteq [a_i, b_i]$, и тем самым $x_i \in [a_i, b_i]$. Значения неизвестных x_j, $j \in Q(i)$, определяются из следующих соотношений:

$$x_i = \sum_{(i,j) \in A} x_j, \tag{2.11}$$

$$a_j^p \le x_j \le b_j^p,\ j \in Q(i). \tag{2.12}$$

Система (2.11),(2.12) совместна в силу того, что $x_i \in [a_i^p, b_i^p]$ и $[a_i^p, b_i^p] \subseteq [\sum_{(i,j) \in A} a_j^p,\ \sum_{(i,j) \in A} b_j^p]$. Построенный таким образом набор значений неизвестных является решением системы (2.8), (2.9). *Теорема доказана.*

На основании конструктивного доказательства теоремы 2.2 можно построить алгоритм поиска допустимого потока древовидной сети.

Алгоритм 2.3. Поиск допустимого потока древовидной сети.

Вход. Проблема $z_{FFT} = z_{FFT}(G; a_i, b_i, i \in V)$.

Шаг 1. Используя рекуррентные соотношения (2.10), вычислить приведенные границы a_i^p, b_i^p, $i \in V$. Если $a_i^p \le b_i^p$, $i \in V$, перейти на шаг 2; иначе проблема Z_{FFT} несовместна и алгоритм завершен.

Шаг 2. Выбрать $x_s \in [a_s^p, b_s^p]$. Перейти на шаг 3.

Шаг 3. Если найдется $i \in V \setminus T$, что величина x_i найдена и не найдены величины x_j, $j \in Q(i)$, то перейти на шаг 4; иначе построенный набор x_i, $i \in V$, является допустимым потоком древовидной сети проблемы z_{FFT}, алгоритм завершен.

Шаг 4. Вычислить величины x_j, $j \in Q(i)$, как решение системы уравнений (2.11),(2.12), для чего воспользуемся следующей процедурой. Определить начальные значения переменных $x_j = a_j^p$, $j \in Q(i)$. Просмотреть в произвольном порядке элементы $j \in Q(i)$, если $\sum_{l \in Q(i)} x_l < x_i$, то $x_j := x_j + \min(b_j^p - x_j, x_i - \sum_{l \in Q(i)} x_l)$ и перейти к следующему элементу множества $Q(i)$; иначе система (2.11),(2.12) решена, и перейти на шаг 3.

Граф G имеет древовидную структуру поэтому, используя обход дерева в ширину, можно упорядочить вершина графа таким образом, что $V = \{i_1, ..., i_{|V|}\}$ и при этом $Q(i_l) \subseteq \{i_{l+1}, ..., i_{|V|}\}$, $l = \overline{1, |V|}$. Данная процедура требует $O(|V|)$ вычислительных операций. При вычислении приведенных границ a_i^p, b_i^p, $i \in V$, на шаге 1 алгоритма 2.3, основанном на рекуррентных соотношениях (2.10), необходимо просматривать вершины упорядоченного множества $\{i_1, ..., i_{|V|}\}$ в обратном порядке, тем самым будет гарантирована корректность вычисления приведенных границ по данным рекуррентным соотношениям. При обработке каждый из вершин i_l просматриваются приведенные границы вершин множества $Q(i_l)$. Так как

$$\sum_{i \in V} Q(i) = O(|V|), \tag{2.13}$$

то шаг 1 алгоритма 2.3 требует $O(|V|)$ вычислительных операций. Выбирать вершины на шаге 3 алгоритма 2.3 следует, просматривая упорядоченное множество $\{i_1, ..., i_{|V|}\}$ в прямом порядке. Тогда для каждой из просматриваемых вершин i_l (кроме вершин множества T)

будет выполнены соответствующие шаги 4 алгоритма. На шаге 4 предложенная процедура решения системы (2.11),(2.12) требует $|Q(i_l)|$ вычислительных операций. В силу (2.13) выполнение цикла шаг 3-4 алгоритма 2.3 требует $O(|V|)$ вычислительных операций. Отсюда:

Утверждение 2.3. Алгоритм 2.3 решения задачи поиска допустимого потока $z_{FFT}(G;a_i,b_i,i\in V)$ требует $O(|V|)$ вычислительных операций.

При исследовании задачи $z_{MCFT}(G;a_i,b_i,c_i,i\in V)$, не уменьшая общности, будем считать, что $c_i=0$, $i\in V\setminus T$. Действительно, если это не так, то для каждой из вершин $i\in T$ рассмотрим путь $j_1,...,j_t$, соединяющий лист i с корнем дерева, здесь $(j_l,j_{l+1})\in A$, $l=\overline{1,t-1}$, и $j_1=s$, $j_t=i$; тогда определим модифицированную стоимость вершины i как $c_i'=\sum_{l=1}^{t}c_i$. Модифицированные стоимость остальных вершин определим равными нулю, $c_i'=0$, $i\in V\setminus T$. Легко увидеть, что задачи $z_{MCFT}(G;a_i,b_i,c_i,i\in V)$ и $z_{MCFT}(G;a_i,b_i,c_i',i\in V)$ эквивалентны.

Введем ряд вспомогательных обозначений необходимых далее при исследовании задачи поиска потока минимальной стоимости в древовидной сети. Обозначим через $t(i)$ путь из корня s к листу i в дереве G, таким образом, $t(i)=(v_0,...,v_k)$, где k – длина пути $t(i)$, $v_0=s$, $v_k=i$, $(v_j,v_{j+1})\in A$, $j=\overline{0,k-1}$, $i\in T$. Через $T(G)$ обозначим множество всех путей из корня к листьям в графе G, $T(G)=\{t(i)\,|\,i\in T\}$. Пусть $t=(v_0,...,v_k)$ – путь в графе G и $v\in V$. Тогда для удобства будем обозначать $v\in t$, если путь t проходит через вершину v, т.е. найдется $j\in\{0,...,k\}$, что $v_j=v$; в противном случае будем обозначать $v\notin t$.

Лемма 2.1. Пусть $i^*\in T$ такой, что $|c_i|\le|c_{i^*}|$, $i\in T$. Если задача $z_{MCFT}(G;a_i,b_i,c_i,i\in V)$ совместна, то существует оптимальное решение x_i^*, $i\in V$, задачи $z_{MCFT}(G;a_i,b_i,c_i,i\in V)$ такое, что выполняется следующее соотношение

$$x_{i^*}^*=\begin{cases}\min\limits_{\overline{j=0,k}}(b_{v_j}^p-\sum\limits_{l=j}^{k-1}\sum\limits_{v\in Q(v_l)\setminus\{v_{l+1}\}}a_v^p), \text{если } c_{i^*}\le 0\\ \max\limits_{\overline{j=0,k}}(a_{v_j}^p-\sum\limits_{l=j}^{k-1}\sum\limits_{v\in Q(v_l)\setminus\{v_{l+1}\}}b_v^p), \text{если } c_{i^*}>0\end{cases}, \tag{2.14}$$

где $t(i^*) = (v_0,...,v_k)$.

Доказательство. Пусть задача $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$ совместна, и x'_i , $i \in V$, ее оптимальное решение задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. Обозначим через $i^* \in T$ лист дерева, для которого выполняется $|c_i| \leq |c_{i^*}|$, $i \in T$. Покажем, что можно построить оптимальное решение задачи, для которого выполняется соотношение (2.14). Далее рассмотрим два случая: $c_{i^*} \leq 0$ и $c_{i^*} > 0$.

1. Пусть $c_{i^*} \leq 0$. Обозначим $h = \min\limits_{j=0,k}(b^p_{v_j} - \sum\limits_{l=k-1}^{j} \sum\limits_{v \in Q(v_l) \setminus \{v_{l+1}\}} a^p_v)$, где $t(i^*) = (v_0,...,v_k)$. Если $x'_{i^*} = h$, то для оптимального решения x'_i, $i \in V$, выполняется соотношение (2.14), и лемма доказана. Если $x'_{i^*} > h$, то x'_i, $i \in V$, не удовлетворяет системе (2.7),(2.8), однако это противоречит тому, что x'_i, $i \in V$, – оптимальное решение задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. Далее рассмотрим случай $x'_{i^*} < h$.

Если $c_{i^*} = 0$, то $c_i = 0$, $i \in T$, и тогда произвольное допустимое решение задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$ будет являться оптимальным решением. Тогда, если $x'_{v_j} < b^p_{v_j}$, $j = \overline{0,k}$, то пусть $\Delta = \min\limits_{j=0,k}(b^p_{v_j} - x'_{v_j})$ и модифицируем решение x'_i , $i \in V$, по следующему правилу $x'_i := x'_i + \Delta$, $i \in t(i^*)$. Если $c_{i^*} > 0$, то покажем от противного, что существует $j \in \{0,...,k\}$ такой, что $x'_{v_j} = b^p_{v_j}$. Предположим, что $x'_{v_j} < b^p_{v_j}$, $j = \overline{0,k}$, и пусть $\Delta = \min\limits_{j=0,k}(b^p_{v_j} - x'_{v_j})$. Тогда построим поток x''_i, $i \in V$, по следующему алгоритму

– $x''_i := x'_i$, $i \in V$,

– $x''_i := x''_i + \Delta$, $i \in t(i^*)$.

Несложно увидеть, что x''_i , $i \in V$, удовлетворяет системе (2.7),(2.8). Т.к. $\Delta > 0$, то выполняется $\sum\limits_{i \in V} c_i x''_i < \sum\limits_{i \in V} c_i x'_i$, что противоречит оптимальности решения x'_i, $i \in V$. Получаем противоречие, предположение неверно. Тогда обозначим $j' = \max\{ j \mid x'_{v_j} = b^p_{v_j}, j \in \{0,...,k\}\}$. При этом $j' < k$, т.к. в противном случае $x'_{i^*} = h$.

Далее покажем от противного, что найдется $j'' \in \{j',...,k-1\}$ и $v'' \in Q(v_{j''}) \setminus \{v_{j''+1}\}$, что $x_{v''} > a_{v''}^p$. Предположим, что $x_v = a_v^p$, $v \in Q(v_j) \setminus \{v_{j+1}\}$, $j = \overline{j',k-1}$. Справедливо соотношение

$$b_{v_{j'}}^p = x'_{v_{j'}} = x'_{i^*} + \sum_{l=j'}^{k-1} \sum_{v \in Q(v_l) \setminus \{v_{l+1}\}} x'_v = x'_{i^*} + \sum_{l=j'}^{k-1} \sum_{v \in Q(v_l) \setminus \{v_{l+1}\}} a_v^p,$$

и, следовательно, $x'_{i^*} = b_{v_{j'}}^p - \sum_{l=j'}^{k-1} \sum_{v \in Q(v_l) \setminus \{v_{l+1}\}} a_v^p \geq \min_{j=\overline{0,k}} (b_{v_j}^p - \sum_{l=j}^{k-1} \sum_{v \in Q(v_l) \setminus \{v_{l+1}\}} a_v^p) = h$. Однако $x'_{i^*} < h$, получаем противоречие, предположение неверно. Далее пусть $j'' \in \{j',...,k-1\}$, $v'' \in Q(v_{j''}) \setminus \{v_{j''+1}\}$ и $x_{v''} > a_{v''}^p$. Тогда найдется путь t' в дереве G от v'' до некоторого листа из T, что $\Delta' = \min_{v \in t'} \{x_v - a_v^p\} > 0$. Обозначим через t'' путь в дереве G от $v_{j''+1}$ до i^*. По построению $\Delta'' = \min_{v \in t''} \{b_v^p - x_v\} > 0$.

Определим x_i^*, $i \in V$, по следующему алгоритму

– $x_i^* := x'_i$, $i \in V$,

– $x_i^* := x_i^* - \min(\Delta', \Delta'')$, $i \in t'$,

– $x_i^* := x_i^* + \min(\Delta', \Delta'')$, $i \in t''$.

По построению x_i^*, $i \in V$, удовлетворяет системе (2.7),(2.8). Кроме того, т.к. $|c_i| \leq |c_{i^*}|$, $i \in T$, и $c_{i^*} \leq 0$, то выполняется $c_{i^*} \leq c_i$, $i \in T$. Следовательно, $\sum_{i \in V} c_i x_i^* \leq \sum_{i \in V} c_i x'_i$. В силу оптимальности решения x'_i, $i \in V$, набор x_i^*, $i \in V$, также является оптимальным решением задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. По построению $x_{i^*}^* > x'_{i^*}$. Если $x_{i^*}^* = h$, то соотношение (2.14) выполняется. В противном случае определим $x'_i := x_i^*$, $i \in V$, и повторим построение, описанное выше. Несложно увидеть, что такое повторение возможно не более $|V|$ раз. Следовательно, за конечное число шагов будет построено оптимальное решение x_i^*, $i \in V$, удовлетворяющее соотношению (2.14).

2. Пусть $c_{i^*} > 0$. Обозначим $h = \max_{j=\overline{0,k}} (a_{v_j}^p - \sum_{l=j}^{k-1} \sum_{v \in Q(v_l) \setminus \{v_{l+1}\}} b_v^p)$, где $t(i^*) = (v_0,...,v_k)$. Если $x'_{i^*} = h$, то для оптимального решения x'_i, $i \in V$, выполняется соотношение (2.14), и лемма доказана. Если $x'_{i^*} < h$, то x'_i, $i \in V$, не удовлетворяет системы (2.7),(2.8), однако это

противоречит тому, что x'_i, $i \in V$, – оптимальное решение задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. Далее рассмотрим случай $x'_{i^*} > h$.

Доказательство данного случая можно провести по аналогии с доказательством случая 1. По аналогии можно показать, что существует $j \in \{0,...,k\}$ такой, что $x'_{v_j} = a^p_{v_j}$, и тогда обозначим $j' = \max\{ j \mid x'_{v_j} = a^p_{v_j}, j \in \{0,...,k\}\}$. Также по аналогии можно показать, что найдется $j'' \in \{j',...,k-1\}$ и $v'' \in Q(v_{j''}) \setminus \{v_{j''+1}\}$, что $x_{v''} < b^p_{v''}$. Тогда существует путь t' в дереве G от v'' до некоторого листа из T, что $\Delta' = \min_{v \in t'}\{b^p_v - x_v\} > 0$. Через t'' обозначим путь в дереве G от $v_{j''+1}$ до i^*. По построению $\Delta'' = \min_{v \in t''}\{x_v - a^p_v\} > 0$. Далее определим x^*_i, $i \in V$, по следующему алгоритму

– $x^*_i := x'_i$, $i \in V$,

– $x^*_i := x^*_i + \min(\Delta', \Delta'')$, $i \in t'$,

– $x^*_i := x^*_i - \min(\Delta', \Delta'')$, $i \in t''$.

По построению x^*_i, $i \in V$, удовлетворяет системе (2.7),(2.8). Кроме того, т.к. $|c_i| \le |c_{i^*}|$, $i \in T$, и $c_{i^*} > 0$, то выполняется $c_{i^*} \ge c_i$, $i \in T$. Следовательно, $\sum_{i \in V} c_i x^*_i \le \sum_{i \in V} c_i x'_i$. В силу оптимальности решения x'_i, $i \in V$, набор x^*_i, $i \in V$, также является оптимальным решением задачи $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. По построению $x^*_{i^*} < x'_{i^*}$. Если $x^*_{i^*} = h$, то соотношение (2.14) выполняется. В противном случае определим $x'_i := x^*_i$, $i \in V$, и повторим построение, описанное выше. Как и в случае 1, такое повторение возможно не более $|V|$ раз. Следовательно, за конечное число шагов будет построено оптимальное решение x^*_i, $i \in V$, удовлетворяющее соотношению (2.14). *Лемма доказана.*

Доказанная лемма позволяет предложить следующий алгоритм поиска потока минимальной стоимости в древовидной сети.

Алгоритм 2.4. Поиск потока минимальной стоимости древовидной сети.

Вход. Задача $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$.

Шаг 1. Используя рекуррентные соотношения (2.10), вычислить приведенные границы a^p_i, b^p_i, $i \in V$. Если $a^p_i \le b^p_i$, $i \in V$, перейти на шаг 2; иначе задача $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$ несовместна, и алгоритм завершен.

Шаг 2. Упорядочить множество листьев T по невозрастанию модуля стоимости, $T = \{i_1, ..., i_{|T|}\}$, $|c_{i_j}| \geq |c_{i_{j+1}}|$, $j = \overline{1, |T|-1}$. Далее пусть $j = 1$, переход на шаг 3.

Шаг 3. Определить $x^*_{i_j}$, используя соотношение (2.14). Если $j = |T|$, перейти на шаг 4; иначе пересчитать пропускные способности вершин пути $t(i_j)$ по следующему правилу $a_i := \max(0, a_i - x^*_{i_j}), b_i := b_i - x^*_{i_j}, i \in t(i_j)$, удалить из графа G вершину i_j, $j := j+1$, пересчитать приведенные границы a^p_i, b^p_i, $i \in V$, и переход на шаг 3.

Шаг 4. Используя найденные значения x^*_i, $i \in T$, и условие баланса (2.7), определить x^*_i, $i \in V \setminus T$. Алгоритм завершен.

Как уже отмечалось выше, расчет приведенных границ на шаге 1 может быть осуществлен, используя $O(|V|)$ вычислительных операций. Сортировка на шаге 2 требует $O(|V|^2)$ вычислительных операций. Выполнение вычислений, используя соотношение (2.14) и пересчет приведенных границ на шаге 3 может быть выполнено, используя $O(|V|)$ вычислительных операций. Шаг 3 повторяется $O(|V|)$ раз. Выполнение расчета на шаге 4 требует $O(|V|)$ вычислительных операций. Отсюда:

Утверждение 2.4. Алгоритм 2.4 поиска потока минимальной стоимости древовидной сети $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$ требует $O(|V|^2)$ вычислительных операций.

2.3. Поток в несовместной транспортной сети

Рассматривается проблема исследования несовместной сетевой потоковой модели, представляющей собой систему линейных неравенств транспортного типа. Вопросы исследования несовместных систем линейных неравенств обсуждаются в [59, 109, 117, 155, 159]. Ряд работ посвящен исследованию несовместных потоковых моделей. Так, в [170] приводится обзор результатов исследования задачи поиска "наименее" несовместных потоков для различных мер несовместности. Проблема поиска потока в сети сводится к поиску максимального потока, величина которого определяется величиной минимального разреза и служит критерием несовместности исходной сети. В [113] рассматривается задача поиска разреза минимальной мощности, являющегося причиной несовместности сети. В данном разделе рассматривается задача поиска потока (циркуляции) в несовместной сети, заключающаяся в поиске потока, удовлетворяющего балансным ограничениям и минимизирующего штрафы за разрешенные изменения/нарушения пропускных способностей дуг [2, 12, 30, 111].

Рассмотрим проблему поиска допустимой циркуляции $Z_{FC}(G; a_{ij} b_{ij}, (i,j) \in A)$, где $G = (V, A)$. Система ограничений проблемы представляет собой систему ограничений (2.5),(2.6). Здесь условия (2.5) описывают балансные ограничения, (2.6) – ограничения пропускных способностей дуг. Введем вспомогательные обозначения: u_{ij}^a, u_{ij}^b – параметры, определяющие дуги, для которых разрешены изменения пропускных способностей, так u_{ij}^a (u_{ij}^b) равно 1, если для дуги (i, j) разрешено изменение нижней (верхней) пропускной способности, и равно 0 в противном случае, $u_{ij}^a, u_{ij}^b \in \{0,1\}$; d_{ij}, e_{ij} – штрафы за изменение на единицу нижней и верхней пропускных способностей дуги (i, j) соответственно, $d_{ij}, e_{ij} > 0$, $(i, j) \in A$; p_{ij}, q_{ij} – неизвестные величины, на которые будут изменены нижняя и верхняя пропускные способности дуги (i, j), соответственно, $(i, j) \in A$.

Пусть система (2.5),(2.6) несовместна. Тогда задача поиска потока в несовместной сети заключается в нахождение значений x_{ij}, p_{ij}, q_{ij}, $(i, j) \in A$, являющихся решением следующей задачи линейного программирования:

$$\sum_{(i,j)\in A} x_{ij} - \sum_{(j,i)\in A} x_{ji} = 0, i \in V, \tag{2.15}$$

$$a_{ij} - u_{ij}^a p_{ij} \le x_{ij} \le b_{ij} + u_{ij}^b q_{ij}, (i, j) \in A, \tag{2.16}$$

$$x_{ij}, p_{ij}, q_{ij} \ge 0, (i, j) \in A, \tag{2.17}$$

$$f(p,q) = \sum_{(i,j)\in A} (d_{ij} p_{ij} + e_{ij} q_{ij}) \to \min . \tag{2.18}$$

Далее задачу (2.15)-(2.18) будем обозначать через $z_{FSIN}(G; a_{ij}, b_{ij}, u_{ij}^a, u_{ij}^b d_{ij}, e_{ij}, (i, j) \in A)$ (*FSIN* соответствует англоязычному названию *flow search problem in infeasible network*).

Алгоритм решения задачи $z_{FSIN}(G; a_{ij}, b_{ij}, u_{ij}^a, u_{ij}^b d_{ij}, e_{ij}, (i, j) \in A)$ основан на ее сводимости к вспомогательной задаче поиска циркуляции минимальной стоимости. При доказательстве сводимости приводится схема построения вспомогательной задачи и устанавливается связь между решениями рассматриваемых задач. Проведем доказательство для частного случая задачи поиска потока в несовместной сети, когда изменения нижней и верхней пропускной способности разрешены для всех дуг сети.

Будем предполагать, что $u_{ij}^a = u_{ij}^b = 1$, $(i,j) \in A$, соответствующую задачу будем обозначать через $z_{FSIN}(G; a_{ij}, b_{ij}, d_{ij}, e_{ij}, i \in V)$. Далее будет показано, как обобщить предлагаемый алгоритм на случай, когда часть параметров u_{ij}^a, u_{ij}^b, $(i,j) \in A$, могут принимать нулевые значения.

Лемма 2.2. Пусть $x_{ij}^*, p_{ij}^*, q_{ij}^*$, $(i,j) \in A$ – решение задачи $z_{FSIN}(G; a_{ij}, b_{ij}, d_{ij}, e_{ij}, i \in V)$. Тогда для каждой из дуг $(i,j) \in A$ выполняются следующие условия:

a) если $a_{ij} \le x_{ij}^* \le b_{ij}$, то $p_{ij}^* = 0$, $q_{ij}^* = 0$;

b) если $x_{ij}^* < a_{ij}$, то $p_{ij}^* = a_{ij} - x_{ij}^*$, $q_{ij}^* = 0$;

c) если $b_{ij} < x_{ij}^*$, то $p_{ij}^* = 0$, $q_{ij}^* = x_{ij}^* - b_{ij}$;

d) $p_{ij}^* \le a_{ij}$.

Доказательство. a) Доказательство от противного. Предположим, что выполняются условия леммы, но существует дуга $(i',j') \in A$, что $a_{i'j'} \le x_{i'j'}^* \le b_{i'j'}$ и $p_{i'j'}^* \ne 0$. Тогда из условия (2.17) следует, что $p_{i'j'}^* > 0$. Построим набор значений $x'_{ij} = x_{ij}^*$, $q'_{ij} = q_{ij}^*$, $(i,j) \in A$; $p'_{ij} = p_{ij}^*$, $(i,j) \in A \setminus \{(i',j')\}$, $p'_{i'j'} = 0$. Легко увидеть, что построенный набор значений удовлетворяет системе ограничений (2.15)-(2.17) и $f(p',q') < f(p^*,q^*)$, что противоречит условию леммы. Таким образом, предположение неверно. Случай, когда $a_{i'j'} \le x_{i'j'}^* \le b_{i'j'}$ и $q_{i'j'}^* \ne 0$, рассматривается по аналогии.

b) Доказательство от противного. Предположим, что выполняются условия леммы, но существует дуга $(i',j') \in A$, что $x_{i'j'}^* < a_{i'j'}$ и $p_{i'j'}^* \ne a_{i'j'} - x_{i'j'}^*$. Тогда из условия (2.16) следует, что $a_{i'j'} - x_{i'j'}^* < p_{i'j'}^*$. Построим набор значений $x'_{ij} = x_{ij}^*$, $q'_{ij} = q_{ij}^*$, $(i,j) \in A$; $p'_{ij} = p_{ij}^*$, $(i,j) \in A \setminus \{(i',j')\}$, $p'_{i'j'} = a_{i'j'} - x_{i'j'}^*$. Легко увидеть, что построенный набор значений удовлетворяет системе ограничений (2.15)-(2.17) и $f(p',q') < f(p^*,q^*)$, что противоречит условию леммы. Таким образом, предположение неверно. Случай, когда $x_{i'j'}^* < a_{i'j'}$ и $q_{i'j'}^* \ne 0$ рассматривается по аналогии с пунктом (a).

c) Доказательство аналогично доказательству пункта (b).

d) Доказательство от противного. Предположим, что выполняются условия леммы, но существует дуга $(i',j') \in A$, что $p_{i'j'}^* > a_{i'j'}$. Построим набор значений $x'_{ij} = x_{ij}^*$, $q'_{ij} = q_{ij}^*$,

$(i,j) \in A$; $p'_{ij} = p^*_{ij}$, $(i,j) \in A \setminus \{(i',j')\}$, $p'_{i'j'} = a_{i'j'}$. Легко увидеть, что построенный набор значений удовлетворяет системе ограничений (2.15)-(2.17) и $f(p',q') < f(p^*,q^*)$, что противоречит условию леммы. Таким образом, предположение неверно. *Лемма доказана.*

Введем для каждой из дуг $(i,j) \in A$ вспомогательные переменные y_{ij}, z_{ij}, w_{ij}, которые будем интерпретировать следующим образом. Для заданного ориентированного графа $G = (V, A)$ рассмотрим вспомогательный ориентированный мультиграф с множеством вершин V, в котором для каждой из дуг $(i,j) \in A$ заданного графа строятся (см. рис. 2.1):

– обратная дуга из вершины j в вершину i, поток вдоль которой обозначается через w_{ji};
– две мультидуги из вершины i в вершину j, поток вдоль которых обозначается через y_{ij} и z_{ij}, соответственно.

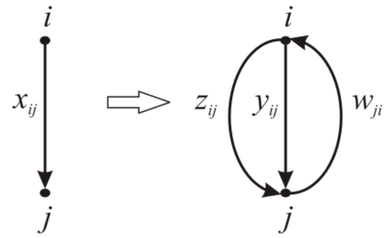

Рисунок 2.1

Построенный мультиграф будет определять сеть, в которой

– нижняя и верхняя пропускные способности дуги, связанной с переменной w_{ji}, составляют 0 и a_{ij} соответственно, стоимость дуги равна d_{ij};
– нижняя и верхняя пропускные способности дуги, связанной с переменной y_{ij}, составляют a_{ij} и b_{ij} соответственно, стоимость дуги равна 0;
– дуга, связанная с переменной z_{ij}, имеет нулевую нижнюю и неограниченную верхнюю пропускные способности, стоимость дуги равна e_{ij}.

Отметим, что в построенной сети, определяемой мультиграфом, величина входящего в вершину i, $i \in V$, потока составляет

$$\sum_{(j,i)\in A} (z_{ji} + y_{ji}) + \sum_{(i,j)\in A} w_{ji} ;$$

величина исходящего из вершины i потока составляет

$$\sum_{(i,j)\in A}(z_{ij}+y_{ij})+\sum_{(j,i)\in A}w_{ij}\ .$$

Рассмотрим вспомогательную задачу поиска циркуляции минимальной стоимости в построенной сети, определяемой мультиграфом:

$$\sum_{(i,j)\in A}(z_{ij}+y_{ij})+\sum_{(j,i)\in A}w_{ij}-\sum_{(j,i)\in A}(z_{ji}+y_{ji})-\sum_{(i,j)\in A}w_{ji}=0, i\in V, \tag{2.19}$$

$$a_{ij}\le y_{ij}\le b_{ij}, (i,j)\in A, \tag{2.20}$$

$$w_{ji}\le a_{ij}, \tag{2.21}$$

$$w_{ji}, y_{ij}, z_{ij}\ge 0, (i,j)\in A, \tag{2.22}$$

$$g(z,w)=\sum_{(i,j)\in A}(e_{ij}z_{ij}+d_{ij}w_{ji})\to \min\ . \tag{2.23}$$

Лемма 2.3. Пусть $w_{ji}^*, y_{ij}^*, z_{ij}^*$, $(i,j)\in A$ – решение задачи (2.19)-(2.23). Тогда $w_{ji}^*\cdot z_{ij}^*=0$, $(i,j)\in A$.

Доказательство. Доказательство от противного. Предположим, что выполняются условия леммы, но существует дуга $(i',j')\in A$, что $w_{j'i'}^*\ne 0$ и $z_{i'j'}^*\ne 0$. Тогда из условия (2.22) следует, что $w_{j'i'}^*>0$, $z_{i'j'}^*>0$. Рассмотрим цикл, образуемый парой дуг, связанных с переменными $w_{j'i'}^*$ и $z_{i'j'}^*$. Пусть $\Delta=\min(w_{j'i'}^*, z_{i'j'}^*)$, по предположению $\Delta>0$. Уменьшим поток вдоль рассматриваемого цикла на величину Δ, т.е. рассмотрим набор значений $y'_{ij}=y_{ij}^*$, $(i,j)\in A$; $z'_{ij}=z_{ij}^*$, $w'_{ji}=w_{ji}^*$ $(i,j)\in A\setminus\{(i',j')\}$; $w'_{j'i'}=w_{j'i'}^*-\Delta$, $z'_{i'j'}=z_{i'j'}^*-\Delta$. Легко увидеть, что построенный набор значений удовлетворяет системе ограничений (2.19)-(2.22) и $g(z',w')<g(z^*,w^*)$, что противоречит условию леммы. Таким образом, предположение неверно. *Лемма доказана.*

Покажем теперь, что задача $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},(i,j)\in A)$ поиска потока в несовместной сети сводится к вспомогательной задаче (2.19)-(2.23) поиска потока минимальной стоимости.

Лемма 2.4. Пусть $x_{ij}^*, p_{ij}^*, q_{ij}^*$, $(i,j)\in A$ – решение задачи $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},(i,j)\in A)$, тогда набор значений $w_{ji}=p_{ij}^*$, $y_{ij}=x_{ij}^*+p_{ij}^*-q_{ij}^*$, $z_{ij}=q_{ij}^*$, $(i,j)\in A$, удовлетворяет системе ограничений (2.19)-(2.22), и при этом $g(z,w)=f(p^*,q^*)$.

Доказательство. Пусть выполняются условия леммы. Тогда рассмотрим набор значений $w_{ji} = p_{ij}^*$, $y_{ij} = x_{ij}^* + p_{ij}^* - q_{ij}^*$, $z_{ij} = q_{ij}^*$, $(i, j) \in A$. Подставим данные значения в ограничения (2.19)-(2.22).

Рассмотрим ограничение (2.19):

$$\sum_{(i,j)\in A}(z_{ij} + y_{ij}) + \sum_{(j,i)\in A} w_{ij} - \sum_{(j,i)\in A}(z_{ji} + y_{ji}) - \sum_{(i,j)\in A} w_{ji} =$$

$$= \sum_{(i,j)\in A}(q_{ij}^* + (x_{ij}^* + p_{ij}^* - q_{ij}^*)) + \sum_{(j,i)\in A} p_{ji}^* - \sum_{(j,i)\in A}(q_{ji}^* + (x_{ji}^* + p_{ji}^* - q_{ji}^*)) - \sum_{(i,j)\in A} p_{ij}^* =$$

$$= \sum_{(i,j)\in A} x_{ij}^* + \sum_{(j,i)\in A} x_{ji}^* = 0,\ i \in V.$$

Таким образом, ограничение (2.19) выполняется.

Далее рассмотрим ограничение (2.20) для фиксированной дуги $(i, j) \in A$:

– если $a_{ij} \le x_{ij}^* \le b_{ij}$, то из пункта (a) леммы 2.2 следует, что $p_{ij}^* = 0$, $q_{ij}^* = 0$, тогда $y_{ij} = x_{ij}^*$, и ограничение (2.20) выполняется;

– если $x_{ij}^* < a_{ij}$, то из пункта (b) леммы 2.2 следует, что $p_{ij}^* = a_{ij} - x_{ij}^*$, $q_{ij}^* = 0$, тогда $y_{ij} = a_{ij}$, и ограничение (2.20) выполняется;

– если $x_{ij}^* > b_{ij}$, то из пункта (c) леммы 2.2 следует, что $p_{ij}^* = 0$, $q_{ij}^* = x_{ij}^* - b_{ij}$, тогда $y_{ij} = b_{ij}$, и ограничение (2.20) выполняется.

Выполнимость ограничения (2.21) автоматически следует из пункта (d) леммы (c).

Неотрицательность величин w_{ji}, z_{ij}, $(i, j) \in A$, непосредственно следует из условия (2.17). Величины y_{ij}, $(i, j) \in A$, являются неотрицательными, так как для них выполняются условия (2.20). Отсюда ограничение (2.22) выполняется.

Наконец рассмотрим значение целевой функции (2.23):

$$g(z, w) = \sum_{(i,j)\in A}(e_{ij} z_{ij} + d_{ij} w_{ji}) = \sum_{(i,j)\in A}(e_{ij} q_{ij}^* + d_{ij} p_{ij}^*) = f(p^*, q^*).$$

Лемма доказана.

Лемма 2.5. Пусть w_{ji}^*, y_{ij}^*, z_{ij}^*, $(i,j)\in A$ – решение задачи (2.19)-(2.23), тогда набор значений $x_{ij}=y_{ij}^*+z_{ij}^*-w_{ji}^*$, $p_{ij}=w_{ji}^*$, $q_{ij}=z_{ij}^*$, $(i,j)\in A$, удовлетворяет системе ограничений (2.15)-(2.17), и при этом $f(p,q)=g(z^*,w^*)$.

Доказательство. Пусть выполняются условия леммы. Тогда рассмотрим набор значений $x_{ij}=y_{ij}^*+z_{ij}^*-w_{ji}^*$, $p_{ij}=w_{ji}^*$, $q_{ij}=z_{ij}^*$, $(i,j)\in A$. Подставим данные значения в ограничения (2.15)-(2.17).

Рассмотрим ограничение (2.15):

$$\sum_{(i,j)\in A}x_{ij}-\sum_{(j,i)\in A}x_{ji}=\sum_{(i,j)\in A}(y_{ij}^*+z_{ij}^*-w_{ji}^*)-\sum_{(j,i)\in A}(y_{ji}^*+z_{ji}^*-w_{ij}^*)=$$

$$=\sum_{(i,j)\in A}(y_{ij}^*+z_{ij}^*)+\sum_{(j,i)\in A}w_{ij}^*-\sum_{(j,i)\in A}(y_{ji}^*+z_{ji}^*)-\sum_{(i,j)\in A}w_{ji}^*=0,\ i\in V.$$

Таким образом, ограничение (2.15) выполняется.

Далее рассмотрим ограничение (2.16) для дуги $(i,j)\in A$. Из леммы 2.3 следует, что выполняется хотя бы одно из двух условий: $z_{ij}^*=0$ или $w_{ji}^*=0$. Из ограничения (2.20) следует, что

– если $z_{ij}^*=0$, то $a_{ij}-p_{ij}=a_{ij}-w_{ji}^*\le y_{ij}^*-w_{ji}^*=x_{ij}\le b_{ij}-w_{ji}^*\le b_{ij}=b_{ij}+q_{ij}$, таким образом, ограничение (2.16) выполняется;

– если $w_{ji}^*=0$, то $a_{ij}-p_{ij}=a_{ij}\le a_{ij}+z_{ij}^*\le y_{ij}^*+z_{ij}^*=x_{ij}^*\le b_{ij}+z_{ij}^*=b_{ij}+q_{ij}$, таким образом, ограничение (2.16) выполняется.

Рассмотрим ограничение (2.17) для дуги $(i,j)\in A$. Воспользуемся здесь также леммой 2.3. Тогда

– если $z_{ij}^*=0$, то $x_{ij}=y_{ij}^*-w_{ji}^*$, и из условий (2.20), (2.21) следует, что $x_{ij}\ge 0$;

– если $w_{ji}^*=0$, то $x_{ij}=y_{ij}^*+z_{ij}^*$, и из условия (2.22) следует, что $x_{ij}\ge 0$.

Неотрицательность величин p_{ij}, q_{ij}, $(i,j)\in A$, непосредственно следует из условия (2.22). Таким образом, ограничение (2.17) выполняется.

Наконец рассмотрим значение целевой функции (2.18):

$$f(p,q)=\sum_{(i,j)\in A}(d_{ij}p_{ij}+e_{ij}q_{ij})=\sum_{(i,j)\in A}(d_{ij}w_{ji}^*+e_{ij}z_{ij}^*)=g(z^*,w^*).$$

Лемма доказана.

Теорема 2.3. Пусть w^*_{ji}, y^*_{ij}, z^*_{ij}, $(i,j)\in A$ – решение задачи (2.19)-(2.23), тогда набор значений $x_{ij} = y^*_{ij} + z^*_{ij} - w^*_{ji}$, $p_{ij} = w^*_{ji}$, $q_{ij} = z^*_{ij}$, $(i,j)\in A$, является решением задачи $Z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},i\in V)$.

Доказательство. Доказательство от противного. Пусть выполняются условия теоремы, но набор значений $x_{ij} = y^*_{ij} + z^*_{ij} - w^*_{ji}$, $p_{ij} = w^*_{ji}$, $q_{ij} = z^*_{ij}$, $(i,j)\in A$, не является оптимальным решением задачи $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},i\in V)$. Из леммы 2.5 следует, что построенный набор значений является решением системы (2.15)-(2.17) и $f(p,q) = g(z^*,w^*)$. Пусть x^*_{ij}, p^*_{ij}, q^*_{ij}, $(i,j)\in A$ – оптимальное решение задачи $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},i\in V)$, следовательно, $f(p^*,q^*) < f(p,q)$. Тогда из леммы 2.3 следует, что можно построить набор значений w'_{ji}, y'_{ij}, z'_{ij}, $(i,j)\in A$, удовлетворяющий системе (2.19)-(2.22), и при этом $g(z',w') = f(p^*,q^*)$. Тогда $g(z',w') = f(p^*,q^*) < f(p,q) = g(z^*,w^*)$, что противоречит условию леммы. Предположение неверно. *Теорема доказана.*

Теорема 2.3 позволяет предложить алгоритм решения задачи поиска потока в несовместной сети, основанный на решении вспомогательной задачи поиска циркуляции минимальной стоимости (2.19)-(2.23).

Алгоритм 2.5. Поиск потока в несовместной сети.

Вход. Задача $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},(i,j)\in A)$.

Шаг 1. Построить вспомогательную задачу (2.19)-(2.23).

Шаг 2. Найти циркуляцию минимальной стоимости w^*_{ji}, y^*_{ij}, z^*_{ij}, $(i,j)\in A$, задачи (2.19)-(2.23), используя алгоритм 2.2.

Шаг 3. Построить решение задачи $z_{FSIN}(G;a_{ij},b_{ij},d_{ij},e_{ij},i\in V)$ по следующему правилу $x_{ij} = y^*_{ij} + z^*_{ij} - w^*_{ji}$, $p_{ij} = w^*_{ji}$, $q_{ij} = z^*_{ij}$, $(i,j)\in A$, алгоритм завершен.

Транспортная сеть, связанная с задачей (2.19)-(2.23), содержит мультидуги. Для большинства потоковых алгоритмов предполагается отсутствие мультидуг сети. Преобразуем рассматриваемую транспортную сеть: для всех мультидуг произведем операцию подразбиения – заменим каждую из соответствующих дуг на две последовательных дуги. Соответствующие преобразования необходимо выполнить перед выполнением шага 2 алгоритма 2.5. Таким образом, при решении задачи

$z_{FSIN}(G; a_{ij}, b_{ij}, d_{ij}, e_{ij}, i \in V)$, где $G = (V, A)$, мы приходим к задаче поиска циркуляции минимальной стоимости в сети, содержащей $O(|V| + |A|)$ вершин и $O(|A|)$ дуг. Тогда, согласно утверждению 2.2:

Утверждение 2.5. Алгоритма 2.5 решения задачи поиска потока в несовместной сети $z_{FSIN}(G; a_{ij}, b_{ij}, d_{ij}, e_{ij}, i \in V)$ требует $\alpha(O(|V| + |A|), O(|V| + |A|))$ + $\beta(O(|V| + |A|), O(|V| + |A|))$ вычислительных операций.

В общем случае при решении задачи $z_{FSIN}(G; a_{ij}, b_{ij}, u_{ij}^a, u_{ij}^b d_{ij}, e_{ij}, i \in V)$ часть параметров u_{ij}^a, u_{ij}^b, $(i, j) \in A$, может принимать нулевые значения. Тогда необходимо модифицировать схему построения вспомогательной сети, определяемой мультиграфом. Для каждой из дуг $(i, j) \in A$ исходного графа G введем дуги мультиграфа, связанные с переменными

– y_{ij}, z_{ij} и w_{ji}, если $u_{ij}^a = 1$, $u_{ij}^b = 1$;

– y_{ij} и w_{ji}, если $u_{ij}^a = 1$, $u_{ij}^b = 0$;

– y_{ij} и z_{ij}, если $u_{ij}^a = 0$, $u_{ij}^b = 1$;

– y_{ij}, если $u_{ij}^a = 0$, $u_{ij}^b = 0$.

Модифицируем соответствующим образом схему построения вспомогательной задачи на шаге 1 алгоритма 2.5. Тогда алгоритм 2.5 может быть применен при решении общей задачи $z_{FSIN}(G; a_{ij}, b_{ij}, u_{ij}^a, u_{ij}^b d_{ij}, e_{ij}, (i, j) \in A)$ поиска потока в несовместной сети.

2.4. Циклическая декомпозиция потока

Рассмотрим задачу поиска циркуляции минимальной стоимости (2.5),(2.6),(2.4). Далее циркуляцией графа G будем называть набор неотрицательных значений x_{ij}, $(i, j) \in A$, удовлетворяющих системе ограничений (2.5). Если дополнительно $x_{ij} \in Z$, $(i, j) \in A$, то циркуляцию будем называть целочисленной. Допустимой циркуляцией (как уже было введено ранее) будем называть набор значений x_{ij}, $(i, j) \in A$, удовлетворяющих системе ограничений (2.5),(2.6). Циркуляцией минимальной стоимости будем называть набор значений x_{ij}, $(i, j) \in A$, являющийся решением задачи (2.5),(2.6),(2.4). Далее в данном разделе исследуется проблема декомпозиции циркуляции на потоки вдоль простых циклов сети [22, 40].

Далее для удобства простой цикл графа G будем обозначать как $C=(i_1,...,i_{k+1})$, где $(i_j,i_{j+1})\in A$, $j=\overline{1,k}$; $i_1=i_{k+1}$; $i_{j'}\neq i_{j''}$ при $j'\neq j''$, $j',j''=\overline{1,k}$. Для определенности будем считать, что $i_1<i_j$, $j=\overline{2,k}$ (здесь будем также предполагать, что $V=\{1,...,|V|\}$). Пусть $(u,v)\in A$, $C=(i_1,...,i_{k+1})$. Тогда для удобства будем обозначать $(u,v)\in C$, если цикл C проходит через дугу (u,v), т.е. найдется $j\in\{1,2,...,k\}$, что $u=i_j$, $v=i_{j+1}$; в противном случае будем обозначать $(u,v)\notin C$. Легко увидеть, что число простых циклов в графе ограничено сверху, например, величиной $\sum_{i=1}^{|V|}C_{|V|}^{i}(i-1)!$ и, таким образом, множество простых циклов в графе является конченым. Множество всех простых циклов графа G будем обозначать через $C(G)$.

Определение 2.9. Пусть x_{ij}, $(i,j)\in A$ – циркуляция графа G. Циклической декомпозицией циркуляции будем называть такой набор неотрицательных значений y_C, $C\in C(G)$, что $\sum_{C\in C(G)|(i,j)\in C}y_C=x_{ij}$, $(i,j)\in A$. Если дополнительно $y_C\in Z$, $C\in C(G)$, то циклическую декомпозицию будем называть целочисленной.

Лемма 2.6. Пусть x_{ij}, $(i,j)\in A$, – ненулевая циркуляция. Тогда существует простой цикл $C\in C(G)$, что $\min_{(u,v)\in C}x_{uv}>0$.

Доказательство. Докажем лемму конструктивно. Пусть выполняются условия леммы, тогда построим алгоритм нахождения требуемого простого цикла C (далее **Алгоритм 2.6**):

Шаг 1. Выберем произвольную дугу $(u,v)\in A$, что $x_{uv}>0$. Пусть $i_1=u$, $i_2=v$, $k=2$. Переход на Шаг 2.

Шаг 2. Рассмотрим множество $Q=\{i\,|\,(i_k,i)\in A;\, x_{i_k i}>0\}$. Если существует j, $j\in\{1,2,....k\}$, что $i_j\in Q$, то $i_{k+1}:=i_j$, $C:=(i_j,...,i_{k+1})$, алгоритм завершен. Иначе выберем произвольный $i\in Q$, далее $i_{k+1}:=i$, $k:=k+1$ и переход на Шаг 2.

Покажем корректность построенного алгоритма. На шаге 1 дугу (u,v) можно выбрать, так как циркуляция ненулевая. Далее необходимо доказать, что на каждом из шагов 2 множество $Q\neq\varnothing$. По построению для каждого из шагов 2 выполняется условие $x_{i_{k-1}i_k}>0$. Тогда из (2.5) следует, что $\sum_{(i_k,j)\in A}x_{i_k j}>0$, отсюда $Q\neq\varnothing$. Конечность работы алгоритма следует из конечности множества вершин графа. По построению, если алгоритм завершил свою работу, то C будет являться простым циклом и $\min_{(u,v)\in C} x_{uv} > 0$. *Лемма доказана.*

Теорема 2.4. Для произвольной циркуляции существует ее циклическая декомпозиция.

Доказательство. Докажем теорему конструктивно. Пусть выполняются условия теоремы, тогда построим алгоритм нахождения циклической декомпозиции y_C, $C \in C(G)$, для заданной циркуляции x_{ij}, $(i, j) \in A$.

Изначально пусть $y_C = 0$, $C \in C(G)$. Введем вспомогательный набор значений $z_{ij} = x_{ij}$, $(i, j) \in A$. Рассмотрим следующий алгоритм (далее **Алгоритм 2.7**):

Шаг 1. Если z_{ij}, $(i, j) \in A$ – нулевая циркуляция, то алгоритм завершен. Иначе, переход на шаг 2.

Шаг 2. При помощи алгоритма 1 выберем простой цикл $C \in C(G)$, что $q := \min_{(u,v)\in C} z_{uv} > 0$. Переход на шаг 3.

Шаг 3. Далее $z_{ij} := z_{ij} - q$, $(i, j) \in C$; $y_C := q$. Переход на шаг 1.

Покажем корректность построенного алгоритма. Покажем, что к концу каждого из шагов 3 набор значений z_{ij}, $(i, j) \in A$, является циркуляцией. Действительно, неотрицательность величин z_{ij}, $(i, j) \in A$, непосредственно следует из выбранного значения q; выполнение условий (2.1) следует из того, что C является циклом.

По построению для каждого из шагов 3 найдется $(u,v) \in C$, что к началу шага 3 $z_{uv} > 0$, и $z_{uv} = 0$ к концу шага 3. Таким образом, величина $|(i, j) | (i, j) \in A, z_{ij} = 0|$ возрастает минимум на единицу к концу каждого из шагов 3. Отсюда следует конечность построенного алгоритма. Кроме того, каждый из циклов $C \in C(G)$ может встретиться не более одного раза на шаге 2. Далее по построению, если алгоритм завершил свою работу, то найденный набор y_C, $C \in C(G)$, является циклической декомпозицией исходной циркуляции. *Теорема доказана.*

При работе алгоритма 2.7 можно потребовать удаление всех дуг с нулевым потоком. Соответствующие действия необходимо выполнить перед началом работы алгоритма 2.7 и на каждом из его шагов 3. Тогда алгоритм 2.6, используемый на шаге 2 алгоритма 2.7, будет требовать $O(|V|)$ вычислительных операций. Действительно, при своевременном удалении дуг с нулевым потоком шаги 1 и 2 алгоритма 2.6 будут требовать $O(1)$ вычислительных операций; а общее число циклов шаг 1, шаг 2 алгоритма 2.6 не превышает $|V|$. Каждый раз после выполнения шага 3 алгоритма 2.7 поток вдоль как минимум одной из дуг становится нулевым. Отсюда общее число циклов шаг 1, шаг 2, шаг 3 алгоритма 2.7 не превышает $|A|$. Таким образом, можно сформулировать следующее утверждение.

Утверждение 2.6. Алгоритм 2.7 построения циклической декомпозиции циркуляции требует $O(|V||A|)$ вычислительных операций.

Согласно схеме работы алгоритма 2.7, в случае целочисленности исходной циркуляции ее циклическая декомпозиция, построенная алгоритмом 2.7, также будет являться целочисленной. Действительно, величина q, определяемая на шаге 2 алгоритма 2.7, будет целочисленной и, следовательно, величины z_{ij}, $(i,j) \in C$ и y_C на шаге 3 также будут целочисленными. Отсюда справедливо следующее следствие.

Следствие 2.3. Для произвольной целочисленной циркуляции ее циклическая декомпозиция, построенная алгоритмом 2.7, также целочисленна.

Заметим, что полученные результаты о циклической декомпозиции справедливы для произвольной циркуляции. В частности результаты справедливы, когда речь идет о допустимой циркуляции или циркуляции минимальной стоимости. Как известно, матрица системы ограничений (2.5),(2.6) абсолютно унимодулярна [51]. Отсюда в случае целочисленных исходных параметров и совместной системы (2.5),(2.6), существует целочисленная допустимая циркуляция, а тем самым и ее целочисленная циклическая декомпозиция.

2.5. Метод ортогональных проекций при решении транспортных систем

Одним из методов решения систем линейных неравенств является итерационные метод ортогональных проекций Агмона-Моцкина [114, 171]. В данном разделе описывается естественная модификация данного алгоритма при решении систем линейных неравенств транспортного типа, предложенная в [83]. Метод ортогональных проекций Агмона-Моцкина решения систем линейных неравенств заключается в следующем. Пусть необходимо найти решение системы

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i, i = \overline{1,m}. \qquad (2.24)$$

Здесь x_j, $j = \overline{1,n}$, – неизвестные; b_i, a_{ij}, $i = \overline{1,m}$, $j = \overline{1,n}$, – заданные параметры. Обозначим $L_i(\vec{x}) = \sum_{j=1}^{n} a_{ij} x_j + b_i, i = \overline{1,m}$. Выберем произвольный вектор $\vec{x}^{(0)} \in R^n$. Далее предположим, что известен вектор $\vec{x}^{(\nu)}$, тогда обозначим через I множество номеров линейных неравенств рассматриваемой системы, которые не выполняются при выбранном векторе $\vec{x}^{(\nu)}$:

$$I = \{ i \mid L_i(\vec{x}^{(\nu)}) < 0, i = \overline{1,m} \}.$$

Если $I = \varnothing$, то задача решена, иначе будем вычислять значение индекса i_0 по следующей формуле:

$$i_0 = \arg\max_{i \in I} \left(-\frac{L_i(\vec{x}^{(\nu)})}{\sum_{j=1}^{n} a_{ij}^2} \right).$$

Тогда очередной вектор итерационного алгоритма определяется следующим образом:

$$\vec{x}^{(\nu+1)} = \vec{x}^{(\nu)} + h\vec{a}_{i_0},$$

где $h = -\frac{L_{i_0}(\vec{x}^{(\nu)})}{\sum_{j=1}^{n} a_{i_0 j}^2}$, а через вектор $\vec{a}_i$ обозначается вектор с компонентами $(a_{i1},...,a_{in})$. В [114] было показано, что если система (2.24) совместна, то описанный итерационный процесс сходится к решению системы.

При исследовании двусторонних систем линейных неравенств транспортного типа в [83] был предложен модифицированный метод ортогональных проекций Агмона-Моцкина, учитывающий данную специфику. Рассмотрим систему линейных двусторонних неравенств транспортного типа

$$b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq c_i, i = \overline{1,m}, \qquad (2.25)$$

где $a_{ij} \in \{-1,0,1\}$, $i = \overline{1,m}$, $j = \overline{1,n}$. Введем обозначения $N_i^+ = \{ j \mid a_{ij} = 1, j = \overline{1,n} \}$, $N_i^- = \{ j \mid a_{ij} = -1, j = \overline{1,n} \}$, $i = \overline{1,m}$. Здесь подмножества N_i^+ и N_i^- определяют компоненты вектора неизвестных, которые входят в уравнение i с коэффициентами 1 и –1 соответственно. Тогда для удобства запишем систему линейных неравенств (2.25) в следующем виде

$$b_i \le \sum_{j \in N_i^+} x_j - \sum_{j \in N_i^-} x_j \le c_i,\ i = \overline{1,m}.$$

Выберем произвольный вектор $\vec{x}^{(0)} \in R^n$. Далее предположим, что известен вектор $\vec{x}^{(\nu)}$, тогда обозначим через s номер первого по порядку ограничения, условия которого нарушены. Если такое ограничение не может быть найдено, то система неравенств решена. Иначе вектор $\vec{x}^{(\nu+1)}$ определяется следующим образом:

$$x_j^{\nu+1} = \begin{cases} x_j^{\nu} + (b_s - \sum_{l \in N_s^+} x_l + \sum_{l \in N_s^-} x_l) \times (|N_s^+| + |N_s^-|)^{-1}, \text{если } b_s > \sum_{l \in N_s^+} x_l - \sum_{l \in N_s^-} x_l, \\ x_j^{\nu} - (\sum_{l \in N_s^+} x_l - \sum_{l \in N_s^-} x_l - c_s) \times (|N_s^+| + |N_s^-|)^{-1}, \text{если } \sum_{l \in N_s^+} x_l - \sum_{l \in N_s^-} x_l > c_s, \end{cases} \quad j = \overline{1,n}.$$

Недостатком метода ортогональных проекций Агмона-Моцкина является то, что для его корректной работы необходимо уметь распознавать совместность системы (2.24). В данном случае возможно введение управляющих параметров: ε – точности и H – количества шагов. Тогда для решения системы (2.24) можно воспользоваться следующей эвристикой: если за H шагом мы с ε-точностью не получили решение системы, то считаем, что система несовместна.

2.6. Многокритериальные транспортные задачи

При решении прикладных задач распределении ресурсов в сложных системах возникает задача определения плана работы системы по различным показателям: группам оборудования, периодам планирования, этапам изготовления, потребляемым ресурсам, видам продукции и т.д [83]. Показатели делятся на жесткие, требующие обязательного выполнения, и желательные, к достижению которых нужно стремиться. Жесткие показатели формализуются в виде ограничений, а желательные – в виде критериев оптимальности. Тогда задача ставится как многокритериальная задача учета желательных показателей с ограничениями в виде учета жестких показателей. Ограничения учета жестких показателей формализуются в виде системы линейных двусторонних неравенств транспортного типа (2.25). Далее приведем описание применяемых критериев оптимальности. Как показывает опыт решения прикладных транспортных задач [84, 91], формализация критериев в виде непрерывных функция для пользователя часто оказывается затруднительной. При решении прикладных задач пользователю предпочтительней оценить заданные показатели, указав границы для величин отклонения, в которых эти величины являются «отличными», «очень хорошими», «хорошими», «удовлетворительными» и др. Тогда формализуемые критерии могут быть представлены в виде кусочно-постоянных функций, разбивающих множество величин отклонения по каждому критерию на области «качества» отклонений. В данном разделе приводится постановка многокритериальной задачи транспортного типа с кусочно-постоянными критериями оптимальности и описывается метод ее решения, основанный на подходе, предложенном в [83].

Пусть определены k показателей, формализуемых в виде линейных функций $\sum_{j=1}^{n} d_{ij} x_j$, $i=\overline{1,k}$; где $d_{ij} \in \{-1,0,1\}$, $i=\overline{1,k}$, $j=\overline{1,n}$. Для каждого из показателей задана совокупность из $p+1$ вложенных друг в друга сегментов $S_i^{(l)}$, $l=\overline{0,p}$, что $S_i^{(l)} \subseteq S_i^{(l+1)}$, $l=\overline{0,p-1}$, $S_i^{(p)}=[-\infty,+\infty]$, $i=\overline{1,k}$. С каждым из таких показателей будем связывать функцию предпочтения $\chi_i(w)$, $i=\overline{1,k}$, которая определена следующим образом:

$$\chi_i(w)=\begin{cases} r, \text{если } w \in S_i^{(r)} \text{ и } w \notin S_i^{(r-1)}, \text{где } r \in \{1,2,\ldots,p\}, \\ 0, \text{если } w \in S_i^{(0)}. \end{cases}$$

Рассмотрим задачу поиска допустимого плана x_j, $j=\overline{1,n}$, удовлетворяющего системе ограничений (2.25) и минимизирующего функции предпочтения для выбранных показателей:

$$\chi_i\Big(\sum_{j=1}^{n} d_{ij} x_j\Big) \to \min, \; i=\overline{1,k}. \tag{2.26}$$

Поставленная задача (2.25), (2.26) является задачей многокритериальной оптимизации, при решении которой будем применять следующие схемы компромисса. Предполагая, что выбранные показатели упорядочены с точки зрения их значимости при определении эффективности функционирования иерархических системы, в качестве схемы компромисса будем рассматривать лексикографическое упорядочивание критериев оптимальности. В случае равнозначности показателей будем рассматривать максиминную свертку критериев.

Формализация схемы компромисса рассматриваемой многокритериальной задачи требует введения некоторых вспомогательных величин. Пусть $a,b \in N$. Введем множество $V_{a,b}=\{(v_1,v_2,\ldots,v_b) \mid v_i=\overline{0,a}, i=\overline{1,b}\}$. Тогда через $V_{a,b}$ будем обозначать множество вершин $a+1$-ичного b-мерного куба. Далее введем $p+1$-ичный k-мерный куб, на котором определим порядок Π. Каждой вершине куба $\vec{v} \in V_{p,k}$ поставим в соответствие систему $\Omega(\vec{v})$. Система $\Omega(\vec{v})$ содержит не зависящие от вершины куба ограничения (2.25) и ограничения, зависящие от вершины $\vec{v}$ следующим образом:

$$\sum_{j=1}^{n} d_{ij} x_j \in S_i^{(v_i)}, \ i = \overline{1,k}. \quad (2.27)$$

Таким образом, $\Omega(\vec{v})$ представляет собой систему ограничений (2.25), (2.27). На множестве вершин куба зададим двузначную функцию $g(\vec{v})$, принимающую значение 1, если система $\Omega(\vec{v})$ совместна, и 0 в противном случае, $\vec{v} \in V_{p,k}$. Через $V = \{\vec{v} \mid \vec{v} \in V_{p,k}, g(\vec{v}) = 1\}$ обозначим множество вершин $p+1$-ичного k-мерного куба, на которых функция g принимает значение, равное 1. Не уменьшая общности, будем считать, что показатели плана упорядочены исходя из их приоритетов. В качестве порядка Π рассмотрим лексикографическое отношение порядка: $\vec{v}^1 \Pi \vec{v}^2$ тогда и только тогда, когда для некоторого $l \in \{1,2,...,k\}$ выполняется условие $v_l^1 < v_l^2$ и одновременно $v_r^1 = v_r^2$, $r = \overline{1, l-1}$.

Тогда будем рассматривать следующую задачу поиска оптимальной вершины куба $\vec{v}^0$:

$$\vec{v}^0 \in V, \quad (2.28)$$

$$\vec{v}^0 \Pi \vec{v}, \ \vec{v} \in V. \quad (2.29)$$

Оптимальная вершина $\vec{v}^0$, являющаяся решением задачи (2.28), (2.29), определяет оптимальное решение многокритериальной задачи (2.25), (2.26) при лексикографической схеме компромисса.

Отметим, что в силу вложенности сегментов $S_i^{(l)}$, $l = \overline{0,p}$, $i = \overline{1,k}$, введенная функция g обладает следующим свойством монотонности.

Утверждение 2.7. Если $\vec{v} \le \vec{u}$, $\vec{v}, \vec{u} \in V_{p,k}$, то $g(\vec{v}) \le g(\vec{u})$.

Монотонность функции g позволяет предложить алгоритм поиска оптимальной вершины куба, основанный на последовательном вычислении компонент оптимальной вершины $\vec{v}^0$. Каждая компонента вычисляется при помощи бинарного поиска, на каждом шаге которого вычисляется значение функции g.

Алгоритм 2.8. Поиск оптимальной вершины куба (лексикографическая свертка).

Вход. Задача (2.28), (2.29).

Шаг 1. Вычислить $g(p, p,..., p)$, если $g(p, p,..., p) = 1$, то пусть $t := 1$ и переход на шаг 2; иначе задача (2.28), (2.29) несовместна, и алгоритм завершен.

Шаг 2. Пусть $v_1^0, v_2^0,..., v_{t-1}^0$ найденные первые $t-1$ компоненты вектора $\vec{v}^0$. Определим t - ую компоненту как решение следующей оптимизационной задачи:

$$y \in \{0,1,..., p\}, \tag{2.30}$$

$$g(v_1^0, v_2^0,..., v_{t-1}^0, y, p, p,..., p) = 1, \tag{2.31}$$

$$y \to \min. \tag{2.32}$$

Оптимальное решение y^0 задачи (2.30)-(2.32) находится при помощи бинарного поиска на множестве $\{0,1,..., p\}$, на каждой итерации которого вычисляется соответствующее значение $g(v_1^0, v_2^0,..., v_{t-1}^0, y, p, p,..., p)$. Пусть $v_t^0 = y^0$, переход на шаг 3.

Шаг 3. Если $t < k$, то $t := t+1$, переход на шаг 2; иначе алгоритм завершен.

Работа алгоритма 2.8 связана с вычислением k компонент вершины $\vec{v}^0$. Каждая из компонент находится при помощи бинарного поиска на множестве мощности $p+1$. На каждой итерации бинарного поиска вычисляется значение функции g, а следовательно проверяется на совместность система вида $\Omega(\vec{v})$. Отсюда:

Утверждение 2.8. Алгоритм 2.8 требует $O(k \log p)$ проверок совместности систем ограничений вида $\Omega(\vec{v})$.

Дополнительно при решении многокритериальной задачи (2.25), (2.26) будем рассматривать максиминную свертку критериев оптимальности. Возникающая при этом оптимизационная задача ставится как задача (2.25), (2.33).

$$\max_{i=\overline{1,k}} \chi_i (\sum_{j=1}^{n} d_{ij} x_j) \to \min. \tag{2.33}$$

Данная задача может быть поставлена как задача (2.28), (2.34) поиска оптимальной вершины куба $\vec{v}^0$.

$$\max_{i=\overline{1,k}} v_i^0 \le \max_{i=\overline{1,k}} v_i, \ \vec{v} \in V. \tag{2.34}$$

Монотонность функции g позволяет предложить алгоритм поиска оптимальной вершины куба, основанный на вычислении минимально возможного общего значения компонент вершины $\vec{v}^0$. Общее значение компонент вычисляется при помощи бинарного поиска, на каждом шаге которого вычисляется значение функции g.

Алгоритм 2.9. Поиск оптимальной вершины куба (максиминная свертка).

Вход. Задача (2.28), (2.34).

Шаг 1. Вычислить $g(p, p,..., p)$, если $g(p, p,..., p) = 1$, то переход на шаг 2; иначе задача (2.28), (2.34) несовместна, и алгоритм завершен.

Шаг 2. Определим оптимальное значение компонент вершины как решение следующей оптимизационной задачи.

$$y \in \{0,1,..., p\}, \tag{2.35}$$

$$g(y, y,..., y) = 1, \tag{2.36}$$

$$y \to \min . \tag{2.37}$$

Оптимальное решение y^0 задачи (2.35)–(2.37) находится при помощи бинарного поиска на множестве $\{0,1,..., p\}$, на каждой итерации которого вычисляется соответствующее значение $g(y, y,..., y)$. Пусть $v_i^0 = y^0$, $i = \overline{1,k}$. Алгоритм завершен.

Работа алгоритма 2.9 связана с вычислением общего значения компонент вершины $\vec{v}^0$, определяемого при помощи бинарного поиска на множестве мощности $p+1$. На каждой итерация бинарного поиска вычисляется значение функции g, а следовательно, проверяется на совместность система вида $\Omega(\vec{v})$. Отсюда:

Утверждение 2.9. Алгоритм 2.9 требует $O(\log p)$ проверок совместности системы ограничений вида $\Omega(\vec{v})$.

Для проверки совместности системы $\Omega(\vec{v})$ можно воспользоваться общими методами решения систем линейных неравенств [109]. Т.к. $\Omega(\vec{v})$ представляет собой систему линейных неравенств транспортного типа, то для ее решения можно также воспользоваться модифицированным методом ортогональных проекций, описанным в пункте 2.5. При решении ряда многоиндексных транспортных задач возникающая система ограничений $\Omega(\vec{v})$ может быть решена потоковыми алгоритмами, соответствующие классы задач будут рассмотрены далее в работе при исследовании вопросов сводимости.

Выводы

Рассмотрены методы решения ряда задач транспортного типа, построены оценки вычислительной сложности. Пусть $\alpha(n,m)$ и $\beta(n,m)$ – количество вычислительных операций, требуемых для решения задачи поиска максимального потока и потока минимальной стоимости заданной величины соответственно, где n и m – количество вершин и дуг сети. Описана и обоснована процедура поиска потока в сети с двусторонними пропускными способностями:

– поиск допустимого потока требует $\alpha(O(|V|), O(|V|+|A|))$ вычислительных операций,

– поиск потока минимальной стоимости требует $\alpha(O(|V|), O(|V|+|A|)) + \beta(O(|V|), O(|V|+|A|))$ вычислительных операций.

Описана и обоснована процедура поиска потока в древовидной сети с двусторонними пропускными способностями:

– поиск допустимого потока требует $O(|V|)$ вычислительных операций,

– поиск потока минимальной стоимости требует $O(|V|^2)$ вычислительных операций.

Разработан и обоснован алгоритм поиска потока в несовместной сети, требующий $\alpha(O(|V|+|A|), O(|V|+|A|)) + \beta(O(|V|+|A|), O(|V|+|A|))$ вычислительных операций. Разработан и обоснован алгоритм построения циклической декомпозиции потока, требующий $O(|V|\cdot|A|)$ вычислительных операций. Описана модификация метода ортогональных проекций решения систем линейных неравенств транспортного типа. Описан и обоснован метод решения k-критериальных задач транспортного типа с кусочно-постоянными (p+1)-значными критериями:

– при лексикографической свертки алгоритм требует $O(k \log p)$ проверок совместности систем транспортного типа,

– при максиминной свертки критериев алгоритм требует $O(\log p)$ проверок совместности систем транспортного типа.

Полученные результаты используются далее в работе при построении алгоритмов решения многоиндексных задач транспортного типа, основанных на сводимости к классу задач поиска потока в сети.

Глава 3. Многоиндексные задачи транспортного типа

Глава посвящена постановкам исследуемых многоиндексных задач транспортного типа. Постановки задач приведены с использованием введенной в [93] и далее развитой в [8, 9, 20] схемы формализации многоиндексных задач. Данная формализация используется в работе при описании результатов исследования многоиндексных задач. Описывается общая формализация схем сводимости, разработанная в [8, 9, 20], применяемая далее при классификации схем сведения многоиндексных задач к классу задач поиска потока в сети.

3.1. Постановки многоиндексных задач транспортного типа

При постановке многоиндексных задач транспортного типа воспользуемся следующей формализацией, основанной на обозначении, введенном в [93]. Пусть $s \in N$ и $N(s) = \{1,2,...,s\}$. Каждому числу l поставим в соответствие параметр j_l, называемый индексом, который принимает значения из множества $J_l = \{1,2,...,n_l\}$, где $n_l \geq 2$, $l \in N(s)$. Там где это не вызывает неоднозначности, будем отождествлять номер индекса l и индекс j_l. Пусть $f = \{k_1, k_2, ..., k_t\} \subseteq N(s)$; $k_i < k_{i+1}$, $i = \overline{1, t-1}$. Набор значений индексов $F_f = (j_{k_1}, j_{k_2}, ..., j_{k_t})_f$ будем называть *t*-индексом. Множество всех *t*-индексов, соответствующих f, определим как $E_f = \{F_f \mid F \in J_{k_1} \times J_{k_2} \times ... \times J_{k_t}\}$. Там, где это не вызывает неоднозначности, будем опускать нижний индекс f и записывать *t*-индекс как $j_{k_1} j_{k_2} ... j_{k_t}$. Компоненту j_i набора F_f будем обозначать $F_f(i) = j_i$, $i \in f$. Пусть $f' \subseteq f'' \subseteq N(s)$, тогда обозначим $F_{f'} = (F_{f''})_{f'}$, если $F_{f'} \in E_{f'}$, $F_{f''} \in E_{f''}$ и $F_{f'}(i) = F_{f''}(i)$, $i \in f'$. Если $F_{f'} \in E_{f'}$, $F_{f''} \in E_{f''}$, где $f', f'' \subseteq N(s)$ и $f' \cap f'' = \varnothing$, то через $F_{f'} F_{f''}$ обозначим такой набор, что $F_{f'} F_{f''} \in E_{f' \cup f''}$ и $(F_{f'} F_{f''})_{f'} = F_{f'}$, $(F_{f'} F_{f''})_{f''} = F_{f''}$. Далее определим $\overline{f} = N(s) \setminus f$, тогда согласно введенному обозначению $F_{N(s)} = F_f F_{\overline{f}}$, если $F_f = (F_{N(s)})_f$ и $F_{\overline{f}} = (F_{N(s)})_{\overline{f}}$. Пусть $M \subseteq 2^{N(s)}$, тогда будем обозначать $\widetilde{M} = \{\overline{f} \mid f \in M\}$.

Каждому набору F_f поставим в соответствие действительное число z_{F_f}, $F_f \in E_f$. Данное отображение множества *t*-индексов E_f в множество действительных чисел назовем *t*-индексной матрицей и обозначим через $\{z_{F_f}\}$. Рассмотрим *s*-индексную матрицу $\{z_{N(s)}\}$ и введем следующее обозначение

$$\sum_{F_f \in E_f} z_{F_f F_{\bar f}} = \sum_{j_{k_1} \in J_{k_1}} \dots \sum_{j_{k_t} \in J_{k_t}} z_{F_f F_{\bar f}}, \ F_{\bar f} \in E_{\bar f}.$$

Введенное обозначение подсумм s-индексной матрицы будем использовать при формализации многоиндексных транспортных задач.

Пусть M – заданное множество, $M \subseteq 2^{N(s)}$; $\{a_{F_{\bar f}}\}$, $\{b_{F_{\bar f}}\}$ – заданные $|\,\bar f\,|$-индексные матрицы свободных коэффициентов, $0 \le a_{F_{\bar f}} \le b_{F_{\bar f}}$, $F_{\bar f} \in E_{\bar f}$, $f \in M$; $\{c_{F_{N(s)}}\}$ – заданная s-индексная матрица коэффициентов целевой функции; $\{x_{F_{N(s)}}\}$ – s-индексная матрица неизвестных. Тогда многоиндексная задача линейного программирования транспортного типа формализуется следующим образом:

$$a_{F_{\bar f}} \le \sum_{F_f \in E_f} x_{F_f F_{\bar f}} \le b_{F_{\bar f}}, F_{\bar f} \in E_{\bar f}, \ f \in M, \tag{3.1}$$

$$x_{F_{N(s)}} \ge 0, \ F_{N(s)} \in E_{N(s)}, \tag{3.2}$$

$$\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} \to \min. \tag{3.3}$$

Задачу (3.1)-(3.3) будем обозначать $w(s; M; n_1,..., n_2; \{a_{F_{\bar f}}\}, \{b_{F_{\bar f}}\}, f \in M; \{c_{F_{N(s)}}\})$, а класс всех многоиндексных задач линейного программирования (3.1)-(3.3) при фиксированном множестве M будем обозначать $W(M)$. Систему ограничений (3.1),(3.2) будем обозначать $d(s; M; n_1,..., n_s; \{a_{F_{\bar f}}\}, \{b_{F_{\bar f}}\}, f \in M)$, а класс всех многоиндексных систем линейных неравенств (3.1),(3.2) при фиксированном множестве M будем обозначать $D(M)$.

Если $w \in W(M)$, то ограничение вида (3.1) задачи w, соответствующее фиксированному множеству $f \in M$ и фиксированном набору $F_{\bar f} \in E_{\bar f}$, обозначим $d(w, f, F_{\bar f})$. Матрицу системы ограничений задачи w обозначим через $Matr(w)$. Строку матрицы $Matr(w)$, определяемую двусторонним неравенством $d(w, f, F_{\bar f})$, обозначим $row(w, f, F_{\bar f})$, $F_{\bar f} \in E_{\bar f}$, $f \in M$. Столбец матрицы $Matr(w)$, соответствующий переменной $x_{F_{N(s)}}$, обозначим $col(w, F_{N(s)})$, $F_{N(s)} \in E_{N(s)}$. Пусть заданы последовательности $f^{(1)},..., f^{(k_1)} \in M$, $F^{(1)}_{\overline{f^{(1)}}} \in E_{\overline{f^{(1)}}},..., F^{(k_1)}_{\overline{f^{(k_1)}}} \in E_{\overline{f^{(k_1)}}}$ и $F^{(1)}_{N(s)},..., F^{(k_2)}_{N(s)} \in E_{N(s)}$, тогда подматрицу, образованную элементами матрицы $Matr(w)$, находящимися на пересечении строк

$row(w, f^{(i)}, F^{(i)}_{f^{(i)}})$, $i=\overline{1,k_1}$ и столбцов $col(w, F^{(j)}_{N(s)})$, $j=\overline{1,k_2}$, будем обозначать $Matr(w; (f^{(i)}, F^{(i)}_{f^{(i)}}),\ i=\overline{1,k_1}; F^{(j)}_{N(s)},\ j=\overline{1,k_2})$.

В ряде задач s-индексная матрица коэффициентов целевой функции $\{c_{F_{N(s)}}\}$ имеет декомпозиционную структуру и может быть представлена в виде подсуммы многоиндексных матриц меньшей размерности. Пусть $H \subseteq 2^{N(s)}$ и заданы $|H|$ многоиндексных матриц коэффициентов $\{d_{F_f}\}$, $f \in H$, что

$$c_{F_{N(s)}} = \sum_{f \in H} d_{(F_{N(s)})_f}\ ,\ F_{N(s)} \in E_{N(s)}.$$

Тогда рассмотрим целевую функцию следующего вида:

$$\sum_{F_{N(s)} \in E_{N(s)}} \sum_{f \in H} d_{(F_{N(s)})_f} x_{F_{N(s)}} \to \min\ . \tag{3.4}$$

Класс всех многоиндексных задач задача линейного программирования (3.1),(3.2),(3.4) при фиксированных множествах M и H будем обозначать $W(M,H)$. Справедливы следующие утверждения.

Утверждение 3.1. Пусть $M,H \subseteq 2^{N(s)}$, тогда $W(M,H) \subseteq W(M)$.

Утверждение 3.2. Пусть $M,H \subseteq 2^{N(s)}$ и $N(s) \in H$, тогда $W(M,H) = W(M)$.

Пусть $f_1 \subseteq f_2 \subseteq N(s)$ и заданы многоиндексные матрицы $\{c_{F_{N(s)}}\}$, $\{d_{F_{f_1}}\}$, для которых выполняются условия $c_{F_{N(s)}} = d_{(F_{N(s)})_{f_1}}$, $F_{N(s)} \in E_{N(s)}$. Тогда существует многоиндексная матрица $\{d_{F_{f_2}}\}$, что $c_{F_{N(s)}} = d_{(F_{N(s)})_{f_2}}$, $F_{N(s)} \in E_{N(s)}$. Такую матрицу можно определить следующим образом $d_{(F_{N(s)})_{f_2}} = d_{(F_{N(s)})_{f_1}}$, $F_{N(s)} \in E_{N(s)}$. Отсюда несложно увидеть, что справедливо следующее утверждение.

Утверждение 3.3. Пусть $M,H_1,H_2 \subseteq 2^{N(s)}$ и для каждого $f_1 \in H_1$ найдется $f_2 \in H_2$, что $f_1 \subseteq f_2$. Тогда $W(M,H_1) \subseteq W(M,H_2)$.

Дополнительно среди задач класса $W(M,H)$ выделим задачи, удовлетворяющие следующему обобщенному неравенству треугольника.

$$d_{(F_{N(s)})_{f'}} \le \sum_{f \in H \setminus \{f'\}} d_{(F_{N(s)})_f}\ ,\ F_{N(s)} \in E_{N(s)},\ f' \in H\ . \tag{3.5}$$

Подкласс всех задач класса $W(M,H)$, удовлетворяющих условию (3.5), при фиксированных множествах M и H будем обозначать $W(M,\Delta H)$.

Особый интерес представляет решение целочисленной многоиндексной транспортной задачи, когда дополнительно необходимо выполнение ограничения целочисленности переменных

$$x_{F_{N(s)}} \in Z_+,\ F_{N(s)} \in E_{N(s)}. \tag{3.6}$$

Задачу (3.1)-(3.3),(3.6) будем обозначать $w_Z(s;M;n_1,...,n_s;\{a_{F_{\bar{f}}}\},\{b_{F_{\bar{f}}}\},\ f \in M)$. Также будем добавлять нижний индекс Z к обозначению класса многоиндексных задач, если дополнительно выполняется условие целочисленности (3.6). Тогда соответствующие классы систем целочисленных линейных неравенств (3.1),(3.6), задач целочисленного линейного программирования (3.1),(3.6),(3.3) и задач целочисленного линейного программирования (3.1),(3.6),(3.4) при фиксированных множествах M и H будем обозначать через $D_Z(M)$, $W_Z(M)$ и $W_Z(M,H)$ соответственно.

Рассмотрим задачу исследования несовместной многоиндексной системы линейных неравенств транспортного типа. Пусть многоиндексная система линейных неравенств несовместна. Известно подмножество двусторонних неравенств системы, для которых разрешены изменения левых и (или) правых границ. Ограничения, для которых разрешены изменения границ, будем называть "желательными", иначе "жесткими". Заданы штрафы за нарушения границ желательных ограничений. Содержательно рассматриваемая задача заключается в определении значений многоиндексной матрицы неизвестных, удовлетворяющих жестким ограничениям и минимизирующих суммарный штраф за нарушения желательных ограничений.

Введем вспомогательные обозначения. Обозначим через $u^a_{F_{\bar{f}}}$ ($u^b_{F_{\bar{f}}}$) параметр, принимающий значение 1, если для неравенства $d(w,f,F_{\bar{f}})$ разрешено изменение нижней (верхней) границы, и значение 0 в противном случае; $e^a_{F_{\bar{f}}}$ ($e^b_{F_{\bar{f}}}$) – штраф за изменение на единицу нижней (верхней) границы неравенства $d(w,f,F_{\bar{f}})$, $e^a_{F_{\bar{f}}},e^b_{F_{\bar{f}}} > 0$; $y^a_{F_{\bar{f}}}$ ($y^b_{F_{\bar{f}}}$) – неизвестная величина, на которую будет изменена нижняя (верхняя) граница неравенства $d(w,f,F_{\bar{f}})$ $F_{\bar{f}} \in E_{\bar{f}}$, $f \in M$, $w \in W(M)$. Пусть система ограничений задачи $w \in W(M)$ несовместна и $w = w(s;M;n_1,...,n_s;\{a_{F_{\bar{f}}}\},\{b_{F_{\bar{f}}}\},\ f \in M;\{c_{F_{N(s)}}\})$. Тогда задача исследования несовместной многоиндексной системы заключается в определении многоиндексных матриц неизвестных $\{y^a_{F_{\bar f}}\}$, $\{y^b_{F_{\bar f}}\}$, $f \in M$ и $\{x_{F_{N(s)}}\}$, являющихся решением следующей задачи линейного программирования:

$$a_{F_{\bar f}} - u^a_{F_{\bar f}} y^a_{F_{\bar f}} \le \sum_{F_f \in E_f} x_{F_f F_{\bar f}} \le b_{F_{\bar f}} + u^b_{F_{\bar f}} y^b_{F_{\bar f}},\ F_{\bar f} \in E_{\bar f},\ f \in M\ , \tag{3.7}$$

$$y^a_{F_{\bar f}}, y^b_{F_{\bar f}} \ge 0,\ F_{\bar f} \in E_{\bar f},\ f \in M\ , \tag{3.8}$$

$$x_{F_{N(s)}} \ge 0\ ,\ F_{N(s)} \in E_{N(s)}\ , \tag{3.9}$$

$$\sum_{f \in M} \sum_{F_{\bar f} \in E_{\bar f}} (e^a_{F_{\bar f}} u^a_{F_{\bar f}} y^a_{F_{\bar f}} + e^b_{F_{\bar f}} u^b_{F_{\bar f}} y^b_{F_{\bar f}}) \to \min\ . \tag{3.10}$$

Класс всех задач задача линейного программирования (3.7)-(3.10) при фиксированном множестве M будем обозначать $S(M)$.

Рассмотрим многокритериальную многоиндексную задачу с кусочно-постоянными критериями оптимальности. Будем ассоциировать многоиндексные подсуммы многоиндексной матрицы неизвестных с определенными многоиндексными показателями задачи, которые будем разделять на «жесткие», требующие обязательного выполнения, и желательные, к достижению которых нужно стремиться. Жесткие показатели формализуются в виде ограничений, а желательные – в виде критериев оптимальности. Тогда задача ставится как многокритериальная задача учета желательных показателей с критериями, определяющими степень удовлетворения желательных. При определении критериев оптимальности будем использовать кусочно-постоянные критерии, ранжируя тем самым ступени удовлетворения желательных ограничений.

С заданными множествами M и H, $M, H \subseteq 2^{N(s)}$, будем связывать жесткие и желательные показатели соответственно. Тогда множество M определяет систему многоиндексных неравенств транспортного типа (3.1),(3.2). Множество H определяет кусочно-постоянные критерии, формализация которых приводится далее. Пусть $H \subseteq 2^{N(s)}$ и $P(H) = \{(f, F_f) \mid F_f \in E_f,\ f \in H\}$. Для каждой из пар (f, F_f) определим $p+1$ вложенных сегментов $S^{(l)}_{f,F_f}$, $l = \overline{0,p}$, что $S^{(l)}_{f,F_f} \subseteq S^{(l+1)}_{f,F_f}$, $l = \overline{0,p-1}$, $S^{(p)}_{f,F_f} = [-\infty, +\infty]$, $(f, F_f) \in P(H)$. Каждая из пар (f, F_f) определяет следующую функцию предпочтения

$$\chi_{f,F_f}(w) = \begin{cases} r, \text{если } w \in S^{(r)}_{f,F_f} \text{ и } w \notin S^{(r-1)}_{f,F_f}, \text{где } r \in \{1,2,\ldots,p\}, \\ 0, \text{если } w \in S^{(0)}_{f,F_f}. \end{cases}$$

Тогда рассмотрим многокритериальную задачу определения s-индексной матрицы неизвестных $\{x_{F_{N(s)}}\}$, удовлетворяющей системе ограничений (3.1),(3.2) и минимизирующей функции предпочтения

$$\chi_{f,F_f}(\sum_{F_{\bar{f}} \in E_{\bar{f}}} x_{F_f F_{\bar{f}}}) \to \min, \ (f, F_f) \in P(H). \tag{3.11}$$

Класс многокритериальных задач (3.1),(3.2),(3.11) будем обозначать $U(M,H)$. Пусть множество $P(H)$ упорядочено по значимости соответствующих критериев. Тогда в качестве схемы компромисса при решении поставленной многокритериальной задачи (3.1),(3.2),(3.11) используем лексикографическое упорядочивание критериев оптимальности, соответствующий класс оптимизационных задач при фиксированных множествах M и H будем обозначать $U_{\prec}(M,H)$. Также в качестве схемы компромисса будем рассматривать максиминную свертку критериев оптимальности, соответствующий класс оптимизационных задач при фиксированных множествах M и H будем обозначать $U_{\max\min}(M,H)$.

3.2. Задачи распределения ресурсов как многоиндексные задачи

Введенную схему формализации многоиндексных задач транспортного типа проиллюстрируем на примере прикладных многоиндексных задач распределения ресурсов, описанных в главе 1. Определим место описанных прикладных задач в классе многоиндексных задач транспортного типа.

Транспортная задача с промежуточными пунктами

Рассмотрим транспортную задачу с промежуточными пунктами, описанную в разделе 1.1. Для данной задачи $s = 3$, $N(s) = \{i, j, k\}$ и $M = \{\{i\},\{k\},\{i, j\},\{j,k\}\}$. Здесь ограничение (1.1) соответствует множеству $\{j,k\}$, ограничение (1.2) – множеству $\{i, j\}$, ограничение (1.3) – множеству $\{k\}$, ограничение (1.4) – множеству $\{i\}$.

Тогда поставленные многоиндексные задачи линейного программирования (1.1)-(1.5),(1.6), задача (1.1)-(1.5),(1.7) и задача (1.1)-(1.5),(1.8) относятся к классу $W(M)$. Можно заметить, что коэффициенты целевых функций (1.6), (1.7) и (1.8) имеют декомпозиционную структуру – они определены в виде суммы многоиндексных матриц коэффициентов. Так для критерия (1.6) коэффициенты целевой функции имеют вид $a_i + c_{ij} + d_{jk}$, $i \in I, j \in J, k \in K$. Отсюда задача (1.1)-(1.5),(1.6) относится к классу $W(M,H)$, где $H = \{\{i\},\{i, j\},\{j,k\}\}$. Задача (1.1)-(1.5),(1.7) относится к классу $W(M,H)$, где $H=\{\{k\}\}$. Задача (1.1)-(1.5),(1.8) относится к классу $W(M,H)$, где $H=\{\{i\},\{k\},\{i,j\},\{j,k\}\}$.

При постановке многокритериальной задачи выбора плана перевозок (1.1)-(1.5),(1.9) в качестве показателей плана, определяющих критерии задачи, выбраны объемы продукта соответствующего потребителям системы. Объем продуктов потребленный элементом k определяется как подсумма $\sum_{i\in I}\sum_{j\in J}x_{ijk}$, $k\in K$. Тогда поставленная многокритериальная задача (1.1)-(1.5),(1.9) относится к классу $U(M,H)$, где $M=\{\{i\},\{k\},\{i,j\},\{j,k\}\}$ – определяет жесткие показатели, формализуемые в виде системы ограничений (1.1)-(1.5), $H=\{\{k\}\}$ – определяет желательные показатели, формализуемые в виде критериев (1.9). Если множество потребителей K упорядочено с точки зрения их приоритета, то при решении многокритериальной задачи (1.1)-(1.5),(1.9) может быть рассмотрена лексикографическая свертка критериев. Соответствующая задача относится к классу $U_{\prec}(M,H)$. В случае равнозначности потребителей в качестве схемы компромисса при решении многокритериальной задачи (1.1)-(1.5),(1.9) может быть рассмотрена максиминная свертка. Соответствующая задача относится к классу $U_{\max\min}(M,H)$.

Задача формирования портфеля заказов

Рассмотрим задачу формирования портфеля заказов, описанную в разделе 1.2. Для данной задачи $s=3$, $N(s)=\{i,j,t\}$ и $M=\{\varnothing,\{j\},\{i,t\},\{j,t\}\}$. Здесь ограничение (1.10) соответствует множеству $\{j,t\}$, ограничение (1.11) – множеству $\{j\}$, ограничение (1.12) – множеству $\{i,t\}$, ограничение (1.13) – множеству $\varnothing$. Тогда поставленная проблема определения допустимого портфеля заказов (1.10)-(1.14) относится к классу $D(M)$.

Пусть система (1.10)-(1.14) несовместна, что может быть вызвано несогласованностью внутренних ограничений предприятия. Вопрос о несогласованности внутренних ограничений связан с исследованием совместности системы (1.10),(1.11),(1.14), которая относится к классу $D(M)$, где $M=\{\{j\},\{j,t\}\}$. Далее рассматриваются две задачи: задача формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам и задача формирования портфеля заказов с возможными нарушениями сроков выполнения работ по заказам.

Задача формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам ставится как задача (1.10),(1.11),(1.13)-(1.17). В поставленной задаче разрешены нарушения ограничения (1.12), определяемого множеством $\{i,t\}$. Отсюда задача (1.10),(1.11),(1.13)-(1.17) относится к классу $S(M)$, где $M=\{\varnothing,\{j\},\{i,t\},\{j,t\}\}$, и при этом $u^a_{F_{\bar f}}=u^b_{F_{\bar f}}=0$, $F_{\bar f}\in E_{\bar f}$, $f\in M\setminus\{i,t\}$ и $u^a_{F_{\bar f}}=u^b_{F_{\bar f}}=1$, $F_{\bar f}\in E_{\bar f}$, $f=\{i,t\}$.

Задача формирования портфеля заказов с возможными нарушениями сроков выполнения работ по заказам ставится как задача линейного программирования (1.10)-(1.12),(1.14),(1.18). Поставленная задача относится к классу $W(M)$, где $M=\{\{j\},\{i,t\},\{j,t\}\}$.

Объемно-календарное планирование переработки газового конденсата

Рассмотрим задачу объемно-календарного планирования переработки газового конденсата, описанную в разделе 1.3. Для данной задачи $s=5$, $N(s)=\{i,j,k,p,t\}$ и $M=\{\{i,j\},\{i,p\},\{j,k,p,t\}\}$. Здесь ограничение (1.19) соответствует множеству $\{j,k,p,t\}$, ограничение (1.20) – множеству $\{i,p\}$, ограничение (1.21) – множеству $\{i,j\}$. Тогда поставленные многоиндексные задачи линейного программирования (1.19)-(1.22),(1.23), задача (1.19)-(1.22),(1.24), задача (1.19)-(1.22),(1.25), задача (1.19)-(1.22),(1.26), задача (1.19)-(1.22),(1.27) и задача (1.19)-(1.22),(1.28) относятся к классу $W(M)$.

Можно заметить, что коэффициенты целевых функций рассматриваемых задач имеют декомпозиционную структуру. Отсюда задача (1.19)-(1.22),(1.23) относится к классу $W(M,H)$, где $H=\{\{k,p,t\}\}$. Задача (1.19)-(1.22),(1.24) относится к классу $W(M,H)$, где $H=\{\{j,k\}\}$. Задача (1.19)-(1.22),(1.25) относится к классу $W(M,H)$, где $H=\{\{i,j\}\}$. Задача (1.19)-(1.22),(1.26) относится к классу $W(M,H)$, где $H=\{\{t\}\}$. Задача (1.19)-(1.22),(1.27) относится к классу $W(M,H)$, где $H=\{\{k,p\}\}$. Задача (1.19)-(1.22),(1.28) относится к классу $W(M,H)$, где $H=\{\{t\},\{i,j\},\{j,k\},\{k,p\},\{k,p,t\}\}$.

Задача составления расписания занятий

Рассмотрим задачу составления расписания занятий, описанную в разделе 1.4. Для данной задачи $s=4$, $N(s)=\{i,j,k,t\}$ и $M=\{\varnothing,\{i,j\},\{j,k\},\{i,k,t\}\}$. Здесь ограничение (1.29) соответствует множеству $\{j,k\}$, ограничение (1.30) – множеству $\{i,k,t\}$, ограничение (1.31) – множеству $\{i,j\}$, ограничения (1.32), (1.33), (1.34) – множеству $\varnothing$, ограничение (1.35) эквивалентно совокупности ограничений (3.12) и дополнительного ограничения целочисленности переменных, при этом ограничение (3.12) также соответствует множеству $\varnothing$.

$$0 \le x_{ijkt} \le 1, \; i \in I, j \in J, k \in K, t \in T. \tag{3.12}$$

Отсюда система целочисленных линейных неравенств (1.29)–(1.35) относится к классу $D_Z(M)$, задача целочисленного линейного программирования (1.29)–(1.36) относится к классу $W_Z(M)$.

Заметим, что с учетом ограничения неотрицательности переменных ограничения (1.32), (1.33), (1.34) могут быть записаны в следующей эквивалентной форме

$$\sum_{k \in K} \sum_{t \in T} x_{ijkt} = 0, \; j \in J \setminus A_i, i \in I, \tag{3.13}$$

$$\sum_{j \in J} \sum_{k \in K} x_{ijkt} = 0, \; t \in T \setminus B_i, i \in I, \tag{3.14}$$

$$\sum_{i \in I} \sum_{j \in J} x_{ijkt} = 0, \; t \in T \setminus C_k, k \in K. \tag{3.15}$$

Кроме того, ограничение (3.12) автоматически выполняется при выполнении ограничения (1.29) и ограничения неотрицательности переменных. Отсюда система целочисленных линейных неравенств (1.29)–(1.35) относится к классу $D_Z(M')$, где $M' = \{\{i, j\}, \{j, k\}, \{k, t\}, \{i, k, t\}\}$. Далее несложно увидеть, что коэффициенты целевой функции (1.36) имеют декомпозиционную структуру. Отсюда задача (1.29)–(1.36) относится к классу $W(M', H)$, где $H = \{\{i, j\}, \{i, t\}, \{k, t\}\}$.

3.3. Общая концепция сводимости многоиндексных задач

При описании результатов сводимости одного класса задач к другому удобно использовать схему формализации сводимости, позволяющую классифицировать требуемые вычислительные затраты и применяемые процедуры при сведении. Приведем формализацию концепции сводимости, которую далее будем использовать при исследовании сводимости классов многоиндексных задач к классу задач поиска потока в сети.

Пусть $A \in R^{n \times m}$, $b, b^-, b^+ \in R^n$, $c \in R^m$ – заданные параметры, $x \in R^m$ – вектор неизвестных. Через $w(A,b,c)$ будем обозначать задачу линейного программирования $\min\{(c,x) \mid Ax \le b, x \ge 0\}$; через $w(A,b^-,b^+,c)$ – задачу линейного программирования $\min\{(c,x) \mid b^- \le Ax \le b^+, x \ge 0\}$. Для удобства через $nrow(A)$ и $ncol(A)$ будем обозначать количество строк и столбцов матрицы A соответственно. Отметим, что задача $w(A,b^-,b^+,c)$ может быть описана с использованием обозначения вида $w(A,b,c)$. Тем не менее будем использовать обозначение $w(A,b^-,b^+,c)$ в случае, когда хотим подчеркнуть, что система ограничений задачи представляет собой систему двусторонних неравенств. Также будем рассматривать задачи целочисленного линейного программирования. Если $w = w(A,b,c)$ – задача линейного программирования, то через w_Z обозначим задачу целочисленного линейного программирования $w_Z = \min\{(c,x) \mid Ax \le b, x \in Z_+^{ncol(A)}\}$. Пусть W – произвольный класс задач линейного программирования. Соответствующий класс задач целочисленного линейного программирования определим как $W_Z = \{w_Z \mid w \in W\}$.

Далее рассмотрим два класса задач линейного программирования W' и W''. На содержательном уровне под сводимостью класса W' к классу W'' понимается возможность построения для произвольной задачи $w' \in W'$ соответствующей задачи $w'' \in W''$ таким образом, чтобы решение задачи w'' определяло решение задачи w'. При формализации схем сведения будем определять временные затраты и (или) конкретные вычислительные процедуры, связанные с:

– построением матрицы системы ограничений задачи w'' по исходным параметрам задачи w';
– построением свободных коэффициентов и коэффициентов целевой функции задачи w'' по исходным параметрам задачи w';
– построением решения задачи w' по решению задачи w''.

Предлагаемое далее обозначение схемы сведения вводится по аналогии с нотацией Р. Грэхема (R. Graham), используемой при классификации задач теории расписаний [129].

Определение 3.1. Будем говорить, что класс W' является $t_1 - s_1 \mid t_2 - s_2 \mid t_3 - s_3$ сводимым к классу W'', если для любой задачи $w' = w(A',b',c') \in W'$ можно за время $O(t_1)$ с использованием процедуры s_1 построить матрицу A'', за время $O(t_2)$ с использованием процедуры s_2 построить векторы b'',c'', что $w'' = w(A'',b'',c'') \in W''$ и при этом

– задача w' совместна (ограничена) тогда и только тогда, когда совместна (ограничена) задача w'';
– если известно оптимальное (допустимое) решение x'' задачи w'', то оптимальное (допустимое) решение x' задачи w' может быть построено за время $O(t_3)$ с использованием процедуры s_3.

Здесь s_1, s_2, s_3 – строковые обозначения вычислительных процедур, связанных с построением матрицы системы ограничений, с построением свободных коэффициентов и коэффициентов целевой функции и с построением решения задачи соответственно.

Замечание. Опционально будем опускать записи вычислительных затрат t_1, t_2, t_3 и (или) записи обозначений вычислительных процедур s_1, s_2, s_3, если при сведении не конкретизируются соответствующие вычислительные затраты и (или) вычислительные процедуры. Задачу w'' (см. определение 3.1) будем называть задачей, соответствующей задаче w'. Решение x' (см. определение 3.1) будем называть решением, соответствующим решению x''. При определении вычислительных затрат t_1, t_2, t_3, если не оговорено иного, будем оценивать, как и в [82], количество вычислительных операций на машине с произвольным доступом к памяти. Иногда для удобства функции оценки вычислительной сложности t_1, t_2, t_3 будем заменять обозначениями L или P, подразумевая линейные или полиномиальные функции от размера матрицы A' соответственно. В случае, когда при определении t_1, t_2, t_3 оценивается вычислительная сложность рассматриваемых вычислительных процедур, будем добавлять верхний индекс «*» в записи соответствующих оценок. В частности будем записывать L^* или P^*, когда речь идет о линейной или полиномиальной оценках вычислительной сложности как функции от размера индивидуальной задачи соответственно. Схематично концепция $t_1 - s_1 \,|\, t_2 - s_2 \,|\, t_3 - s_3$ сводимости представлена на рис. 3.1.

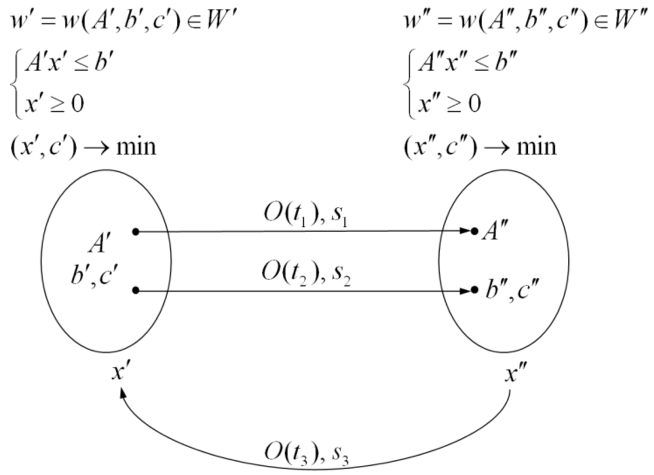

Рисунок 3.1.

В работе исследуется возможность сведения класса многоиндексных задач линейного программирования транспортного типа к классу задач поиска потока минимальной стоимости. В качестве потоковой задачи будем рассматривать задачу поиска циркуляции минимальной стоимости в сети (2.5),(2.6),(2.4), обозначаемую $z_{MCC}(G;a_{ij}b_{ij},c_{ij}(i,j)\in A)$. Класс всех задач (2.5),(2.6),(2.4) будем обозначать W_{Graph}. Также нас будет интересовать возможность сведения к классу задач поиска циркуляции минимальной стоимости в древовидной сети. Под древовидной сетью задачи о циркуляции будем понимать сеть, определяемую графом $G=(V,A)$, для которого существует корневое ориентированное дерево $G'=(V,A')$ с корнем $s\in V$ и множеством листьев $T\subseteq V$, что $A=A'\cup\{(t,s)\,|\,t\in T\}$. Класс задач поиска циркуляции минимальной стоимости в древовидной сети будем обозначать W_{Tree}, здесь $W_{Tree}\subseteq W_{Graph}$.

Далее в работе исследуются вопросы $t_1-s_1\,|\,t_2-s_2\,|\,t_3-s_3$ сводимости классов многоиндексных задач $W(M)$ и $W(M,H)$ к классам задач поиска потока в сети W_{Graph} и W_{Tree}. Применяемые при исследовании сводимости вычислительные процедуры s_1, s_2, s_3 будут определены при описании рассматриваемых схем сведения.

Выводы

Описана общая схема формализации многоиндексных задач, применяемая в работе. Данная схема проиллюстрирована на примере прикладных многоиндексных задач, рассмотренных в главе 1. Полученные в работе теоретические результаты исследования многоиндексных задач будут изложены с использованием данной формализации. Описана общая схема формализации сводимости задач линейного программирования, применяемая далее при исследовании сводимости многоиндексных задач к классу задач поиска потока в сети. Результаты исследования сводимости будут применены в работе при построении алгоритмов решения поставленных многоиндексных задач $D(M)$, $W(M)$, $W(M,H)$, $W(M,\Delta H)$ (и соответственно $D_Z(M)$, $W_Z(M)$, $W_Z(M,H)$, $W_Z(M,\Delta H)$), а также $S(M)$, $U_{\prec}(M,H)$ и $U_{\max\min}(M,H)$.

Глава 4. Сводимость с сохранением соответствия ребер

Глава посвящена исследованию сводимости класса многоиндексных задач к классу задач поиска потока минимальной стоимости в сети с использованием схемы сведения с сохранением соответствия ребер [8, 20, 25, 27, 28, 32, 34, 89]. Особенностью рассматриваемой концепции сводимости является существование соответствия между переменными исходной задачи и дугами вспомогательной сети. Предлагаемая схема сведения гарантирует, что произвольный оптимальный поток вспомогательной сети будет определять такое оптимальное решение исходной задачи, в котором переменным присваивается значение потока вдоль соответствующих дуг сети. В рамках рассматриваемой схемы сведения в данной главе найдены условия сводимости класса многоиндексных задач линейного программирования к классу задач поиска потока в сети и к классу задач поиска потока в древовидной сети. Предлагается конструктивная схема сведения. На основе предложенной схемы сведения строятся алгоритмы решения многоиндексных задач, удовлетворяющих условиям сводимости. Результаты обобщаются при исследовании более широкого класса схем сводимости.

4.1. Концепция $t_1 \,|\, t_2 - equal \,|\, t_3 - edge$ сводимости

Введем схему $t_1 | t_2 - equal | t_3 - edge$ сводимости (сводимости с сохранением соответствия ребер), которая будет применяться в данной главе при исследовании сводимости многоиндексных задач к классу задач поиск потока в сети. Основные особенности предлагаемой схемы:

– при построении вспомогательной сети пропускные способности дуг определяются равными соответствующим свободным коэффициентам двусторонних неравенств системы ограничений, стоимости дуг определяются равными соответствующим коэффициентам целевой функции (такая особенность сведения отражается в процедуре, обозначаемой $equal$);

– при построении решения исходной задачи предполагается существование соответствия между ее переменными и дугами вспомогательной сети, при этом оптимальный поток вспомогательной сети определяет такое оптимальное решение исходной задачи, в котором переменным присваивается значение потока вдоль соответствующих дуг сети (такая особенность сведения отражается в процедуре, обозначаемой $edge$).

Таким образом, рассматриваемая схема сведения является частным случаем схемы, описанной в определении 3.1, для которой $s_2 = equal, s_3 = edge$.

Определение 4.1. Пусть W – класс задач линейного программирования с двусторонней системой линейных неравенств. Будем говорить, что класс W является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу $W' \subseteq W_{Graph}$, если класс W является $t_1 | t_2 | t_3$ сводимым к классу W', и дополнительно произвольная задача $w = w(A, b^-, b^+, c) \in W$, а также соответствующая ей задача $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W'$ удовлетворяют следующим условиям. Найдутся такие инъективные функции $\alpha : \{1,..., nrow(A)\} \to A_G$, $\beta : \{1,..., ncol(A)\} \to A_G$, что

– $l_{\alpha(i)} = b_i^-, u_{\alpha(i)} = b_i^+$, $i = \overline{1, nrow(A)}$; $l_{(u,v)} = 0, u_{(u,v)} = b^*$, $(u,v) \in A_G \setminus \{\alpha(i) \mid i = \overline{1, nrow(A)}\}$, где $b^* = \sum_{k=1}^{nrow(A)} b_k^+$;

– $e_{\beta(i)} = c_i$, $i = \overline{1, ncol(A)}$; $e_{(u,v)} = 0$, $(u,v) \in A_G \setminus \{\beta(i) \mid i = \overline{1, ncol(A)}\}$;

– если x_{ij}, $(i,j) \in A_G$, является оптимальным (допустимым) решением задачи z, то $(x_{\beta(1)},..., x_{\beta(ncol(A))})$ будет являться оптимальным (допустимым) решением задачи w, соответствующим решению задачи z.

Замечание. В качестве величины b^* выбрана величина, значение которой эквивалентно отсутствию (в данном контексте) верхней пропускной способности дуги. Схематично концепция $t_1 | t_2 - equal | t_3 - edge$ сводимости представлена на рис. 4.1.

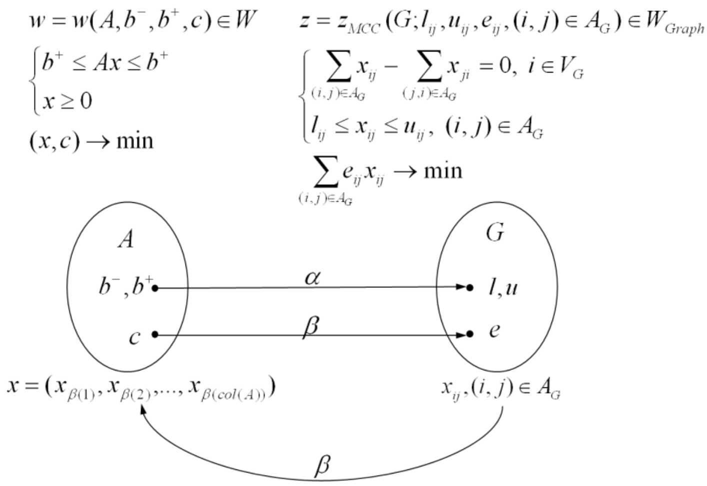

Рисунок 4.1.

Согласно определению 4.1, в случае $t_1 | t_2 - equal | t_3 - edge$ сводимости класса W к классу $W' \subseteq W_{Graph}$ гарантируется, что если $w \in W$, $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}(i, j) \in A_G) \in W_{Graph}$ и z является задачей, соответствующей задаче w, то при построении задачи поиска потока минимальной стоимости z пропускные способности и стоимости дуг в задаче определяются через коэффициенты задачи w, а решение задачи w находится через подмножество компонент решения задачи z. Тогда, согласно утверждению 2.2, можно предложить алгоритм решения задачи w, основанный на решении соответствующей задачи z. Данный алгоритм требует $O(t_1 + t_2 + t_3 + \alpha(O(|V_G|), O(|V_G| + |A_G|)) + \beta(O(|V_G|), O(|V_G| + |A_G|))$ вычислительных операций, где $\alpha(n,m)$ и $\beta(n,m)$ – количество вычислительных операций, требуемых для решения задачи поиска максимального потока z_{MF} и задачи поиска потока минимальной стоимости заданной величины z_{MCF} соответственно в сети с n вершинами и m дугами. Обзор оценок вычислительной сложности для известных потоковых алгоритмов можно найти, например, в [149, 151, 172]. Если дополнительно известно, что $W' \subseteq W_{Tree}$, то для решения задачи z может быть применен алгоритм поиска потока минимальной стоимости в древовидной сети. Тогда, согласно утверждению 2.4, можно предложить алгоритм решения задачи w, основанный на решении соответствующей задачи z поиска потока минимальной стоимости в древовидной сети и имеющий вычислительную сложность $O(t_1 + t_2 + t_3 + |V|^2)$.

Далее в данной главе будут рассмотрены вопросы $t_1 | t_2 - equal | t_3 - edge$ сводимости класса $W(M)$ к классам W_{Graph} и W_{Tree}, а также некоторые обобщения.

4.2. Многоиндексные задачи с 2-вложенной структурой

Вид задач линейного программирования класса $W(M)$ определяется заданным множеством *M*. Поэтому нас будет интересовать нахождение условий, которым должно удовлетворять множество *M*, чтобы класс $W(M)$ являлся $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}.

Теорема 4.1. Пусть $M \subseteq 2^{N(s)}$. Если класс $W(M)$ является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}, то для произвольной задачи $w \in W(M)$ матрица $Matr(w)$ является абсолютно унимодулярной.

Доказательство от противного. Пусть класс $W(M)$ является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}, и найдется такая задача $w \in W(M)$, что матрица $Matr(w)$ не является абсолютно унимодулярной. Согласно [51], условие абсолютной унимодулярности матрицы является необходимым и достаточным условием целочисленности многогранника соответствующей системы линейных неравенств с целочисленными свободными коэффициентами. Отсюда существует задача $w' \in W(M)$, удовлетворяющая следующим условиям:

– $Matr(w') = Matr(w)$,

– свободные коэффициенты задачи w' являются целочисленными,

– система ограничений задачи w' совместна,

– задача w' не имеет целочисленных решений.

Класс $W(M)$ является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}. Тогда рассмотрим задачу $z \in W_{Graph}$, соответствующую задаче w'. По определению 4.1 задача поиска потока минимальной стоимости z совместна и пропускные способности сети в задаче z являются целочисленными. Матрица системы ограничений задачи z является абсолютно унимодулярной, отсюда задача z имеет целочисленное оптимальное решение. Тогда, согласно определению 4.1**,** задача w' также имеет целочисленное оптимальное решение. Получаем противоречие. Таким образом, предположение неверно. *Теорема доказана.*

Определение 4.2. Множество M, $M \subseteq 2^{N(s)}$, называется k-вложенным, если существует разбиение множества M на k подмножеств $M_i = \{f_1^{(i)},..., f_{m_i}^{(i)}\}$, $i = \overline{1,k}$, что $f_j^{(i)} \subseteq f_{j+1}^{(i)}$, $j = \overline{1, m_i - 1}$, $i = \overline{1,k}$.

Лемма 4.1. Пусть $M \subseteq 2^{N(s)}$ и множество M является 1-вложенным, тогда $\sum_{f \in M} | E_{\overline{f}} | = O(| E_{N(s)} |)$.

Доказательство. Пусть $M \subseteq 2^{N(s)}$ и M является 1-вложенным. Тогда по определению 4.2 $M = \{f_1,..., f_m\}$, где $f_j \subseteq f_{j+1}$, $j = \overline{1, m-1}$. Т.к. $f_j \subseteq N(s)$, $j = \overline{1,m}$, и $| N(s) |= s$, то $m \le s+1$. Т.к. $\overline{f_j} \subseteq N(s)$, то $| E_{\overline{f_j}} | \le | E_{N(s)} |$, $j = \overline{1,m}$. Следовательно $\sum_{f \in M} | E_{\overline{f}} | \le s | E_{N(s)} |= O(E_{N(s)})$. *Лемма доказана.*

Было найдено следующее достаточное условие сводимости.

Теорема 4.2. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $L \mid L-equal \mid L-edge$ сводимым к классу W_{Graph}, достаточно, чтобы множество M было 2-вложенным.

Доказательство. Пусть множество $M \subseteq 2^{N(s)}$ является 2-вложенным. Тогда $M = M_1 \cup M_2$, где $M_i = \{f_1^{(i)},...,f_{m_i}^{(i)}\}$, $f_j^{(i)} \subseteq f_{j+1}^{(i)}$, $j=\overline{1,m_i-1}$, $i=\overline{1,2}$. Рассмотрим произвольную задачу $w = w(s;M;n_1,...,n_s;\{a_{F_{\overline{f}}}\},\{b_{F_{\overline{f}}}\}, f \in M) \in W(M)$. Не уменьшая общности, будем считать (для удобства описания схемы сведения), что

– $\varnothing \in M_1$ и тогда $f_1^{(1)} = \varnothing$,

– $|M_1| \geq 2$,

– $M_2 \neq \varnothing$,

в противном случае, добавим недостающие двусторонние неравенства, в качестве нижней границы которых выберем ноль, в качестве верхней – достаточно большую величину, например $\sum\limits_{f \in M}\sum\limits_{F_{\overline{f}} \in E_{\overline{f}}} b_{F_{\overline{f}}}$. Далее приведем процедуру построения задачи $z = z_{MCC}(G;l_{ij},u_{ij},e_{ij},(i,j) \in A_G) \in W_{Graph}$, соответствующей задаче w.

Построим ориентированный граф $G = (V_G, A_G)$ с множеством вершин $V_G = \{v_{F_{\overline{f}}} \mid F_{\overline{f}} \in E_{\overline{f}},\ f \in M\} \cup \{s,t\}$ и множеством дуг $A_G = A' \cup A_1' \cup A_1'' \cup A_2' \cup A_2''$, где

– $A_1' = \{(v_{(F_{\overline{f_i^{(1)}}})_{\overline{f_{i+1}^{(1)}}}}, v_{F_{\overline{f_i^{(1)}}}}) \mid F_{\overline{f_i^{(1)}}} \in E_{\overline{f_i^{(1)}}}, i = \overline{1,m_1-1}\}$;

– $A_1'' = \{(s, v_{F_{\overline{f_{m_1}^{(1)}}}}) \mid F_{\overline{f_{m_1}^{(1)}}} \in E_{\overline{f_{m_1}^{(1)}}}\}$

– $A_2' = \{(v_{F_{\overline{f_i^{(2)}}}}, v_{F_{(F_{\overline{f_i^{(2)}}})_{\overline{f_{i+1}^{(2)}}}}}) \mid F_{\overline{f_i^{(2)}}} \in E_{\overline{f_i^{(2)}}}, i = \overline{1,m_2-1}\}$;

– $A_2'' = \{(v_{F_{\overline{f_{m_2}^{(2)}}}}, t) \mid F_{\overline{f_{m_2}^{(2)}}} \in E_{\overline{f_{m_2}^{(2)}}}\}$;

– $A' = \{t,s\} \cup \{(v_{F_{\overline{f_1^{(1)}}}}, v_{(F_{\overline{f_1^{(1)}}})_{\overline{f_1^{(2)}}}}) \mid F_{\overline{f_1^{(1)}}} \in E_{\overline{f_1^{(1)}}}\}$.

Определим функцию α следующим образом:

– каждому ограничению $d(w, f_i^{(1)}, F_{\overline{f_i^{(1)}}})$ поставим в соответствие дугу $(v_{(F_{\overline{f_i^{(1)}}})_{\overline{f_{i+1}^{(1)}}}}, v_{F_{\overline{f_i^{(1)}}}})$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{\overline{f_i^{(1)}}}}$ и $b_{F_{\overline{f_i^{(1)}}}}$ соответственно, $F_{\overline{f_i^{(1)}}} \in E_{\overline{f_i^{(1)}}}$, $i = \overline{1, m_1 - 1}$;

– каждому ограничению $d(w, f_{m_1}^{(1)}, F_{\overline{f_{m_1}^{(1)}}})$ поставим в соответствие дугу $(s, v_{F_{\overline{f_{m_1}^{(1)}}}})$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{\overline{f_{m_1}^{(1)}}}}$ и $b_{F_{\overline{f_{m_1}^{(1)}}}}$ соответственно, $F_{\overline{f_{m_1}^{(1)}}} \in E_{\overline{f_{m_1}^{(1)}}}$;

– каждому ограничению $d(w, f_i^{(2)}, F_{\overline{f_i^{(2)}}})$ поставим в соответствие дугу $(v_{F_{\overline{f_i^{(2)}}}}, v_{(F_{\overline{f_i^{(2)}}})_{\overline{f_{i+1}^{(2)}}}})$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{\overline{f_i^{(2)}}}}$ и $b_{F_{\overline{f_i^{(2)}}}}$ соответственно, $F_{\overline{f_i^{(2)}}} \in E_{\overline{f_i^{(2)}}}$, $i = \overline{1, m_2 - 1}$;

– каждому ограничению $d(w, f_{m_2}^{(2)}, F_{\overline{f_{m_2}^{(2)}}})$ поставим в соответствие дугу $(v_{F_{\overline{f_{m_2}^{(2)}}}}, t)$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{\overline{f_{m_2}^{(2)}}}}$ и $b_{F_{\overline{f_{m_2}^{(2)}}}}$ соответственно, $F_{\overline{f_{m_2}^{(2)}}} \in E_{\overline{f_{m_2}^{(2)}}}$.

Определим функцию β следующим образом: каждой переменной $x_{F_{N(s)}}$ поставим в соответствие дугу $(v_{(F_{N(s)})_{\overline{f_2^{(1)}}}}, v_{F_{N(s)}})$, $F_{N(s)} \in E_{N(s)}$. Напомним, что $f_1^{(1)} = \varnothing$, и следовательно, $N(s) = \overline{f_1^{(1)}}$.

Покажем, что построенная задача z совместна тогда и только тогда, когда совместна исходная задача w. Действительно, пусть $x_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$ – допустимое решение системы ограничений задачи w. Определим циркуляцию y_{ij}, $(i, j) \in A$, в графе G следующим образом:

– $y_{v_{(F_{N(s)})_{\overline{f_2^{(1)}}}} v_{F_{N(s)}}} = x_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$;

– поток вдоль оставшихся дуг, т.е. дуг из множества $A_G \setminus \{(v_{(F_{N(s)})_{\overline{f_2^{(1)}}}}, v_{F_{N(s)}}) \mid F_{N(s)} \in E_{N(s)}\}$, рассчитывается однозначно через ограничение баланса (2.5) согласно структуре построенного графа G.

По построению и с учетом ограничения (3.1) справедливы следующие соотношения

$$a_{F_{\overline{f_i^{(1)}}}} \le y_{v_{(F_{\overline{f_i^{(1)}}})_{\overline{f_{i+1}^{(1)}}}} v_{F_{\overline{f_i^{(1)}}}}} = \sum_{F_{f_i^{(1)}} \in E_{f_i^{(1)}}} x_{F_{f_i^{(1)}} F_{\overline{f_i^{(1)}}}} \le b_{F_{\overline{f_i^{(1)}}}},\ F_{\overline{f_i^{(1)}}} \in E_{\overline{f_i^{(1)}}},\ i = \overline{1, m_1 - 1}; \tag{4.1}$$

$$a_{F_{\overline{f_{m_1}^{(1)}}}} \le y_{s v_{F_{\overline{f_{m_1}^{(1)}}}}} = \sum_{F_{f_{m_1}^{(1)}} \in E_{f_{m_1}^{(1)}}} x_{F_{f_{m_1}^{(1)}} F_{\overline{f_{m_1}^{(1)}}}} \le b_{F_{\overline{f_{m_1}^{(1)}}}},\ F_{\overline{f_{m_1}^{(1)}}} \in E_{\overline{f_{m_1}^{(1)}}}; \tag{4.2}$$

$$a_{F_{\overline{f_i^{(2)}}}} \le y_{v_{F_{\overline{f_i^{(2)}}}} v_{(F_{\overline{f_i^{(2)}}})_{\overline{f_{i+1}^{(2)}}}}} = \sum_{F_{f_i^{(2)}} \in E_{f_i^{(2)}}} x_{F_{f_i^{(2)}} F_{\overline{f_i^{(2)}}}} \le b_{F_{\overline{f_i^{(2)}}}},\ F_{\overline{f_i^{(2)}}} \in E_{\overline{f_i^{(2)}}},\ i = \overline{1, m_2 - 1}; \tag{4.3}$$

$$a_{F_{\overline{f_{m_2}^{(2)}}}} \le y_{v_{F_{\overline{f_{m_2}^{(2)}}}} t} = \sum_{F_{f_{m_2}^{(2)}} \in E_{f_{m_2}^{(2)}}} x_{F_{f_{m_2}^{(2)}} F_{\overline{f_{m_2}^{(2)}}}} \le b_{F_{\overline{f_{m_2}^{(2)}}}},\ F_{\overline{f_{m_2}^{(2)}}} \in E_{\overline{f_{m_2}^{(2)}}}. \tag{4.4}$$

Тогда в соответствии с введенной функцией α набор $y_{ij}, (i, j) \in A,$ будет являться допустимой циркуляцией задачи z.

Далее, пусть $y_{ij},\ (i, j) \in A_G$ – допустимая циркуляция задачи z. Согласно введенной функции β построим следующий набор значений переменных: $x_{F_{N(s)}} = y_{v_{(F_{N(s)})_{\overline{f_2^{(1)}}}}, v_{F_{N(s)}}}$, $F_{N(s)} \in E_{N(s)}$. По построению, с учетом введенной функции α и с учетом ограничения (2.6) , справедливы соотношения (4.1)–(4.4). Отсюда построенный набор является допустимым решением задачи w.

Пусть $y_{ij},\ (i, j) \in A_G,$ – циркуляция минимальной стоимости в задаче z. Используя функцию β, построим следующий набор значений переменных $x_{F_{N(s)}} = y_{v_{(F_{N(s)})_{\overline{f_2^{(1)}}}}, v_{F_{N(s)}}}$, $F_{N(s)} \in E_{N(s)}$, который (как показано выше) является допустимым решением задачи w. Теперь покажем от противного, что построенный набор будет оптимальным решением задачи w. Действительно, пусть это не так и $x'_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$, – оптимальное решение задачи w. Тогда по предположению $\sum\limits_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} > \sum\limits_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}}$. По построению $\sum\limits_{(i,j) \in A_G} c_{ij} y_{ij} = \sum\limits_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}}$. Далее (аналогично описанной выше схеме) определим следующую циркуляцию

– $y'_{v_{(F_{N(s)})_{\overline{f_2^{(1)}}}} v_{F_{N(s)}}} = x'_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$;

– поток вдоль оставшихся дуг, т.е. дуг из множества $A_G \setminus \{(v_{(F_{N(s)})_{\overline{f_2^{(1)}}}}, v_{F_{N(s)}}) \mid F_{N(s)} \in E_{N(s)}\}$, рассчитывается однозначно через ограничение баланса (2.5) согласно структуре построенного графа G.

Как уже было показано выше, построенный таким образом набор y'_{ij}, $(i,j) \in A_G$, является допустимой циркуляцией задачи z. При этом $\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} y'_{F_{N(s)}}$. $\sum_{(i,j) \in A_G} c_{ij} y_{ij} = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} > \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} = \sum_{(i,j) \in A_G} c_{ij} y'_{ij}$, получаем противоречие, т.к. y_{ij}, $(i,j) \in A_G$, – циркуляция минимальной стоимости в задаче z. Таким образом, предположение неверно, и построенный набор $x_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$, является оптимальным решением задачи w.

Далее проведем анализ сложности вычислительных процедур, связанных с построением задачи z и построением решения задачи w по решению задачи z. Заметим, что исходная задача w содержит $|E_{N(s)}|$ переменных. По построению граф G в задаче z содержит $|V_G| = 2 + \sum_{t=1}^{2} \sum_{i=1}^{m_t} |E_{\overline{f_i^{(t)}}}|$ вершин и $|A_G| = 1 + \sum_{t=1}^{2} \sum_{i=1}^{m_t} |E_{\overline{f_i^{(t)}}}| + |E_{N(s)}|$ дуг. По определению 4.2 множества M_1 и M_2 являются 1-вложенными. Тогда по лемме 4.1 выполняется $|V_G| = O(|E_{N(s)}|)$, $|A_G| = O(|E_{N(s)}|)$. Отсюда построение задачи z требует линейных от размера матрицы $Matr(w)$ вычислительных операций. Схема построения решения задачи w по решению задачи z определяется предложенным выше отображением β и, следовательно, также требует линейных от размера $Matr(w)$ вычислительных операций. Отсюда $W(M)$ является $L \mid L-equal \mid L-edge$ сводимым к классу W_{Graph}. *Теорема доказана.*

Конструктивная схема доказательства теоремы 4.2 в случае 2-вложенности множества M позволяет для задач класса $W(M)$ и для систем линейных неравенств класса $D(M)$ предложить алгоритм решения, основанный на сводимости к классу W_{Graph}.

Алгоритм 4.1. Решение 2-вложенных многоиндексных задач.

Вход. Задача $w \in W(M)$ (система $w \in D(M)$), где M – 2-вложенное.

Шаг 1. Используя конструктивную схему, применяемую при доказательстве теоремы 4.2, построить задачу $z \in W_{Graph}$, соответствующую w. Перейти на шаг 2.

Шаг 2. Найти оптимальное решение (допустимое решение) задачи z, используя алгоритм 2.2 (алгоритм 2.1). Если задача z несовместна, то w также несовместна, и алгоритм завершен; иначе переход на шаг 3.

Шаг 3. Используя отображение β, описанное при доказательстве теоремы 4.2, найти решение w согласно определению 4.1. Алгоритм завершен.

Матрица системы ограничений задачи $z \in W_{Graph}$ абсолютно унимодулярна, поэтому для совместной задачи z с целочисленными параметрами гарантируется существование целочисленного решения [51]. При этом конструктивная схема построения задачи z, применяемая при доказательстве теоремы 4.2, гарантирует целочисленность ее параметров при целочисленности параметров задачи (системы) w. Согласно шагу 3 алгоритма 4.1, в случае целочисленности решения задачи z решение задачи (системы) w также является целочисленным. Отсюда алгоритм 4.1 применим также при исследовании целочисленных задач: $W_Z(M)$ и $D_Z(M)$. Для общности отметим, что если $w \in W_Z(M)$ ($w \in D_Z(M)$) содержит нецелочисленные свободные коэффициенты, то в силу целочисленности коэффициентов матрицы системы ограничений w, задача (система) w при помощи округления параметров может быть преобразована к эквивалентной постановке с целочисленными параметрами.

Заметим, что для задачи $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Graph}$, соответствующей задаче w, согласно доказательству теоремы 4.2, выполняется $|V_G| = O(|E_{N(s)}|)$, $|A_G| = O(|E_{N(s)}|)$. Отсюда с учетом утверждений 2.1, 2.2 можно сформулировать следующее следствие.

Следствие 4.1. Пусть множество $M \subseteq 2^{N(s)}$ является 2-вложенным, тогда алгоритм 4.1 решения задач класса $W(M)$ и класса $W_Z(M)$ (систем класса $D(M)$ и класса $D_Z(M)$) требует $\alpha(O(|E_{N(s)}|), O(|E_{N(s)}|)) + \beta(O(|E_{N(s)}|), O(|E_{N(s)}|))$ ($\alpha(O(|E_{N(s)}|), O(|E_{N(s)}|))$) вычислительных операций.

Здесь, напомним, $\alpha(n,m)$ и $\beta(n,m)$ – количество вычислительных операций алгоритма решения задачи поиска максимального потока и алгоритма решения задачи поиска потока минимальной стоимости заданной величины соответственно в сети с n вершинами и m дугами. Применим для поиска потока минимальной стоимости заданной величины алгоритм, предложенный в работе Орлина [172], обладающий оценкой $O(m\log n(m+n\log n))$. Для поиска максимального потока воспользуемся алгоритмом Слейтора-Тарьяна [181], обладающей оценкой $O(nm\log n)$. Тогда, согласно следствию 4.1, можно сформулировать следующее утверждение.

Утверждение 4.1. Пусть множество $M \subseteq 2^{N(s)}$ является 2-вложенным, тогда существует алгоритм решения задач класса $W(M)$ и класса $W_Z(M)$ (систем класса $D(M)$ и класса $D_Z(M)$), требующий $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ ($O(|E_{N(s)}|^2 \log |E_{N(s)}|)$) вычислительных операций.

Далее покажем, что условие 2-вложенности является необходимым и достаточным условием сводимости многоиндексной транспортной задачи к задаче поиска потока минимальной стоимости.

Определение 4.3. Пусть $M \subseteq 2^{N(s)}$ и $g \subseteq N(s)$, тогда обозначим $M(g) = \{f \cap g \mid f \in M\}$.

Определение 4.4. Пусть $s_1 \le s_2$ и $M_1 \subseteq 2^{N(s_1)}$, $M_2 \subseteq 2^{N(s_2)}$. Тогда обозначим $M_1 \prec M_2$, если существует подмножество $g \subseteq N(s_2)$, $|g| = s_1$ и биективная функция $\pi : g \to N(s_1)$, что $M_1 \subseteq \bigcup_{f \in M_2(g)} \{\{\pi(i) \mid i \in f\}\}$.

Содержательно определение 4.4 означает, что при «игнорировании» индексов множества $N(s_2) \setminus g$ для любой задачи класса $W(M_1)$ существует задача класса $W(M_2)$ содержащая (с точностью до перенумерации индексов π) ограничения соответствующей задачи класса $W(M_1)$.

Лемма 4.2. Пусть $s_1 \le s_2$, $M_1 \subseteq 2^{N(s_1)}$, $M_2 \subseteq 2^{N(s_2)}$ и $M_1 \prec M_2$ тогда, если существует задача $w_1 \in W(M_1)$, что матрица $Matr(w_1)$ не является абсолютно унимодулярной, то существует задача $w_2 \in W(M_2)$, что матрица $Matr(w_2)$ также не является абсолютно унимодулярной.

Доказательство. Пусть выполняются условия леммы. Тогда по определению 4.4 существует подмножество $g \subseteq N(s_2)$, $|g| = s_1$ и биективная функция $\pi : g \to N(s_1)$, что $M_1 \subseteq \bigcup_{f \in M_2(g)} \{\{\pi(i) \mid i \in f\}\}$.

Покажем, что для любой задачи $w_1 \in W(M_1)$ существует задача $w_2 \in W(M_2)$, что $Matr(w_1)$ является подматрицей $Matr(w_2)$. Рассмотрим произвольную задачу

$w_1 \in W(M_1)$, $w_1 = w(s_1; n'_1,...,n'_{s_1}; \{a'_{F_{\bar f}}\},\{b'_{F_{\bar f}}\}, f \in M_1; \{c'_{F_{N(s_1)}}\})$ и выберем $w_2 \in W(M_2)$, $w_2 = w(s_2; n''_1,...,n''_{s_2}; \{a''_{F_{\bar f}}\},\{b''_{F_{\bar f}}\}, f \in M_2; \{c''_{F_{N(s_2)}}\})$ такую, что $n''_i = n'_{\pi(i)}$, $i \in g$. Далее рассмотрим произвольную строку $row(w_1, f_1, F_{\bar f_1})$ матрицы $Matr(w_1)$. Строка $row(w_1, f_1, F_{\bar f_1})$ также может быть задана как подматрица $Matr(w_1; f_1, F_{\bar f_1}; F_{N(s_1)}, F_{N(s_1)} \in E_{N(s_1)})$ размера $1 \times |E_{N(s)}|$. Выберем $f_2 \in M_2$ так, что $f_1 = \{\pi(i) \mid i \in f_2 \cap g\}$; и выберем $F_{\bar f_2} \in E_{\bar f_2}$, что $F_{\bar f_2}(i) = F_{\bar f_1}(\pi(i))$, $i \in \overline{f_2} \cap g$ и $F_{\bar f_2}(i) = 1$, $i \in \overline{f_2} \setminus g$. В строке $row(w_2, f_2, F_{\bar f_2})$ матрицы $Matr(w_2)$ рассмотрим элементы, находящиеся на пересечение со столбцами $\{col(w_2, F_{N(s_2)}) \mid F_{N(s_2)} = F_g(1,1,...,1)_{\bar g}$, где $F_g \in E_g\}$. Можно увидеть, что матрицы $Matr(w_1; (f_1, F_{\bar f_1}); F_{N(s_1)} \in E_{N(s_1)})$ и $Matr(w_2; (f_2, F_{\bar f_2}); F_g(1,1,...,1)_{\bar g}, F_g \in E_g)$ размера $1 \times |E_{N(s_1)}|$ совпадают с точностью до перестановки столбцов, которая определяется функцией π. Отсюда матрица $Matr(w_1)$ является подматрицей $Matr(w_2)$.

Следовательно, если существует задача $w_1 \in W(M_1)$, что матрица $Matr(w_1)$ не является абсолютно унимодулярной (т.е. содержит минор отличный от $0, 1, -1$), то существует задача $w_2 \in W(M_2)$ такая, что $Matr(w_2)$ также содержит минор отличный от $0, 1, -1$. *Лемма доказана.*

Лемма 4.3. Пусть $M \subseteq 2^{N(s)}$. Если выполняется одно из следующих трех условий:

1. $s = 3$, $M = \{\{1,2\},\{1,3\},\{2,3\}\}$;
2. $s = 3$, $M = \{\{1\},\{2\},\{3\}\}$;
3. $s = 4$, $M = \{\{1,2\},\{2,3\},\{1,4\}\}$;

то существует $w \in W(M)$, что матрица $Matr(w)$ не является абсолютно унимодулярной.

Доказательство. 1. Пусть $s = 3$, $M = \{\{1,2\},\{1,3\},\{2,3\}\}$. По определению $n_1, n_2, n_3 \ge 2$, тогда выберем произвольную задачу $w \in W(M)$ и рассмотрим ее подматрицу

$$H = Matr(w;$$
$$(\{1,2\},(1)_{\{3\}}),(\{1,3\},(1)_{\{2\}}),(\{2,3\},(1)_{\{1\}});$$

$$(1,1,2)_{N(3)},(1,2,1)_{N(3)},(2,1,1)_{N(3)}).$$

Можно увидеть, что $H=\begin{pmatrix}0&1&1\\1&0&1\\1&1&0\end{pmatrix}$ и $\det H=2$.

2. Пусть $s=3$, $M=\{\{1\},\{2\},\{3\}\}$. Рассмотрим произвольную задачу $w\in W(M)$ такую, что $n_1,n_2,n_3\geq 3$ и рассмотрим ее подматрицу

$$H=\ Matr(w;$$
$$(\{1\},(1,3)_{\{2,3\}}),(\{1\},(2,2)_{\{2,3\}}),(\{1\},(2,3)_{\{2,3\}}),(\{3\},(1,1)_{\{1,2\}}),(\{3\},(2,2)_{\{1,2\}}),$$
$$(\{2\},(1,2)_{\{1,3\}}),(\{2\},(3,3)_{\{1,3\}});$$
$$(1,1,2)_{N(3)},(1,1,3)_{N(3)},(1,2,2)_{N(3)},(2,2,2)_{N(3)},(2,2,3)_{N(3)},(3,1,3)_{N(3)},(3,2,3)_{N(3)}).$$

Можно увидеть, что $H=\begin{pmatrix}0&1&0&0&0&1&0\\0&0&1&1&0&0&0\\0&0&0&0&1&0&1\\1&1&0&0&0&0&0\\0&0&0&1&1&0&0\\1&0&1&0&0&0&0\\0&0&0&0&0&1&1\end{pmatrix}$ и $\det H=2$.

3. Пусть $s=4$, $M=\{\{1,2\},\{2,3\},\{1,4\}\}$. По определению $n_1,n_2,n_3\geq 2$, тогда выберем произвольную задачу $w\in W(M)$ и рассмотрим ее подматрицу

$$H=\ Matr(w;$$
$$(\{1,2\},(1,1)_{\{3,4\}}),(\{2,3\},(1,1)_{\{1,4\}}),(\{2,3\},(1,2)_{\{1,4\}}),(\{1,4\},(1,1)_{\{2,3\}}),(\{1,4\},(1,2)_{\{2,3\}});$$
$$(1,1,1,2)_{N(4)},(1,1,2,1)_{N(4)},(1,1,2,2)_{N(4)},(1,2,1,1)_{N(4)},(2,1,1,1)_{N(4)}).$$

Можно увидеть, что $H=\begin{pmatrix}0&0&0&1&1\\0&1&0&1&0\\1&0&1&0&0\\1&0&0&0&1\\0&1&1&0&0\end{pmatrix}$ и $\det H=-2$. *Лемма доказана.*

Замечание. Матрицы, используемые при доказательстве леммы 4.3, были получены при помощи параллельной программной системы, выполнявшейся на супер-ЭВМ вычислительного центра коллективного пользования ФГУП «РФЯЦ-ВНИИЭФ» [25].

Лемма 4.4. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы множество M было 1-вложенным, необходимо и достаточно, чтобы для любых $f_1, f_2 \in M$ существовали $k, l \in \{1,2\}$, $k \neq l$, что $f_k \subseteq f_l$.

Доказательство. Доказательство необходимости. Пусть множество M является 1-вложенным, тогда, согласно 4.2, множество M может быть представлено следующим образом $M = \{f_1, ..., f_m\}$, где $f_j \subseteq f_{j+1}$, $j = \overline{1, m-1}$. Рассмотрим произвольные $j_1, j_2 \in \{1,2,...,m\}$. Если $j_1 \leq j_2$, то $f_{j_1} \subseteq f_{j_2}$, иначе $f_{j_2} \subseteq f_{j_1}$.

Доказательство достаточности. Пусть для любых $f_1, f_2 \in M$ существуют $k, l \in \{1,2\}$, $k \neq l$, что $f_k \subseteq f_l$. Упорядочим элементы множества M по неубыванию их мощности, $M = \{f_{t_1}, f_{t_2}, ..., f_{t_{|M|}}\}$, $|f_{t_j}| \leq |f_{t_{j+1}}|$, $j = \overline{1, |M|-1}$. Т.к. $f_{t_j} \neq f_{t_{j+1}}$ и $|f_{t_j}| \leq |f_{t_{j+1}}|$, то $f_{t_{j+1}} \not\subset f_{t_j}$, тогда по условию $f_{t_j} \subseteq f_{t_{j+1}}$, $j = \overline{1, |M|-1}$. Отсюда по определению 4.2 множество M является 1-вложенным. *Лемма доказана.*

Теорема 4.3. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы множество M было 2-вложенным, необходимо и достаточно, чтобы для любых $f_1, f_2, f_3 \in M$ существовали $k, l \in \{1,2,3\}$, $k \neq l$, что $f_k \subseteq f_l$.

Доказательство. Доказательство необходимости. Пусть множество M является 2-вложенным, тогда, согласно определению 4.2, существует разбиение множества M на 2 подмножества $M_1 = \{f_1^{(1)}, ..., f_{m_1}^{(1)}\}$, $M_2 = \{f_1^{(2)}, ..., f_{m_2}^{(2)}\}$, что $f_j^{(i)} \subseteq f_{j+1}^{(i)}$, $j = \overline{1, m_i - 1}$, $i = \overline{1,2}$. Рассмотрим произвольные $f_1, f_2, f_3 \in M$. Существуют $k, l \in \{1,2,3\}$, $k \neq l$ и $t \in \{1,2\}$, что $f_k, f_l \in M_t$. Если $|f_k| \leq |f_l|$, то $f_k \subseteq f_l$, иначе $f_l \subseteq f_k$.

Доказательство достаточности. Пусть для любых $f_1, f_2, f_3 \in M$ существуют $k, l \in \{1,2,3\}$, $k \neq l$, что $f_k \subseteq f_l$. Проведем доказательство конструктивно, построим алгоритм нахождения разбиения $M_1 = \{f_1^{(1)}, ..., f_{m_1}^{(1)}\}$, $M_2 = \{f_1^{(2)}, ..., f_{m_2}^{(2)}\}$ множества M такого, что $f_j^{(i)} \subseteq f_{j+1}^{(i)}$, $j = \overline{1, m_i - 1}$, $i = \overline{1,2}$.

Алгоритм 4.2. Разбиение множества M на два 1-вложенных множества.

Вход. Множество $M \subseteq 2^{N(s)}$, удовлетворяющее условию теоремы.

Шаг 1. Упорядочим элементы множества M по неубыванию их мощности, $M = \{g_1, g_2, ..., g_{|M|}\}$, $|g_j| \le |g_{j+1}|$, $j = \overline{1, |M|-1}$. Далее пусть $M_1, M_2 := \varnothing$, $j := 1$. Переход на шаг 2.

Шаг 2. Упорядочим элементы множеств M_1, M_2 по неубыванию их мощности, $M_1 = \{f_1^{(1)}, ..., f_{m_1}^{(1)}\}$, $M_2 = \{f_1^{(2)}, ..., f_{m_2}^{(2)}\}$, где $|f_l^{(i)}| \le |f_{l+1}^{(i)}|$, $l = \overline{1, m_i - 1}$, $i = \overline{1,2}$. Если $j > |M|$, то алгоритм завершен; иначе переход на шаг 3.

Шаг 3. Если существует $l \in \{1,2\}$, что $f_{m_l}^{(l)} \subseteq g_j$, то $M_l := M_l \cup \{g_j\}$, $j := j+1$, переход на шаг 2; иначе переход на шаг 4.

Шаг 4. Обозначим $I_l = \{f_i^{(l)} \mid f_i^l \not\subset g_j, i \in \{1, ..., m_l\}\}$, $i_l^* = \min\limits_{i | f_i^{(l)} \in I_l} i$, $l = \overline{1,2}$. Рассмотрим множество $I = I_1 \cup I_2$ и упорядочим его элементы по неубываню их мощности $I = \{q_1, q_2, ..., q_{|I|}\}$, $|q_s| \le |q_{s+1}|$, $s = \overline{1, |I|-1}$. Пусть $t = \begin{cases} 1, \text{если } q_1 \in I_1 \\ 2, \text{если } q_1 \in I_2 \end{cases}$, $r = \begin{cases} 1, \text{если } q_1 \notin I_1 \\ 2, \text{если } q_1 \notin I_2 \end{cases}$. Далее $M_t := M_t \cup I_r$, $M_r := M_r \setminus I_r \cup \{g_j\}$, $j := j+1$, переход на шаг 2.

Приведем доказательство корректности построенного алгоритма. По построению к началу выполнения каждого из очередных шагов 2 справедливо $M_1 \cap M_2 = \varnothing$ и очередной элемент g_j добавляется в M_1 либо в M_2. Таким образом, после завершения работы алгоритма M_1, M_2 является разбиением множества M. Далее покажем, что перед каждым из шагов 2 множества M_1, M_2 являются 1-вложенными.

Изначально $M_1, M_2 = \varnothing$ и, согласно определению 4.2, являются 1-вложенными. Пусть $M_l = \{f_1^{(l)}, ..., f_{m_l}^{(l)}\}$, $f_i^{(l)} \subseteq f_{i+1}^{(l)}$, $i = \overline{1, m_l - 1}$, $l = \overline{1,2}$. Если на шаге 3 найдется $l \in \{1,2\}$, что $f_{m_l}^{(l)} \subseteq g_j$, то $M_l := M_l \cup \{g_j\}$ и, т.к. $f_i^{(l)} \subseteq f_{i+1}^{(l)}$, $i = \overline{1, m_l - 1}$, то множество M_l будет являться 1-вложенным. В случае перехода на шаг 4 справедливо $f_{m_1}^{(1)}, f_{m_2}^{(2)} \not\subset g_j$, тогда $f_{m_1}^{(1)} \in I_1$, $f_{m_2}^{(2)} \in I_2$, следовательно, $I_1, I_2 \ne \varnothing$ и существуют величины i_1^*, i_2^*. Т.к. множества

M_1, M_2 являются 1-вложенными, то $I_1 = \{f_{i_1^*}^{(1)},..., f_{m_1}^{(1)}\}$, $I_2 = \{f_{i_2^*}^{(2)},..., f_{m_2}^{(2)}\}$. Схематично множества M_1 и M_2, как подмножества $2^{N(s)}$, можно отобразить следующим образом:

Рисунок 4.2.

По построению $q \not\subset g_j$, $q \in I$. С другой стороны, согласно построенному алгоритму, $|g_j| \geq q$, $q \in I$, отсюда $g_j \not\subset q$, $q \in I$. Рассмотрим произвольные $q' \in I_1$, $q'' \in I_2$. Как уже было показано $g_j \not\subset q', q''$ и $q', q'' \not\subset g_j$, тогда, согласно условию леммы, выполняется одно из следующих соотношений: $q' \subseteq q''$ или $q'' \subseteq q'$. Отсюда по лемме 4.4. множество I является 1-вложенным и $q_s \subseteq q_{s+1}$, $s = \overline{1, |I|-1}$. По построению $q_1 \in I_t$. Далее либо $M_t \setminus I_t = \varnothing$, либо $M_t \setminus I_t = \{f_1^{(t)},..., f_{i_t^*-1}^{(t)}\}$ и следовательно $f_{i_t^*-1}^{(t)} \subseteq f_{i_t^*}^{(t)} = q_1$. При этом $I \cup M_t \setminus I_t = M_t \cup I_r$. Отсюда построенное множество $M_t := M_t \cup I_r$ является 1-вложенным. Далее либо $M_r \setminus I_r = \varnothing$, либо $M_r \setminus I_r = \{f_1^{(r)},..., f_{i_r^*-1}^{(r)}\}$ и по построению $f_{i_r^*-1}^{(r)} \subseteq g_j$. Отсюда построенное множество $M_r := M_r \setminus I_r \cup \{g_j\}$ является 1-вложенным. Схематично построенные на шаге 4 множества M_1 и M_2, как подмножества $2^{N(s)}$, можно отобразить следующим образом:

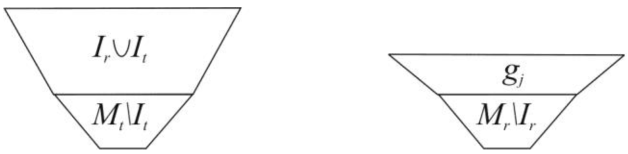

Рисунок 4.3.

Следовательно, после завершения работы алгоритма построенные множества M_1, M_2 представляют собой разбиение множества M и являются 1-вложенными. *Теорема доказана.*

Следствие 4.2. Пусть $M \subseteq 2^{N(s)}$. Если $|M| \geq 2s+1$, то множество M не является 2-вложенным. Если $|M| \geq s+2$, то множество M не является 1-вложенным.

Доказательство. Пусть $|M| \geq 2s+1$. Т.к. $M \subseteq 2^{N(s)}$ и $N(s)=\{1,...,s\}$, то найдутся три различных множества $f_1, f_2, f_3 \in M$, что $|f_1|=|f_2|=|f_3|$. Тогда $f_k \not\subset f_l$ для любых $k,l \in \{1,2,3\}$, $k \neq l$. Отсюда по теореме 4.3. множество M не является 2-вложенным.

Пусть $|M| \geq s+2$. Т.к. $M \subseteq 2^{N(s)}$ и $N(s)=\{1,...,s\}$, то найдутся два различных множества $f_1, f_2 \in M$, что $|f_1|=|f_2|$. Тогда $f_k \not\subset f_l$ для любых $k,l \in \{1,2\}$, $k \neq l$. Отсюда по лемме 4.4 множество M не является 2-вложенным. *Следствие доказано.*

Заметим, что критерий 2-вложенности множества M, сформулированный в теореме 4.3, и следствие 4.2 позволяют предложить следующий алгоритм проверки два вложенности множество M.

Алгоритм 4.3. Проверка 2-вложенности множества M.

Вход. Множество $M \subseteq 2^{N(s)}$.

Шаг 1. Если $|M| \geq 2s+1$, то множество M не 2-вложенное, алгоритм завершен; иначе переход на шаг 2.

Шаг 2. Если для каждой тройки $f_1, f_2, f_3 \in M$ выполняется условие теоремы 4.3, то множество M является 2-вложенным; иначе множество M не является 2-вложенным. Алгоритм завершен.

Утверждение 4.2. Алгоритм 4.3 проверки 2-вложенности множества M требует $O(s^4)$ вычислительных операций.

Доказательство. Шаг 1 алгоритма связан с подсчетом числа элементов множества M, причем если $|M| \geq 2s+1$, то алгоритм завершается. Таким образом, шаг 1 требует $O(s)$ вычислительных операций. Шаг 2 алгоритма связан с перебором всех троек $f_1, f_2, f_3 \in M$, причем, на данном шаге $|M| \leq 2s$. Отсюда общее число таких троек не превышает $8s^3$. Для каждой тройки множеств f_1, f_2, f_3 проверяется вложенность каждого из множеств в остальные, всего 6 проверок. В силу того, что $M \subseteq 2^{N(s)}$, выполняется: $|f| \leq s$ для каждого $f \in M$. Тогда проверка вложенности одного множества может быть осуществлена, используя $O(s)$ вычислительных операций. Таким образом, шаг 2 требует $O(s^4)$ вычислительных операций. *Утверждение доказано.*

Проведем анализ сложности алгоритма разбиения 2-вложенного множества M на два 1-вложенных множества, используемого при доказательстве теоремы 4.3.

Утверждение 4.3. Алгоритм 4.2 разбиения 2-вложенного множества M на два 1-вложенных множества требует $O(s^3)$ вычислительных операций.

Доказательство. Согласно следствию 4.2**,** т.к. множество M является 2-вложенным, то $|M| \le 2s$. Напомним, что $M \subseteq 2^{N(s)}$, и следовательно выполняется: $|f| \le s$ для каждого $f \in M$. Тогда проверка вложенности пар множеств из M может быть осуществлено, используя $O(s)$ вычислительных операций.

Шаг 1 алгоритма связан с сортировкой множества M по неубыванию мощности множеств, входящих в M. Данная сортировка может быть осуществлена, используя $O(s^2)$ вычислительных операций (более точная оценка не изменит итоговую оценку алгоритма). Далее алгоритм состоит из $|M| = O(s)$ повторений шагов 2, 3, 4. Шаг 2 связан с сортировкой множеств M_1, M_2 по неубыванию мощности множеств, входящих в M_1, M_2. Выполнение данной сортировки на шаге 2 можно избежать, если выполнять необходимую сортировку при модификации множеств M_1, M_2 на шагах 3 и 4 (сортировка была указана на шаге 2 лишь для краткости изложения доказательства теоремы 4.3). Шаг 3 связан с проверкой условий $f_{m_1}^{(1)} \subseteq g_j$, $f_{m_2}^{(2)} \subseteq g_j$ и следовательно, требует $O(s)$ вычислительных операций. Отметим, что при выполнении шага 4 справедливо соотношение $|M_1| + |M_2| = j$. Тогда построение множеств I_1, I_2 требует $O(js)$ вычислительных операций. Т.к. $I \subseteq M_1 \cup M_2$, то $|I| \le j$, следовательно, его построение и упорядочивание требует $O(j^2)$ вычислительных операций. Определение t, r может быть осуществлено, используя $O(1)$ вычислительных операций. Далее модификация множеств M_1, M_2 может быть осуществлена, используя $O(j)$ вычислительных операций. Таким образом, выполнение алгоритма 4.2 требует $O(s^2)$ + $O(s)O(s)$ + $\sum_{j=1}^{2s}(O(s) + O(js) + O(j^2))$ = $O(s^3)$ вычислительных операций. *Утверждение доказано.*

Далее рассмотрим произвольные множества $f_1, f_2, f_3 \subseteq N(s)$, для которых выполняется следующее условие:

$$f_1 \not\subseteq f_2 f_3, \qquad f_2 \not\subseteq f_1 f_3, \qquad f_3 \not\subseteq f_1 f_2. \tag{4.5}$$

Условие (4.5) выполняется тогда и только тогда, когда существуют такие элементы $a_{ij} \in N(s)$, $j \in \{1,2,3\} \setminus \{i\}$, $i \in \{1,2,3\}$, что

$$a_{12} \in f_1, a_{12} \notin f_2, \quad a_{21} \in f_2, a_{21} \notin f_1, \quad a_{31} \in f_3, a_{31} \notin f_1,$$

$$a_{13} \in f_1, a_{13} \notin f_3, \quad a_{23} \in f_2, a_{23} \notin f_3, \quad a_{32} \in f_3, a_{32} \notin f_2.$$

Множество $A = \{a_{ij} \mid j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}\}$ будем называть множеством, разделяющим f_1, f_2, f_3. Обозначим через $A(f_1, f_2, f_3)$ множество всех разделяющих f_1, f_2, f_3 множеств. Из теоремы 4.3, используя понятие разделяющего множества, можно сформулировать следующее следствие.

Следствие 4.3. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы множество M было 2-вложенным, необходимо и достаточно, чтобы $A(f_1, f_2, f_3) = \varnothing$ для любых $f_1, f_2, f_3 \in M$.

Пусть $A \in A(f_1, f_2, f_3)$, тогда $A = \{a_{ij} \mid j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}\}$ и $a_{ij} \in f_i$, $a_{ij} \notin f_j$, $j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}$. Обозначим через $p(A)$ следующую величину

$$p(A) = |\{a_{12}\} \cup \{a_{13}\}| + |\{a_{21}\} \cup \{a_{23}\}| + |\{a_{31}\} \cup \{a_{32}\}|.$$

Пусть $d^*(f_1, f_2, f_3) = \min\limits_{A \in A(f_1, f_2, f_3)} |A|$. Тогда рассмотрим задачу выбора среди множеств, разделяющих f_1, f_2, f_3, множества мощности $d^*(f_1, f_2, f_3)$ с максимальным значением величины $p(A)$:

$$A^*(f_1, f_2, f_3) = \operatorname*{arg\,max}_{A \in A(f_1, f_2, f_3) | d^*(f_1, f_2, f_3) = |A|} p(A). \tag{4.6}$$

Решение $A^*(f_1, f_2, f_3)$ задачи (4.6) обладает следующим важным свойством.

Лемма 4.5. Пусть $f_1, f_2, f_3 \subseteq N(s)$, $A(f_1, f_2, f_3) \neq \varnothing$ и $A^*(f_1, f_2, f_3) = \{a_{ij}^* \mid j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}\}$, где $a_{ij}^* \in f_i$, $a_{ij}^* \notin f_j$, $j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}$. Если $a_{ij}^* \in f_k$, то $a_{ij}^* = a_{kj}^*$, где $i, j, k \in \{1,2,3\}$, $i \neq j$, $i \neq k$, $j \neq k$.

Доказательство от противного. Предположим, что выполняются условия леммы, и найдутся такие $i, j, k \in \{1,2,3\}$, $i \neq j$, $i \neq k$, $j \neq k$, что $a_{ij}^* \in f_k$, но $a_{ij}^* \neq a_{kj}^*$. Рассмотрим множество $A' = \{a'_{st} \mid s \in \{1,2,3\} \setminus \{t\}, t \in \{1,2,3\}\}$, построенное следующим образом

$$a'_{ij} = a_{ij}^*, \quad a'_{ji} = a_{ji}^*, \quad a'_{ki} = a_{ki}^*,$$

$$a'_{ik} = a_{ik}^*, \quad a'_{jk} = a_{jk}^*, \quad a'_{kj} = a_{ij}^*.$$

По предположению $a'_{kj} = a^*_{ij} \in f_k$, при этом по построению $a'_{kj} = a^*_{ij} \notin f_j$. Отсюда $a'_{st} \in f_s, a'_{st} \notin f_t, t \in \{1,2,3\} \setminus \{s\}, s \in \{1,2,3\}$ и $A' \in A(f_1, f_2, f_3)$. Далее проведем доказательство, рассмотрев следующие два возможных случая:

1. Пусть $a^*_{kj} = a^*_{ki}$. Т.к. $a'_{ki} = a^*_{ki} = a^*_{kj}$, $a'_{kj} = a'_{ij} = a^*_{ij}$, то $|A'| =$

$$\begin{aligned}
&|\{a'_{ij}\} \cup \{a'_{ik}\} \cup \{a'_{ji}\} \cup \{a'_{jk}\} \cup \{a'_{ki}\} \cup \{a'_{kj}\}| = \\
&|\{a'_{ij}\} \cup \{a'_{ik}\} \cup \{a'_{ji}\} \cup \{a'_{jk}\} \cup \{a'_{ki}\}| = \\
&|\{a^*_{ij}\} \cup \{a^*_{ik}\} \cup \{a^*_{ji}\} \cup \{a^*_{jk}\} \cup \{a^*_{ki}\}| = \\
&|\{a^*_{ij}\} \cup \{a^*_{ik}\} \cup \{a^*_{ji}\} \cup \{a^*_{jk}\} \cup \{a^*_{ki}\} \cup \{a^*_{kj}\}| = |A^*(f_1, f_2, f_3)|.
\end{aligned}$$

По построению $a'_{kj} = a^*_{ij} \in f_i$ и $a'_{ki} = a^*_{ki} \notin f_i$, таким образом, $a'_{kj} \neq a'_{ki}$. По условию $a^*_{kj} = a^*_{ki}$. Отсюда $p(A') =$

$$|\{a'_{ij}\} \cup \{a'_{ik}\}| + |\{a'_{ji}\} \cup \{a'_{jk}\}| + |\{a'_{ki}\} \cup \{a'_{kj}\}| =$$

$$|\{a^*_{ij}\} \cup \{a^*_{ik}\}| + |\{a^*_{ji}\} \cup \{a^*_{jk}\}| + 2$$

и $p(A^*(f_1, f_2, f_3)) =$

$$\begin{aligned}
&|\{a^*_{ij}\} \cup \{a^*_{ik}\}| + |\{a^*_{ji}\} \cup \{a^*_{jk}\}| + |\{a^*_{ki}\} \cup \{a^*_{kj}\}| = \\
&|\{a^*_{ij}\} \cup \{a^*_{ik}\}| + |\{a^*_{ji}\} \cup \{a^*_{jk}\}| + 1.
\end{aligned}$$

Следовательно, $p(A') = p(A^*(f_1, f_2, f_3)) + 1$. Получаем противоречие.

2. Пусть $a^*_{kj} \neq a^*_{ki}$. По предположению $a^*_{kj} \neq a^*_{ij}$. По построению $a^*_{kj} \in f_k$ и $a^*_{ik} \notin f_k$, тогда $a^*_{kj} \neq a^*_{ik}$. Далее по построению $a^*_{kj} \notin f_j$, $a^*_{ji}, a^*_{jk} \in f_j$, тогда $a^*_{kj} \neq a^*_{ji}, a^*_{jk}$. Отсюда $|A'| =$

$$\begin{aligned}
&|\{a'_{ij}\} \cup \{a'_{ik}\} \cup \{a'_{ji}\} \cup \{a'_{jk}\} \cup \{a'_{ki}\} \cup \{a'_{kj}\}| = \\
&|\{a'_{ij}\} \cup \{a'_{ik}\} \cup \{a'_{ji}\} \cup \{a'_{jk}\} \cup \{a'_{ki}\}| = \\
&|\{a^*_{ij}\} \cup \{a^*_{ik}\} \cup \{a^*_{ji}\} \cup \{a^*_{jk}\} \cup \{a^*_{ki}\}| = \\
&= |A^*(f_1, f_2, f_3) \setminus \{a^*_{kj}\}| = |A^*(f_1, f_2, f_3)| - 1.
\end{aligned}$$

Получаем противоречие. Таким образом, предположение неверно. *Лемма доказана.*

Лемма 4.6. Пусть $M \subseteq 2^{N(s)}$. Если множество M не является 2-вложенным, то выполняется одно из следующих трех условий:

– $\{\{1,2\},\{1,3\},\{2,3\}\} \prec M$,

– $\{\{1\},\{2\},\{3\}\} \prec M$,

– $\{\{1,2\},\{2,3\},\{1,4\}\} \prec M$.

Доказательство. Пусть множество $M \subseteq 2^{N(s)}$ не является 2-вложенным. Тогда, согласно следствию 4.3, найдутся такие $f_1, f_2, f_3 \in M$, что $A(f_1, f_2, f_3) \neq \varnothing$. Рассмотрим множество $A^*(f_1, f_2, f_3) = \{a^*_{ij} \mid j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}\}$, где $a^*_{ij} \in f_i$, $a^*_{ij} \notin f_j$, $j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}$, являющееся решением задачи (4.6). Схематично будем представлять возможные комбинации множества $A^*(f_1, f_2, f_3)$ в виде графа G_f с множеством вершин $V = \{ij \mid j \in \{1,2,3\} \setminus \{i\}, i \in \{1,2,3\}\}$ и множеством дуг $\{(ij,kl) \mid a^*_{ij} = a^*_{kl}, ij \neq kl, ij, kl \in V\}$. Проведем доказательство, рассмотрев следующее возможные два случая.

1. Пусть для каждого $i \in \{1,2,3\}$ существует $t_i \in \{1,2,3\} \setminus \{i\}$, что $a^*_{it_i} \notin f_{k_i}$, где $k_i \in \{1,2,3\} \setminus \{i, t_i\}$. Рассмотрим элементы $a^*_{1t_1}, a^*_{2t_2}, a^*_{3t_3}$. По построению

$$a^*_{1t_1} \notin f_2, f_3, \quad a^*_{2t_2} \notin f_1, f_3, \quad a^*_{3t_3} \notin f_1, f_2.$$

И следовательно $a^*_{1t_1} \neq a^*_{2t_2}, a^*_{3t_3}$, $a^*_{2t_2} \neq a^*_{1t_1}, a^*_{3t_3}$, $a^*_{3t_3} \neq a^*_{1t_1}, a^*_{2t_2}$. Тогда подграф графа G_f, индуцированный подмножеством вершин $\{1t_1, 2t_2, 3t_3\}$, будет иметь вид, представленный на рис. 4.4.

Рисунок 4.4.

Отсюда $\{\{a^*_{1t_1}\},\{a^*_{2t_2}\},\{a^*_{3t_3}\}\} \subseteq M(\{a^*_{1t_1}, a^*_{2t_2}, a^*_{3t_3}\})$ и, по определению 4.4, $\{\{1\},\{2\},\{3\}\} \prec M$.

2. Пусть существует $i \in \{1,2,3\}$, что для каждого $j \in \{1,2,3\} \setminus \{i\}$, выполняется $a^*_{ij} \in f_k$, где $k \in \{1,2,3\} \setminus \{i, j\}$. Не уменьшая общности, будем считать, что $i = 1$, в противном случае перенумеруем элементы. Тогда $a^*_{12} \in f_3$, $a^*_{13} \in f_2$. Согласно лемме 4.5, $a^*_{12} = a^*_{32}$, $a^*_{13} = a^*_{23}$. По построению $a^*_{kl} \in f_k$, $a^*_{kl} \notin f_l$, $l \in \{1,2,3\} \setminus \{k\}$, $k \in \{1,2,3\}$. Следовательно, $a^*_{kl} \neq a^*_{lm}$, $m, k \in \{1,2,3\} \setminus \{l\}$, $l \in \{1,2,3\}$. По построению $a^*_{12} \in f_3$, $a^*_{13} \notin f_3$, отсюда $a^*_{12} \neq a^*_{13}$. Также по построению $a^*_{23} = a^*_{13} \in f_1$, $a^*_{21} \notin f_1$, отсюда $a^*_{21} \neq a^*_{23}$;

$a^*_{32} = a^*_{12} \in f_1$, $a^*_{31} \notin f_1$, отсюда $a^*_{31} \neq a^*_{32}$. Далее возможен один из двух подслучаев: $a^*_{21} = a^*_{31}$ или $a^*_{21} \neq a^*_{31}$.

2.1. Пусть $a^*_{21} = a^*_{31}$. Тогда граф G_f будет иметь вид, представленный на рис. 4.5.

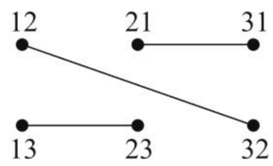

Рисунок 4.5.

Отсюда $\{\{a^*_{12}, a^*_{13}\}, \{a^*_{13}, a^*_{21}\}, \{a^*_{12}, a^*_{21}\}\} \subseteq M(\{a^*_{12}, a^*_{13}, a^*_{21}\})$ и, по определению 4.4, $\{\{1,2\},\{1,3\},\{2,3\}\} \prec M$.

2.2. Пусть $a^*_{21} \neq a^*_{31}$. Тогда граф G_f будет иметь вид, представленный на рис. 4.6.

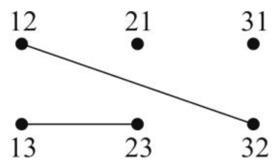

Рисунок 4.6.

Отсюда $\{\{a^*_{12}, a^*_{13}\}, \{a^*_{13}, a^*_{21}\}, \{a^*_{12}, a^*_{31}\}\} \subseteq M(\{a^*_{12}, a^*_{13}, a^*_{21}, a^*_{31}\})$ и, по определению 4.4, $\{\{1,2\},\{2,3\},\{1,4\}\} \prec M$. *Лемма доказана.*

Теорема 4.4. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}, необходимо, чтобы множество M было 2-вложенным.

Доказательство. Проведем доказательство от противного. Пусть класс $W(M)$ является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Graph}, однако множество M не является 2-вложенным. Согласно лемме 4.6, выполняется одно из следующих трех условий:

– $\{\{1,2\},\{1,3\},\{2,3\}\} \prec M$;

– $\{\{1\},\{2\},\{3\}\} \prec M$;

– $\{\{1,2\},\{2,3\},\{1,4\}\} \prec M$.

Тогда, согласно леммам 4.2, 4.3**,** найдется $w \in W(M)$, что $Matr(w)$ не является абсолютно унимодулярной. С другой стороны, из теоремы 4.1 следует, что матрица $Matr(w)$ является абсолютно унимодулярной. Получаем противоречие, таким образом, предположение неверно. *Теорема доказана.*

Согласно теоремам 4.2 и 4.4, найденное условие 2-вложенности является необходимыми и достаточным условием сводимости (согласно введенной концепции сведения) многоиндексных задач к задаче поиска потока минимальной стоимости. Более того, оказывается, что предложенная при конструктивном доказательстве теоремы 4.2 схема сведения класса многоиндексных задач $W(M)$ с 2-вложенным множеством M, требующая линейных вычислительных затрат, является (в рамках рассматриваемой концепции $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости к классу W_{Graph}) оптимальной в том смысле, что

– сведение с использованием вычислительных затрат асимптотически меньших линейных невозможно в связи с необходимостью считывания входных данных задачи (асимптотически меньшая оценка понимается в смысле оценки $o()$, «о малое», принятой при анализе сложности алгоритмов [179]);
– сколь угодное увеличение вычислительных затрат на сведение не приводится к расширению класса многоиндексных задач, сводимых к классу W_{Graph}.

Таким образом, в совокупности теоремы 4.2 и 4.4 представляют собой исчерпывающий результат исследования $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости классов многоиндексных задач $W(M)$ к классу задач поиска потока минимальной стоимости W_{Graph}.

Приведем пример многоиндексной транспортной задачи с 2-вложенной структурой, на котором проиллюстрируем схему сведения, предложенную при доказательстве теоремы 4.2.

$$a_{j_3}^- \le \sum_{j_1 \in J_1} \sum_{j_2 \in J_2} x_{j_1 j_2 j_3} \le a_{j_3}^+ , \; j_3 \in J_3 , \tag{4.7}$$

$$b_{j_1}^- \le \sum_{j_2 \in J_2} \sum_{j_3 \in J_3} x_{j_1 j_2 j_3} \le b_{j_1}^+ , \; j_1 \in J_1 , \tag{4.8}$$

$$c_{j_2 j_3}^- \le \sum_{j_1 \in J_1} x_{j_1 j_2 j_3} \le c_{j_2 j_3}^+ , \; j_2 \in J_2 , \; j_3 \in J_3 , \tag{4.9}$$

$$x_{j_1 j_2 j_3} \ge 0 , \; j_1 \in J_1 , \; j_2 \in J_2 , \; j_3 \in J_3 , \tag{4.10}$$

$$\sum_{j_1\in J_1}\sum_{j_2\in J_2}\sum_{j_3\in J_3} d_{j_1j_2j_3}x_{j_1j_2j_3} \to \min . \qquad (4.11)$$

Задача (4.7)-(4.11) относится к классу многоиндексных задач $W(M)$, где $M=\{\{1\},\{1,2\},\{2,3\}\}$ и $s=3$. Множество M является 2-вложенным, т.к. существует разбиение $M_1=\{\{1\},\{1,2\}\}$, $M_2=\{\{2,3\}\}$, удовлетворяющее определению 4.2. Отсюда, согласно теореме 4.2, класс $W(M)$ является $L\,|\,L-equal\,|\,L-edge$ сводимым к классу W_{Graph}. Далее рассмотрим схему сведения, используемой при конструктивном доказательстве теоремы 4.2, на примере задачи (4.7)-(4.11). Пусть $|J_1|=|J_2|=|J_3|=2$. Приведем транспортную сеть, определяющую задачу поиска потока минимальной стоимости, соответствующую исходной многоиндексной задаче (см. рисунок 4.7). На рисунке 4.7 для ряда дуг приведены их пропускные способности и/или стоимости. Дуги, у которых не указаны пропускные способностей, имеют нулевую нижнюю и неограниченную верхнюю пропускные способности. Дуги, у которых не указаны стоимости, имеют нулевую стоимость. Согласно доказательству теоремы 4.2 и алгоритму 4.1, решение исходной многоиндексной задачи определяется следующим образом: каждой переменной $x_{i_1i_2i_3}$ многоиндексной задачи (4.7)-(4.11) присваивается значение потока вдоль дуги $(v_{(j_2,j_3)_{\{2,3\}}}, v_{(j_1,j_2,j_3)_{\{1,2,3\}}})$, который в свою очередь определяется как поток минимальной стоимости данной сети.

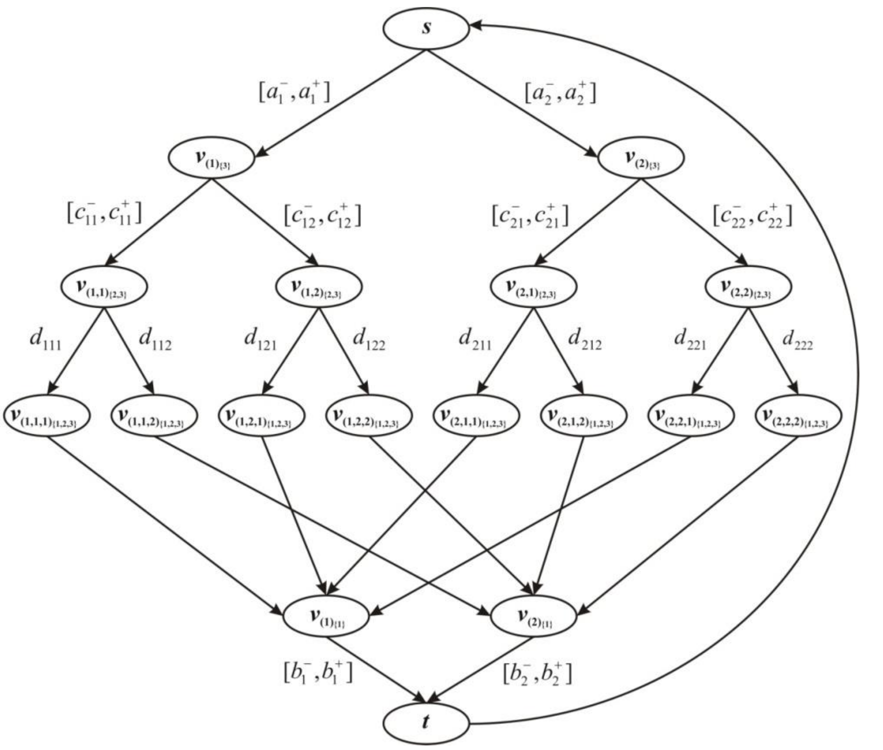

Рисунок 4.7.

Далее проиллюстрируем применимость метода исследования 2-вложенных многоиндексных задач класса $W(M)$, основанного на сводимости к классу задач поиска потока в сети, при решении ряда рассмотренных ранее многоиндексных задач. Пусть $M \subseteq 2^{N(s)}$ является 2-вложенным. Согласно теореме 4.2, класс задач $W(M)$ сводим к поиску потока в сети с $O(|E_{N(s)}|)$ вершинами и $O(|E_{N(s)}|)$ дугами. В рамках исследуемой схемы сведения пропускные способности дуг вспомогательной сети моделируют двусторонние ограничения исходной многоиндексной задачи. В случае несовместности многоиндексной задачи, соответствующая потоковая задача также является несовместной. Тогда можно построить алгоритм решения задач класса $S(M)$, основанный на сводимости к поиску потока в сети по схеме, предложенной при доказательстве теоремы 4.2**,** и решении соответствующей задачи поиска потока в несовместной сети при помощи алгоритма 2.5. Применим для поиска потока минимальной стоимости заданной величины алгоритм, предложенный в работе Орлина [172], для поиска максимального потока воспользуемся алгоритмом Слейтора-Тарьяна [181]. Данные алгоритмы используются в алгоритме 2.2 (на шаге 2 алгоритма 2.5). Тогда, согласно утверждению 2.5, можно сформулировать следующий результат.

Утверждение 4.4. Пусть множество $M \subseteq 2^{N(s)}$ является 2-вложенным, тогда существует алгоритм решения задач класса $S(M)$, требующий $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ вычислительных операций.

Рассмотрим задачу $U_{\prec}(M,H)$, $M,H \subseteq 2^{N(s)}$. Для решения задачи $U_{\prec}(M,H)$ может быть применен алгоритм 2.8, который, согласно утверждению 2.8, требует $O(k \log p)$ проверок совместности системы ограничений вида $\Omega(\vec{v})$. Здесь $k = \sum_{f \in H} |E_f|$ и система ограничений вида $\Omega(\vec{v})$ относится к классу $D(M \cup \widetilde{H})$. Также отметим, что $|H| \le 2^s$, $|E_f| \le |E_{N(s)}|$, отсюда $k = O(2^s |E_{N(s)}|) = O(|E_{N(s)}|)$. Пусть множество $M \cup \widetilde{H}$ является 2-вложенным. Тогда, согласно утверждению 4.1, соответствующая системы вида $\Omega(\vec{v})$ может быть решена, используя $O(|E_{N(s)}|^2 \log |E_{N(s)}|)$ вычислительных операций. Отсюда

Утверждение 4.5. Пусть $M,H \subseteq 2^{N(s)}$ и множество $M \cup \widetilde{H}$ является 2-вложенным, тогда существует алгоритм решения задач класса $U_{\prec}(M,H)$, требующий $O(|E_{N(s)}|^3 \log |E_{N(s)}| \log p)$ вычислительных операций.

Дополнительно рассмотрим задачу $U_{\max\min}(M,H)$, решение которой, согласно утверждению 2.9, связано с решением $O(\log p)$ систем вида $\Omega(\vec{v})$. Таким образом, справедливо утверждение:

Утверждение 4.6. Пусть $M,H \subseteq 2^{N(s)}$ и множество $M \cup \widetilde{H}$ является 2-вложенным, тогда существует алгоритм решения задач класса $U_{\max\min}(M,H)$, требующий $O(|E_{N(s)}|^2 \log|E_{N(s)}| \log p)$ вычислительных операций.

В качестве замечания отметим, что если множество $M \subseteq 2^{N(s)}$ не является 2-вложенным, то возможно отыскание 2-вложенного подмножества $M' \subseteq M$ и применение для решения систем класса $D(M')$ алгоритма 4.1. Полученное решение может быть использовано в качестве начальной точки при решении систем класса $D(M)$ итерационным методом ортогональных проекций, описанным в главе 2.

Дополнительно проиллюстрируем полученные результаты при решении ряда прикладных многоиндексных задач, поставленных в главе 1. Транспортная задача с промежуточными пунктами (1.1)-(1.5),(1.8) относится к классу $W(M)$, где $M=\{\{i\},\{k\},\{i,j\},\{j,k\}\}$. При этом $s=3$ и $|E_{N(s)}|=|I\|J\|K|$. Заметим, что множество M является 2-вложенным, т.к. существует разбиение $M_1=\{\{i\},\{i,j\}\}$, $M_2=\{\{k\},\{j,k\}\}$, удовлетворяющее определению 4.2. Отсюда, согласно утверждению 4.1, транспортная задача с промежуточными пунктами может быть решена, используя $O(|I|^2|J|^2|K|^2\log^2(|I\|J\|K|))$ вычислительных операций. Многокритериальная задача выбора плана перевозок (1.1)-(1.5),(1.9) относится к классу $U(M,H)$, где $H=\{\{k\}\}$. Рассмотрим соответствующие задачи $U_{\prec}(M,H)$ и $U_{\max\min}(M,H)$. Заметим, что множество $M \cup \widetilde{H}=\{\{i\},\{k\},\{i,j\},\{j,k\}\}=M$, как уже было показано, является 2-вложенным. Таким образом, согласно утверждениям 4.5, 4.6, задачи $U_{\prec}(M,H)$ и $U_{\max\min}(M,H)$ могут быть решены, используя $O(|I|^3|J|^3|K|^3\log(|I\|J\|K|)\log p)$ и $O(|I|^2|J|^2|K|^2\log(|I\|J\|K|)\log p)$ вычислительных операций соответственно.

Проблема формирования допустимого портфеля заказов (1.10)-(1.14) относится к классу $D(M)$, где $M=\{\varnothing,\{j\},\{i,t\},\{j,t\}\}$. При этом $s=3$ и $|E_{N(s)}|=|I\|J\|T|$. Множество M является 2-вложенным, т.к. существует разбиение $M_1=\{\varnothing,\{j\},\{j,t\}\}$, $M_2=\{\{i,t\}\}$, удовлетворяющее определению 4.2. Отсюда, согласно утверждению 4.1, проблема определения допустимого портфеля заказов может быть решена, используя $O(|I|^2|J|^2|T|^2 \log(|I||J||T|))$ вычислительных операций. В случае несовместности проблемы формирования допустимого портфеля заказов ставится задача (1.10),(1.11),(1.13)-(1.17) формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам, относящаяся к классу $S(M)$. Тогда, согласно утверждению 4.4, проблема формирования портфеля заказов с возможными нарушениями требуемых объемов работ по заказам может быть решена, используя $O(|I|^2|J|^2|T|^2 \log^2(|I||J||T|))$ вычислительных операций. Рассматривается также задача формирования портфеля заказов с возможными нарушениями сроков выполнения работ (1.10)-(1.12),(1.14),(1.18), относящаяся к классу $W(M')$, где $M' = \{\{j\},\{i,t\},\{j,t\}\}$. Множество M' является 2-вложенным, отсюда, согласно утверждению 4.1, задача формирования портфеля заказов с возможными нарушениями сроков выполнения работ может быть решена, используя $O(|I|^2|J|^2|T|^2 \log^2(|I||J||T|))$ вычислительных операций.

4.3. Многоиндексные задачи с 1-вложенной структурой

Далее будем рассматривать вопросы нахождения условий, которым должно удовлетворять множество *M*, чтобы класс $W(M)$ являлся $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Tree}. Как было определено ранее, класс W_{Tree} представляет собой класс задач поиска циркуляции минимальной стоимости в древовидной сети, здесь под древовидной сетью понимается сеть, представляющая собой корневое ориентированное дерево, дополненное дугами из листьев в корень. Таким образом, $W_{Tree} \subseteq W_{Graph}$. Исследование сводимости к классу W_{Tree} представляет особый интерес, т.к. задачи данного класса эквивалентны задачам поиска потока минимальной стоимости в древовидной сети $z_{MCFT}(G; a_i, b_i, c_i, i \in V)$. Таким образом, согласно утверждениям 2.3, 2.4:

Утверждение 4.7. Существует поиска оптимального (допустимого) решения задачи $z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Tree}$, требующий $O(|V_G|^2)$ $(O(|V_G|))$ вычислительных операций.

Пусть множество $M \subseteq 2^{N(s)}$ является 1-вложенным, согласно определению 4.2, множество M при этом также будет является и 2-вложенным. Тогда, применяя схему сведения, описываемую при доказательстве теоремы 4.2, для задач $w \in W(M)$ будет построена соответствующая задача $z \in W_{Graph}$. Отметим, что если M является 1-вложенным, то при доказательстве теоремы 4.2 можно считать, что $M_1 = M$, $M_2 = \varnothing$. Несложно увидеть, что в данном случае сеть, описывающая задачу z, будет иметь древовидную структуру, и тем самым $z \in W_{Tree}$. Тогда справедливо следующее следствие.

Следствие 4.4. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $L \mid L-equal \mid L-edge$ сводимым к классу W_{Tree}, достаточно, чтобы множество M было 1-вложенным.

Конструктивная схема доказательства теоремы 4.2 в случае 1-вложенности множества M позволяет для задач класса $W(M)$ и для систем линейных неравенств класса $D(M)$ предложить алгоритм решения, основанный на сводимости к классу W_{Tree}.

Алгоритм 4.4. Решение 1-вложенных многоиндексных задач.

Вход. Задача $w \in W(M)$ (система $w \in D(M)$), где M – 1-вложенное.

Шаг 1. Используя конструктивную схему, применяемую при доказательстве теоремы **4.2** (необходимо положить $M_1 = M$, $M_2 = \varnothing$), построить задачу $z \in W_{Tree}$, соответствующую w. Перейти на шаг 2.

Шаг 2. Найти оптимальное решение (допустимое решение) задачи z, используя алгоритм, основанный на решении задачи поиска потока в древовидной сети (см. утверждение 4.7). Если задача z несовместна, то w также несовместна, алгоритм завершен; иначе переход на шаг 3.

Шаг 3. Используя отображение β, описанное при доказательстве теоремы 4.2, найти решение w, согласно определению 4.1. Алгоритм завершен.

По аналогии с алгоритмом 4.1 алгоритм 4.4 применим также при исследовании целочисленного случая. Заметим, что для задачи $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Tree}$, соответствующей задаче w, согласно доказательству теоремы 4.2, выполняется: $|V_G| = O(|E_{N(s)}|)$, $|A_G| = O(|E_{N(s)}|)$. Отсюда, с учетом утверждения 4.7 справедливо следующее утверждение.

Утверждение 4.8. Пусть множество $M \subseteq 2^{N(s)}$ является 1-вложенным, тогда существует алгоритм решения задач класса $W(M)$ и класса $W_Z(M)$ (систем класса $D(M)$ и класса $D_Z(M)$), требующий $O(|E_{N(s)}|^2)$ ($O(|E_{N(s)}|)$) вычислительных операций.

Можно предложить следующий алгоритм проверки 1-вложенности множества M.

Алгоритм 4.5. Проверка 1-вложенности множества M.

Вход. Множество $M \subseteq 2^{N(s)}$.

Шаг 1. Если $|M| \geq s+2$, то множество M не является 1-вложенным, и алгоритм завершен; иначе переход на шаг 2.

Шаг 2. Упорядочим элементы множества M по неубыванию их мощности, $M = \{f_1, ..., f_{|M|}\}$, $|f_j| \leq |f_{j+1}|$, $j = \overline{1, |M|-1}$. Переход на шаг 2.

Шаг 3. Если выполняется условие $f_j \subseteq f_{j+1}$, $j = \overline{1, |M|-1}$, то множество M является 1-вложенным; иначе множество M не является 1-вложенным. Алгоритм завершен.

Утверждение 4.9. Алгоритм 4.5 корректно решает задачу проверки 1-вложенности множества M и требует $O(s^2)$ вычислительных операций.

Доказательство. Если $|M| \geq s+2$ на шаге 1, то, согласно следствию 4.2, множество M не является 1-вложенным. Далее пусть выполнен шаг 2 алгоритма и элементы множества M упорядочены по неубыванию их мощности, $M = \{f_1, ..., f_{|M|}\}$, $|f_j| \leq |f_{j+1}|$, $j = \overline{1, |M|-1}$. Согласно лемме 4.4, для того, чтобы множество M было 1-вложенным, необходимо и достаточно, чтобы для любых $i, j \in \{1, ..., |M|\}$, $i \neq j$, выполнялось $f_i \subseteq f_j$ или $f_j \subseteq f_i$. Пусть $i < j$, тогда по построению $|f_i| \leq |f_j|$. Следовательно, $f_j \not\subset f_i$. Таким образом, для того, чтобы множество M было 1-вложенным, необходимо и достаточно, чтобы для любых $i, j \in \{1, ..., |M|\}$, $i < j$, выполнялось $f_i \subseteq f_j$. Тогда по свойству транзитивности данный критерий может быть записан в следующей эквивалентной постановке. Для того, чтобы множество M было 1-вложенным, необходимо и достаточно, чтобы $f_j \subseteq f_{j+1}$, $j = \overline{1, |M|-1}$. Данное условие проверяется на шаге 3 алгоритма. Следовательно, алгоритм 4.5 корректно решает задачу проверки 1-вложенности множества M.

Шаг 1 алгоритма связан с подсчетом мощности множества M и проверкой условия $|M| \geq s+2$, следовательно, может быть выполнен, используя $O(s)$ вычислительных операций. На шаге 2 алгоритма $|M| \leq s+1$, следовательно, выполняемая сортировка множества M может быть осуществлена, используя $O(s^2)$ вычислительных операций. Шаг 3 связан с $O(s)$ проверками вложенности множеств, мощность которых не превосходит s. Следовательно, каждая такая проверка вложенности требует $O(s)$ вычислительных операций. Таким образом, алгоритм требует $O(s^2)$ вычислительных операций. *Утверждение доказано.*

Далее покажем, что условие 1-вложенности является необходимым и достаточным условием сводимости многоиндексной транспортной задачи к задаче поиска циркуляции минимальной стоимости в древовидной сети.

Рассмотрим задачу $z_{MCC}(G';l_{ij},u_{ij},e_{ij}(i,j)\in A')\in W_{Tree}$. Пусть $G'=(V,A')$, тогда, согласно определению класса W_{Tree}, существует корневое ориентированное дерево $G=(V,A)$ с корнем $s\in V$ и множеством листьев $T\subseteq V$, что $A'=A\cup\{(t,s)\,|\,t\in T\}$. Обозначим через $t(i)$ путь из корня s к листу i в дереве G, таким образом, $t(i)=(v_0,...,v_k)$, где k – длина пути $t(i)$, $v_0=s$, $v_k=i$, $(v_j,v_{j+1})\in A$, $j=\overline{0,k-1}$, $i\in T$. Тогда обозначим через $T(G)$ множество всех путей из корня к листьям в графе G, т.е. $T(G)=\{t(i)\,|\,i\in T\}$.

Пусть $t=(v_0,...,v_k)$ – путь в графе G и $e\in A$. Тогда для удобства будем обозначать $e\in t$, если путь t проходит через дугу e, т.е. найдется $j\in\{0,...,k-1\}$, что $(v_j,v_{j+1})=e$; в противном случае, будем обозначать $e\notin t$. Далее пусть $e_1,e_2\in A$, тогда будем обозначать $e_1\prec e_2$, если существуют $t=(v_0,...,v_k)\in T(G)$ и $j_1,j_2\in\{0,...,k-1\}$, $j_1<j_2$, что $(v_{j_1},v_{j_1+1})=e_1$, $(v_{j_2},v_{j_2+1})=e_2$; в противном случае, будем обозначать $e_1\not\prec e_2$.

Лемма 4.7. Пусть $z=z_{MCC}(G';l_{ij},u_{ij},e_{ij},(i,j)\in A')\in W_{Tree}$ и существуют такие дуги $e_1,e_2,e_3,e_4,e_5\in A'$, что

$$l_{e_1}=0,\ u_{e_1}=1,\quad l_{e_4}=4,\ u_{e_4}=4,$$
$$l_{e_2}=0,\ u_{e_2}=2,\quad l_{e_5}=5,\ u_{e_5}=5,$$
$$l_{e_3}=0,\ u_{e_3}=3,\quad l_e=0,\ u_e\in\{0,d\},\ e\in A'\setminus\{e_1,e_2,e_3,e_4,e_5\},$$

где $d\ge\sum_{i=1}^{5}u_{e_i}$. Тогда либо задача z несовместна, либо существует допустимое решение $x_e, e\in A'$, задачи z такое, что $\{1,2,3\}\not\subset\{x_e\,|\,e\in A'\}$.

Доказательство. Пусть для задачи $z = z_{MCC}(G'; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A') \in W_{Tree}$ выполняются условия леммы. Тогда $G' = (V, A')$ и существует корневое ориентированное дерево $G = (V, A)$ с корнем $s \in V$ и множеством листьев $T \subseteq V$, что $A' = A \cup \{(t,s) \mid t \in T\}$. Не уменьшая общности, будем считать, что $e_1, e_2, e_3, e_4, e_5 \in A \subseteq A'$ и $l_e = 0, u_e = d$, $e \in A' \setminus A$; в противном случае, всегда можно перейти к эквивалентной постановке задачи z, удовлетворяющей данным требованиям.

Покажем, что выполняется условие (4.12), иначе задача z несовместна.

$$e_4 \bar{\prec} e_5, \tag{4.12}$$

Действительно, предположим, что $e_4 \prec e_5$. Тогда из древовидности сети следует, что задача z несовместна.

Далее пусть условие (4.12) выполняется. Отсюда выполняется одно из следующих двух условий

– $e_5 \prec e_4$; (4.13)

– $e_4 \notin t$ или $e_5 \notin t$ для любого пути $t \in T(G)$. (4.14)

Отдельно рассмотрим два случая: (4.13) и (4.14).

1. Пусть $e_5 \prec e_4$. Тогда найдется путь $t_1 \in T(G)$, что $e_5 \in t_1$, $e_4 \notin t_1$ и $\min\limits_{e \in t_1} u_e \ge 1$, иначе задача z несовместна. Далее обозначим $T_1 = \{t \mid t \in T(G),\ e_4 \in t,\ e_5 \in t\}$. Тогда $T_1 \ne \varnothing$ и

$$\sum_{t \in T_1} \min_{e \in t} u_e \ge 4, \tag{4.15}$$

иначе задача z несовместна. Схематично данный случай приведен на рис. 4.8.

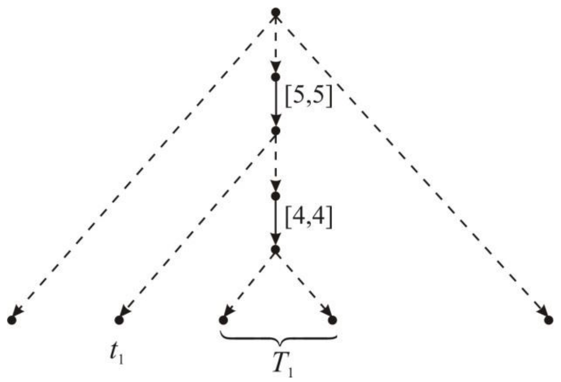

Рисунок 4.8.

Далее пусть условие (4.15) выполняется. Тогда возможны лишь два подслучая :

a. существует $t_2 \in T_1$, что $\min\limits_{e \in t_2} u_e = 4$,

b. $\min\limits_{e \in t} u_e < 4$, $t \in T_1$.

1.a. Пусть существует $t_2 \in T_1$, что $\min\limits_{e \in t_2} u_e = 4$. Тогда построим допустимый поток x_e, $e \in A'$, в задаче z по следующему алгоритму:

Шаг 1. $x_e := 0$, $e \in A$;

Шаг 2. $x_e := x_e + 1$, $e \in t_1$;

Шаг 3. $x_e := x_e + 4$, $e \in t_2$;

Шаг 4. x_e, $e \in A' \setminus A$, определяется однозначно через (2.5).

Для полученного допустимого решения справедливо: $x_e \in \{0,1,4,5\}$, $e \in A'$. Тогда $2,3 \notin \{x_e \mid e \in A'\}$.

1.b. Пусть $\min\limits_{e \in t} u_e < 4$, $t \in T_1$. Из условий леммы и условия (4.15) следует, что существуют $t_2, t_3 \in T_1$, что $\min\limits_{e \in t_2} u_e \in \{1,2\}$, $\min\limits_{e \in t_3} u_e = 3$. Тогда построим допустимый поток x_e, $e \in A'$, в задаче z по следующему алгоритму:

Шаг 1. $x_e := 0$, $e \in A$;

Шаг 2. $x_e := x_e + 1$, $e \in t_1$;

Шаг 3. $x_e := x_e + 1$, $e \in t_2$;

Шаг 4. $x_e := x_e + 3$, $e \in t_3$;

Шаг 5. x_e, $e \in A' \setminus A$, определяется однозначно через (2.5).

Для полученного допустимого решения справедливо: $x_e \in \{0,1,3,4,5\}$, $e \in A'$. Тогда $2 \notin \{x_e \mid e \in A'\}$.

2. Пусть $e_4 \notin t$ или $e_5 \notin t$ для любого пути $t \in T(G)$. Далее обозначим $T_1 = \{t \mid t \in T(G), e_4 \in t\}$, $T_2 = \{t \mid t \in T(G), e_5 \in t\}$, тогда $T_1, T_2 \neq \varnothing$ и

$$\sum_{t\in T_1}\min_{e\in t} u_e \ge 4\,,\ \sum_{t\in T_2}\min_{e\in t} u_e \ge 5\,, \qquad (4.16)$$

иначе задача z несовместна. Схематично данный случай приведен на рис. 4.9.

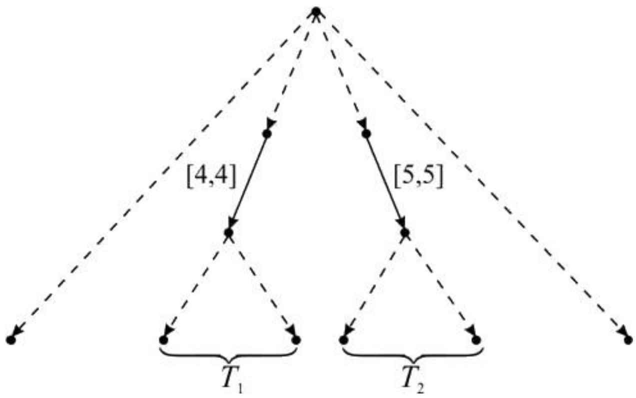

Рисунок 4.9.

Далее возможны лишь четыре подслучая:

a. Существуют $t_1 \in T_1$ и $t_2 \in T_2$, что $\min_{e\in t_1} u_e \ge 4$ и $\min_{e\in t_2} u_e \ge 5$;

b. существует $t_1 \in T_1$, что $\min_{e\in t_1} u_e \ge 4$ и $\min_{e\in t} u_e < 5$, $t \in T_2$;

c. существует $t_1 \in T_2$, что $\min_{e\in t_1} u_e \ge 5$ и $\min_{e\in t} u_e < 4$, $t \in T_1$;

d. $\min_{e\in t} u_e < 4$, $t \in T_1$, и $\min_{e\in t} u_e < 5$, $t \in T_2$.

2.a. Пусть существуют $t_1 \in T_1$ и $t_2 \in T_2$, что $\min_{e\in t_1} u_e \ge 4$ и $\min_{e\in t_2} u_e \ge 5$. Тогда построим допустимый поток x_e, $e \in A'$, в задаче z по следующему алгоритму:

Шаг 1. $x_e := 0$, $e \in A$;

Шаг 2. $x_e := x_e + 4$, $e \in t_1$;

Шаг 3. $x_e := x_e + 5$, $e \in t_2$;

Шаг 5. x_e, $e \in A' \setminus A$, определяется однозначно через (2.5).

Для полученного допустимого решения справедливо $x_e \in \{0,4,5,9\}$, $e \in A'$. Тогда $1,2,3 \notin \{x_e \mid e \in A'\}$.

2.b. Пусть существуют $t_1 \in T_1$, что $\min_{e \in t_1} u_e \geq 4$ и $\min_{e \in t} u_e < 5$, $t \in T_2$. Из условий леммы и условия (4.16) следует, что существуют $t_2, t_3 \in T_2$, что $\min_{e \in t_2} u_e = 2$, $\min_{e \in t_3} u_e = 3$. Тогда построим допустимый поток x_e, $e \in A'$, в задаче z по следующему алгоритму:

Шаг 1. $x_e := 0$, $e \in A$;

Шаг 2. $x_e := x_e + 4$, $e \in t_1$;

Шаг 3. $x_e := x_e + 2$, $e \in t_2$;

Шаг 4. $x_e := x_e + 3$, $e \in t_3$;

Шаг 5. x_e, $e \in A' \setminus A$, определяется однозначно через (2.5).

Для полученного допустимого решения справедливо $x_e \in \{0,2,3,4,5,9\}$, $e \in A'$. Тогда $1 \notin \{x_e \mid e \in A'\}$.

2.c. Пусть существуют $t_1 \in T_2$, что $\min_{e \in t_1} u_e \geq 5$ и $\min_{e \in t} u_e < 4$, $t \in T_1$. Из условий леммы и условия (4.16) следует, что существуют $t_2, t_3 \in T_2$, что $\min_{e \in t_2} u_e \in \{1,2\}$, $\min_{e \in t_3} u_e = 3$. Тогда построим допустимый поток x_e, $e \in A'$, в задаче z по следующему алгоритму:

Шаг 1. $x_e := 0$, $e \in A$;

Шаг 2. $x_e := x_e + 5$, $e \in t_1$;

Шаг 3. $x_e := x_e + 1$, $e \in t_2$;

Шаг 4. $x_e := x_e + 3$, $e \in t_3$;

Шаг 5. x_e, $e \in A' \setminus A$, определяется однозначно через (2.5).

Для полученного допустимого решения справедливо $x_e \in \{0,1,3,4,5,9\}$, $e \in A'$. Тогда $2 \notin \{x_e \mid e \in A'\}$.

2.d. Пусть $\min_{e \in t} u_e < 4$, $t \in T_1$, и $\min_{e \in t} u_e < 5$, $t \in T_2$. Тогда из условий леммы следует $\sum_{t \in T_1} \min_{e \in t} u_e + \sum_{t \in T_2} \min_{e \in t} u_e \leq u_1 + u_2 + u_3 = 6$, что противоречит условию (4.16).

Получаем, что задача z, удовлетворяющая условиям леммы, либо несовместна, либо имеет допустимое решение $x_e, e \in A'$, такое, что $\{1,2,3\} \not\subset \{x_e \mid e \in A'\}$. *Лемма доказана.*

Лемма 4.8. Пусть $M \subseteq 2^{N(s)}$. Если класс $W(M)$ является $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Tree}, то класс $W(M \cup \{\varnothing\})$ также является $t_1' \mid t_2' - equal \mid t_3' - edge$ сводимым к классу W_{Tree}.

Доказательство. Пусть выполняются условия леммы. Если $\varnothing \in M$, то $M \cup \{\varnothing\} = M$ и лемма доказана. Далее пусть $\varnothing \notin M$. Рассмотрим задачу $w' = w(A', b'^-, b'^+, c') \in W(M \cup \{\varnothing\})$. Т.к. $M \subseteq M \cup \{\varnothing\}$, то найдется задача $w = w(A, b^-, b^+, c) \in W(M)$, что

- $col(A) = col(A')$, $row(A) + col(A') = row(A')$;
- многоиндексные переменные задачи w совпадают с многоиндексными переменными задачи w', тогда, не уменьшая общности, можно считать, что столбцы матриц A, A' с одинаковыми номерами связаны с одними и теми же многоиндексными переменными задач w и w';
- задача w' содержит все ограничения задачи w, тогда, не уменьшая общности, можно считать, что $A_{ij} = A'_{ij}$, $b_i^- = b_i'^-$, $b_i^+ = b_i'^+$, $i = \overline{1, row(A)}$, $j = \overline{1, col(A)}$;
- $c = c'$.

При этом, не уменьшая общности, будем считать, что

- $A'_{row(A)+i,i} = 1$, $i = \overline{1, col(A')}$,
- $A'_{row(A)+i,j} = 0$, $j \in \overline{1, col(A')} \setminus \{i\}$, $i = \overline{1, col(A')}$.

Таким образом, ограничения задачи w', определяемые элементом $\varnothing \in M \cup \{\varnothing\}$, представляющие собой двусторонние ограничения на переменные задачи w', задаются нижними строками матрицы A'.

Т.к. $W(M)$ является $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Tree}, то, согласно определению 4.1, существует задача $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Tree}$, соответствующая задаче w. И при этом существует пара функций α, β, удовлетворяющих условиям определения 4.1. Далее опишем схему построения задачи $z' = z_{MCC}(G'; l'_{ij}, u'_{ij}, e'_{ij}, (i,j) \in A'_{G'}) \in W_{Tree}$, соответствующей задаче w'.

Задача w' отличаются от задачи w лишь наличием двусторонних ограничений на переменные. И при этом функция β определяет дуги задачи z, соответствующие переменным исходной задачи. Таким образом, каждую из таких дуг можно заменить на пару последовательных дуг, первая из которых будет соответствовать переменной задачи, а вторая двустороннему ограничению на эту переменную. Тогда для построения задачи z' модифицируем задачу z следующим образом. Пусть изначально $z' = z$. Для дуги $\beta(i) = (u,v)$ преобразуем граф G' следующим образом:

– $V_{G'} := V_{G'} \cup \{p_i\}$,

– $A_{G'} := A_{G'} \setminus \{(u,v)\} \cup \{(u,p_i),(p_i,v)\}$,

где p_i – новая вершина, $i = \overline{1,col(A)}$. Далее определим функции $\alpha' : \{1,2,...,row(A')\} \to A_{G'}$, $\beta' : \{1,2,...,col(A')\} \to A_{G'}$ следующим образом:

– $\alpha'(i) = \begin{cases} \alpha(i), \text{если } \alpha(i) \notin \{\beta(j) \mid j = \overline{1,col(A)}\} \\ (u,p_j), \text{если существует } j \in \{1,...,col(A)\}, \text{что } \alpha(i) = \beta(j) = (u,v) \end{cases}$, $i = \overline{1,row(A)}$;

– $\alpha'(row(A)+i) = (p_i,v)$, где $\beta(i) = (u,v)$, $i = \overline{1,col(A')}$;

– $\beta'(i) = (u,p_i)$, где $\beta(i) = (u,v)$, $i = \overline{1,col(A')}$.

Параметры $l'_{ij}, u'_{ij}, e'_{ij}, (i,j) \in A'_{G'}$, задачи z' определяются через функции α' и β', согласно определению 4.1.

Таким образом, по построению задача z' является задачей, соответствующей задаче w'. Сеть, соответствующая задаче z', имеет по построению древовидную структуру в силу древовидности структуры сети задачи z. Отсюда $z' \in W_{Tree}$. Следовательно класс $W(M \cup \{\varnothing\})$ является $t'_1 \mid t'_2 - equal \mid t'_3 - edge$ сводимым к классу W_{Tree}. *Лемма доказана.*

Теорема 4.5. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Tree}, необходимо, чтобы множество M было 1-вложенным.

Доказательство. Доказательство от противного, пусть M не является 1-вложенным, и класс $W(M)$ является $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Tree}.

Т.к. множество M не является 1-вложенным, то, согласно лемме 4.4, найдутся такие $f', f'' \in M$, что $f' \not\subset f''$, $f'' \not\subset f'$. Тогда существуют $k', k'' \in N(s)$, $k' \neq k''$ такие, что

$$k' \in f',\ k' \notin f'',$$

$$k'' \in f'',\ k'' \notin f'.$$

Рассмотрим s-индексные наборы $F'_{N(s)}$, $F''_{N(s)}$, $F'''_{N(s)}$, где

$$F'_{N(s)}(k') = 2,\ F'_{N(s)}(i) = 1,\ i \in N(s) \setminus \{k'\},$$

$$F''_{N(s)}(k'') = 2,\ F''_{N(s)}(i) = 1,\ i \in N(s) \setminus \{k''\},$$

$$F'''_{N(s)}(i) = 1,\ i \in N(s).$$

Далее рассмотрим многоиндексные наборы $F'_{\overline{f'}}$ и $F''_{\overline{f''}}$, где

$$F'_{\overline{f'}}(i) = 1,\ i \in \overline{f'},$$

$$F''_{\overline{f''}}(i) = 1,\ i \in \overline{f''}.$$

Пусть $w \in W(M \cup \varnothing)$ и рассмотрим следующую подматрицу H матрицы $Matr(w)$ (т.е. подматрицу матрицы системы ограничений задачи w):

$$H = \quad Matr(w;$$

$$(f', F'_{\overline{f'}}), (f'', F''_{\overline{f''}}), (\varnothing, F'_{N(s)}), (\varnothing, F''_{N(s)}), (\varnothing, F'''_{N(s)});$$

$$F'_{N(s)}, F''_{N(s)} F'''_{N(s)}).$$

Для произвольной задачи $w \in W(M \cup \varnothing)$ справедливо:

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

Тогда выберем задачу $w' \in W(M \cup \varnothing)$ со следующей системой ограничений:

$$4 \le \sum_{F_{f'} \in E_{f'}} x_{F_{f'} F'_{\overline{f'}}} \le 4;\ 0 \le \sum_{F_{f'} \in E_{f'}} x_{F_{f'} F_{\overline{f'}}} \le d,\ F_{\overline{f'}} \in E_{\overline{f'}} \setminus \{F'_{\overline{f'}}\}; \tag{4.17}$$

$$5 \le \sum_{F_{f''} \in E_{f''}} x_{F_{f''} F''_{\overline{f''}}} \le 5;\ 0 \le \sum_{F_{f''} \in E_{f''}} x_{F_{f''} F_{\overline{f''}}} \le d,\ F_{\overline{f''}} \in E_{\overline{f''}} \setminus \{F''_{\overline{f''}}\}; \tag{4.18}$$

$$0 \le x_{F'_{N(s)}} \le 1;\ 0 \le x_{F''_{N(s)}} \le 2;\ 0 \le x_{F'''_{N(s)}} \le 3; \tag{4.19}$$

$$0 \le x_{F_{N(s)}} \le 0,\ F_{N(s)} \in E_{N(s)} \setminus \{F'_{N(s)}, F''_{N(s)}, F''_{N(s)}\};$$

$$0 \le \sum_{F_f \in E_f} x_{F_f F_{\overline{f}}} \le d,\ F_{\overline{f}} \in E_{\overline{f}},\ f \in M \setminus \{f_1, f_2, \varnothing\}; \tag{4.20}$$

где $d = 1+2+3+4+5 = 15$. Тогда система (4.17)–(4.20) эквивалентна следующей системе линейных неравенств

$$\begin{pmatrix} 4 \\ 5 \\ 0 \\ 0 \\ 0 \end{pmatrix} \le H \begin{pmatrix} x_{F'_{N(s)}} \\ x_{F'_{N(s)}} \\ x_{F'_{N(s)}} \end{pmatrix} \le \begin{pmatrix} 4 \\ 5 \\ 1 \\ 2 \\ 3 \end{pmatrix},$$

$$x_{F_{N(s)}} = 0,\ F_{N(s)} \in E_{N(s)} \setminus \{F'_{N(s)}, F''_{N(s)}, F'''_{N(s)}\}.$$

Отсюда задача w' имеет единственное решение

$$x_{F'_{N(s)}} = 1,\ x_{F''_{N(s)}} = 2,\ x_{F'''_{N(s)}} = 3,$$

$$x_{F_{N(s)}} = 0,\ F_{N(s)} \in E_{N(s)} \setminus \{F'_{N(s)}, F''_{N(s)}, F''_{N(s)}\}.$$

По предположению класс $W(M)$ является $t_1 | t_2 - equal | t_3 - edge$ сводимым к классу W_{Tree}. Тогда, согласно лемме 4.8, класс $W(M \cup \{\varnothing\})$ также является $t'_1 | t'_2 - equal | t'_3 - edge$ сводимым к классу W_{Tree}. Рассмотрим задачу $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Tree}$, соответствующую описанной выше задаче w'. Согласно определению 4.1, задача z удовлетворяет условию леммы 4.7. По лемме 4.7, задача z либо несовместна, либо имеет допустимое решение x_{ij}, $(i,j) \in A_G$, такое, что $\{1,2,3\} \not\subset \{x_{ij} \mid (i,j) \in A_G\}$. Задача w' совместна и имеет единственное решение $x_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$, при этом $\{1,2,3\} \subseteq \{x_{F_{N(s)}} \mid F_{N(s)} \in E_{N(s)}\}$. Однако, согласно определению 4.1, должно выполняться условие $\{x_{F_{N(s)}} \mid F_{N(s)} \in E_{N(s)}\} \subseteq \{x_{ij} \mid (i,j) \in A_G\}$. Получаем противоречие. Таким образом, предположение неверно. *Теорема доказана.*

Согласно следствию 4.4 и теореме 4.5, найденное условие 1-вложенности является необходимыми и достаточным условием сводимости (согласно введенной концепции сведения) многоиндексных задач к задаче поиска потока минимальной стоимости в древовидной сети. Более того, оказывается (как и для 2-вложенных множеств при сводимости к классу W_{Graph}), что предложенная при конструктивном доказательстве теоремы 4.2 схема сведения класса многоиндексных задач $W(M)$ с 1-вложенным множеством M, требующая линейных вычислительных затрат, является (в рамках рассматриваемой концепции $t_1 | t_2 - equal | t_3 - edge$ сводимости к классу W_{Tree}) оптимальной в том смысле, что

– сведение с использованием вычислительных затрат асимптотически меньших линейных невозможно;
– сколь угодное увеличение вычислительных затрат на сведение не приводится к расширению класса многоиндексных задач сводимых к классу W_{Tree}.

Таким образом, в совокупности следствие 4.4 и теорема 4.5 представляют собой исчерпывающий результат исследования $t_1 | t_2 - equal | t_3 - edge$ сводимости классов многоиндексных задач $W(M)$ к классу задач поиска потока минимальной стоимости в древовидной сети W_{Tree}.

Далее проиллюстрируем применимость метода исследования 1-вложенных многоиндексных задач $W(M)$, основанного на сводимости к классу задач поиска потока в древовидной сети, при решении ряда рассмотренных ранее многоиндексных задач. Рассмотрим задачу $U_{\prec}(M,H)$ и $U_{\max\min}(M,H)$ $M,H \subseteq 2^{N(s)}$. Согласно утверждениям 2.8 и 2.9, решение данных задач связано с последовательным решением систем ограничений вида $\Omega(\vec{v})$, относящейся к классу $D(M \cup \widetilde{H})$. Далее пусть множество $M \cup \widetilde{H}$ является 1-вложенным. Тогда, согласно утверждению 4.8, соответствующая система вида $\Omega(\vec{v})$ может быть решена, используя $O(|E_{N(s)}|)$ вычислительных операций. Отсюда справедливы следующие утверждения.

Утверждение 4.10. Пусть $M,H \subseteq 2^{N(s)}$ и множество $M \cup \widetilde{H}$ является 1-вложенным, тогда существует алгоритм решения задач класса $U_{\prec}(M,H)$, требующий $O(|E_{N(s)}|^2 \log p)$ вычислительных операций.

Утверждение 4.11. Пусть $M, H \subseteq 2^{N(s)}$ и множество $M \cup \widetilde{H}$ является 1-вложенным, тогда существует алгоритм решения задач класса $U_{\max\min}(M, H)$, требующий $O(|E_{N(s)}| \log p)$ вычислительных операций.

Дополнительно проиллюстрируем полученные результаты при решении прикладной многоиндексной задачи, поставленной в главе 1. Рассмотрим проблему формирования допустимого портфеля заказов (1.10)-(1.14). В случае несовместности системы ограничений (1.10)-(1.14) ставится задача идентификации причин несовместности. Так несовместность может быть вызвана несогласованностью внутренних ограничений предприятия. Вопрос о несогласованности внутренних ограничений связан с исследованием совместности системы (1.10),(1.11),(1.14), которая относится к классу $D(M)$, где $M = \{\{j\}, \{j, t\}\}$. При этом $s = 3$ и $|E_{N(s)}| = |I||J||T|$. Заметим, что по определению 4.2 множество M является 1-вложенным. Отсюда, согласно утверждению 4.8, исследование несогласованности внутренних ограничений предприятия (при формировании портфеля заказов) может быть осуществлено, используя $O(|I||J||T|)$ вычислительных операций.

4.4. Условия $t_1 | t_2 - Z | t_3 - Z$ сводимости

Введем схему $t_1 | t_2 - Z | t_3 - Z$ сводимости (сводимости с сохранением целочисленности), представляющую собой обобщение рассмотренной ранее схемы $t_1 | t_2 - equal | t_3 - edge$ сводимости. Для данной схемы сведения будут обобщены результаты сводимости, полученные ранее для $t_1 | t_2 - equal | t_3 - edge$ сводимости. Основные особенности предлагаемой схемы:

– при построении вспомогательной сети пропускные способности дуг определяются таким образом, что пропускные способности являются целочисленными в случае целочисленности свободных коэффициентов двусторонних неравенств системы ограничений исходной задачи (такая особенность сведения отражается в процедуре, обозначаемой *Z*);

– при построении решения исходной задачи строится целочисленное решение в случае целочисленности потока в сети (такая особенность сведения отражается в процедуре, обозначаемой *Z*).

Таким образом, рассматриваемая схема сведения является частным случаем схемы, описанной в определении 3.1, для которой $s_2 = Z$, $s_3 = Z$.

Определение 4.5. Пусть W – класс задач линейного программирования с двусторонней системой линейных неравенств. Будем говорить, что класс W является $t_1 \mid t_2 - Z \mid t_3 - Z$ сводимым к классу $W' \subseteq W_{Graph}$, если класс W является $t_1 \mid t_2 \mid t_3$ сводимым к классу W', и выполняются следующие условия. Пусть $w = w(A, b^-, b^+, c) \in W$, $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W'$ является задачей, соответствующей задаче w. Далее пусть x_{ij}, $(i,j) \in A_G$, является оптимальным (допустимым) решением задачи z, а x_i', $i = \overline{1, col(A)}$ является соответствующим ему оптимальным (допустимым) решением задачи w. Тогда:

– если $b^-, b^+ \in Z^{row(A)}$, то $l_{ij}, u_{ij} \in Z$, $(i,j) \in A_G$,

– если $x_{ij} \in Z,\ (i,j) \in A_G$, то $x_i' \in Z$, $i = \overline{1, col(A)}$.

Схематично концепция $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости представлена на рис. 4.10.

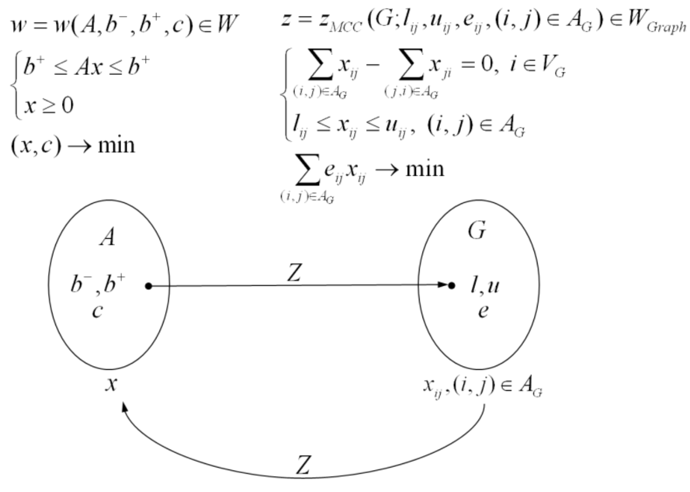

Рисунок 4.10.

Несложно увидеть, что исследуемая ранее схема $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости является частным случаем более общей схемы $t_1 \mid t_2 - Z \mid t_3 - Z$ сводимости. Действительно, согласно определению 4.1 в случае $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости выполняются условия целочисленности, описанные в определении 4.5. При этом схема $t_1 \mid t_2 - Z \mid t_3 - Z$ сводимости описывает более широкий класс схем сведения (по сравнению с $t_1 | t_2 - equal | t_3 - edge$ сводимостью), т.к. накладывает лишь условие целочисленности на процедуру сведения. Таким образом, согласно теореме 4.2, справедливо следующее следствие.

Следствие 4.5. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $L | L - Z | L - Z$ сводимым к классу W_{Graph} достаточно, чтобы множество M было 2-вложенным.

Далее покажем, что условие 2-вложенности является необходимым и достаточным условием $P^* | P^* - Z | P^* - Z$ сводимости многоиндексной транспортной задачи к задаче поиск потока минимальной стоимости, иначе неверной является известная гипотеза о неравенстве классов P и NP (см., например, [52]).

Теорема 4.6. Пусть $M \subseteq 2^{N(s)}$. Если класс $W(M)$ является $P^* | P^* - Z | P^* - Z$ сводимым к классу W_{Graph}, то класс $W_Z(M)$ разрешим за полиномиальное время.

Доказательство. Пусть выполняются условия теоремы. Тогда для каждой задачи $w \in W(M)$ можно построить соответствующую ей задачу $z \in W_{Graph}$. В силу $P^* | P^* - Z | P^* - Z$ сводимости количество вершин и дуг, а также значения пропускных способностей в соответствующей потоковой задаче z полиномиально зависят от размера задачи w. При этом, согласно определению 4.5, пропускные способности дуг в задаче z при целочисленных свободных коэффициентах двусторонних ограничений задачи w также являются целочисленными. Для общности отметим, что если $w \in W_Z(M)$ содержит нецелочисленные свободные коэффициенты, то в силу целочисленности коэффициентов матрицы системы ограничений задачи w, задача w при помощи округления параметров может быть преобразована к эквивалентной постановке с целочисленными параметрами.

Как известно, матрица системы ограничений задачи линейного программирования z является абсолютно унимодулярной [51]. Тогда в силу целочисленности пропускных способностей дуг, целочисленное решение задачи z может быть найдено за полиномиальное время [102]. Отсюда, согласно определению 4.5, построенное за полиномиальное время решение задачи w также будет являться целочисленным. Следовательно, найденное решение также будет являться решением задачи w_Z. Таким образом, класс задач целочисленного линейного программирования $W_Z(M)$ разрешим за полиномиальное время. *Теорема доказана.*

Следствие 4.6. Пусть $M \subseteq 2^{N(s)}$. Если класс $W_Z(M)$ является NP-трудным, то класс $W(M)$ не является $P^* \mid P^* - Z \mid P^* - Z$ сводимым к классу W_{Graph}, в противном случае $P = NP$.

Доказательство. Пусть класс задач $W_Z(M)$ является NP-трудным, и класс $W(M)$ является $P^* \mid P^* - Z \mid P^* - Z$ сводимым к классу W_{Graph}. Тогда по теореме 4.6 класс $W_Z(M)$ полиномиально разрешимым. Отсюда, согласно концепции NP-трудности, $P = NP$ [52]. *Следствие доказано.*

Лемма 4.9. Пусть $s_1 \le s_2$, $M_1 \subseteq 2^{N(s_1)}$, $M_2 \subseteq 2^{N(s_2)}$ и $M_1 \prec M_2$ тогда, если класс $W_Z(M_1)$ является NP-трудным, то класс $W_Z(M_2)$ также является NP-трудным.

Доказательство. Пусть выполняются условия леммы. Проведем доказательство путем полиномиального сведения NP-трудного класса задач $W_Z(M_1)$ к классу $W_Z(M_2)$.

По определению 4.4 существует подмножество $g \subseteq N(s_2)$, $|g| = s_1$ и биективная функция $\pi : g \to N(s_1)$, что $M_1 \subseteq \bigcup\limits_{f \in M_2(g)} \{\{\pi(i) \mid i \in f\}\}$. Обозначим $P(f_1) = \{f_2 \mid f_1 = \{\pi(i) \mid i \in f_2 \cap g\}, f_2 \in M_2\}$, $f_1 \in M_1$. Далее рассмотрим произвольную задачу $w_1 \in W_Z(M_1)$, $w_1 = w_Z(s_1; n'_1, n'_2, ..., n'_{s_1}; \{a'_{F_{\overline{f}}}\}, \{b'_{F_{\overline{f}}}\}, f \in M_1; \{c'_{F_{N(s_1)}}\})$ и построим задачу $w_2 \in W(M_2)$, $w_2 = w_Z(s_2; n''_1, n''_2, ..., n''_{s_2}; \{a''_{F_{\overline{f}}}\}, \{b''_{F_{\overline{f}}}\}, f \in M_2; \{c''_{F_{N(s_2)}}\})$, удовлетворяющую следующим условиям:

– $n''_i = n'_{\pi(i)}$, $i \in g$,

– $n''_i = 2$, $i \in N(s_2) \setminus g$,

– $a''_{F_{\overline{f_2}}} = a'_{F_{\overline{f_1}}}$, $b'_{F_{\overline{f_2}}} = b''_{F_{\overline{f_1}}}$, где $F_{\overline{f_2}} = (F_{\overline{f_2}})_{\overline{f_2} \cap g}(1,...,1)_{\overline{f_2} \setminus g}$, $(F_{\overline{f_2}})_{\overline{f_2} \cap g}(i) = F_{\overline{f_1}}(\pi(i))$, $i \in \overline{f_2} \cap g$, $F_{\overline{f_2}} \in E_{\overline{f_2}}$, $F_{\overline{f_1}} \in E_{\overline{f_1}}$, $f_2 \in P(f_1)$, $f_1 \in M_1$,

– $a''_{F_{\overline{f_2}}} = 0$, $b'_{F_{\overline{f_2}}} = b^*$, где $F_{\overline{f_2}} \in E_{\overline{f_2}}$, $f_2 \in M_2 \setminus (\bigcup\limits_{f \in M_1} P(f))$ или $(F_{\overline{f_2}})_{\overline{f_2} \setminus g} \ne (1,...,1)_{\overline{f_2} \setminus g}$, $\overline{f_2} \setminus g \ne \varnothing$, $F_{\overline{f_2}} \in E_{\overline{f_2}}$, $f_2 \in \bigcup\limits_{f \in M_1} P(f)$,

– $c''_{F_{N(s_2)}} = c'_{F_{N(s_1)}}$, где $F_{N(s_2)} = F_g(1,...,1)_{\overline{g}}$, $F_g(i) = F_{N(s_{s_1})}(\pi(i))$, $i \in g$, $F_g \in E_g$, $F_{N(s_1)} \in E_{N(s_1)}$,

– $c''_{F_{N(s_2)}} = c^* + 1$, $F_{N(s_2)} \in E_{N(s_2)} \setminus \{F_g(1,...,1)_{\overline{g}} \mid F_g \in E_g\}$, здесь $c^* = b^* \sum_{F_{N(s_1)} \in E_{N(s_1)}} | c'_{F_{N(s_1)}} |$, $b^* = \sum_{f \in M_1} \sum_{F_{\bar f} \in E_{\bar f}} b_{F_{\bar f}}$.

Несложно увидеть, что по построению выполняются следующие условия:

a. если задача w_2 несовместна, то задача w_1 также несовместна;

b. пусть $x''_{F_{N(s_2)}}, F_{N(s_2)} \in E_{N(s_2)}$ – решение задачи w_2, и $F''^* = \sum_{F_{N(s_2)} \in E_{N(s_2)}} c_{F_{N(s_2)}} x''_{F_{N(s_2)}}$, тогда

b.1. если $F''^* > c^*$, то задача w_1 несовместна;

b.2. если $F''^* \le c^*$, то решение задачи w_1 может быть построено по следующему правилу:

$$x'_{F_{N(s_1)}} = x''_{F_{N(s_2)}}, \text{ где } F_{N(s_2)} = F_g(1,...,1)_{\bar g}, \; F_g(i) = F_{N(s_{s_1})}(\pi(i)), \; i \in g, \; F_{N(s_1)} \in E_{N(s_1)}.$$

Количество переменных в задаче w_1 равно $n' = \prod_{i=1}^{s_1} n'_i$, по построению количество переменных в задаче w_2 равно $n'' = \prod_{i=1}^{s_2} n''_i = 2^{s_2 - s_1} \prod_{i=1}^{s_1} n'_i = O(n')$. Пусть $A' = Matr(w_1)$, $A'' = Matr(w_2)$. Тогда число ограничений в задачах w_1 и w_2 равны $nrow(Matr(w_1))$ и $nrow(Matr(w_2))$ соответственно. При этом

$$nrow(Matr(w_1)) = \sum_{f \in M_1} | E_{\bar f} | \le \sum_{f \in 2^{N(s_1)}} | E_{\bar f} | \le (\prod_{i=1}^{s_1} n'_i)(\sum_{j=0}^{s_1} C^j_{s_1}) = 2^{s_1} \prod_{i=1}^{s_1} n'_i = O(n'),$$

$$nrow(Matr(w_2)) = \sum_{f \in M_2} | E_{\bar f} | \le \sum_{f \in 2^{N(s_2)}} | E_{\bar f} | \le (\prod_{i=1}^{s_2} n''_i)(\sum_{j=0}^{s_2} C^j_{s_2}) = 2^{s_2} 2^{s_2 - s_1} \prod_{i=1}^{s_1} n'_i = O(n').$$

Таким образом,

$$O(n') \le nrow(A) \cdot ncol(A) \le O(n'^2),$$

$$O(n') \le nrow(A') \cdot ncol(A') \le O(n'^2).$$

Также отметим, что параметры b^*, c^* вычислимы за полиномиальное время от размера задачи w_1. Отсюда выполненное сведение класса $W_Z(M_1)$ к классу $W_Z(M_2)$ реализуемо за полиномиальное время. Тогда из *NP*-трудности класса $W_Z(M_1)$ следует *NP*-трудность класса $W_Z(M_2)$ [52]. *Лемма доказана*.

Лемма 4.10. Пусть $M \subseteq 2^{N(s)}$. Если выполняется одно из следующих трех условий:

1. $s=3$, $M=\{\{1,2\},\{1,3\},\{2,3\}\}$,

2. $s=3$, $M=\{\{1\},\{2\},\{3\}\}$,

3. $s=4$, $M=\{\{1,2\},\{2,3\},\{1,4\}\}$,

то класс $W_Z(M)$ является *NP*-трудным.

Доказательство. 1. Пусть $s=3$, $M=\{\{1,2\},\{1,3\},\{2,3\}\}$. Согласно [52], класс трехиндексных аксиальных задач о назначениях (4.21)-(4.25) является *NP*-трудным.

$$\sum_{i=1}^{n}\sum_{j=1}^{n} y_{ijk}=1,\ k=\overline{1,n}, \tag{4.21}$$

$$\sum_{i=1}^{n}\sum_{k=1}^{n} y_{ijk}=1,\ j=\overline{1,n}, \tag{4.22}$$

$$\sum_{j=1}^{n}\sum_{k=1}^{n} y_{ijk}=1,\ i=\overline{1,n}, \tag{4.23}$$

$$y_{ijk}\in\{0,1\},\ i,j,k=\overline{1,n}, \tag{4.24}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk}y_{ijk}\to\min, \tag{4.25}$$

где $e_{ijk}\ge 0,\ i,j,k=\overline{1,n}.$ Далее рассмотрим ограничение

$$y_{ijk}\in Z_+,\ i,j,k=\overline{1,n}. \tag{4.26}$$

Легко увидеть, что система ограничений (4.21),(4.24) эквивалентна системе ограничений (4.21),(4.26). Таким образом, задача (4.21)-(4.25) может быть поставлена в виде (4.21)-(4.23),(4.26),(4.25). В свою очередь задача (4.21)-(4.23),(4.26),(4.25) включена в рассматриваемый класс $W_Z(M)$. Следовательно, класс $W_Z(M)$ является *NP*-трудным.

2. Пусть $s=3$, $M=\{\{1\},\{2\},\{3\}\}$. Согласно [146], класс трехиндексных планарных задач о назначениях (4.27)-(4.29),(4.24),(4.25) является *NP*-трудным.

$$\sum_{i=1}^{n} y_{ijk}=1,\ j,k=\overline{1,n}, \tag{4.27}$$

$$\sum_{j=1}^{n} y_{ijk}=1,\ i,k=\overline{1,n}, \tag{4.28}$$

$$\sum_{k=1}^{n} y_{ijk} = 1,\ i, j = \overline{1, n}. \tag{4.29}$$

По аналогии с пунктом 1 доказательства, задача (4.27)-(4.29),(4.24),(4.25) может быть поставлена в виде (4.27)-(4.29),(4.26),(4.25), которая в свою очередь включена в рассматриваемый класс $W_Z(M)$. Следовательно, класс $W_Z(M)$ является *NP*-трудным.

3. Пусть $s = 4$, $M = \{\{1,2\},\{2,3\},\{1,4\}\}$. Проведем доказательство путем полиномиального сведения к классу $W_Z(M)$ класса трехиндексных планарных задач о назначениях (4.27)-(4.29),(4.24),(4.25), который, согласно [146], является *NP*-трудным.

Задаче (4.27)-(4.29),(4.24),(4.25) поставим в соответствие задачу $w = w_Z(s; M; n, n, n, n; \{a_{F_{\bar f}}\}, \{b_{F_{\bar f}}\}, f \in M; \{c_{F_{N(s)}}\}) \in W_Z(M)$ такую, что

$$a_{F_{\bar f}} = b_{F_{\bar f}} = 1, F_{\bar f} \in E_{\bar f},\ f \in M,$$

$$c_{i_1 i_2 i_3 i_4} = \begin{cases} e_{i_2 i_3 i_4}, \text{ если } i_1 = i_2 \\ \Delta, \text{ иначе} \end{cases},\ i_1, i_2, i_3, i_4 = \overline{1, n},$$

где $\Delta = 1 + \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk}$. Тогда задача w представляет собой задачу целочисленного линейного программирования (4.30)-(4.34).

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n} x_{i_1 i_2 i_3 i_4} = 1,\ i_3, i_4 = \overline{1, n}, \tag{4.30}$$

$$\sum_{i_2=1}^{n}\sum_{i_3=1}^{n} x_{i_1 i_2 i_3 i_4} = 1,\ i_1, i_4 = \overline{1, n}, \tag{4.31}$$

$$\sum_{i_1=1}^{n}\sum_{i_4=1}^{n} x_{i_1 i_2 i_3 i_4} = 1,\ i_2, i_3 = \overline{1, n}, \tag{4.32}$$

$$x_{i_1 i_2 i_3 i_4} \in Z_+,\ i_1, i_2, i_3, i_4 = \overline{1, n}, \tag{4.33}$$

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n} c_{i_1 i_2 i_3 i_4} x_{i_1 i_2 i_3 i_4} \to \min. \tag{4.34}$$

Заметим, что в силу, например, условий (4.30) и (4.33), если $x_{i_1 i_2 i_3 i_4}$, $i_1, i_2, i_3, i_4 = \overline{1, n},$ является допустимым решением задачи w, то $x_{i_1 i_2 i_3 i_4} \in \{0,1\}$, $i_1, i_2, i_3, i_4 = \overline{1, n}$.

Далее пусть $\{x^*_{F_{N(s)}}\}$ – оптимальное решение задачи w. Покажем, от противного, что выполняются следующие условия

$$x^*_{i_1 i_2 i_3 i_4} = 0 \text{, если } i_1 \neq i_2 \text{, } i_1, i_2, i_3, i_4 = \overline{1,n}. \tag{4.35}$$

Предположим, что условие (4.35) не выполняется, т.е. существуют $i_1, i_2, i_3, i_4 \in \{1,...,n\}$, $i_1 \neq i_2$, что $x^*_{i_1 i_2 i_3 i_4} = 1$. Отсюда

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n} c_{i_1 i_2 i_3 i_4} x^*_{i_1 i_2 i_3 i_4} \geq \Delta .$$

Рассмотрим матрицу значений переменных $\{x_{F_{N(s)}}\}$, определенную следующим образом

$$x_{i_1 i_1 i_3 i_4} = \begin{cases} 1, \text{ если } i_1 = i_3 + i_4 - 1, i_3 + i_4 \leq n+1 \text{ или } i_1 = i_3 + i_4 - n - 1, i_3 + i_4 > n+1 \\ 0, \text{ иначе} \end{cases}, \; i_1, i_3, i_4 = \overline{1,n},$$

$$x_{i_1 i_2 i_3 i_4} = 0, \; i_1 \neq i_2, \; i_1, i_2, i_3, i_4 = \overline{1,n}.$$

Покажем, что построенная матрица $\{x_{F_{N(s)}}\}$ является допустимым решением задачи w. С учетом $i_3, i_4 \in \{1,...,n\}$ выполняется: если $i_3 + i_4 \leq n+1$, то $1 \leq i_3 + i_4 - 1 \leq n$; если $i_3 + i_4 > n+1$, то $1 = n + 2 - n - 1 \leq i_3 + i_4 - n - 1 \leq n + n - n - 1 = n - 1$. Для удобства обозначим $\pi(i_3, i_4) = \begin{cases} i_3 + i_4 - 1, \text{если } i_3 + i_4 \leq n+1 \\ i_3 + i_4 - n - 1, \text{иначе} \end{cases}$, $i_3, i_4 = \overline{1,n}$. Как показано выше, $\pi(i_3, i_4) \in \{1,...,n\}$, $i_3, i_4 = \overline{1,n}$. Отсюда

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n} x_{i_1 i_2 i_3 i_4} = \sum_{i_1=1}^{n} x_{i_1 i_1 i_3 i_4} = x_{\pi(i_3,i_4)\pi(i_3,i_4) i_3 i_4} = 1, \; i_3, i_4 = \overline{1,n}.$$

Таким образом, $\{x_{F_{N(s)}}\}$ удовлетворяет (4.30). Обозначим $\Pi'(i_1, i_4) = \{i_3 \mid i_1 = \pi(i_3, i_4), i_3 \in \{1,...,n\}\}$, $i_1, i_4 = \overline{1,n}$. Пусть $i_1, i_4 \in \{1,...,n\}$. Покажем от противного, что $|\Pi'(i_1, i_4)| \leq 1$. Предположим что найдутся $i'_3, i''_3 \in \Pi'(i_1, i_4)$, $i'_3 \neq i''_3$, тогда если $i'_3, i''_3 \leq n - i_4 + 1$ или $i'_3, i''_3 > n - i_4 + 1$, то по построению $\pi(i'_3, i_4) \neq \pi(i''_3, i_4)$; если $i'_3 \leq n - i_4 + 1 < i''_3$, то с учетом $i''_3 - i'_3 \leq n - 1$ выполняется $\pi(i'_3, i_4) = i'_3 + i_4 - 1$, $\pi(i''_3, i_4) = i''_3 + i_4 - n - 1 \leq i'_3 + n - 1 + i_4 - n - 1 = \pi(i'_3, i_4) - 1$ и $\pi(i'_3, i_4) \neq \pi(i''_3, i_4)$, получаем противоречие, предположение неверно и следовательно $|\Pi'(i_1,i_4)|\le 1$. Рассмотрим множество

$$\{\pi(i_3,i_4)\,|\,i_3=\overline{1,n}\}=$$

$$\{i_3+i_4-1\,|\,1\le i_3\le n-i_4+1, i_3\in N\}\cup\{i_3+i_4-n-1\,|\,n-i_4+1<i_3\le n, i_3\in N\}=$$

$$\{i_4,...,n\}\cup\{1,...,i_4-1\}=\{1,...,n\}.$$

Тогда $\Pi'(i_1,i_4)\ne\varnothing$, и следовательно $|\Pi'(i_1,i_4)|=1$. Отсюда

$$\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}x_{i_1i_2i_3i_4}=\sum_{i_3=1}^{n}x_{i_1i_1i_3i_4}=\sum_{i_3\in\Pi'(i_1,i_4)}x_{i_1i_1i_3i_4}=1,\ i_1,i_4=\overline{1,n}.$$

Таким образом, $\{x_{F_{N(s)}}\}$ удовлетворяет (4.31). Обозначим $\Pi''(i_1,i_3)=\{i_4\,|\,i_1=\pi(i_3,i_4), i_4\in\{1,...,n\}\}$, по аналогично можно показать, что $|\Pi''(i_1,i_3)|=1$, $i_1,i_3=\overline{1,n}$. Отсюда

$$\sum_{i_1=1}^{n}\sum_{i_4=1}^{n}x_{i_1i_2i_3i_4}=\sum_{i_4=1}^{n}x_{i_2i_2i_3i_4}=\sum_{i_4\in\Pi''(i_2,i_3)}x_{i_2i_2i_3i_4}=1,\ i_2,i_3=\overline{1,n}.$$

Таким образом, $\{x_{F_{N(s)}}\}$ удовлетворяет (4.32). Следовательно, $\{x_{F_{N(s)}}\}$ является допустимым решением задачи w. При этом

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n}c_{i_1i_2i_3i_4}x_{i_1i_2i_3i_4}<\Delta\le\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n}c_{i_1i_2i_3i_4}x^*_{i_1i_2i_3i_4}.$$

Получаем противоречие, предположение неверно и условие (4.35) выполняется.

Далее обозначим $y^*_{ijk}=x^*_{iijk}$, $i,j,k=\overline{1,n}$. В силу условия (4.35) и ограничений (4.30)-(4.33) набор y^*_{ijk}, $i,j,k=\overline{1,n}$, является допустимым решением задачи (4.27)-(4.29),(4.24),(4.25) и по построению

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n}c_{i_1i_2i_3i_4}x^*_{i_1i_2i_3i_4}=\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}e_{ijk}y^*_{ijk}.$$

Далее покажем от противного, что y^*_{ijk}, $i,j,k=\overline{1,n}$, является оптимальным решением задачи (4.27)-(4.29),(4.24),(4.25). Предположим, что существует допустимое решение y'_{ijk}, $i,j,k=\overline{1,n}$, задачи (4.27)-(4.29),(4.24),(4.25), что

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk} y'_{ijk} < \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk} y^*_{ijk} .$$

Тогда опередим $\{x'_{F_{N(s)}}\}$ следующим образом

$$x'_{i_1 i_1 i_3 i_4} = y_{i_1 i_3 i_4} ,\ i_1, i_3, i_4 = \overline{1,n} ,$$

$$x'_{i_1 i_2 i_3 i_4} = 0 ,\ i_1 \neq i_2 ,\ i_1, i_2, i_3, i_4 = \overline{1,n} .$$

Несложно увидеть, что в силу условий (4.27)-(4.29),(4.24) построенная многоиндексная матрица $\{x'_{F_{N(s)}}\}$ является допустимым решением задачи w, и при этом

$$\sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n} c_{i_1 i_2 i_3 i_4} x'_{i_1 i_2 i_3 i_4} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk} y'_{ijk} < \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} e_{ijk} y^*_{ijk} = \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\sum_{i_3=1}^{n}\sum_{i_4=1}^{n} c_{i_1 i_2 i_3 i_4} x^*_{i_1 i_2 i_3 i_4} .$$

Получаем противоречие, предположение неверно и y^*_{ijk}, $i, j, k = \overline{1,n}$, является оптимальным решением задачи (4.27)-(4.29),(4.24),(4.25).

Таким образом, *NP*-трудный класс трехиндексных планарных задач о назначениях (4.27)-(4.29),(4.24),(4.25) полиномиально сводим к классу $W_Z(M)$. Отсюда класс $W_Z(M)$ является *NP*-трудным [52]. *Лемма доказана.*

Теорема 4.7. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $P^* \mid P^* - Z \mid P^* - Z$ сводимым к классу W_{Graph}, необходимо и достаточно, чтобы множество *M* было 2-вложенным, в противном случае *P*=*NP*.

Доказательство. Пусть множество $M \subseteq 2^{N(s)}$ является 2-вложенным, то, согласно следствию 4.5, класс $W(M)$ является $L \mid L - equal \mid L - edge$ сводимым к классу W_{Graph} и, следовательно, является $P^* \mid P^* - Z \mid P^* - Z$ сводимым к классу W_{Graph}.

Далее пусть множество *M* не является 2-вложенным, тогда, согласно лемме 4.6, выполняется одно из следующих условий

– $\{\{1,2\},\{1,3\},\{2,3\}\} \prec M$,

– $\{\{1\},\{2\},\{3\}\} \prec M$,

– $\{\{1,2\},\{2,3\},\{1,4\}\} \prec M$, и, согласно леммам 4.9, 4.10, класс $W_Z(M)$ является NP-трудным. Отсюда на основании следствия 4.6 класс $W(M)$ не является $P^* | P^* - Z | P^* - Z$ сводимым к классу W_{Graph}, в противном случае $P = NP$. *Теорема доказана.*

Полученный результат является дополнительным обоснованием оптимальности условия 2-вложенности, как условия сводимости класса $W(M)$ к классу W_{Graph}. Действительно, согласно теоремам 4.2 и 4.4, найденное условие 2-вложенности является необходимыми и достаточным условием $t_1 | t_2 - equal | t_3 - edge$ сводимости. При этом оказывается, что в случае полиномиальных вычислительных затрат применение более общего класса схем сведения, описываемых в рамках концепции $P^* | P^* - Z | P^* - Z$ сводимости, не приводит к расширению класса сводимых задач, иначе $P = NP$.

Выводы

Предложена схема $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости (сводимости с сохранением соответствия ребер) класса задач линейного программирования к классу задач поиска потока в сети W_{Graph}. При исследовании $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости класса $W(M)$ к классу W_{Graph} установлено:

- для того, чтобы класс $W(M)$ являлся $L \mid L - equal \mid L - edge$ сводимым к классу W_{Graph}, достаточно, чтобы множество M являлось 2-вложенным;
- для того, чтобы класс $W(M)$ являлся $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Graph}, необходимо, чтобы множество M является 2-вложенным;
- пусть множество M является 2-вложенным, тогда существует алгоритм решения задач класса $W(M)$ и систем класса $D(M)$, требующий $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ и $O(|E_{N(s)}|^2 \log |E_{N(s)}|)$ вычислительных операций соответственно, данный алгоритм применим при исследовании целочисленного случая;
- пусть множество M является 2-вложеннм, тогда существует алгоритм решения задач класса $S(M)$, требующий $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ вычислительных операций;
- пусть множество $M \cup \tilde{H}$ является 2-вложенным, тогда существуют алгоритмы решения задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, требующие $O(|E_{N(s)}|^3 \log |E_{N(s)}| \log p)$ и $O(|E_{N(s)}|^2 \log |E_{N(s)}| \log p)$ вычислительных операций соответственно.

При исследовании $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости класса $W(M)$ к классу W_{Tree} установлено:

- для того, чтобы класс $W(M)$ являлся $L \mid L - equal \mid L - edge$ сводимым к классу W_{Graph}, достаточно, чтобы множество M было 1-вложенным;
- для того, чтобы класс $W(M)$ являлся $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Graph}, необходимо, чтобы множество M являлось 1-вложенным;
- пусть множество M является 1-вложеннм, тогда существует алгоритм решения задач класса $W(M)$ и систем класса $D(M)$, требующий $O(|E_{N(s)}|^2$ и $O(|E_{N(s)}|)$

вычислительных операций соответственно, данный алгоритм также применим при исследовании целочисленного случая;

– пусть множество $M \cup \widetilde{H}$ является 1-вложенным, тогда существуют алгоритмы решения задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, требующие $O(|E_{N(s)}|^2 \log p)$ и $O(|E_{N(s)}| \log p)$ вычислительных операций соответственно.

Сформулирована схема $t_1 | t_2 - Z | t_3 - Z$ сводимости класса $W(M)$ к классу W_{Graph}, являющаяся обобщением схемы $t_1 | t_2 - equal | t_3 - edge$ сводимости. Установлено, что для того, чтобы класс $W(M)$ являлся $P^* | P^* - Z | P^* - Z$ сводимым к классу W_{Graph}, необходимо и достаточно, чтобы множество *M* было 2-вложенным, в противном случае *P=NP*.

Глава 5. Сводимость с сохранением соответствия циклов

Глава посвящена исследованию сводимости класса многоиндексных задач к классу задач поиска потока минимальной стоимости в сети с использованием схемы сведения с сохранением соответствия циклов [5, 7, 9, 15, 22]. Особенностью рассматриваемой концепции сводимости является существование соответствия между переменными исходной задачи и циклами вспомогательной сети. Предлагаемая схема сведения гарантирует, что произвольный оптимальный поток вспомогательной сети будет определять такое оптимальное решение исходной задачи, в котором переменным присваиваются значения потока вдоль соответствующих простых циклов. В рамках рассматриваемой схемы сведения в данной главе найдены условия сводимости класса многоиндексных задач линейного программирования к классу задач поиска потока в сети. Предлагается конструктивная схема сведения. На основе предложенной схемы сведения строятся алгоритмы решения многоиндексных задач, удовлетворяющих условиям сводимости. Результаты применяются при исследовании ряда *NP*-трудных целочисленных многоиндексных задач.

5.1. Концепция $t_1 \mid t_2 - equal \mid t_3 - cycle$ сводимости

Введем схему $t_1 | t_2 - equal | t_3 - cycle$ сводимости (сводимости с сохранением соответствия циклов), которая будет применяться в данной главе при исследовании сводимости многоиндексных задач к классу задач поиск потока в сети. Основные особенности предлагаемой схемы:

– при построении вспомогательной сети пропускные способности дуг определяются равными соответствующим свободным коэффициентам двусторонних неравенств системы ограничений (такая особенность сведения отражается в процедуре, обозначаемой $equal$);

– при построении решения исходной задачи предполагается существование соответствия между ее переменными и простыми циклами вспомогательной сети, при этом оптимальный поток вспомогательной сети определяет такое оптимальное решение исходной задачи, в котором переменным присваивается значение потока вдоль соответствующих циклов сети, величина потока вдоль простых циклов определяется через циклическую декомпозицию потока (такая особенность сведения отражается в процедуре, обозначаемой $cycle$).

Таким образом, рассматриваемая схема сведения является частным случаем схемы описанной в определении 3.1, для которой $s_2 = equal$, $s_3 = cycle$.

Определение 5.1. Пусть W – класс задач линейного программирования с двусторонней системой линейных неравенств. Будем говорить, что класс W является $t_1 | t_2 - equal | t_3 - cycle$ сводимым к классу W_{Graph}, если класс W является $t_1 | t_2 | t_3$ сводимым к классу W_{Graph}, и дополнительно произвольная задача $w = w(A, b^-, b^+, c) \in W$, а также соответствующая ей задача $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Graph}$ удовлетворяют следующим условиям. Найдутся такие инъективные функции $\alpha : \{1, ..., nrow(A)\} \to A_G$, $\beta : \{1, ..., ncol(A)\} \to C(G)$, что

– $l_{\alpha(i)} = b_i^-, u_{\alpha(i)} = b_i^+$, $i = \overline{1, nrow\,(A)}$; $l_{(u,v)} = 0, u_{(u,v)} = b^*$, $(u,v) \in A_G \setminus \{\alpha(i) \mid i = \overline{1, nrow(A)}\}$, где $b^* = \sum_{k=1}^{nrow(A)} b_k^+$;

– если циркуляция x_{ij}, $(i,j) \in A_G$, является оптимальным (допустимым) решением задачи z и y_C, $C \in C(G)$, является циклической декомпозицией данной циркуляции, то $(y_{\beta(1)}, ..., y_{\beta(ncol(A))})$ будет являться оптимальным (допустимым) решением задачи w, соответствующим решению задачи z.

Замечание. В качестве величины b^* выбрана величина, значение которой эквивалентно отсутствию (в данном контексте) верхней пропускной способности дуги. Схематично концепция $t_1 | t_2 - equal | t_3 - edge$ сводимости представлена на рис. 5.1.

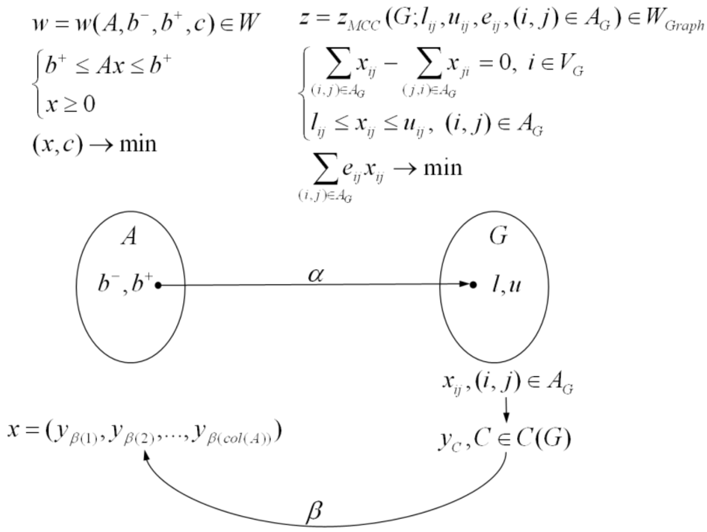

Рисунок 5.1.

Согласно определению 5.1, в случае $t_1 | t_2 - equal | t_3 - cycle$ сводимости класса W к классу W_{Graph} гарантируется, что если $w \in W$, $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}(i,j) \in A_G) \in W_{Graph}$ и z является задачей, соответствующей задаче w, то при построении задачи поиска потока минимальной стоимости z пропускные способности в задаче определяются через свободные коэффициенты задачи w, а решение задачи w находится через подмножество компонент циклической декомпозиции решения задачи z. Тогда, согласно утверждениям 2.2 и 2.6, можно предложить алгоритм решения задачи w, основанный на решении соответствующей задачи z и имеющий вычислительную сложность $O(t_1 + t_2 + t_3 + \alpha(O(|V_G|), O(|V_G| + |A_G|)) + \beta(O(|V_G|), O(|V_G| + |A_G|) + O(|V_G||A_G|))$ вычислительных операций, где $\alpha(n,m)$ и $\beta(n,m)$ – количество вычислительных операций, требуемых для решения задачи поиска максимального потока z_{MF} и задачи поиска потока минимальной стоимости заданной величины z_{MCF} соответственно в сети с n вершинами и m дугами. Как было отмечено ранее, обзор оценок вычислительной сложности для известных потоковых алгоритмов можно найти, например, в [149, 151, 172].

Далее в данной главе будут рассмотрены вопросы $t_1 | t_2 - equal | t_3 - cycle$ сводимости класса $W(M,H)$ к классу W_{Graph}.

5.2. Многоиндексные задачи с декомпозиционной структурой

Согласно следующей теореме, выделение подклассов многоиндексных задач, сводимых (в рамках рассматриваемой концепции сводимости) к потоковым задачам, позволяет выделить в NP-трудном классе целочисленных многоиндексных задач полиномиально разрешимые подклассы.

Теорема 5.1. Пусть $M, H \subseteq 2^{N(s)}$. Если класс $W(M,H)$ является $P^* | P^* - equal | P^* - cycle$ сводимым к классу W_{Graph}, то класс $W_Z(M,H)$ разрешим за полиномиальное время.

Доказательство. Пусть выполняются условия теоремы. Тогда для каждой задачи $w \in W(M,H)$ можно построить соответствующую ей задачу $z \in W_{Graph}$. В силу $P^* | P^* - equal | P^* - cycle$ сводимости количество вершин и дуг, а также значения пропускных способностей в соответствующей потоковой задаче z полиномиально зависят от размера задачи w. При этом, согласно определению 5.1, пропускные способности дуг в задаче z при целочисленных свободных коэффициентах двусторонних ограничений задачи w также являются целочисленными. Для общности отметим, что если $w \in W_Z(M,H)$ содержит нецелочисленные свободные коэффициенты, то в силу целочисленности коэффициентов матрицы системы ограничений задачи w, задача w при помощи округления параметров может быть преобразована к эквивалентной постановке с целочисленными параметрами.

Как известно, матрица системы ограничений задачи линейного программирования z является абсолютно унимодулярной [51]. Тогда в силу целочисленности пропускных способностей дуг, целочисленное решение задачи z может быть найдено за полиномиальное время [102]. Согласно следствию 2.3 и утверждению 2.6, целочисленная декомпозиция целочисленного решения задачи z может быть найдена за полиномиальное время. Тогда, согласно определению 5.1, построенное за полиномиальное время решение задачи w также будет являться целочисленным. Следовательно, найденное решение также будет являться решением задачи w_Z. Таким образом, класс задач целочисленного линейного программирования $W_Z(M,H)$ разрешим за полиномиальное время. *Теорема доказана.*

По аналогии со следствием 4.6 можно доказать следующий результат.

Следствие 5.1. Пусть $M,H \subseteq 2^{N(s)}$. Если класс $W_Z(M,H)$ является *NP*-трудным, то класс $W(M,H)$ не является $P^* \mid P^* - equal \mid P^* - cycle$ сводимым к классу W_{Graph}, в противном случае $P = NP$.

Отметим, что конструктивная схема $L \mid L - equal \mid L - edge$ сводимости, описываемая при доказательстве теоремы 4.2**,** обладает следующей особенностью. В случае 2-вложенности множества M задаче $w \in W(M)$ ставится в соответствие задача $z \in W_{Graph}$. Согласно определению 4.1, оптимальный поток задачи z определяет такое оптимальное решение задачи w, в котором переменным присваивается значение потока вдоль соответствующих дуг, определяемых функцией β. При этом в построенной при доказательстве теоремы 4.2 задаче z через каждую из дуг, определяемых функцией β, проходит единственный простой цикл. Отсюда справедливо следующее следствие.

Следствие 5.2. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $L \mid L - equal \mid L - cycle$ сводимым к классу W_{Graph}, достаточно, чтобы множество *M* было 2-вложенным.

На основании теоремы 5.1 и лемм 4.6, 4.9, 4.10 справедливо следствие.

Следствие 5.3. Пусть $M \subseteq 2^{N(s)}$. Для того, чтобы класс $W(M)$ являлся $P^* \mid P^* - equal \mid P^* - cycle$ сводимым к классу W_{Graph}, необходимо и достаточно, чтобы множество M было 2-вложенным, в противном случае P=NP.

Сформулированные выше следствия 5.2 и 5.3 обуславливают дальнейшие исследования сводимости частных подклассов класса $W(M)$. Таким образом, основное внимание в данной главе будет уделено вопросам исследования классов многоиндексных задач $W(M,H)$, которые могут быть сведены согласно концепции, введенной в определении 5.1, к потоковым задачам. Покажем, что таким подклассом является класс многоиндексных транспортных задач со специальной декомпозиционной структурой.

Определение 5.2. Пусть $M \subseteq 2^{N(s)}$ и $f_1,..., f_k$ – разбиение множества $N(s)$. Будем говорить, что множество M является $f_1,..., f_k$-декомпозиционным, если $M \subseteq \{ f_i \mid i = \overline{1,k}\} \cup \{ f_i \cup f_{i+1} \mid i = \overline{1,k-1}\}$.

Лемма 5.1. Пусть $f_1,..., f_k$ – разбиение множества $N(s)$, тогда $\sum_{i=1}^{k} | E_{f_i} | \le | E_{N(s)} |$.

Доказательство. Пусть $f_1,..., f_k$ – разбиение множества $N(s)$. Тогда $| E_{f_i} | = \prod_{j \in f_i} n_j$, $i = \overline{1,k}$, и $| E_{N(s)} | = \prod_{l=1}^{s} n_l = \prod_{i=1}^{k} \prod_{j \in f_i} n_j = \prod_{i=1}^{k} | E_{f_i} |$. Обозначим для удобства $| E_{f_i} | = m_i$, $i = \overline{1,k}$, тогда $| E_{N(s)} | = \prod_{i=1}^{k} m_i$. Так как $n_l \ge 2$, $l = \overline{1,s}$, то $m_i \ge 2$, $i = \overline{1,k}$.

Несложно увидеть, что $k \le 2^{k-1} \le \dfrac{\prod_{i=1}^{k} m_k}{\max_{i=\overline{1,k}} m_i}$. Далее, $\sum_{i=1}^{k} m_i \le k \max_{i=\overline{1,k}} m_i \le \dfrac{\prod_{i=1}^{k} m_k}{\max_{i=\overline{1,k}} m_i} \max_{i=\overline{1,k}} m_i = \prod_{i=1}^{k} m_k$. *Лемма доказана.*

Лемма 5.2. Пусть $f_1,..., f_k$ – разбиение множества $N(s)$, тогда $\sum_{i=1}^{k-1} | E_{f_i} || E_{f_{i+1}} | \le | E_{N(s)} |$.

Доказательство. Пусть $f_1,...,f_k$ – разбиение множества $N(s)$. По аналогии с доказательством леммы 5.1 обозначим для удобства $|E_{f_i}|=m_i \ge 2$, $i=\overline{1,k}$, тогда $|E_{N(s)}|=\prod_{i=1}^{k} m_i$. Очевидно при $k=1$ лемма верна. Далее пусть $k \ge 2$.

Несложно увидеть, что $k-1 \le 2^{k-2} \le \dfrac{\prod_{i=1}^{k} m_k}{\max\limits_{i=\overline{1,k-1}} m_i m_{i+1}}$. Далее $\sum_{i=1}^{k-1} m_i m_{i+1} \le (k-1)\max\limits_{i=\overline{1,k-1}} m_i m_{i+1} \le \dfrac{\prod_{i=1}^{k} m_k}{\max\limits_{i=\overline{1,k-1}} m_i m_{i+1}} \max\limits_{i=\overline{1,k-1}} m_i m_{i+1} = \prod_{i=1}^{k} m_k$. *Лемма доказана.*

Теорема 5.2. Пусть $M,H \subseteq 2^{N(s)}$, $f_1,...,f_k$ – разбиение множества $N(s)$. Для того чтобы класс $W(M,H)$ являлся $L|L-equal\,\||E_{N(s)}|^2-cycle$ сводимым к классу W_{Graph} достаточно, чтобы множества $\tilde{M}$ и H были $f_1,...,f_k$-декомпозиционными.

Доказательство. Пусть $f_1,...,f_k$ – разбиение множества $N(s)$ и множества $\tilde{M}$, H являются $f_1,...,f_k$-декомпозиционными. Рассмотрим произвольную задачу $w=w(s;M;n_1,...,n_2;\{a_{F_{\bar{f}}}\},\{b_{F_{\bar{f}}}\}, f\in M;\{c_{F_{N(s)}}\})\in W(M,H)$. Согласно определению 5.2 выполняется
$M \subseteq \{\overline{f_i}\,|\,i=\overline{1,k}\}\cup\{\overline{f_i \cup f_{i+1}}\,|\,i=\overline{1,k-1}\}, H \subseteq \{f_i\,|\,i=\overline{1,k}\}\cup\{f_i \cup f_{i+1}\,|\,i=\overline{1,k-1}\}$. Тогда заданы $|H|$ многоиндексных матриц коэффициентов $\{d_{F_f}\}$, $f\in H$, что $c_{F_{N(s)}}=\sum_{f\in H} d_{(F_{N(s)})_f}$. Не уменьшая общности, будем считать, что $M=\{\overline{f_i}\,|\,i=\overline{1,k}\}\cup\{\overline{f_i \cup f_{i+1}}\,|\,i=\overline{1,k-1}\}$, в противном случае добавим недостающие двусторонние неравенства, в качестве нижней границы которых выберем ноль, в качестве верхней – достаточно большую величину, например $\sum_{F_{\overline{f'}}\in E_{\overline{f'}}} b_{F_{\overline{f'}}}$, где f' – произвольный элемент из M. Также, не уменьшая общности, будем считать, что $H=\{f_i\,|\,i=\overline{1,k}\}\cup\{f_i \cup f_{i+1}\,|\,i=\overline{1,k-1}\}$, в противном случае пусть $d_{F_f}=0$, $F_f\in E_f$, $f\in(\{f_i\,|\,i=\overline{1,k}\}\cup\{f_i \cup f_{i+1}\,|\,i=\overline{1,k-1}\})\setminus H$. Далее приведем процедуру построения задачи $z=z_{MCC}(G;l_{ij},u_{ij},e_{ij},(i,j)\in A_G)\in W_{Graph}$, соответствующей задаче w.

Построим ориентированный граф G с множеством вершин $V_G=\{v'_{F_{f_i}}, v''_{F_{f_i}} \mid F_{f_i}\in E_{f_i}, i=\overline{1,k}\}\cup\{s,t\}$ и множеством дуг $A_G=A_1\cup A_2\cup A_3$, где

– $A_1=\{(v'_{F_{f_i}}, v''_{F_{f_i}}) \mid F_{f_i}\in E_{f_i}, i=\overline{1,k}\}$;

– $A_2=\{(v''_{F_{f_i}}, v'_{F_{f_{i+1}}}) \mid F_{f_i}\in E_{f_i}, F_{f_{i+1}}\in E_{f_{i+1}}, i=\overline{1,k-1}\}$;

– $A_3=\{(s,v'_{F_{f_1}}) \mid F_{f_1}\in E_{f_1}\}\cup\{(v''_{F_{f_k}},t) \mid F_{f_k}\in E_{f_k}\}\cup\{(t,s)\}$.

Определим функцию α следующим образом:

– каждому ограничению $d(w,\overline{f_i},F_{f_i})$ поставим в соответствие дугу $(v'_{F_{f_i}}, v''_{F_{f_i}})$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{f_i}}$ и $b_{F_{f_i}}$ соответственно, $F_{f_i}\in E_{f_i}$, $i=\overline{1,k}$;

– каждому ограничению $d(w,\overline{f_i\cup f_{i+1}},F_{f_i\cup f_{i+1}})$ поставим в соответствие дугу $(v''_{(F_{f_i\cup f_{i+1}})_{f_i}}, v'_{(F_{f_i\cup f_{i+1}})_{f_{i+1}}})$, таким образом, в задаче z нижняя и верхняя границы данной дуги будут составлять $a_{F_{f_i\cup f_{i+1}}}$ и $b_{F_{f_i\cup f_{i+1}}}$ соответственно, $F_{f_i\cup f_{i+1}}\in E_{f_i\cup f_{i+1}}$, $i=\overline{1,k-1}$.

Определим стоимости дуг в задаче z следующим образом:

– $e_{v'_{F_{f_i}}v''_{F_{f_i}}}=d_{F_{f_i}}, F_{f_i}\in E_{f_i}, i=\overline{1,k}$,

– $e_{v''_{F_{f_i}}v'_{F_{f_{i+1}}}}=d_{F_{f_i}F_{f_{i+1}}}$, $F_{f_i}\in E_{f_i}$, $F_{f_{i+1}}\in E_{f_{i+1}}$, $i=\overline{1,k-1}$,

– $e_{ij}=0,\ (i,j)\in A_3$.

По условию $f_1,...,f_k$ является разбиением множества $N(s)$. Отсюда каждый элемент $F_{N(s)}\in E_{N(S)}$ однозначным образом представим в виде $F_{N(s)}=F_{f_1}...F_{f_k}$, где $F_{f_i}=(F_{N(s)})_{f_i}$, $i=\overline{1,k}$. Введем следующее обозначение $C_{F_{N(s)}}=(s,v'_{F_{f_1}},v''_{F_{f_1}},...,v'_{F_{f_k}},v''_{F_{f_k}},t,s)$. По построению $C(G)=\{C_{F_{N(s)}} \mid F_{N(s)}\in E_{N(s)}\}$. Тогда определим функцию β следующим образом: каждой переменной $x_{F_{N(s)}}$ поставим в соответствие простой цикл $C_{F_{N(s)}}$, $F_{N(s)}\in E_{N(s)}$.

Покажем, что построенная задача z совместна тогда и только тогда, когда совместна исходная задача w. Действительно, пусть $x_{F_{N(s)}}$, $F_{N(s)}\in E_{N(s)}$ – допустимое решение системы ограничений задачи w. Определим $y_{C_{F_{N(s)}}}=x_{F_{N(s)}}$, $F_{N(s)}\in E_{N(s)}$, далее пусть $z_{ij} = \sum_{C\in C(G)|(i,j)\in C} y_C$, $(i,j)\in A_G$. Набор $z_{ij},(i,j)\in A_G$, будет являться циркуляцией в графе G и по построению:

$$z_{v'_{F_{f_i}} v''_{F_{f_i}}} = \sum_{F_{\overline{f_i}} \in E_{\overline{f_i}}} x_{F_{\overline{f_i}} F_{f_i}} ,\ F_{f_i} \in E_{f_i} ,\ i=\overline{1,k} ;$$

$$z_{v''_{(F_{f_i\cup f_{i+1}})_{f_i}} v'_{(F_{f_i\cup f_{i+1}})_{f_{i+1}}}} = \sum_{F_{\overline{f_i\cup f_{i+1}}} \in E_{\overline{f_i\cup f_{i+1}}}} x_{F_{\overline{f_i\cup f_{i+1}}} F_{f_i\cup f_{i+1}}} ,\ F_{f_i\cup f_{i+1}} \in E_{f_i\cup f_{i+1}} ,\ i=\overline{1,k-1}.$$

Тогда в соответствии с введенной функцией α набор $z_{ij},(i,j)\in A,$ будет являться допустимой циркуляцией задачи z .

Далее, пусть z_{ij}, $(i,j)\in A_G$ – допустимая циркуляция задачи v . Согласно теореме 2.4, для данной циркуляции существует ее циклическая декомпозиции y_C , $C\in C(G)$. Согласно введенной функции β, построим следующий набор значений переменных: $x_{F_{N(s)}} = y_{C_{F_{N(s)}}}$, $F_{N(s)} \in E_{N(s)}$. Тогда

$$a_{F_{f_i}} \le \sum_{F_{\overline{f_i}} \in E_{\overline{f_i}}} y_{C_{F_{\overline{f_i}} F_{f_i}}} = z_{v'_{F_{f_i}} v''_{F_{f_i}}} \le b_{F_{f_i}} ,\ F_{f_i} \in E_{f_i} ,\ i=\overline{1,k} ,$$

$$a_{F_{f_i\cup f_{i+1}}} \le \sum_{F_{\overline{f_i\cup f_{i+1}}} \in E_{\overline{f_i\cup f_{i+1}}}} y_{C_{F_{\overline{f_i\cup f_{i+1}}} F_{f_i\cup f_{i+1}}}} = z_{v''_{(F_{f_i\cup f_{i+1}})_{f_i}} v'_{(F_{f_i\cup f_{i+1}})_{f_{i+1}}}} \le b_{F_{f_i\cup f_{i+1}}} ,\ F_{f_i\cup f_{i+1}} \in E_{f_i\cup f_{i+1}} ,\ i=\overline{1,k-1}.$$

Отсюда построенный набор является допустимым решением задачи w.

Пусть z_{ij}, $(i,j)\in A_G$, – циркуляция минимальной стоимости в задаче z и y_C , $C\in C(G)$ – ее циклическая декомпозиция. Используя функцию β , построим следующий набор значений переменных $x_{F_{N(s)}} = y_{C_{F_{N(s)}}}$, $F_{N(s)} \in E_{N(s)}$, который (как уже было показано) является допустимым решением задачи w . Теперь покажем от противного, что построенный набор будет оптимальным решением задачи w . Действительно, пусть это не так и $x'_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$ – оптимальное решение задачи w . Тогда по предположению $\sum_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} > \sum_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}}$. По построению $\sum_{(i,j)\in A_G} e_{ij} z_{ij} = \sum_{(i,j)\in A_G} e_{ij} \sum_{C\in C(G)|(i,j)\in C} y_C =$ $= \sum_{C\in C(G)} y_C \sum_{(i,j)\in C} e_{ij} = \sum_{F_{N(s)}\in E_{N(s)}} y_{C_{F_{N(s)}}} \sum_{f\in H} d_{(F_{N(s)})_f} = \sum_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}}$. Определим $y'_{C_{F_{N(s)}}} = x'_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$, и $z'_{ij} = \sum_{C\in C(G)|(i,j)\in C} y'_C$, $(i,j)\in A_G$. Как уже было показано ранее, построенный таким образом набор z'_{ij}, $(i,j)\in A_G$, является допустимой циркуляцией задачи z . При этом $\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} = \sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} y'_{C_{F_{N(s)}}} = \sum\limits_{F_{N(s)}\in E_{N(s)}} y'_{C_{F_{N(s)}}} \sum\limits_{f\in H} d_{(F_{N(s)})_f} = \sum\limits_{C\in C(G)} y'_C \sum\limits_{(i,j)\in C} e_{ij} =$ $= \sum\limits_{(i,j)\in A_G} e_{ij} \sum\limits_{C\in C(G)|(i,j)\in C} y'_C = \sum\limits_{(i,j)\in A_G} e_{ij} z'_{ij}$. Тогда $\sum\limits_{(i,j)\in A_G} e_{ij} z_{ij} = \sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} > \sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} =$ $= \sum\limits_{(i,j)\in A_G} e_{ij} z'_{ij}$, получаем противоречие. Таким образом, предположение неверно, и построенный набор $x_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$, является оптимальным решением задачи w.

Далее проведем анализ сложности вычислительных процедур, связанных с построением задачи z и построением решения задачи w по решению задачи z. Заметим, что исходная задача w содержит $|E_{N(s)}|$ переменных. По построению граф G в задаче z содержит $|V| = 2 + 2\sum\limits_{i=1}^{k} |E_{f_i}|$ вершин и $|A| = 1 + |E_{f_1}| + |E_{f_k}| + \sum\limits_{i=1}^{k} |E_{f_i}| + \sum\limits_{i=1}^{k-1} |E_{f_i}||E_{f_{i+1}}|$ дуг. Применяя леммы 5.1 и 5.2, можно получить следующие оценки $|V| = O(|E_{N(s)}|)$, $|A| = O(|E_{N(s)}|)$.

Отсюда построение задачи z требует линейных от размера матрицы $Matr(w)$ вычислительных операций. Схема построения решения задачи w по решению задачи z определяется предложенным выше отображением β и связана с построением циклической декомпозиции циркуляции, которое, согласно утверждению 2.6, требует $O(|V||A|) = O(|E_{N(s)}|^2)$ вычислительных операций. Отсюда класс $W(M,H)$ является $L|L-equal\,\|E_{N(s)}|^2-cycle$ сводимым к классу W_{Graph}. *Теорема доказана.*

Конструктивная схема доказательства теоремы 5.2 позволят для задач класса $W(M,H)$ и для систем линейных неравенств класса $D(M)$, где $\tilde{M}$ и H являются $f_1,..., f_k$-декомпозиционными, $f_1,..., f_k$ – разбиение множества $N(s)$, предложить алгоритм решения, основанный на сводимости к классу W_{Graph}.

Алгоритм 5.1. Решение многоиндексных задач с декомпозиционной структурой.

Вход. Задача $w \in W(M,H)$ (система $w \in D(M)$), где $\tilde{M}$ и H являются $f_1,..., f_k$-декомпозиционными, $f_1,..., f_k$ – разбиение множества $N(s)$.

Шаг 1. Используя конструктивную схему, применяемую при доказательстве теоремы 5.2, построить задачу $z \in W_{Graph}$, соответствующую w. Перейти на шаг 2.

Шаг 2. Найти решение (допустимое решение) задачи z, используя алгоритм 2.2. (алгоритм 2.1). Если задача z несовместна, то w также несовместна, и алгоритм завершен; иначе переход на шаг 3.

Шаг 3. Построить циклическую декомпозицию решения, найденного на шаге 2, применяя алгоритм 2.7. Перейти на шаг 4.

Шаг 4. Используя отображение β, описанное при доказательстве теоремы 5.2, найти решение w согласно определению 5.1. Алгоритм завершен.

Матрица системы ограничений задачи $z \in W_{Graph}$ абсолютно унимодулярна, поэтому для совместной задачи z с целочисленными параметрами гарантируется существование целочисленного решения [51]. При этом конструктивная схема построения задачи z, применяемая при доказательстве теоремы 5.2, гарантирует целочисленность ее параметров при целочисленности параметров задачи (системы) w. Согласно следствию 2.3, для произвольной целочисленной циркуляции ее циклическая декомпозиция, построенная алгоритмом 2.7 (применяемым на шаге 3 алгоритма 5.1), является целочисленной. Отсюда на шаге 4 алгоритма 5.1 в случае целочисленности решения задачи z решение задачи (системы) w также является целочисленным. Отсюда алгоритм 5.1 применим также при исследовании целочисленных задач: $W_Z(M,H)$ и $D_Z(M)$. Как уже было отмечено при построении алгоритма 4.1, если $w \in W_Z(M,H)$ ($w \in D_Z(M)$) содержит нецелочисленные свободные коэффициенты, то в силу целочисленности коэффициентов матрицы системы ограничений w, задача (система) w при помощи округления параметров может быть преобразована к эквивалентной постановке с целочисленными параметрами.

Заметим, что для задачи $z = z_{MCC}(G; l_{ij}, u_{ij}, e_{ij}, (i,j) \in A_G) \in W_{Graph}$, соответствующей задаче w, согласно доказательству теоремы 5.2, выполняется $|V_G| = O(|E_{N(s)}|)$, $|A_G| = O(|E_{N(s)}|)$. Отсюда с учетом утверждений 2.1, 2.2, 2.6 можно сформулировать следующее следствие.

Следствие 5.4. Пусть $M, H \subseteq 2^{N(s)}$, множества $\tilde{M}$ и H являются $f_1,\ldots,f_k$-декомпозиционными, $f_1,\ldots,f_k$ – разбиение множества $N(s)$. Тогда алгоритм 5.1 решения задач класса $W(M,H)$ и класса $W_Z(M,H)$ (систем класса $D(M)$ и класса $D_Z(M)$) требует

$\alpha(O(|E_{N(s)}|),O(|E_{N(s)}|))$ + $\beta(O(|E_{N(s)}|),O(|E_{N(s)}|))$ + $O(|E_{N(s)}|^2)$ $(\alpha(O(|E_{N(s)}|),O(|E_{N(s)}|))+O(|E_{N(s)}|^2))$ вычислительных операций.

Здесь, напомним, $\alpha(n,m)$ и $\beta(n,m)$ – количество вычислительных операций алгоритма решения задачи поиска максимального потока и алгоритма решения задачи поиска потока минимальной стоимости заданной величины соответственно в сети с n вершинами и m дугами. Применим, как и при исследовании алгоритма 4.1, для поиска потока минимальной стоимости заданной величины алгоритм, предложенный в работе Орлина [172], для поиска максимального потока воспользуемся алгоритмом Слейтора-Тарьяна [181]. Тогда, согласно следствию 5.4, можно сформулировать следующее утверждение.

Утверждение 5.1. Пусть $M,H\subseteq 2^{N(s)}$, множества $\tilde{M}$ и H являются $f_1,...,f_k$-декомпозиционными, $f_1,...,f_k$ – разбиение множества $N(s)$. Тогда существует алгоритм решения задач класса $W(M,H)$ и класса $W_Z(M,H)$ (систем класса $D(M)$ и класса $D_Z(M)$), требующий $O(|E_{N(s)}|^2\log^2|E_{N(s)}|)$ ($O(|E_{N(s)}|^2\log|E_{N(s)}|)$) вычислительных операций.

Схема $L|L-equal\,||E_{N(s)}|^2-cycle$ сводимости, предложенная при доказательстве теоремы 5.2, также является схемой $P^*|P^*-equal|P^*-cycle$ сводимости. На основании теоремы 5.1 справедливо следующее следствие.

Следствие 5.5. Пусть $M,H\subseteq 2^{N(s)}$, множества $\tilde{M}$ и H являются $f_1,...,f_k$-декомпозиционными, $f_1,...,f_k$ – разбиение множества $N(s)$. Тогда класса задач целочисленного линейного программирования $W_Z(M,H)$ разрешим за полиномиальное время.

Таким образом, полученные результаты сводимости позволили выделить новый полиномиально разрешимый подкласс $W_Z(M,H)$, сформулированный в следствие 5.5, в *NP*-трудном классе многоиндексных задач целочисленного линейного программирования.

Приведем пример многоиндексной транспортной задачи с декомпозиционной структурой, на котором проиллюстрируем схему сведения, предложенную при доказательстве теоремы 5.2.

$$a_{j_3}^-\le\sum_{j_1\in J_1}\sum_{j_2\in J_2}x_{j_1j_2j_3}\le a_{j_3}^+,\ j_3\in J_3, \tag{5.1}$$

$$b_{j_2}^- \le \sum_{j_1 \in J_1} \sum_{j_3 \in J_3} x_{j_1 j_2 j_3} \le b_{j_2}^+ , \; j_2 \in J_2 , \tag{5.2}$$

$$c_{j_1}^- \le \sum_{j_2 \in J_2} \sum_{j_3 \in J_3} x_{j_1 j_2 j_3} \le c_{j_1}^+ , \; j_1 \in J_1 , \tag{5.3}$$

$$d_{j_1 j_2}^- \le \sum_{j_3 \in J_3} x_{j_1 j_2 j_3} \le d_{j_1 j_2}^+ , \; j_1 \in J_1 , \; j_2 \in J_2 , \tag{5.4}$$

$$x_{j_1 j_2 j_3} \ge 0 , \; j_1 \in J_1 , \; j_2 \in J_2 , \; j_3 \in J_3 , \tag{5.5}$$

$$\sum_{j_1 \in J_1} \sum_{j_2 \in J_2} \sum_{j_3 \in J_3} (u_{j_2 j_3} + v_{j_2} + w_{j_3}) x_{j_1 j_2 j_3} \to \min . \tag{5.6}$$

Задача (5.1)-(5.6) относится к классу многоиндексных задач $W(M)$, где $M = \{\{3\}, \{1,2\}, \{1,3\}, \{2,3\}\}$ и $s = 3$. Многоиндексная матрица стоимостей $\{c_{F_{N(s)}}\}$ задачи (5.1)-(5.6), определяемая выражением $u_{j_2 j_3} + v_{j_2} + w_{j_3}$, может быть представлена в виде $c_{F_{N(s)}} = \sum_{f \in H} d_{(F_{N(s)})_f}$, где $H = \{\{2\}, \{3\}, \{2,3\}\}$. Таким образом, задача (5.1)-(5.6) относится к классу $W(M, H)$. Рассмотрим разбиение $f_1 = \{1\}$, $f_2 = \{2\}$, $f_3 = \{3\}$ множества $N(3)$. Несложно увидеть, что $\tilde{M}, H \subseteq \{ f_i \mid i = \overline{1,3}\} \cup \{ f_i \cup f_{i+1} \mid i = \overline{1,2}\}$ и, согласно определению 5.2, множества $\tilde{M}$ и H являются $\{1\},\{2\},\{3\}$-декомпозиционными. Отсюда, согласно теореме 5.2, класс $W(M,H)$ является $L \mid L-equal \,\|E_{N(s)}|^2 -cycle$ сводимым к классу W_{Graph}, здесь $|E_{N(s)}| = |J_1\|J_2\|J_3|$. Далее рассмотрим схему сведения, используемой при конструктивном доказательстве теоремы 5.2, на примере задачи (5.1)-(5.6). Пусть $|J_1| = |J_2| = |J_3| = 2$. Приведем транспортную сеть, определяющую задачу поиска потока минимальной стоимости, соответствующую исходной многоиндексной задаче.

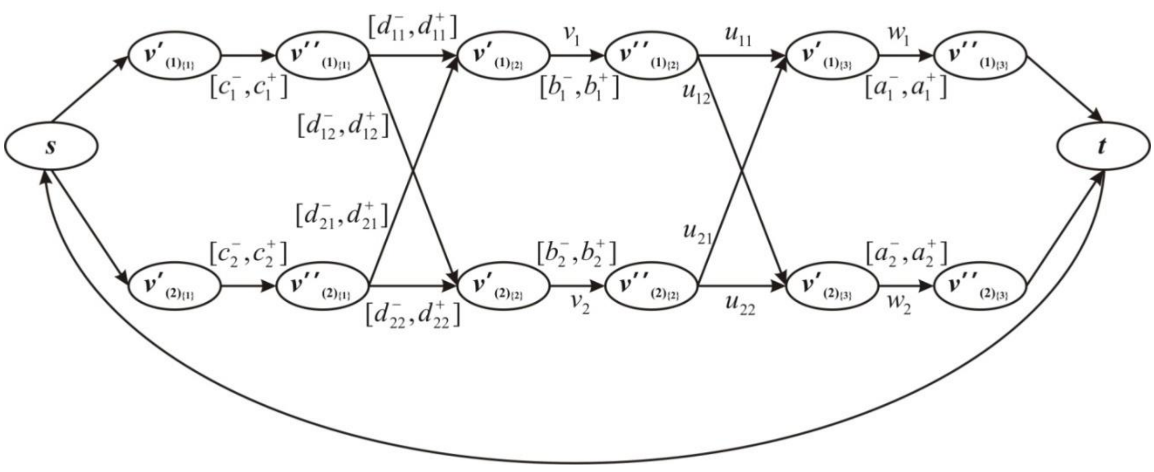

Рисунок 5.2.

На рисунке 5.2 для ряда дуг приведены их пропускные способности и стоимости. Дуги, у которых не указаны пропускные способностей, имеют нулевую нижнюю и неограниченную верхнюю пропускные способности. Дуги, у которых не указаны стоимости, имеют нулевую стоимость. Согласно доказательству теоремы 5.2 и алгоритму 5.1, решение исходной многоиндексной задачи определяется следующим образом: каждой переменной $x_{j_1 j_2 j_3}$ многоиндексной задачи (5.1)-(5.6) присваивается значение потока вдоль соответствующего простого цикла $(s, v'_{(j_1)_{\{1\}}}, v''_{(j_1)_{\{1\}}}, v'_{(j_2)_{\{2\}}}, v''_{(j_2)_{\{2\}}}, v'_{(j_3)_{\{3\}}}, v''_{(j_3)_{\{3\}}}, t, s)$, которое, в свою очередь, получается в результате циклической декомпозиции потока минимальной стоимости данной сети.

Далее проиллюстрируем применимость метода исследования декомпозиционных многоиндексных $W(M,H)$, основанного на сводимости к классу задач поиска потока в сети, при решении ряда рассмотренных ранее многоиндексных задач. Пусть $M \subseteq 2^{N(s)}$ и множества $\tilde{M}$ и H являются $f_1,..., f_k$-декомпозиционными, где $f_1,..., f_k$ – разбиение множества $N(s)$. Согласно теореме 5.2, класс задач $W(M,H)$ сводим к поиску потока в сети с $O(|E_{N(s)}|)$ вершинами и $O(|E_{N(s)}|)$ дугами. В рамках исследуемой схемы сведения пропускные способности дуг вспомогательной сети моделируют двусторонние ограничения исходной многоиндексной задачи. В случае несовместности ограничений потоковой задачи для ее исследования может быть применен, как и при обосновании утверждения 4.4, алгоритм поиска потока в несовместной сети.

Утверждение 5.2. Пусть $M \subseteq 2^{N(s)}$ и множество $\tilde{M}$ является $f_1,..., f_k$-декомпозиционным, где $f_1,..., f_k$ – разбиение множества $N(s)$. Тогда существует алгоритм решения задач класса $S(M)$, требующий $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ вычислительных операций.

Рассмотрим задачи $U_{\prec}(M,H)$ и $U_{\max\min}(M,H)$, $M,H \subseteq 2^{N(s)}$. Тогда аналогично утверждениям 4.5 и 4.6, используя утверждение 5.1, справедливо:

Утверждение 5.3. Пусть $M,H \subseteq 2^{N(s)}$ и множество $\tilde{M} \cup H$ является $f_1,..., f_k$-декомпозиционным, где $f_1,..., f_k$ – разбиение множества $N(s)$. Тогда существует алгоритм решения задач класса $U_{\prec}(M,H)$, требующий $O(|E_{N(s)}|^3 \log |E_{N(s)}| \log p)$ вычислительных операций.

Утверждение 5.4. Пусть $M,H \subseteq 2^{N(s)}$ и множество $\widetilde{M} \cup H$ является $f_1,\ldots,f_k$-декомпозиционным, где $f_1,\ldots,f_k$ – разбиение множества $N(s)$. Тогда существует алгоритм решения задач класса $U_{\max\min}(M,H)$, требующий $O(|E_{N(s)}|^2 \log|E_{N(s)}| \log p)$ вычислительных операций.

Дополнительно проиллюстрируем полученные результаты при решении ряда прикладных многоиндексных задач, поставленных в главе 1. Общая математическая модель проблемы объемно-календарного планирования переработки газового конденсата (1.19)-(1.22) относится к классу $D(M)$, где $M = \{\{i,j\},\{i,p\},\{j,k,p,t\}\}$. При этом $s=5$, $N(s)=\{i,j,k,p,t\}$ и $|E_{N(s)}|=|I\|J\|K\|P\|T|$. Рассмотрим разбиение $f_1=\{i\}$, $f_2=\{j\}$, $f_3=\{k,t\}$, $f_4=\{p\}$ множества $N(s)$. Заметим, что, согласно определению 5.2, множество $\widetilde{M}$ является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным, т.к. $\widetilde{M} \subseteq \{f_i \mid i=\overline{1,4}\} \cup \{f_i \cup f_{i+1} \mid i=\overline{1,3}\}$. Отсюда, согласно утверждению 5.1, система (1.19)-(1.22) может быть решена, используя $O(|I|^2|J|^2|K|^2|P|^2|T|^2 \log(|I\|J\|K\|P\|T|))$ вычислительных операций. Ранее были рассмотрены различные постановки задачи объемно-календарного планирования переработки газового конденсата, различающиеся выбором критерия. Задача (1.19)-(1.22),(1.23) относится к классу $W(M,H_1)$, где $H_1=\{\{k,p,t\}\}$. Легко увидеть, что, согласно определению 5.2, множество H_1 является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным, т.к. $H_1 \subseteq \{f_i \mid i=\overline{1,4}\} \cup \{f_i \cup f_{i+1} \mid i=\overline{1,3}\}$. Задача (1.19)-(1.22),(1.24) относится к классу $W(M,H_2)$, где $H_2=\{\{j,k\}\}$. Согласно утверждению 3.3, выполняется $W(M,H_2) \subseteq W(M,H_2')$, где $H_2'=\{\{j,k,t\}\}$ и множество H_2' является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным по определению 5.2. Задача (1.19)-(1.22),(1.25) относится к классу $W(M,H_3)$, где $H_3=\{\{i,j\}\}$. Согласно определению 5.2, множество H_3 является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным. Задача (1.19)-(1.22),(1.26) относится к классу $W(M,H_4)$, где $H_4=\{\{t\}\}$. Согласно утверждению 3.3, выполняется $W(M,H_4) \subseteq W(M,H_4')$, где $H_4'=\{\{k,t\}\}$ и H_4' является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным по определению 5.2. Задача (1.19)-(1.22),(1.27) относится к классу $W(M,H_5)$, где $H_5=\{k,p\}$. Согласно утверждению 3.3, выполняется $W(M,H_5) \subseteq W(M,H_5')$, где $H_5'=\{\{k,p,t\}\}$ и H_5' является $\{i\},\{j\},\{k,t\},\{p\}$-

декомпозиционным по определению 5.2. Задача (1.19)-(1.22),(1.28) относится к классу $W(M,H_6)$, где $H_6=\{\{t\},\{i,j\},\{j,k\},\{k,p\},\{k,p,t\}\}$. Согласно утверждению 3.3, выполняется $W(M,H_6)\subseteq W(M,H_6')$, где $H_6'=\{\{i,j\},\{j,k,t\},\{k,p,t\}\}$ и H_6' является $\{i\},\{j\},\{k,t\},\{p\}$-декомпозиционным по определению 5.2. Отсюда, согласно утверждению 5.1, рассмотренные задачи объемно-календарного планирования переработки газового конденсата система могут быть решены, используя $|I|^2|J|^2|K|^2|P|^2|T|^2\log^2(|I||J||K||P||T|)$ вычислительных операций.

Задача составления расписания занятий (1.29)-(1.36), как было показано в главе 3, относится к классу $W=(M,H)$, $M=\{\{i,j\},\{j,k\},\{k,t\},\{i,k,t\}\}$, $H=\{\{i,j\},\{i,t\},\{k,t\}\}$. При этом $s=4$, $N(s)=\{i,j,k,t\}$ и $|E_{N(s)}|=|I||J||K||T|$. Рассмотрим разбиение $f_1=\{j\}$, $f_2=\{i\}$, $f_3=\{t\}$, $f_4=\{k\}$ множества $N(s)$. Заметим, что, согласно определению 5.2, множества $\tilde{M}$ и H являются $\{j\},\{i\},\{t\},\{k\}$-декомпозиционным, т.к. $\tilde{M},H\subseteq\{f_i\,|\,i=\overline{1,4}\}\cup\{f_i\cup f_{i+1}\,|\,i=\overline{1,3}\}$. Отсюда, согласно утверждению 5.1, задача составления расписания занятий (1.29)-(1.36) может быть решена, используя $O(|I|^2|J|^2|K|^2|T|^2\log^2(|I||J||K||T|))$ вычислительных операций. Предлагаемый метод решения задачи составления расписания основан на полученных результатах сводимости многоиндексных задач к классу задач поиска потока в сети. Отметим, что в ряде случаев анализ схемы сведения для конкретных прикладных задач приводит к уточнению оценок сложности алгоритма. Так, применяя схему сведения, описанную при доказательстве теоремы 5.2, мы строим соответствующую задачу поиска потока $z=z_{MCC}(G;l_{ij},u_{ij},e_{ij},(i,j)\in A_G)\in W_{Graph}$. При этом $|V_G|=O(|I|+|J|+|K|+|T|)$, $|A_G|=O(|I|+|J|+|K|+|T|+\sum_{i\in I}|A_i|+\sum_{i\in I}|B_i|+\sum_{k\in K}|C_k|)$, где из множества дуг A_G предварительно исключили дуги с нулевой верхней пропускной способностью. Обозначим $d=|I|+|J|+|K|+|K|+|T|+\sum_{i\in I}|A_i|+\sum_{i\in I}|B_i|+\sum_{k\in K}|C_k|$. Отсюда, согласно следствию 5.4, задача составления расписания занятий (1.29)-(1.36) может быть решена, используя $O(d^2\log^2 d)$ вычислительных операций, что уточняет полученную в утверждение 5.1 общую оценку сложности алгоритма.

5.3. Декомпозиционные *NP*-трудные многоиндексные задачи

В общей постановке класс целочисленных многоиндексных транспортных задач является *NP*-трудным уже в трехиндексном случае [52]. Для задач данного класса не существует полиномиальных ε-приближенных алгоритмов, иначе *P*=*NP*, данный результат также справедлив уже в трехиндексном случае [131]. Более того, можно показать отсутствие эффективных приближенных алгоритмов для класса целочисленных многоиндексных задач, обладающих определенными декомпозиционными свойствами. Сформулируем используемое определение ε-приближенного алгоритма, приведенное в [52]. Пусть W – класс задач минимизации; V – алгоритм, который для любой задачи $w \in W$ возвращает допустимое решение $V(w)$ задачи w; $c(V(w))$ – значение критерия задачи w на допустимом решении $V(w)$; $OPT(w)$ – оптимальное значение критерия задачи w. Тогда алгоритм V будем называть ε-приближенным алгоритмом для задач класса W, где ε – неотрицательная константа, если выполняется следующее условие:

$$c(V(w)) \le (1+\varepsilon)OPT(w),\ w \in W.$$

Рассмотрим далее ряд декомпозиционных классов многоиндексных задач, связанные с ними вопросы сложности и вопросы построения приближенных алгоритмов, основанных на полученных результатах $t_1 \,|\, t_2 - equal \,|\, t_3 - cycle$ сводимости.

Утверждение 5.5. Пусть $M = \{\overline{f_i} \,|\, i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \,|\, i = \overline{1,k}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \ge 3$. Тогда для задач класса $W_Z(M,H)$ не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon \ge 0$, иначе $P = NP$.

Доказательство. Проведем доказательство от противного. Пусть выполняются условия утверждения, и для задач класса $W_Z(M,H)$ существует полиномиальный ε-приближенный алгоритм для некоторого $\varepsilon \ge 0$.

Рассмотрим трехиндексную аксиальную задачу о назначениях с матрицей стоимостей специального вида. Пусть $m \in N$, $u_{ij}^{(1)}, u_{ip}^{(2)}, u_{jp}^{(3)} \ge 0$, $i, j, p = \overline{1,m}$. Тогда задача ставится как следующая задача целочисленного линейного программирования:

$$\sum_{i=1}^{m}\sum_{j=1}^{m} y_{ijp} = 1,\ p = \overline{1,m}, \tag{5.7}$$

$$\sum_{i=1}^{m}\sum_{k=1}^{m} y_{ijp} = 1,\ j = \overline{1,m}, \tag{5.8}$$

$$\sum_{j=1}^{m}\sum_{p=1}^{m} y_{ijp}=1,\ i=\overline{1,m}, \tag{5.9}$$

$$y_{ijp}\in\{0,1\},\ i,j,p=\overline{1,m}, \tag{5.10}$$

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{p=1}^{m}(u_{ij}^{(1)}+u_{ip}^{(2)}+u_{jp}^{(3)})y_{ijp}\to\min. \tag{5.11}$$

В [131] было показано, что для класса задач (5.7)-(5.11) не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon\geq 0$, иначе *P=NP*.

Несложно увидеть, что если $k=3$, то класс $W_Z(M,H)$ включает в себя класс задач (5.7)-(5.11). Отсюда из существования полиномиального ε-приближенного алгоритма для задач класса $W_Z(M,H)$ следует существование полиномиального ε-приближенного алгоритм для класса задач (5.7)-(5.11). Получаем противоречие. Далее пусть $k\geq 4$.

Рассмотрим задачу (5.7)-(5.11) и выберем задачу $w=w_Z(s;M;n_1,...,n_s;\{a_{F_{\bar f}}\},\{b_{F_{\bar f}}\},\ f\in M;\{c_{F_{N(s)}}\})\in W_Z(M,H)$ такую, что $|E_{f_l}|=m$, $l=\overline{1,k}$ (не уменьшая общности, будем считать, что такой выбор возможен). Перенумеруем элементы множества E_{f_l}: $E_{f_l}=\{F_{f_l}^{(1)},...,F_{f_l}^{(m)}\}$, $l=\overline{1,k}$. Так как $w\in W_Z(M,H)$, то $c_{F_{N(s)}}=\sum_{f\in H} d_{(F_{N(s)})_f}$, $F_{N(s)}\in E_{N(s)}$, где $\{d_{F_f}\}$, $f\in H$ – заданные матрицы коэффициентов. Определим значения свободных коэффициентов двусторонних неравенств в задаче w следующим образом:

$$a_{F_{f_l}}=b_{F_{f_l}}=1,\ F_{f_l}\in E_{f_l},\ l=\overline{1,k}.$$

Матрицы $\{d_{F_f}\}$, $f\in H$, задающие матрицу коэффициентов целевой функции $\{c_{F_{N(s)}}\}$, определим следующим образом:

$$d_{F_{f_1}^{(i)}F_{f_2}^{(j)}}=u_{ij}^{(1)},\ d_{F_{f_2}^{(j)}F_{f_3}^{(p)}}=u_{jp}^{(3)},\ d_{F_{f_3}^{(p)}F_{f_4}^{(i)}}=u_{ip}^{(2)},\ i,j,p=\overline{1,m},$$

$$d_{F_{f_l}^{(i)}F_{f_{l\,\mathrm{mod}k+1}}^{(j)}}=\begin{cases}0, \text{если } i=j\\ \Delta, \text{иначе}\end{cases},\ i,j=\overline{1,m},\ l=\overline{4,k},$$

где $\Delta=1+(1+\lceil\varepsilon\rceil)\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{p=1}^{m}(u_{ij}^{(1)}+u_{ip}^{(2)}+u_{jp}^{(3)})$.

Пусть y^*_{ijp}, $i,j,p=\overline{1,m}$, и u^* – оптимальное решение и оптимальное значение критерия задачи (5.7)-(5.11) соответственно. Т.к. $f_1,...,f_k$ – разбиение множества $N(s)$, то

$$E_{N(s)}=\{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)}\mid i_l=\overline{1,m}, l=\overline{1,k}\}.$$

Рассмотрим многоиндексную матрицу $\{x^*_{F_{N(s)}}\}$, построенную по следующему правилу:

$$x^*_{F_{f_1}^{(i_1)},...,F_{f_k}^{(i_k)}}=\begin{cases} y^*_{i_1i_2i_3}, \text{ если } i_1=i_l, l=\overline{4,k} \\ 0, \text{ иначе}\end{cases}, \ i_1,...,i_k=\overline{1,m}.$$

Несложно увидеть, что $\{x^*_{F_{N(s)}}\}$ является допустимым решением задачи w и $u^*=\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x^*_{F_{N(s)}}$. Покажем от противного, что $\{x^*_{F_{N(s)}}\}$ является оптимальным решением задачи w. Предположим, что $\{x'_{F_{N(s)}}\}$ является оптимальным решением задачи w и $\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x'_{F_{N(s)}}<u^*$. Несложно увидеть, что $x'_{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)}}=0,$ если существует $l\in\{4,...,k\},$ что $i_1\neq i_l$, $i_1,...,i_k=\overline{1,m};$ в противном случае $\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x'_{F_{N(s)}}\geq\Delta>u^*$, что противоречит предположению. Тогда рассмотрим набор

$$y'_{ijp}=x'_{F_{f_1}^{(i)}F_{f_2}^{(j)}F_{f_3}^{(p)}F_{f_4}^{(i)}...F_{f_k}^{(i)}},\ i,j,p=\overline{1,m}.$$

Несложно увидеть, что y'_{ijp}, $i,j,p=\overline{1,m}$, является допустимым решением задачи (5.7)-(5.11) и $\sum\limits_{i=1}^m\sum\limits_{j=1}^m\sum\limits_{p=1}^m(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y'_{ijp}=\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x'_{F_{N(s)}}<u^*$. Отсюда y^*_{ijp}, $i,j,p=\overline{1,m}$, не является оптимальным решением задачи. Получаем противоречие, предположение неверно и, следовательно, $\{x^*_{F_{N(s)}}\}$ является оптимальным решением задачи w. Таким образом, оптимальное значение критерия задачи w равно u^*.

Далее применим для задачи w существующий (по предположению) полиномиальный ε-приближенный алгоритм. Пусть $\{x'_{F_{N(s)}}\}$ решение найденное данным алгоритмом. Тогда $\{x'_{F_{N(s)}}\}$ является допустимым решением задачи w и $\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x'_{F_{N(s)}}\leq(1+\varepsilon)u^*$. При этом $\Delta>(1+\varepsilon)\sum\limits_{i=1}^m\sum\limits_{j=1}^m\sum\limits_{p=1}^m(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})\geq(1+\varepsilon)u^*$. Тогда $\sum\limits_{F_{N(s)}\in E_{N(s)}} c_{F_{N(s)}}x'_{F_{N(s)}}<\Delta$ и следовательно $x'_{F_{f_1}^{(i_1)},...,F_{f_k}^{(i_k)}}=0,$ если существует $l\in\{4,...,k\},$ что

$i_1 \neq i_l$, $i_1,...,i_k = \overline{1,m}$; в противном случае $\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} \geq \Delta$, что противоречит полученной выше оценке. Тогда набор $y'_{ijp} = x'_{F^{(i)}_{f1} F^{(j)}_{f2} F^{(p)}_{f3} F^{(i)}_{f4} ... F^{(i)}_{fk}}$, $i, j, p = \overline{1,m}$, будет являться допустимым решением задачи (5.7)-(5.11) и $\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{p=1}^{m}(u^{(1)}_{ij} + u^{(2)}_{ip} + u^{(3)}_{jp}) y'_{ijp} = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} \leq (1+\varepsilon)u^*$. Отсюда для класса задач (5.7)-(5.11) существует полиномиальный ε-приближенный алгоритм, заключающийся в построении задачи w по описанной выше схеме, ее решении полиномиальным ε-приближенным алгоритмом и построении приближенного решения исходной задачи по описанному выше правилу. Получаем противоречие, предположение неверно. *Утверждение доказано.*

Пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$, $M' = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$. Тогда класс задач $W_Z(M', H)$ включен в класс $W_Z(M)$. Отсюда, согласно утверждению 5.5, справедливо:

Утверждение 5.6. Пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$. Тогда для задач класса $W_Z(M)$ не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon \geq 0$, иначе $P = NP$.

Далее пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$. Тогда рассмотрим класс задач $W_Z(M)$. Согласно утверждению 5.6, для данного класса не существует полиномиальных ε-приближенных алгоритмов при выполнении известной гипотезы о неравенстве классов P и NP. Данный факт обуславливает поиск эвристических методов решения задач класса $W_Z(M)$. В качестве такого метода может быть предложен следующий алгоритм, основанный на полученном в теореме 5.2 результате сводимости. Идея алгоритма заключается в выборе среди задач, для которых могут быть применены результаты сводимости, задачи наиболее «близкой» к исходной задаче.

Алгоритм 5.2. Эвристический алгоритм решение многоиндексных задач с декомпозиционной системой ограничений.

Вход. Задача $w = w_Z(s; M; n_1,..., n_2; \{a_{F_{\bar{f}}}\}, \{b_{F_{\bar{f}}}\}, f \in M; \{c_{F_{N(s)}}\}) \in W_Z(M)$, где $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$.

Шаг 1. Пусть $H=\{f_i \mid i=\overline{1,k}\}\cup\{f_i\cup f_{i+1}\mid i=\overline{1,k-1}\}$. Тогда определим многоиндексные матрицы $\{d^*_{F_f}\}$, $f\in H$, и связанную с ними многоиндексную матрицу $\{e^*_{F_{N(s)}}\}$ как решение следующей оптимизационной задачи

$$e_{F_{N(s)}}=\sum_{f\in H} d_{(F_{N(s)})_f}\ ,\ F_{N(s)}\in E_{N(s)}, \tag{5.12}$$

$$dist(\{e_{F_{N(s)}}\},\{c_{F_{N(s)}}\})\to \min, \tag{5.13}$$

где $dist$ некоторая функция «близости» многоиндексных матриц. Перейти на шаг 2.

Шаг 2. Решить задачу $w'=w_Z(s;M;n_1,...,n_2;\{a_{F_{\bar f}}\},\{b_{F_{\bar f}}\},f\in M;\{e^*_{F_{N(s)}}\})$, используя алгоритм 5.1. Полученное решение будет являться приближенным решением задачи w. Алгоритм завершен.

Корректность алгоритма 5.2 обосновывается следующим образом. Пусть $H=\{f_i \mid i=\overline{1,k}\}\cup\{f_i\cup f_{i+1}\mid i=\overline{1,k-1}\}$. Согласно определению 5.2, множества $\tilde M$ и H являются $f_1,...,f_k$-декомпозиционными. Тогда $w'\in W_Z(M,H)$. И, следовательно, для решения задачи w' может быть применен алгоритм 5.1. Системы ограничений задач w и w' совпадают. Тем самым решение задачи w' будет являться допустимым решением задачи w.

Конкретная реализация алгоритма 5.2 определяется выбором функции близости $dist$. Так в качестве такой меры близости может быть выбрана сумма квадратов отклонений компонент матриц. Тогда задача (5.12),(5.13) может быть поставлена как следующая задача квадратичной безусловной оптимизации:

$$\sum_{F_{N(s)}\in E_{N(s)}}(c_{F_{N(s)}}-\sum_{f\in H} d_{(F_{N(s)})_f})^2\to \min. \tag{5.14}$$

Для решения задачи (5.14) может быть применен метод минимизации суммы квадратов невязок [60]. Алгоритм 5.2 с использованием на шаге 1 решения задачи (5.14) будем называть алгоритмом 5.2 с минимизацией сумм квадратов отклонений.

В качестве функции близости $dist$ также может быть выбран максимум относительного отклонения сверху компонент матриц. Тогда задача (5.12),(5.13) может быть поставлена как следующая задача линейного программирования:

$$c_{F_{N(s)}}\le \sum_{f\in H} d_{(F_{N(s)})_f}\le \Delta c_{F_{N(s)}},\ \ F_{N(s)}\in E_{N(s)}, \tag{5.15}$$

$$\Delta \to \min. \tag{5.16}$$

Далее алгоритм 5.2 с использованием на шаге 1 решения задачи (5.15),(5.16) будем называть алгоритмом 5.2 с минимизацией максимума относительного отклонения. Для решения задачи (5.15),(5.16) могут быть применены общие методы решения задач линейного программирования [102]. Для приближенного алгоритма 5.2 с минимизацией максимума относительного отклонения справедлива следующая оценка отклонения от оптимума.

Теорема 5.3. Пусть c^* – оптимальное значение критерия задачи w, c' – значение критерия задачи w на приближенном решение, найденном алгоритмом 5.2 с минимизацией максимума относительного отклонения, Δ^* – оптимальное значение критерия в задаче (5.15),(5.16), соответствующей задаче w, тогда $c' \le \Delta^* c^*$, $w \in W_Z(M)$, $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$, где $f_1,..., f_k$ – разбиение множества $N(s)$, $k \ge 3$.

Доказательство. Пусть $w = w_Z(s; M; n_1,..., n_s; \{a_{F_{\bar{f}}}\}, \{b_{F_{\bar{f}}}\}, f \in M; \{c_{F_{N(s)}}\}) \in W_Z(M)$ и $\{x^*_{F_{N(s)}}\}$ – оптимальное решения задачи w, $\{x'_{F_{N(s)}}\}$ – приближенное решение задачи w, найденное алгоритмом 5.2 с минимизацией максимума относительного отклонения. Тогда $c^* = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^*_{F_{N(s)}}$, $c' = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}}$. Далее пусть $d^*_{F_F}$, $F_f \in E_f$, $f \in H$, Δ^* – решение задачи (5.15)-(5.16), соответствующей задаче w. Согласно алгоритму 5.2, многоиндексная матрица $\{x'_{F_{N(s)}}\}$ является оптимальным решением задачи $w' = w_Z(s; M; n_1,..., n_2; \{a_{F_{\bar{f}}}\}, \{b_{F_{\bar{f}}}\}, f \in M; \{e^*_{F_{N(s)}}\})$, где $e^*_{F_{N(s)}} = \sum_{f \in H} d^*_{(F_{N(s)})_f}$, $F_{N(s)} \in E_{N(s)}$. Согласно условию (3.2) выполняется $x'_{F_{N(s)}}, x^*_{F_{N(s)}} \ge 0$, $F_{N(s)} \in E_{N(s)}$. Согласно (5.15) выполняется $c_{F_{N(s)}} \le \sum_{f \in H} d^*_{(F_{N(s)})_f} \le \Delta^* c_{F_{N(s)}}$, $F_{N(s)} \in E_{N(s)}$. Отсюда,

$$c' = \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x'_{F_{N(s)}} \le \sum_{F_{N(s)} \in E_{N(s)}} \left(\sum_{f \in H} d^*_{(F_{N(s)})_f} \right) x'_{F_{N(s)}} = \sum_{F_{N(s)} \in E_{N(s)}} e^*_{F_{N(s)}} x'_{F_{N(s)}} \le$$

$$\sum_{F_{N(s)} \in E_{N(s)}} e^*_{F_{N(s)}} x^*_{F_{N(s)}} = \sum_{F_{N(s)} \in E_{N(s)}} \left(\sum_{f \in H} d^*_{(F_{N(s)})_f} \right) x^*_{F_{N(s)}} \le \sum_{F_{N(s)} \in E_{N(s)}} \Delta^* c_{F_{N(s)}} x^*_{F_{N(s)}} = \Delta^* c^*.$$

Теорема доказана.

Заметим, что оценка отклонения от оптимума, сформулированная в теореме 5.3, не является константой, т.к. зависит от исходных данных задачи w. Обратное, согласно утверждению 5.6, означало бы равенство классов P и NP.

Утверждение 5.7. Пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$. Тогда класс задач $W_Z(M, \Delta H)$ является NP-трудным.

Доказательство. Пусть выполняются условия утверждения. Рассмотрим трехиндексную аксиальную задачу о назначениях (5.7)-(5.11), матрица стоимостей которой удовлетворяет неравенству треугольника:

$$0 \leq u_{ij}^{(1)} \leq u_{ip}^{(2)} + u_{jp}^{(3)},\ i, j, p = \overline{1,m}, \tag{5.17}$$

$$0 \leq u_{ip}^{(2)} \leq u_{ij}^{(1)} + u_{jp}^{(3)},\ i, j, p = \overline{1,m}, \tag{5.18}$$

$$0 \leq u_{jp}^{(3)} \leq u_{ij}^{(1)} + u_{ip}^{(2)},\ i, j, p = \overline{1,m}. \tag{5.19}$$

В [131] было показано, что класс задач (5.7)-(5.11) с матрицей стоимостей, удовлетворяющей условию (5.17)-(5.19), является NP-трудным. Проведем далее доказательство путем полиномиального сведения к классу $W_Z(M, \Delta H)$ класса задач (5.7)-(5.11) с матрицей стоимостей, удовлетворяющей условию (5.17)-(5.19).

Несложно увидеть, что если $k = 3$, то класс $W_Z(M, \Delta H)$ включает в себя класс задач (5.7)-(5.11) с матрицей стоимостей, удовлетворяющей условию (5.17)-(5.19). Отсюда класс $W_Z(M, \Delta H)$ является NP-трудным.

Далее пусть $k \geq 4$. Рассмотрим задачу (5.7)-(5.11) с матрицей стоимостей, удовлетворяющей условию (5.17)-(5.19), и выберем задачу $w = w_z(s; M; n_1,..., n_s; \{a_{F_{\bar f}}\}, \{b_{F_{\bar f}}\}, f \in M; \{c_{F_{N(s)}}\}) \in W_Z(M, H)$ такую, что $|E_{f_l}| = m$, $l = \overline{1,k}$ (не уменьшая общности, будем считать, что такой выбор возможен). Перенумеруем элементы множества E_{f_l}: $E_{f_l} = \{F_{f_l}^{(1)},..., F_{f_l}^{(m)}\}$, $l = \overline{1,k}$. Так как $w_z \in W_Z(M, H)$, то $c_{F_{N(s)}} = \sum_{f \in H} d_{(F_{N(s)})_f}$, $F_{N(s)} \in E_{N(s)}$, где $\{d_{F_f}\}$, $f \in H$ – заданные матрицы коэффициентов. Определим значения свободных коэффициентов двусторонних неравенств в задаче w_Z следующим образом:

$$a_{F_{f_l}} = b_{F_{f_l}} = 1,\ F_{f_l} \in E_{f_l},\ l = \overline{1,k}.$$

Матрицы $\{d_{F_f}\}$, $f \in H$, задающие матрицу коэффициентов целевой функции $\{c_{F_{N(s)}}\}$, определим следующим образом:

$$d_{F_{f_1}^{(i)}F_{f_2}^{(j)}} = u_{ij}^{(1)} + \Delta,\ d_{F_{f_2}^{(j)}F_{f_3}^{(p)}} = u_{jp}^{(3)} + \Delta,\ d_{F_{f_3}^{(p)}F_{f_4}^{(i)}} = u_{ip}^{(2)} + \Delta,\ i, j, p = \overline{1,m},$$

$$d_{F_{f_l}^{(i)}F_{f_{l \bmod k+1}}^{(j)}} = \begin{cases} 0, \text{если } i = j \\ \Delta, \text{иначе} \end{cases},\ i, j = \overline{1,m},\ l = \overline{4,k},$$

где $\Delta = 1 + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{p=1}^{m}(u_{ij}^{(1)} + u_{ip}^{(2)} + u_{jp}^{(3)})$. Заметим, что $f_1,..., f_k$ – разбиение множества $N(s)$ и таким образом, $E_{N(s)} = \{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)} \mid i_l = \overline{1,m}, l = \overline{1,k}\}$. По построению

$$d_{(F_{N(s)})_{f'}} \le 2\Delta \le \sum_{f \in H \setminus \{f'\}} d_{(F_{N(s)})_f},\ F_{N(s)} \in E_{N(s)},\ f' \in H.$$

Следовательно, матрица коэффициентов целевой функции удовлетворяет обобщенному неравенству треугольника (3.5). Отсюда $w \in W_Z(M, \Delta H)$.

Пусть $\{x^*_{F_{N(s)}}\}$ является оптимальным решением задачи w. Тогда покажем от противного, что выполняется следующее условие:

$x^*_{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)}} = 0,$ если существует $l \in \{4,...,k\}$, что $i_1 \ne i_l$, $i_1,...,i_k = \overline{1,m}$. (5.20)

Предположим, что найдутся $i_1,...,i_k \in \{1,...,m\}$ и $l \in \{4,...,k\}$, что $x^*_{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)}} = 1$. Тогда по построению

$$\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^*_{F_{N(s)}} \ge (3m+1)\Delta.$$

Определим $\{x_{F_{N(s)}}\}$ следующим образом:

$$x_{F_{f_1}^{(i_1)}...F_{f_k}^{(i_k)}} = \begin{cases} 1, \text{если } i_1 = i_l, l = \overline{2,k} \\ 0, \text{иначе} \end{cases},\ i_1,...,i_k = \overline{1,m}.$$

Несложно увидеть, что $\{x_{F_{N(s)}}\}$ является допустимым решением задачи w и

$$\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x_{F_{N(s)}} < (3m+1)\Delta \le \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^*_{F_{N(s)}}.$$

Получаем противоречие, предположение неверно. Следовательно, условие (5.20) выполняется.

Определим y^*_{ijp}, $i,j,p=\overline{1,m}$, следующим образом: $y^*_{ijp}=x^*_{F^{(i)}_{f_1}F^{(j)}_{f_2}F^{(p)}_{f_3}F^{(i)}_{f_4}\ldots F^{(i)}_{f_k}}$, $i,j,p=\overline{1,m}.$ Согласно условию (5.20) набор y^*_{ijp}, $i,j,p=\overline{1,m}$, является допустимым решением задачи (5.7)-(5.11). При этом $\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y^*_{ijp}=\sum_{F_{N(s)}\in E_{N(s)}}c_{F_{N(s)}}x^*_{F_{N(s)}}-3m\Delta$.

Докажем от противного, что y^*_{ijp}, $i,j,p=\overline{1,m}$, является оптимальным решением задачи (5.7)-(5.11). Предположим, что y_{ijp}, $i,j,p=\overline{1,m},$ – оптимальное решение задачи (5.7)-(5.11) и выполняется

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y_{ijp}<\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y^*_{ijp}.$$

Тогда построим $\{x_{F_{N(s)}}\}$ следующим образом

$$x_{F^{(i_1)}_{f_1}\ldots F^{(i_k)}_{f_k}}=\begin{cases}y_{i_1i_2i_3}, \text{если } i_1=i_l,\ l=\overline{4,k}\\ 0, \text{иначе}\end{cases},\ i_1,\ldots,i_k=\overline{1,m}.$$

Несложно увидеть, что $\{x_{F_{N(s)}}\}$ является допустимым решением задачи w и

$$\sum_{F_{N(s)}\in E_{N(s)}}c_{F_{N(s)}}x_{F_{N(s)}}=\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y_{ijp}+3m\Delta.$$

Отсюда

$$\sum_{F_{N(s)}\in E_{N(s)}}c_{F_{N(s)}}x_{F_{N(s)}}=\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y_{ijp}+3m\Delta<$$
$$\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m}(u^{(1)}_{ij}+u^{(2)}_{ip}+u^{(3)}_{jp})y^*_{ijp}+3m\Delta=\sum_{F_{N(s)}\in E_{N(s)}}c_{F_{N(s)}}x^*_{F_{N(s)}}.$$

Однако $\{x^*_{F_{N(s)}}\}$ является оптимальным решением задачи W. Получаем противоречие, предположение неверно. Следовательно, y^*_{ijp}, $i,j,p=\overline{1,m}$, – оптимальное решение задачи (5.7)-(5.11).

Таким образом, *NP*-трудный класс класса задач (5.7)-(5.11) с матрицей стоимостей, удовлетворяющей условию (5.17)-(5.19) полиномиально сводим к классу $W_Z(M,\Delta H)$. Отсюда класс $W_Z(M,\Delta H)$ является *NP*-трудным [52]. *Утверждение доказано.*

Далее пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$. Тогда, согласно утверждению 5.5, для задач класса $W_Z(M,H)$ не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon \geq 0$, иначе $P = NP$. Рассмотрим подкласс $W_Z(M,\Delta H)$, содержащий задачи класса $W_Z(M,H)$, удовлетворяющие обобщенному неравенству треугольника (3.5). Согласно утверждению 5.7, класс $W_Z(M,\Delta H)$ является NP-трудным. Данный факт обуславливает поиск приближенных методов решения задач класса $W_Z(M,\Delta H)$. В качестве такого метода может быть предложен следующий алгоритм, основанный на полученном в теореме 5.2 результате сводимости.

Алгоритм 5.3. Приближенный алгоритм решение многоиндексных задач с декомпозиционной структурой и матрицей стоимостей, удовлетворяющей обобщенному неравенству треугольника.

Вход. Задача $w = w_Z(s; M; n_1,..., n_2; \{a_{F_{\bar{f}}}\}, \{b_{F_{\bar{f}}}\}, f \in M; \{c_{F_{N(s)}}\}) \in W_Z(M,\Delta H)$, где $M = \{f_i \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$.

Шаг 1. Пусть $\{d_{F_f}\}$, $f \in H$ – заданные многоиндексные матрицы такие, что $c_{F_{N(s)}} = \sum_{f \in H} d_{(F_{N(s)})_f}$, $F_{N(s)} \in E_{N(s)}$. Определим многоиндексную матрицу $\{c^{(i)}_{F_{N(s)}}\}$ следующим образом

$$c^{(i)}_{F_{N(s)}} = \sum_{f \in H \setminus \{f_i \cup f_{i \bmod k+1}\}} d_{(F_{N(s)})_f}, \quad F_{N(s)} \in E_{N(s)}, \tag{5.21}$$

и обозначим $w_i = w_Z(s; M; n_1,..., n_2; \{a_{F_{\bar{f}}}\}, \{b_{F_{\bar{f}}}\}, f \in M; \{c^{(i)}_{F_{N(s)}}\})$, $i = \overline{1,k}$. Переход на шаг 2.

Шаг 2. Решим задачу w_i при помощи алгоритма 5.1, если задача w_i несовместна, то задача w также несовместна и алгоритм завершен, $i = \overline{1,k}$. Далее пусть $\{x^{(i)}_{F_{N(s)}}\}$ – оптимальное решение задачи w_i, $i = \overline{1,k}$. Переход на шаг 3.

Шаг 3. Определим i^* следующим образом

$$i^* = \arg\min_{i=\overline{1,k}} \Big(\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \Big).$$

Тогда $\{x^{(i^*)}_{F_{N(s)}}\}$ является приближенным решением задачи w. Алгоритм завершен.

Далее приведем обоснование корректности алгоритма 5.3 и покажем, что построенный алгоритм является ε-приближенным алгоритмом для задач рассматриваемого класса $W_Z(M,\Delta H)$.

Лемма 5.3. Пусть $A\in R^{n\times m}$, $b\in R^n$, $c_0,c_1,...,c_l\in R^m$, $d\in R$, $dc_0=c_1+...+c_l$ и $g_i^*=\min\{(c_i,x)\mid Ax\le b, x\in Z^m\}$, $i=\overline{0,l}$. Тогда $\sum_{i=1}^{l} g_i^*\le dg_0^*$.

Доказательство. Пусть выполняются условия леммы. Обозначим через $x^*_{(i)}$ оптимальное решение задачи $\min\{(c_i,x)\mid Ax\le b, x\in Z^m\}$, $i=\overline{0,l}$. Тогда выполняется

$$g_i^*=(c_i,x^*_{(i)})\le(c_i,x^*_{(0)}),\ i=\overline{0,l}.$$

Отсюда

$$\sum_{i=1}^{l} g_i^*=\sum_{i=1}^{l}(c_i,x^*_{(i)})\le\sum_{i=1}^{l}(c_i,x^*_{(0)})=(dc_0,x^*_{(0)})=dg_0^*.$$

Лемма доказана.

Теорема 5.4. Пусть $M=\{\overline{f_i}\mid i=\overline{1,k}\}$, $H=\{f_i\cup f_{i\bmod k+1}\mid i=\overline{1,k}\}$, $f_1,...,f_k$ – разбиение множества $N(s)$, $k\ge 3$. Тогда алгоритм 5.3 является $(k-2)/k$-приближенным алгоритмом для задач класса $W_Z(M,\Delta H)$.

Доказательство. Пусть $M=\{\overline{f_i}\mid i=\overline{1,k}\}$, $H=\{f_i\cup f_{i\bmod k+1}\mid i=\overline{1,k}\}$, $f_1,...,f_k$ – разбиение множества $N(s)$, $k\ge 3$ и $w=w_Z(s;M;n_1,...,n_2;\{a_{F_{\bar f}}\},\{b_{F_{\bar f}}\}, f\in M;\{c_{F_{N(s)}}\})\in W_Z(M,\Delta H)$. Тогда существуют многоиндексные матрицы $\{d_{F_f}\}$, $f\in H$, что $c_{F_{N(s)}}=\sum_{f\in H} d_{(F_{N(s)})_f}$, $F_{N(s)}\in E_{N(s)}$. Согласно шагу 1 алгоритма, строятся k задач $w_i=w_Z(s;M;n_1,...,n_2;\{a_{F_{\bar f}}\},\{b_{F_{\bar f}}\}, f\in M;\{c^{(i)}_{F_{N(s)}}\})$, где многоиндексная матрица $\{c^{(i)}_{F_{N(s)}}\}$ определяется согласно соотношению (5.21), $i=\overline{1,k}$.

Пусть $i\in\{1,...,k\}$. Рассмотрим задачу w_i. Обозначим $H_i=H\setminus\{f_i\cup f_{i\bmod k+1}\}$. По построению $w_i\in W_Z(M,H_i)$. Согласно определению 5.2, множества $\tilde{M}$ и H_i являются

$f_{i+1},\dots,f_k,f_1,\dots,f_i$-декомпозиционными. Следовательно, алгоритм 5.1 может быть применен для решения задачи w_i на шаге 2. Системы ограничений задач w и w_i совпадают, таким образом, задача w совместна тогда и только тогда, когда совместна задача w_i.

Далее пусть $\{x^{(i)}_{F_{N(s)}}\}$ является оптимальным решением задачи w_i, $i=\overline{1,k}$, $\{x^{*}_{F_{N(s)}}\}$ – оптимальным решением задачи w. Т.к. системы ограничений задач w и w_i совпадают, то $\{x^{(i)}_{F_{N(s)}}\}$ является допустимым решением задачи w, $i=\overline{1,k}$. Согласно соотношению (5.21),

$$\sum_{i=1}^{k} c^{(i)}_{F_{N(s)}} = \sum_{i=1}^{k}\sum_{f\in H\setminus\{f_i\cup f_{i\bmod k+1}\}} d_{(F_{N(s)})_f} = (k-1)\sum_{f\in H} d_{(F_{N(s)})_f} = (k-1)c_{F_{N(s)}},\ F_{N(s)}\in E_{N(s)}.$$

Отсюда, согласно лемме 5.3,

$$\sum_{i=1}^{k}\sum_{F_{N(s)}\in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \le (k-1)\sum_{F_{N(s)}\in E_{F_{N(s)}}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}.$$

Покажем от противного, что

$$\min_{i=\overline{1,k}} \sum_{F_{N(s)}\in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \le \frac{(k-1)}{k}\sum_{F_{N(s)}\in E_{F_{N(s)}}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}. \tag{5.22}$$

Предположим, что $\min\limits_{i=\overline{1,k}} \sum\limits_{F_{N(s)}\in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} > \frac{(k-1)}{k}\sum\limits_{F_{N(s)}\in E_{F_{N(s)}}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}$, тогда $\sum\limits_{i=1}^{k}\sum\limits_{F_{N(s)}\in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \ge k\min\limits_{i=\overline{1,k}} \sum\limits_{F_{N(s)}\in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} > (k-1)\sum\limits_{F_{N(s)}\in E_{F_{N(s)}}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}$. Получаем противоречие, предположение неверно и соотношение (5.22) выполняется.

По условию теоремы $c_{F_{N(s)}} = \sum\limits_{f\in H} d_{(F_{N(s)})_f}$, $F_{N(s)}\in E_{N(s)}$, тогда

$$c_{F_{N(s)}} = \sum_{f\in H\setminus\{f_i\cup f_{i\bmod k+1}\}} d_{(F_{N(s)})_f} + d_{(F_{N(s)})_{(f_i\cup f_{i\bmod k+1})}} = c^{(i)}_{F_{N(s)}} + d_{(F_{N(s)})_{(f_i\cup f_{i\bmod k+1})}},\ F_{N(s)}\in E_{N(s)},\ i=\overline{1,k}.$$

С другой стороны, согласно обобщенному неравенству треугольника (3.5) выполняется

$$d_{(F_{N(s)})_{(f_i\cup f_{i\bmod k+1})}} \le \sum_{f\in H\setminus\{f_i\cup f_{i\bmod k+1}\}} d_{(F_{N(s)})_f} = c^{(i)}_{F_{N(s)}},\ F_{N(s)}\in E_{N(s)},\ i=\overline{1,k}.$$

Следовательно,

$$c_{F_{N(s)}} = c^{(i)}_{F_{N(s)}} + d_{(F_{N(s)})_{(f_i \cup f_{i \bmod k+1})}} \le 2c^{(i)}_{F_{N(s)}},\ F_{N(s)} \in E_{N(s)},\ i = \overline{1,k}.$$

Тогда, в силу ограничения неотрицательности переменных (3.2), выполняется

$$\sum_{F_{N(s)} \in E_{F_{N(s)}}} c_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \le 2 \sum_{F_{N(s)} \in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}},\ i = \overline{1,k}.$$

Отсюда с использованием соотношения (5.22)

$$\min_{i=\overline{1,k}} \sum_{F_{N(s)} \in E_{F_{N(s)}}} c_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \le 2 \min_{i=\overline{1,k}} \sum_{F_{N(s)} \in E_{F_{N(s)}}} c^{(i)}_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \le 2\frac{(k-1)}{k} \sum_{F_{N(s)} \in E_{F_{N(s)}}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}.$$

Согласно шагу 3 алгоритма $i^* = \arg\min_{i=\overline{1,k}} \Big(\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^{(i)}_{F_{N(s)}} \Big)$. Тогда

$$\sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^{(i^*)}_{F_{N(s)}} \le \left(1 + \frac{k-2}{k}\right) \sum_{F_{N(s)} \in E_{N(s)}} c_{F_{N(s)}} x^{*}_{F_{N(s)}}.$$

Следовательно, алгоритм 5.3 является $(k-2)/k$-приближенным алгоритмом для задач класса $W(M,\Delta H)$. *Теорема доказана.*

Алгоритм 5.3 основан на решении k задач, каждая из которых может быть решена алгоритмом 5.1. При этом т.к. $f_1,...,f_k$ – разбиение множества $N(s)$, то $k \le s$. Тогда на основании утверждения 5.1 справедлива следующая оценка сложности алгоритма 5.3.

Утверждение 5.8. Пусть $M = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1,...,f_k$ – разбиение множества $N(s)$, $k \ge 3$. Тогда алгоритм 5.3 поиска приближенного решения задач класса $W_Z(M,\Delta H)$ требует $O(|E_{N(s)}|^2 \log^2 |E_{N(s)}|)$ вычислительных операций.

Заметим, что построенный $(k-2)/k$-приближенный алгоритм для задач класса $W(M,\Delta H)$ обобщает, полученный в работе [131] результат, связанный с построением приближенного алгоритма для трехиндексных аксиальных задач о назначениях, удовлетворяющих неравенству треугольника. Несложно увидеть, что класс трехиндексных аксиальных задач о назначениях, удовлетворяющих неравенству треугольника, относится к классу $W(M,\Delta H)$, где $M = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$, $f_1 = \{1\}, f_2 = \{2\}, f_3 = \{3\}$ – разбиение множества $N(s)$, $k = s = 3$.

В данном случае алгоритм 5.3 будет представлять собой 1/3-приближенный алгоритм для класса трехиндексных аксиальных задач о назначениях, удовлетворяющих неравенству треугольника. Данная оценка для рассматриваемой задачи о назначениях совпадает с оценкой, полученной в [131]. Однако в отличие от метода, предложенного в работе [131], алгоритм 5.3 применим при исследовании гораздо более широкого класса задач – целочисленных многоиндексных задач линейного программирования транспортного типа, обладающих декомпозиционной структурой и удовлетворяющих обобщенному неравенству треугольника.

Выводы

Предложена схема $t_1 | t_2 - equal | t_3 - cycle$ сводимости (сводимости с сохранением соответствия циклов) класса задач линейного программирования к классу задач поиска потока в сети W_{Graph}. При исследовании $t_1 | t_2 - equal | t_3 - cycle$ сводимости класса $W(M)$ к классу W_{Graph} установлено:

– для того, чтобы класс $W(M,H)$ являлся $L | L - equal \| E_{N(s)} |^2 - cycle$ сводимым к классу W_{Graph}, достаточно, чтобы множества $\tilde{M}$ и H были $f_1,..., f_k$-декомпозиционными;

– пусть множества $\tilde{M}$, H являются $f_1,..., f_k$-декомпозиционными, тогда существует алгоритм решения задач класса $W(M,H)$ и систем класса $D(M)$, требующий $O(| E_{N(s)} |^2 \log^2 | E_{N(s)} |)$ и $O(| E_{N(s)} |^2 \log | E_{N(s)} |)$ вычислительных операций соответственно, данный алгоритм применим при исследовании целочисленного случая, что позволяет выделить новый полиномиально разрешимый подкласс в NP-трудном классе многоиндексных задач целочисленного линейного программирования;

– пусть множество $\tilde{M}$ является $f_1,..., f_k$-декомпозиционным, тогда существует алгоритм решения задач класса $S(M)$, требующий $O(| E_{N(s)} |^2 \log^2 | E_{N(s)} |)$ вычислительных операций;

– пусть множество $\tilde{M} \cup H$ является $f_1,..., f_k$-декомпозиционным, тогда существуют алгоритмы решения задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, требующие $O(| E_{N(s)} |^3 \log | E_{N(s)} | \log p)$ и $O(| E_{N(s)} |^2 \log | E_{N(s)} | \log p)$ вычислительных операций соответственно;

где $f_1,..., f_k$ – разбиение множества $N(s)$.

Результаты исследования $t_1 | t_2 - equal | t_3 - cycle$ сводимости применены также при построении приближенных алгоритмов решения ряда *NP*-трудных многоиндексных задач, обладающих декомпозиционной структурой:

– рассмотрен класс задач $W_Z(M)$, для которого не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon \ge 0$, иначе $P = NP$, $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$; для задач класса $W_Z(M)$ предложен полиномиальный приближенный алгоритм, находящий допустимое решение, отклоняющееся от оптимального значения критерия не более чем в Δ^* раз, где Δ^* является оптимальным значение критерия вспомогательной задачи линейного программирования;

– рассмотрен NP-трудный класс задач $W_Z(M,\Delta H)$, где $M=\{\overline{f_i} \mid i=\overline{1,k}\}$, $H=\{f_i \cup f_{i \bmod k+1} \mid i=\overline{1,k}\}$; для задач класса $W_Z(M,\Delta H)$ предложен полиномиальный $(k-2)/k$-приближенный алгоритм;

где $f_1,...,f_k$ – разбиение множества $N(s)$, $k \geq 3$.

Заключение

Основные результаты диссертационной работы заключаются в следующем. При исследовании вопросов сводимости класса многоиндексных задач линейного программирования транспортного типа к классу задач поиска потока минимальной стоимости в сети были предложены и исследованы две схемы сводимости:

– сводимость с сохранением соответствия ребер ($t_1 \mid t_2 - equal \mid t_3 - edge$ сводимость),

– сводимость с сохранением соответствия циклов ($t_1 \mid t_2 - equal \mid t_3 - cycle$ сводимость).

При исследовании $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости класса многоиндексных задач $W(M)$ к классу задач поиска потока минимальной стоимости в сети W_{Graph} был получен исчерпывающий ответ вопроса сводимости:

– для того, чтобы класс $W(M)$ являлся $L \mid L - equal \mid L - edge$ сводимым к классу W_{Graph}, достаточно, чтобы множество M было 2-вложенным,

– для того, чтобы класс $W(M)$ являлся $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимым к классу W_{Graph}, необходимо, чтобы множество M было 2-вложенным.

Более того, предложенная конструктивная схема сводимости класса $W(M)$ в случае 2-вложенности множества M является оптимальной (сведение с асимптотически меньшими вычислительными затратами невозможно и сколько угодное увеличение вычислительных затрат на сведение не приводится к расширению класса сводимых задач). Были разработаны алгоритмы решения следующих многоиндексных задач, основанные на полученных результатах сводимости:

– задач класса $W(M)$ и систем класса $D(M)$, а также задач соответствующих целочисленных классов $W_Z(M)$ и $D_Z(M)$, где множество M является 2-вложенным,

– задач класса $S(M)$, где множество M является 2-вложенным,

– задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, где множество $M \cup \widetilde{H}$ является 2-вложенным.

При исследовании $t_1 \mid t_2 - equal \mid t_3 - edge$ сводимости класса многоиндексных задач $W(M)$ к классу задач поиска потока минимальной стоимости в древовидной сети W_{Tree} был также получен исчерпывающий ответ вопроса сводимости:

– для того, чтобы класс $W(M)$ являлся $L\,|\,L-equal\,|\,L-edge$ сводимым к классу W_{Tree}, достаточно, чтобы множество *M* было 1-вложенным,

– для того, чтобы класс $W(M)$ являлся $t_1\,|\,t_2-equal\,|\,t_3-edge$ сводимым к классу W_{Tree}, необходимо, чтобы множество *M* было 1-вложенным.

Предложенная конструктивная схема сводимости класса $W(M)$ в случае 1-вложенности множества *M* здесь также является оптимальной. Были разработаны алгоритмы решения следующих многоиндексных задач, основанные на полученных результатах сводимости:

– задач класса $W(M)$ и систем класса $D(M)$, а также задач соответствующих целочисленных классов $W_Z(M)$ и $D_Z(M)$, где множество M является 1-вложенным,

– задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, где множество $M \cup \tilde{H}$ является 1-вложенным.

Была предложена схема $t_1\,|\,t_2-Z\,|\,t_3-Z$ сводимости класса $W(M)$ к классу W_{Graph}, являющаяся обобщением схемы $t_1\,|\,t_2-equal\,|\,t_3-edge$ сводимости. При исследовании $t_1\,|\,t_2-Z\,|\,t_3-Z$ сводимости класса многоиндексных задач $W(M)$ к классу задач поиска потока минимальной стоимости в сети W_{Graph} были получены следующие результаты:

– для того, чтобы класс $W(M)$ являлся $L\,|\,L-Z\,|\,L-Z$ сводимым к классу W_{Graph}, достаточно, чтобы множество *M* было 2-вложенным,

– для того, чтобы класс $W(M)$ являлся $P^*\,|\,P^*-Z\,|\,P^*-Z$ сводимым к классу W_{Graph}, необходимо и достаточно, чтобы множество *M* было 2-вложенным, в противном случае *P=NP*.

При исследовании $t_1\,|\,t_2-equal\,|\,t_3-cycle$ сводимости класса многоиндексных задач $W(M,H)$ к классу задач поиска потока минимальной стоимости в сети W_{Graph} было установлено: для того, чтобы класс $W(M,H)$ являлся $L\,|\,L-equal\,\|E_{N(s)}|^2-cycle$ сводимым к классу W_{Graph}, достаточно, чтобы множество чтобы множества $\tilde{M}$ и H были $f_1,\ldots,f_k$-декомпозиционными, где $f_1,\ldots,f_k$ – разбиение множества $N(s)$. Были разработаны алгоритмы решения следующих многоиндексных задач, основанные на полученных результатах сводимости:

– задач класса $W(M,H)$ и систем класса $D(M)$, а также задач соответствующих целочисленных классов $W_Z(M,H)$ и $D_Z(M)$, где множества M и H являются $f_1,..., f_k$-декомпозиционными,

– задач класса $S(M)$, где множество $\tilde{M}$ является $f_1,..., f_k$-декомпозиционным;

– задач класса $U_{\prec}(M,H)$ и задач класса $U_{\max\min}(M,H)$, где множество $\tilde{M} \cup H$ является $f_1,..., f_k$-декомпозиционным,

здесь $f_1,..., f_k$ – разбиение множества $N(s)$. Построенные результаты позволили выделить новый полиномиально разрешимый класс (класс $W_Z(M,H)$, где множества $\tilde{M}$ и H являются $f_1,..., f_k$-декомпозиционными, $f_1,..., f_k$ – разбиение множества $N(s)$) в NP-трудном классе многоиндексных задач целочисленного линейного программирования. Полученные результаты сводимости применены также при построении приближенных алгоритмов решения ряда *NP*-трудных многоиндексных задач, обладающих декомпозиционной структурой:

– рассмотрен класс задач $W_Z(M)$, для которого не существует полиномиального ε-приближенного алгоритма для любых $\varepsilon \geq 0$, иначе $P = NP$, $M = \{\overline{f_i} \mid i = \overline{1,k}\} \cup \{\overline{f_i \cup f_{i+1}} \mid i = \overline{1,k-1}\}$; для задач класса $W_Z(M)$ предложен полиномиальный приближенный алгоритм, находящий допустимое решение, отклоняющееся от оптимального значения критерия не более чем в Δ^* раз, где Δ^* является оптимальным значение критерия вспомогательной задачи линейного программирования;

– рассмотрен *NP*-трудный класс задач $W_Z(M,\Delta H)$, где $M = \{\overline{f_i} \mid i = \overline{1,k}\}$, $H = \{f_i \cup f_{i \bmod k+1} \mid i = \overline{1,k}\}$; для задач класса $W_Z(M,\Delta H)$ предложен полиномиальный $(k-2)/k$-приближенный алгоритм;

здесь $f_1,..., f_k$ – разбиение множества $N(s)$, $k \geq 3$.

Применимость разработанных подходов была также проиллюстрирована на примере прикладных многоиндексных задач распределения ресурсов. Построенные алгоритмы были использованы при разработке программных систем, прошедших апробацию на ряде промышленных предприятий.

Список литературы

1. Адельсон-Вельский Г.М., Диниц Е.А., Карзанов А.В. Потоковые алгоритмы. – М.: Наука. 1975.

2. Афраймович Л.Г. Задача поиска потока в несовместной транспортной сети // Проблемы теоретической кибернетики. Тезисы докладов XV международной конференции. 2008. С. 8.

3. Афраймович Л.Г. Задачи распределения однородного ресурса в иерархических системах // VIII Нижегородская сессия молодых ученых. Математические науки. Тезисы докладов. Саров. 2003. С. 41–42.

4. Афраймович Л.Г. Задачи распределения однородного ресурса в многоуровневых иерархических системах // IX Нижегородская сессия молодых ученых. Технические науки. Тезисы докладов. Нижний Новгород. 2004. С. 5–6.

5. Афраймович Л.Г. Метод решения целочисленных многоиндексных транспортных задач с декомпозиционной структурой // Доклады Одесского семинара по дискретной математике. № 11. 2011. С. 4–14.

6. Афраймович Л.Г. Минимизация затрат при распределении однородного ресурса в иерархических системах с двусторонними ограничениями // КоГраф 2002. Материалы докладов всероссийской конференции. – Нижний Новгород. 2002. С. 81–83.

7. Афраймович Л.Г. Многоиндексные задачи линейного программирования с декомпозиционной структурой // VI Московская международная конференция по исследованию операций (ORM2010): Москва, 19-23 октября 2010 г.: Труды – М.: МАКС Пресс, 2010. С. 280–281.

8. Афраймович Л.Г. Многоиндексные транспортные задачи с 2-вложенной структурой // Автоматика и телемеханика. 2013. № 1. С. 116–134.

9. Афраймович Л.Г. Многоиндексные транспортные задачи с декомпозиционной структурой // Автоматика и телемеханика. 2012. № 1. С. 130–147.

10. Афраймович Л.Г. Модифицированные потоковые алгоритмы распределения однородного ресурса в иерархических системах // Математика и кибернетика 2003. Сборник научных статей юбилейной научно-технической конференции факультета ВМК ННГУ и НИИ ПМК. Нижний Новгород. 2003. С. 23–25.

11. Афраймович Л.Г. Оптимальное распределение однородного ресурса в задачах управления производством // Технологии Microsoft в теории и практике программирования. Труды Всероссийской конференции студентов, аспирантов и молодых ученых. Центральный регион. – М.: Издательство МГТУ им Н.Э. Баумана. 2005. С. 31–32.

12. Афраймович Л.Г. Оптимальные преобразования перестраиваемых иерархических систем распределения однородного ресурса // IX Нижегородская сессия молодых ученых. Математические науки. Тезисы докладов. Саров. 2004. С 6–7.

13. Афраймович Л.Г. Потоковые алгоритмы исследования совместности иерархических систем распределения ресурсов с ограничениями // Вестник Нижегородского государственного университета. Математическое моделирование и оптимальное управление. 2006. 2(31). С. 129-138.

14. Афраймович Л.Г. Потоковые алгоритмы распределения ресурсов в иерархических системах // Тезисы докладов НТК «Технические, программные и математические аспекты управления сложными распределенными системами». Нижний Новгород. 2003. С. 6.

15. Афраймович Л.Г. Приближенный алгоритм решения многоиндексных транспортных задач с декомпозиционной структурой // Проблемы теоретической кибернетики. Материалы XVI Международной конференции (Нижний Новгород, 20-25 июня 2011 г.) – Нижний Новгород: изд-во Нижегородского госуниверситета, 2011. С. 42-45.

16. Афраймович Л.Г. Проблема существования решения в многоресурсных иерархических системах // Вестник ВГАВТ. Межвузовская серия Моделирование и оптимизация сложных систем. 2005. Вып. 14. с. 122–127.

17. Афраймович Л.Г. Равномерное перераспределение ресурсов в иерархических системах транспортного типа // Технологии Microsoft в теории и практике программирования. Материалы конференции. – Нижний Новгород: Изд-во Нижегородского госуниверситета. 2007. С. 320–321.

18. Афраймович Л.Г. Распределение ресурсов в иерархических системах транспортного типа с ограничениями. Построение математических моделей и их исследование // Труды НГТУ. Системы обработки информации и управления. 2005. Т. 45. Вып. 23. С. 27–34.

19. Афраймович Л.Г. Сведение системы линейных неравенств транспортного типа к задаче поиска максимального потока в сети при дополнительных ограничениях // Проблемы теоретической кибернетики. Тезисы докладов XIV Международной конференции. – М.: Изд-во механико-математического факультета МГУ. 2005. С. 13.

20. Афраймович Л.Г. Трехиндексные задачи линейного программирования с вложенной структурой // Автоматика и телемеханика. 2011. № 8. С. 109–120.

21. Афраймович Л.Г. Учебно-методическое пособие по курсу «Модели и методы эффективного использования распределенных вычислительных систем» при изучении темы «Задачи статической балансировки». – Нижний Новгород: Нижегородский госуниверситет. 2011.

22. Афраймович Л.Г. Циклическая сводимость многоиндексных систем линейных неравенств транспортного типа // Известия РАН. Теория и системы управления. 2010. № 4. С. 83–90.

23. Афраймович Л.Г. Батищев Д.И., Костюков В.Е., Прилуцкий М.Х., Шагалиев Р.М. Статическая балансировка параллельных методов моделирования газодинамических процессов // Параллельные вычислительные технологии (ПаВТ'2009): Труды международной научной конференции – Челябинск: Изд. ЮУрГУ, 2009. С. 364–369.

24. Афраймович Л.Г., Куликов М.С., Прилуцкий М.Х. Распределение производительности купола по газовым скважинам // Технологии Microsoft в теории и практике программирования. Материалы конференции. – Нижний Новгород: Изд-во Нижегородского госуниверситета. 2009. С. 350–352.

25. Афраймович Л.Г. Катеров А.С. Трех- и четырехиндексные задачи с вложенной структурой // Математическое моделирование. Оптимальное управление. Вестник Нижегородского университета им. Н.И. Лобачевского. 2012. № 3 (1). С. 163–169.

26. Афраймович Л.Г. Прилуцкий М.Х. Методические указания для самостоятельной работы студентов по курсу «Моделирование сложных систем» при изучении темы «Распределение ресурсов в многоиндексных иерархических системах». – Нижний Новгород: Нижегородский государственный университет. 2006.

27. Афраймович Л.Г., Прилуцкий М.Х. Многоиндексные задачи оптимального планирования производства // Автоматика и телемеханика. 2010. №. 10. С. 148–155.

28. Афраймович Л.Г., Прилуцкий М.Х. Многоиндексные задачи распределения ресурсов в иерархических системах // Автоматика и телемеханика. 2006. № 6. С.194–205.

29. Афраймович Л.Г., Прилуцкий М.Х. Многопродуктовые потоки в древовидных сетях // Известия РАН. Теория и системы управления. 2008. № 2. С. 57–63.

30. Афраймович Л.Г., Прилуцкий М.Х. Поиск потока в несовместных транспортных сетях // Управление большими системами. 2009. Вып. 24. С. 147–168.

31. Афраймович Л.Г. Прилуцкий М.Х. Распределение ресурсов в иерархических системах транспортного типа. Учебно-методический материал по программе повышения квалификации «Новые подходы в исследованиях и разработках информационно-телекоммуникационных систем и технологий». – Нижний Новгород: Нижегородский госуниверситет. 2007.

32. Афраймович Л.Г., Прилуцкий М.Х. Распределение ресурсов в несовместных многоиндексных иерархических системах // Дискретные модели в теории управляющих систем: VIII Международная конференция, Москва, 6-9 апреля 2009 г. Труды.: МАКС Пресс. 2009. С. 18–23.

33. Афраймович Л.Г., Прилуцкий М.Х. Сводимость задачи распределения ресурсов в иерархических системах древовидной структуры к потоковым задачам // Технологии Microsoft в теории и практике программирования. Материалы конференции. – Нижний Новгород: Изд-во Нижегородского госуниверситета. 2006. С. 25–26.

34. Афраймович Л.Г., Прилуцкий М.Х. Трехиндексные транспортные задачи с вложенной структурой // Материалы X Международного семинара «Дискретная математика и ее приложения» – М.: Изд-во механико-математического факультета МГУ, 2010. С. 286–288.

35. Афраймович Л.Г., Прилуцкий М.Х., Бухвалова И.Р., Старостин Н.В., Филимонов А.В. Оптимизационные задачи оперативного управления работой компрессорной станцией // Электронный журнал «Исследовано в России». 2008. 032. С. 375–382. http://zhurnal.ape.relarn.ru/articles/2008/032.pdf

36. Афраймович Л.Г., Прилуцкий М.Х., Шумилов В.Б., Старостин Н.В., Филимонов А.В. Оптимизационные задачи объемно-календарного планирования для предприятий по переработке газового конденсата // Электронный журнал «Исследовано в России». 2008. 031. С. 365–374. http://zhurnal.ape.relarn.ru/articles/2008/031.pdf

37. Афраймович Л.Г., Прилуцкий М.Х., Шумилов В.Б., Старостин Н.В., Филимонов А.В. Оптимизационные задачи планирования транспорта газа в

магистральном газопроводе // Электронный журнал «Исследовано в России». 2008. 033. С. 383–391. http://zhurnal.ape.relarn.ru/articles/2008/033.pdf

38. Бабенко М.А. О потоках в простых двунаправленных и кососимметрических сетях // Проблемы передачи информации. 2006. Т. 42. Вып. 4. С. 104–120.

39. Березнев В.А. О полиномиальной сложности одной модификации симплекс-метода // Журнал вычислительной математики и математической физики. 2004. Т. 44. № 7. С. 1244–1260.

40. Берж К. Теория графов и ее приложения. – М.: ИЛ. 1962.

41. Булавский В.А., Звягина Р.А., Яковлева М.А. Численные методы линейного программирования. Специальные задачи. – М.: Наука. 1977.

42. Верховский Б.С. Многомерные задачи линейного программирования типа транспортной // Доклады АН СССР. 1963. Т. 151. № 3. С. 515–518.

43. Гейл Д. Теория линейных экономических моделей. – М.: Мир. 1969.

44. Гимади Э.Х., Глазков Ю.В. Об асимптотически точном алгоритме решения одной модификации трехиндексной планарной задачи о назначениях // Дискретный анализ и исследование операций. Сер. 2. 2006. Т. 13. № 1. С. 10–26.

45. Гимади Э.Х., Коркишко Н.М. Об одном алгоритме решения трехиндексной аксиальной задачи о назначениях на одноциклических подстановках // Дискретный анализ и исследование операций. Сер. 1. 2003. Т. 10. № 2. С. 56–65.

46. Гимади Э.Х., Сердюков А.И. Аксиальные трехиндексные задачи о назначении и коммивояжера: быстрые приближенные алгоритмы и их вероятностный анализ // Известия высших учебных заведений. Математика. 1999. № 12. С. 19–25.

47. Голиков А.И., Евтушенко Ю.Г. Нахождение нормального решения задачи линейного программирования / / Динамика неоднородных систем. № 10. С. 104–117.

48. Гольштейн Е.Г., Юдин Д.Б. Задачи линейного программирования транспортного типа. – М.: Наука. 1969.

49. Гольштейн Е.Г., Юдин Д.Б. Новые направления в линейном программировании М: Советское радио, 1966.

50. Голиков А.И., Евтушенко Ю.Г., Моллаверди Н. Применение метода Ньютона к решению задач линейного программирования большой размерности // Журнал вычислительной математики и математической физики. 2004. Т. 44. № 9. С. 1564–1573.

51. Гофман А.Д., Краскал Д.Б. Целочисленные граничные точки выпуклых многогранников // Линейные неравенства и смежные вопросы. – М.: ИЛ. 1959. С. 325–347.

52. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир. 1982.

53. Давидсон М. Р., Малашенко Ю. Е., Новикова Н. М. и др. Математические постановки задач восстановления и обеспечения живучести для многопродуктовых сетей. – М.: ВЦ РАН. 1993.

54. Диниц Е.А. Алгоритм решения задачи о максимальном потоке в сети со степенной оценкой // Доклады АН СССР. 1970. Т. 194. № 4. С. 754–757.

55. Дичковская С.А., Кравцов М.К. Исследование полиномиальных алгоритмов решения многокритериальной трехиндексной планарной задачи о назначениях // Журнал вычислительной математики математической физики. 2007. Т. 47. № 6. С. 1077–1086.

56. Дичковская С.А., Кравцов М.К. Исследование полиномиальных алгоритмов решения трехиндексной планарной проблемы выбора // Журнал вычислительной математики математической физики. 2006. Т. 46. № 2. С. 222–228.

57. Емеличев В.А., Ковалев М.М., Кравцов М.К. Многогранники, графы, оптимизация. – М.: Наука. 1981.

58. Емеличев В.А., Кравцов М.К. Полиэдральные аспекты многоиндексных аксиальных транспортных задач // Дискретная математика, 1991. Т. 3. Вып. 2. С. 3–24.

59. Зуховицкий С.И., Авдеева Л.И. Линейное и выпуклое программирование. – М.: Наука. 1967.

60. Канатников А.Н., Крищенко А.П. Линейная алгебра. – М.: Изд-во МГТУ им. Н.Э. Баумана. 2002.

61. Канторович Л.В. Математические методы организации и планировании производства. – Л.: Изд-во ЛГУ, 1939.

62. Канторович Л.В. Экономический расчёт наилучшего использования ресурсов. – М.: Изд-во АН СССР, 1960.

63. Карзанов А.В. Нахождение максимального потока в сети методом предпотоков // Доклады АН СССР. 1974. Т. 215. №1. С. 49–52.

64. Ковалев М.М. Матроиды в дискретной оптимизации. – М: Едиториал УРСС. 2003.

65. Костюков В.Е., Прилуцкий М.Х. Распределение ресурсов в иерархических системах. Оптимизационные задачи добычи, транспорта газа и переработки газового конденсата. – Нижний Новгород: Изд-во Нижегородского государственного университета. 2010.

66. Кравцов В.М., Кравцов М.К., Лукшин Е.В. О типах (3n−2)-нецелочисленных вершин многогранника трехиндексной аксиальной задачи выбора // Известия высших учебных заведений. Математика. 2002. № 12. С. 84–90.

67. Кравцов М.К., Крачковский А.П. Асимптотический подход к решению многоиндексной аксиальной транспортной задачи // Журнал вычислительной математики и математической физики. 1998. Т. 38. № 7. С. 1133–1139.

68. Кравцов М.А., Крачковский А.П. О некоторых свойствах трехиндексных транспортных многогранников // Дискретная математика. 1999. Т. 11. Вып. 3. С. 109–125.

69. Кравцов М.А., Крачковский А.П. О полиномиальном алгоритме нахождения асимптотически оптимального решения трехиндексной планарной проблемы выбора // Журнал вычислительной математики математической физики. 2001. Т. 41. № 2. С. 342–345.

70. Кравцов М.К., Кравцов В.М. О типах максимально нецелочисленных вершин релаксационного многогранника четырехиндексной аксиальной задачи о назначениях // Известия высших учебных заведений. Математика. 2012. № 3. С. 9–16 .

71. Кравцов М.К., Кравцов В.М., Лукшин Е.В. О нецелочисленных вершинах многогранника трехиндексной аксиальной задачи о назначениях // Дискретная математика. 2001. Т. 13. Вып. 2. С. 120–143.

72. Кравцов М.К., Лукшин Е.В. О нецелочисленных вершинах многогранника трехиндексной аксиальной транспортной задачи // Автоматика и телемеханика. 2004. № 3. С. 71–79.

73. Кропанов В.А., Рублев В.С. Равномерное назначение работ минимальной стоимости // Дискретная математика. 2001. Т. 13. Вып. 4. 2001. С. 144–156.

74. Литвак Б.Г., Раппопорт А.М. Задачи линейного программирования, допускающие сетевую постановку // Экономика и математические методы. 1970. Т. 6. Вып. 4. С. 594–604.

75. Мазалов В.В. Математическая теория игр и приложения. – Санкт-Петербург-Москва-Краснодар: Лань, 2010.

76. Малашенко Ю. Е., Новикова Н. М. Потоковые задачи анализа уязвимости многопродуктовых сетей. – М.: ВЦ АН СССР. 1989.

77. Меламед И.И., Сигал И.Х. Вычислительное исследование алгоритмов решения бикритериальных задач дискретного программирования // Журнал вычислительной математики и математической физики. 2000. Т. 4. № 11. С. 1602–1610.

78. Меламед И.И., Сигал И.Х. Вычислительное исследование трехкритериальных задач о деревьях и назначениях // Журнал вычислительной математики и математической физики. 1998. Т. 38. № 10. С. 1780–1787.

79. Миронов А.А., Цурков В.И. Замкнутые транспортные модели с минимаксным критерием // Автоматика и телемеханикаю. 2002. № 3. С. 50–61.

80. Миронов А.А., Цурков В.И. Минимакс в транспортных задачах. – М.: Наука. 1997.

81. Нестеров Ю.Е. Метод линейного программирования с трудоемкостью $O(n^3L)$ операций // Экономика и математические методы. 1988. Т. 24. № 16. С. 174–176.

82. Пападимитриу Х., Стайглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. – М.: Мир. 1985.

83. Прилуцкий М.Х. Многокритериальное распределение однородного ресурса в иерархических системах // Автоматика и телемеханика. 1996. № 2. С. 24–29.

84. Прилуцкий М.Х. Многокритериальные многоиндексные задачи объемно-календарного планирования // Известия РАН. Теория и системы управления. 2007. № 1. С. 78–82.

85. Прилуцкий М.Х. Распределение однородного ресурса в иерархических системах древовидной структуры // Труды международной конференции «Идентификация систем и задачи управления SICPRO’2000». – М.: ИПУ им. В.А. Трапезникова РАН. 2000. С. 2038–2049.

86. Прилуцкий М.Х., Афраймович Л.Г. Оптимальное распределение однородного ресурса в иерархических системах с доходами // Вестник ВГАВТ. Межвузовская серия Моделирование и оптимизация сложных систем. 2004. Вып. 9. С. 56–63.

87. Прилуцкий М.Х., Афраймович Л.Г. Параллельные алгоритмы распределения ресурсов в иерархических системах с лексикографическим упорядочиванием элементов // Материалы второго Международного научно-практического семинара «Высокопроизводительные параллельные вычисления на кластерных системах». – Нижний Новгород: Издательство Нижегородского госуниверситета. 2002. С. 243–248.

88. Прилуцкий М.Х., Афраймович Л.Г. Параллельные структуры потоковых и итерационных алгоритмов распределения ресурса в иерархических системах // Материалы третьего Международного научно-практического семинара «Высокопроизводительные параллельные вычисления на кластерных системах». – Нижний Новгород: Издательство Нижегородского госуниверситета. 2003. С. 140–145.

89. Прилуцкий М.Х., Афраймович Л.Г. Условия совместности многоиндексных систем транспортного типа // Электронный журнал "Исследовано в России", 70. 2005. С. 762–767. http://zhurnal.ape.relarn.ru/articles/2005/070.pdf

90. Прилуцкий М.Х., Афраймович Л.Г., М.С. Куликов. Об одном классе многокритериальных задач квадратичного программирования транспортного типа // Математическое моделирование. Оптимальное управление. Вестник Нижегородского университета им. Н.И. Лобачевского. 2009. № 6 (1). С. 178–183.

91. Прилуцкий М.Х. Власов С.Е. Многокритериальные задачи объемного планирования. Лексикографические схемы // Информационные технологии. 2005. № 7. С. 61–66.

92. Прилуцкий М.Х. Куликова Е.А. Построение Парето-области для многокритериальных задач распределения ресурсов с кусочно-постоянными функциями критериев // Системы управления и информационные технологию. 2011. Т. 44. № 2. С. 16–21.

93. Раскин Л.Г., Кириченко И.О. Многоиндексные задачи линейного программирования. – М.: Радио и связь. 1982.

94. Рублев В.С., Смирнов А.В. Задача целочисленного сбалансирования трехмерной матрицы и алгоритмы ее решения // Моделирование и анализ информационных систем. 2010. Т. 17. № 2. С. 72–98.

95. Рублев В.С., Смирнов А.В. NP-полнота задачи сбалансирования трехмерной матрицы // Доклады Академии наук. 2010. Т. 435. № 3. С. 314–316.

96. Рублев В.С., Чаплыгина Н.Б. Выбор критерия оптимизации в задаче о равномерном назначении // Дискретная математика. 2005. Т. 17. Вып. 4. С. 150–157.

97. Серая О.В., Дунаевская О.И. Многоиндексные нелинейные транспортные задачи // Інформаційно-керуючі системи на залізничному транспорті. 2009. №5. С.25–30.

98. Сергеев С.И. Новые нижние границы для трипланарной задачи назначения. Использование классической модели // Автоматика и телемеханика. 2008. № 12. С. 53–75.

99. Сергеев С.И. Улучшенные нижние границы для решения квадратичной задачи назначения // Автоматика и телемеханика. 2004. № 11. С. 49–63.

100. Сигал И.Х., Иванова А.П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы. Учебное пособие. – М.: Физмалит. 2007.

101. Смирнов А.В. Задача целочисленного сбалансирования трехмерной матрицы и сетевая модель // Моделирование и анализ информационных систем. 2009. Т. 16. № 3. С. 70–76.

102. Схрейвер А. Теория линейного и целочисленного программирования: в 2 т. – М.: Мир. 1991.

103. Триус Е.Б. Задачи математического программирования транспортного типа. М.: Советское радио. 1967.

104. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. – М.: Мир. 1984.

105. Финкельштейн Ю.Ю. Приближенные методы и прикладные задачи дискретного программирования. М.: Наука. 1976.

106. Форд Л., Фалкерсон Д. Потоки в сетях. – М.: Мир. 1966.

107. Ху Т. Целочисленное программирование и потоки в сетях. – М.: Мир. 1974.

108. Хачиян Л.Г. Полиномиальный алгоритм в линейном программировании // Доклады АН СССР. 1979. Т. 244. № 5. С. 1093–1096.

109. Черников С.Н. Линейные неравенства. – М.: Мир. 1966.

110. Afraimovich L.G. Generalized model of homogeneous resource distribution in hierarchy systems // VI International congress on mathematical modeling. Book of abstracts. Nizhny Novgorod. 2004. P. 65.

111. Afraimovich L.G. Reconstructing hierarchy systems of homogeneous resource distribution // Избранные вопросы современной математики: Тез. Междунар. науч. конф., приуроченной к 200-летию со дня рождения великого немецкого математика Карла Густава Якоби и 750-летию со дня основания г. Калининграда (Кенигсберга). – Калининград: Изд-во КГУ. 2005. С. 64–66.

112. Afraimovich L.G., Prilutskii M.Kh. Multi-commodity min-cost flow problem in a directed tree structured graph // V Московская международная конференция по исследованию операций (ORM 2007), посвященная 90-летию со дня рождения академика Н.Н. Моисеева. – М.:Макс Пресс. 2007. С. 233–234.

113. Aggarwal C., Ahuja R.K., Hao J., Orlin J.B. Diagnosing infeasibilities in network flow problems // Mathematical Programming. 1998. V. 81. N 3. P. 263–280.

114. Agmon S. The relaxation method for linear inequalities // Canadian Journal of Mathematics. 1954. V. 6. № 3. P. 382–392.

115. Ahuja R.K., Magnati T.L., Orlin J.B. Network flows: theory, algorithms, and applications. Prentice Hall. 1993.

116. Alighanbari M., How J.P. Cooperative task assignment of unmanned aerial vehicles in adversarial environments // Proceedings of the American Control Conference. 2005. V. 7. P. 4661–4666.

117. Amaldi E., Pfetsch M.E., Leslie E.T. On the maximum feasible subsystem problem, IISs and IIS-hypergraphs // Mathematical Programming, 2003. V. 95. N 3. P. 533–554.

118. Arbib C., Pacciarelli D., Smriglio S. A. A three-dimensional matching model for perishable production scheduling // Discrete Applied Mathematics. 1999. V. 92. P. 1–15.

119. Balas E., Saltzman M.J. An algorithm for the three-index assignment problem // Operations Research. 1991. V. 39. P. 150–161.

120. Bandelt H.J., Crama Y., Spieksma F.C.R. Approximation algorithms for multidimensional assignment problems with decomposable costs // Discrete Applied Mathematics. 1994. V. 49. P. 25–50.

121. Bandopadhyaya L., Puri M.C. Impaired flow multi-index transportation problem with axial constraints // Journal of Australian Mathematical Society. Series B. 29(3). 1988. P. 296–309.

122. Borradaile G., Klein P., Mozes S., Nussbaum Y., Wulff-Nilsen C. Multiple-source multiple-sink maximum flow in directed planar graphs in near-linear time // Proceedings of 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS). Palm Springs. 2011. P. 170–179.

123. Briskorn D., Drexl A., Spieksma F.C.R. Round robin tournaments and three index assignment // 4OR: a Quarterly Journal of Operations Research. 2010. V. 8. P. 365–374.

124. Burkard R.E., Çela E. Heuristics for biquadratic assignment problems and their computational comparison // European Journal of Operational Research. 1995. V. 83. P. 283–300.

125. Burkard R.E., Cela E., Pardalos P.M., Pitsoulis L. The quadratic assignment problem / Pardalos P.P., Resende M.G.C. (Eds.). Handbook of Combinatorial Optimization. – Dordrecht: Kluwer Academic Publishers. 1998. P. 241–238.

126. Burkard R.E., Dell'Amico M., Martello S. Assignment Problems. – Philadelphia: SIAM. 2009.

127. Burkard R. E., Rudolf R., Woeginger G. J. Computational investigation on 3-dimensional axial assignment problems // Belgian Journal of Operations Research, Statistics and Computer Science. 1992. V. 32. P. 85–98.

128. Burkard R.E., Rudolf R., Woeginger G.J. Three-dimensional axial assignment problems with decomposable cost coefficients // Discrete Applied Mathematics. 1996. V. 65. P. 123–139.

129. Chen B., Potts C.N., Woeginger G.J. A review of machine scheduling. Complexity, algorithms and approximability / Handbook of Combinatorial Optimization. Kluwer Academic Publishers. 1998. V. 3. P. 21–169.

130. Cosares S., Hochbaum D.S. Strongly polynomial algorithms for the quadratic transportation problem with a fixed number of sources // Mathematics of Operations Research archive. 1994. V. 19. P. 94–111.

131. Crama Y., Spieksma F.C.R. Approximation algorithms for three-dimensional assignment problems with triangle inequalities // European Journal of Operational Research. 1992. V. 60. P. 273–279.

132. Daskalai S., Birbas T., Housos E. An integer programming formulation for a case study in university timetabling // European Journal of Operational Research. 2004. V. 153. P. 117–135.

133. Dantzig G.B. Linear programming and extensions. – Princeton, NJ: Princeton University Press. 1963.

134. De Loera J., Hemmecke R., Onn S., Weismantel R. N-fold integer programming // Discrete Optimization. 2008. V. 5. P. 231–241.

135. De Loera J. A., Kim E.D., Onn S., Santos F. Graphs of transportation polytopes // Journal of Combinatorial Theory. Series A. 2009. V. 116. N. 8. P. 1306–1325.

136. De Loera J., Onn S. All linear and integer programs are slim 3-way transportation programs // SIAM Journal on Optimization. 2006. V. 17(3). P. 806–821.

137. De Loera J., Onn S. All Rational polytopes are transportation polytopes and all polytopal integer sets are contingency tables // Integer Programming and Combinatorial Optimization. Lecture Notes in Computer Science. 2004. V. 3064. P. 338–351.

138. De Loera J., Onn S. The complexity of three-way statistical tables // SIAM Journal on Computing. 2004. V. 33. P. 819–836.

139. Delona J., Salomonb J., Sobolevskiĭc A. Local matching indicators for concave transport costs // Comptes Rendus Mathematique. 2010. V. 348. P. 901–905.

140. Diaconis P., Gangolli A. Rectangular arrays with fixed margins // Discrete Probability and Algorithms. The IMA Volumes in Mathematics and its Applications. 1995. V. 72. P 15–41.

141. Dokka T., Kouvela A., Spieksma F.C.R. Approximating the multi-level bottleneck assignment problem // Operations Research Letters. 2012. V. 40. P. 282–286.

142. Duncan G.T., Fienberg S.E., Krishnan R., Padman R., Roehrig S.F. Disclosure limitation methods and information loss for tabular data / P. Doyle, J. Lane, J. Theeuwes, and L. Zayatz (Eds.). Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies. 2001. Elsevier. P. 135–166.

143. Edmonds J., Karp R.M. Theoretical improvements in algorithmic efficiency for network flow problems // Journal of the ACM. 1972. V. 19. N 2. P. 248–264.

144. Eisenbrand F. Fast integer programming in fixed dimension // Algorithms - ESA 2003. Lecture Notes in Computer Science. 2003. V. 2832. P. 196–207.

145. Franz L.S., Miller J.L. Scheduling medical residents to rotations: Solving the large-scale multiperiod staff assignment problem // Operations Research. 1993. V. 41. N 2. P. 269–279.

146. Frieze A.M. Complexity of a 3-dimensional assignment problem // European Journal of Operational Research. 1983. V. 13(2). P. 161–164.

147. Frieze A.M., Yadegar J. An algorithm for solving 3-dimensional assignment problems with application to scheduling a teaching practice // The Journal of the Operational Research Society. 1981. V. 32. N. 11. P. 989–995.

148. Gairing M., Lucking T., Mavronicolas M., Monien B. Computing Nash equilibria for scheduling on restricted parallel links // Proceedings of 36th annual ACM symposium on theory of computing. P. 613–622.

149. Galil Z., Tardos E. An mincost flow algorithm // Journal of the ACM. 1988. V. 35. N. 2. P. 374–386.

150. Glover F. Flows in arborescences // Management Science. 1971. V. 17. N 9. P. 568–586.

151. Goldberg A.V., Rao S. Beyond the flow decomposition barrier // Journal of the ACM. 1998. V. 45. N 5. P. 783–797.

152. Goossens D., Polyakovskiy S., Spieksma F.C.R., Woeginger G.J. The approximability of three-dimensional assignment problems with bottleneck objective // Optimization Letters. 2010. V. 4. P. 7–16.

153. Gülpinar N., Gutin G., Mitra G., Zverovitch A. Extracting pure network submatrices in linear programs using signed graphs // Discrete Applied Mathematics. 2004. V. 137. N. 3. P. 359–372.

154. Gunawan A., Ng K.M., Poh K.L. Solving the teacher assignment-course scheduling problem by a hybrid algorithm // International Journal of Computer and Information Engineering. 2007. V. 1. N 2. P. 136–141.

155. Chinneck J.W. Dravnieks E.R. Locating minimal infeasible constraint sets in linear programs // ORSA Journal on Computing. 1991. V. 3. N 3. P. 157–168.

156. Goldberg A. V., Tarjan R. E. Solving minimum-cost flow problems by successive approximation // Mathematics of Operations Research, 1990. V. 15. N 3. P. 430–466.

157. Junginger W. On representatives of multi-index transportation problems // European Journal of Operational Research. 1993. V. 66(3). P. 353–371.

158. Karmarkar N. A new polynomial time algorithm for linear programming // Combinatorica. 1984. V. 4. P. 373–395.

159. Kämpke T. The geometry of linear infeasibility // Applied Mathematics and Computation. 2002. V. 129. N 2-3. P. 317–337.

160. Klee V., Minty G.J. How good is the simplex algorithm? / Shisha O. (Ed.). Inequalities III. – New York, NY: Academic Press. 1972. P. 159–175.

161. Klinz B., Woeginger G.J. A new efficiently solvable special case of the three-dimensional axial bottleneck assignment problem // Lecture Notes in Computer Science. 1996. V. 1120. P. 150–162.

162. Krokhmal P., Murphey R., Pardalos P., Uryasev S., Zrazhevski G. Robust decision making: addressing uncertainties / Butenko et al. (Eds.). Cooperative Control: Models, Applications and Algorithms. Kluwer Academic Publishers. 2003. P. 165–185

163. Lenstra H.W. Jr. Integer programming with a fixed number of variables // Mathematics of Operations Research. 1983. V. 8. N. 4. P. 538–548.

164. Lim A., Rodrigues B., Zhang X. Scheduling sports competitions at multiple venues – Revisited // European Journal of Operational Research. 2006. V. 175. P. 171–186.

165. Lin Y. A recognition problem in converting linear programming to network flow models // Applied Mathematics - A Journal of Chinese Universities. 1993. V. 8. N. 1. P. 76–85.

166. Luby M., Nisan N. A parallel approximation algorithm for positive linear programming // Proceedings of 25th annual ACM symposium on Theory of computing. STOC'93. 1993. P. 448–457.

167. Magos D. Tabu search for the planar three-index assignment problem // Journal of Global Optimization. 1996. V. 8. P. 35–48.

168. Malvestuto F.M., Mezzini M., Moscarini M. Auditing sum-queries to make a statistical database secure // ACM Transactions on Information and System Security. 2006. V. 9. P. 31–60.

169. Martens M., Skutella M. Flows on few paths: algorithms and lower bounds // Networks. 2006. V. 48. N 2. P. 68-76.

170. McCormick S.T. How to compute least infeasible flows // Mathematical Programming. 1997. V. 78. N 2. P. 179–194.

171. Motzkin T.S., Schoenberg I.J. The relaxation method for linear inequalities // Canadian Journal of Mathematics. 1954. V. 6. N 3. P. 393–404.

172. Orlin J.B. A Faster strongly polynomial minimum cost flow algorithm // Operations research. 1993. V. 41. N 2. P. 338–350.

173. Orlin J.B. Max flows in O(nm) time, or better // Proceedings of the 45th annual ACM symposium on theory of computing. 2013. P. 765–774.

174. Pentico D.W. Assignment problems: A golden anniversary survey // European Journal of Operational Research. 2007. V. 176. P. 774–793.

175. Poore A.B. Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking // Computational Optimization and Applications. 1994. V. 3. N 1. P. 27–57.

176. Pusztaszeri J.F., Rensing P.E., Liebling T.M. Tracking elementary particles near their primary vertex: A combinatorial approach // Journal of Global Optimization. 1996. V. 9. P. 41–64.

177. Queyranne M., Spieksma F.C.R. Approximation algorithms for multi-index transportation problems with decomposable costs // Discrete Applied Mathematics. 1997. V. 76. P. 239–253.

178. Shapley L.S., Shubik M. The assignment game: the core // International Journal of Game Theory. 1972. V. 1. P. 111–130.

179. Sipser M. Introduction to the theory of computation. Boston: PWS Publishing Company. 1997.

180. Skutella M. An introduction to network flows over time // Research Trends in Combinatorial Optimization. 2009. P. 451-482.

181. Sleator D.D., Tarjan R.E. A data structure for dynamic trees // Journal of Computer and System Sciences. 1983. V. 26. P. 362–391.

182. Spieksma F.C.R. Multi index assignment problems: complexity, approximation, applications / P.M. Pardalos, L.S. Pitsoulis (Eds.). Nonlinear Assignment Problems. Algorithms and Applications. – Dordrecht: Kluwer Academic Publishers. 2000. P. 1–11.

183. Spieksma F.C.R., Woeginger G.J. Geometric three-dimensional assignment problems // European Journal of Operational Research. 1996. V. 91. P. 611–618.

184. Storms P.P.A., Spieksma F.C.R. An LP-based algorithm for the data association problem in multitarget tracking // Computers and Operation Research. 2003. V. 30. N 7. P. 1067–1085.

185. Vartak M.N., Geetha S. Specially structured precedence constraints in three-dimensional bottleneck assignment problems // Journal of the Operational Research Society. 1990. V. 41. N 4. P. 339–344.

186. Vlach M. Conditions for the existence of solutions of the three-dimensional planar transportation problem // Discrete Applied Mathematics. 1986. V. 13. P. 61–78.

Приложения

Приложение 1

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2010614640

Программное обеспечение «Заказ-О» (ПО «Заказ-О»)

Правообладатель(ли): ***Федеральное государственное унитарное предприятие «Российский федеральный ядерный центр - Всероссийский научно-исследовательский институт экспериментальной физики» - ФГУП «РФЯЦ-ВНИИЭФ» (RU)***

Автор(ы): ***Прилуцкий Михаил Хаимович, Старостин Николай Владимирович, Афраймович Лев Григорьевич, Филимонов Андрей Викторович, Фотин Сергей Валентинович, Дикарев Константин Игоревич (RU)***

Заявка № **2010612907**

Дата поступления **26 мая 2010 г.**

Зарегистрировано в Реестре программ для ЭВМ

14 июля 2010 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

Приложение 2

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2010614637

Программное обеспечение «Нагнетатель» (ПО «Нагнетатель»)

Правообладатель(ли): ***Федеральное государственное унитарное предприятие «Российский федеральный ядерный центр - Всероссийский научно-исследовательский институт экспериментальной физики» - ФГУП «РФЯЦ-ВНИИЭФ» (RU)***

Автор(ы): ***Прилуцкий Михаил Хаимович, Старостин Николай Владимирович, Афраймович Лев Григорьевич, Филимонов Андрей Викторович, Фотин Сергей Валентинович, Дикарев Константин Игоревич (RU)***

Заявка № **2010612840**

Дата поступления **26 мая 2010 г.**

Зарегистрировано в Реестре программ для ЭВМ

14 июля 2010 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

Приложение 3

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2011614445

Программное обеспечение «Проектировщик-1»
(ПО «Проектировщик-1»)

Правообладатель(ли): ***Федеральное государственное унитарное предприятие «Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт экспериментальной физики» - ФГУП «РФЯЦ-ВНИИЭФ» (RU)***

Автор(ы): ***Прилуцкий Михаил Хаимович, Старостин Николай Владимирович, Афраймович Лев Григорьевич, Филимонов Андрей Викторович, Фотин Сергей Валентинович, Дикарев Константин Игоревич (RU)***

Заявка № **2011612459**

Дата поступления **11 апреля 2011 г.**

Зарегистрировано в Реестре программ для ЭВМ

6 июня 2011 г.

Руководитель Федеральной службы по интеллектуальной собственности, патентам и товарным знакам

Б.П. Симонов

Приложение 4

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2012614244

Программное обеспечение «Проектировщик-2»
(ПО «Проектировщик-2»)

Правообладатель(ли): ***Федеральное государственное унитарное предприятие «Российский федеральный ядерный центр – Всероссийский научно-исследовательский институт экспериментальной физики» - ФГУП «РФЯЦ-ВНИИЭФ» (RU)***

Автор(ы): ***Прилуцкий Михаил Хаимович, Старостин Николай Владимирович, Афраймович Лев Григорьевич, Филимонов Андрей Викторович, Дикарев Константин Игоревич (RU)***

Заявка № **2012611929**

Дата поступления **20 марта 2012 г.**

Зарегистрировано в Реестре программ для ЭВМ
12 мая 2012 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

Приложение 5

УТВЕРЖДАЮ

Директор НПК - Главный конструктор

ФГУП "РФЯЦ-ВНИИЭФ"

_______________ С.Ф. Перетрухин

М.П.

СПРАВКА

об использовании результатов диссертационной работы Л.Г.Афраймовича

«**Потоковые методы решения многоиндексных задач транспортного типа**»

при решении задач планирования и проектирования в газоперерабатывающей и газотранспортной отрасли.

Программные системы «Проектировщик-1», «Проектировщик-2» – диалоговые программные системы проектирования и модернизации магистральных газопроводных систем и программная система «Заказ-О» – система поддержки принятия эффективных решений при формировании портфеля заказов на продукцию переработки газового конденсата, были апробированы ФГУП "РФЯЦ-ВНИИЭФ" в период 2010-2012 г.г. при решении соответствующих задач планирования и проектирования в газоперерабатывающей и газотранспортной области.

Разработанные при создании программных систем алгоритмы основаны на предложенных в диссертационной работе Л.Г. Афраймовича «Потоковые методы решения многоиндексных задач транспортного типа» методах исследования многоиндексных транспортных задач. Проведенные вычислительные эксперименты показали эффективность прилагаемых подходов при принятии решения в задачах планирования и проектирования.

Начальник комплексного
научно-исследовательского отдела В.Л. Хробостов

Приложение 6

ОТКРЫТОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО
«ОПЫТНОЕ КОНСТРУКТОРСКОЕ БЮРО МАШИНОСТРОЕНИЯ
имени И.И. Африкантова»

Бурнаковский проезд, 15
г. Н. Новгород, 603074
Тел. (831)2752640

ОАО «ОКБМ АФРИКАНТОВ»
ПРЕДПРИЯТИЕ ГОСКОРПОРАЦИИ «РОСАТОМ»

Факс (831)2418772,
http://www.okbm.nnov.ru
E-mail: okbm@okbm.nnov.ru

СПРАВКА

об использовании результатов диссертационной работы Л.Г.Афраймовича

«Потоковые методы решения многоиндексных задач транспортного типа»

при решении задач объёмно-календарного планирования.

В 2012 году программный модуль расчета оптимального плана производства для диалоговой системы объемно-календарного планирования производственных мощностей, функционирующей на предприятии, составляющий прикладную часть диссертационной работы Л.Г.Афраймовича «Потоковые методы решения многоиндексных задач транспортного типа», был передан для опытной эксплуатации в ОАО «ОКБМ Африкантов». Программа прошла апробацию при решении задач объёмно-календарного планирования производственных мощностей предприятия. Полученные результаты свидетельствуют об адекватности математических моделей, заложенных в основу созданного программного продукта, условиям реального производства.

Заместитель директора по ИТ В.В. Штарев

Подпись В.В. Штарева удостоверяю:

Главный ученый секретарь А.В. Беспалов

Приложение 7

УТВЕРЖДАЮ

Заместитель директора по качеству и информационным технологиям

_______________ Д.В. Седаков

СПРАВКА

об использовании результатов диссертационной работы Л.Г.Афраймовича

«**Потоковые методы решения многоиндексных задач транспортного типа**»

для планирования процесса изготовления сложных изделий.

В период с 2007 по 2012 гг. версии программной системы ПО «Кристалл», включающие прикладную часть диссертационной работы Л.Г. Афраймовича «Потоковые методы решения многоиндексных задач транспортного типа», были переданы для эксплуатации в ФГУП «ФНПЦ НИИИС им. Ю.Е. Седакова». Программное обеспечение было использовано при долгосрочном планировании процесса изготовления кристаллов СБИС. Проведенные тесты позволяют говорить об эффективности прилагаемых подходов при решении поставленной задачи долгосрочного планирования. Полученные результаты составления графиков выполнения заказов при изготовлении кристаллов СБИС свидетельствуют об адекватности применяемых математических моделей.

Начальник НИО прикладной математики,
к.т.н., с.н.с. _______________ В.Ф. Морозов

Старший научный сотрудник, к.т.н. _______________ В.С. Власов

Приложение 8

УТВЕРЖДАЮ
Проректор ННГУ по научной работе
профессор Гурбатов С.Н.

____

"26" марта 2013 г.

АКТ ВНЕДРЕНИЯ РЕЗУЛЬТАТОВ ДИССЕРТАЦИОННОЙ РАБОТЫ Л.Г. АФРАЙМОВИЧА «ПОТОКОВЫЕ МЕТОДЫ РЕШЕНИЯ МНОГОИНДЕКСНЫХ ЗАДАЧ ТРАНСПОРТНОГО ТИПА» В УЧЕБНЫЙ ПРОЦЕСС ФАКУЛЬТЕТА ВМК ННГУ

Материалы диссертационной работы Л.Г. Афраймовича ««Потоковые методы решения многоиндексных задач транспортного типа» были внедрены в учебный процесс факультета вычислительной математики и кибернетики ННГУ в 2006/2007 учебном году при преподавании курса «Моделирование сложных систем», 2010/2011 учебном году при преподавании курса «Модели и методы эффективного использования распределенных вычислительных систем» и спецсеминара магистров кафедры ИАНИ.

Результаты, полученные в диссертационной работе Л.Г. Афраймовича, нашли свое отражение в следующих учебно-методических материалах:

Афраймович Л.Г. Учебно-методическое пособие по курсу «Модели и методы эффективного использования распределенных вычислительных систем» при изучении темы «Задачи статической балансировки». – Нижний Новгород: Нижегородский госуниверситет. 2011. – 13 с.

Афраймович Л.Г. Прилуцкий М.Х. Методические указания для самостоятельной работы студентов по курсу «Моделирование сложных систем» при изучении темы «Распределение ресурсов в многоиндексных иерархических системах». – Нижний Новгород: Нижегородский государственный университет. 2006. – 18 с.

Афраймович Л.Г. Прилуцкий М.Х. Распределение ресурсов в иерархических системах транспортного типа. Учебно-методический материал по программе повышения квалификации «Новые подходы в исследованиях и разработках информационно-телекоммуникационных систем и технологий». – Нижний Новгород: Нижегородский госуниверситет. 2007. – 80 с.

Декан факультета ВМК,
профессор В.П. Гергель

Зав. кафедрой ИАНИ факультета ВМК,
профессор М.Х. Прилуцкий