

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

На правах рукописи



Рассказова Варвара Андреевна

**Математическое моделирование в задачах планирования
и организации железнодорожных перевозок методами
теории графов и комбинаторной оптимизации и
численные методы их решения**

Специальность 05.13.18 —

«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
доктор физико-математических наук, профессор
Кибзун Андрей Иванович

Москва — 2017

Оглавление

Введение	4
Глава 1. Математическое моделирование в задачах	
планирования и организации железнодорожных	
перевозок	
1.1. Основные используемые понятия теории графов	17
1.2. Теоретико–графовые модели в задачах планирования и	
организации железнодорожных перевозок	23
1.2.1. Теоретико–графовая модель для решения задачи	
формирования бесконфликтного набора нормативных	
нитек	23
Ориентированный мультиграф	23
Неориентированный граф конфликтов	28
1.2.2. Теоретико–графовая модель для решения задачи о	
назначении и перемещении локомотивов	30
Глава 2. Задача планирования железнодорожных перевозок на	
этапе формирования бесконфликтного набора	
нормативных нитек	
2.1. Постановка	35
2.2. Решение	39
2.2.1. Алгоритм «Бегущая волна»	39
2.2.2. Алгоритм расшифровки монотонной булевой функции . .	43
Жадный поиск	46
Поиск с возвратом	53
Глава 3. Задача организации железнодорожных перевозок на	
этапе назначения и перемещения локомотивов	
3.1. Постановка	55
3.1.1. Теоретико–множественный подход	55
3.1.2. Теоретико–графовый подход	60
3.2. Решение	66

3.2.1. Алгоритм назначения и перемещения локомотивов	67
3.2.2. Алгоритм покрытия	72
Глава 4. Проблемно–ориентированные программные комплексы	84
4.1. Программный комплекс для решения задачи формирования бесконфликтного набора нормативных ниток	84
4.2. Программный комплекс для решения задачи о назначении и перемещении локомотивов	100
Заключение	120
Литература	122

Введение

Актуальность темы. В работе исследуются задача планирования железнодорожных перевозок на этапе формирования бесконфликтного набора нормативных ниток, и задача организации железнодорожных перевозок на этапе назначения и перемещения локомотивов без учёта ограничений на использование и техническое обслуживание локомотивов. Ввиду высокой комбинаторной сложности прикладных задач управления перевозками актуальной представляется область разработки математических моделей, в рамках которых исследуемые задачи могут быть решены с помощью классического математического аппарата. Как правило точное решение задач комбинаторной оптимизации большой размерности является избыточным с точки зрения практической реализации решения, и, кроме того, поиск точного решения требует колоссальных вычислительных затрат. В этой связи особый интерес представляет область разработки эффективных вычислительных алгоритмов поиска приближённого решения. В то же время любой эвристический подход требует подтверждения работоспособности, и, таким образом, неотъемлемым этапом исследования выступает разработка комплекса прикладных программ, на основе которого осуществляется ряд вычислительных экспериментов, подтверждающих эффективность предложенных подходов к решению исследуемых задач.

В работах [56], [64], [62], [28], [48], [47], [3], [55], [31], [19] получили обоснование графовый и комбинаторный методы математического моделирования в приложении к решению прикладных задач управления железнодорожными перевозками. Разработанные в диссертации теоретико–графовые модели, кроме структурных свойств железнодорожных сетей, учитывают также и комбинаторный характер исследуемых задач, что позволяет значительно расширить область разработки вычислительных алгоритмов решения.

Задача планирования железнодорожных перевозок исследуется, в том числе посредством методов теории графов и комбинаторной оптимизации, в контексте задач теории расписаний в работах [36], [37], [38], [52], [53]. В рамках разработанной теоретико–графовой модели исследуемая прикладная задача планирования сводится к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом. В работах [4], [5], [61], [6], [7], [8], [23], [24], [25], [26], [27], [39], [40], [41], [45], [42], [43], [44] получены важные резуль-

таты в теории противоречивых систем условий, множество максимальных совместных подсистем которых может быть специальным образом поставлено в соответствие множеству максимальных по размеру клик графа. Этот подход получил продолжение в разработке вычислительных алгоритмов для формирования верхнего нуля монотонной булевой функции, особенностью которых является оценка точности приближённого решения.

В работах [30], [1], [29], [2] задача организации железнодорожных перевозок на этапе назначения и перемещения локомотивов сводится к задаче стохастического программирования. Постановка задачи организации без учёта ограничений на использование и техническое обслуживание локомотивов имеет определённое практическое обоснование в части нижней оценки точности решения. В рамках разработанной теоретико–графовой модели, исследуемая прикладная задача организации сводится к задаче покрытия вершин ориентированного графа минимальным числом путей. Структурные свойства специфического ориентированного графа совместимости заданий на перевозку позволяют ограничиться рассмотрением множества максимальных по включению путей для покрытия вершин графа, и, таким образом, размерность задачи может быть существенно снижена.

В области разработки алгоритмов комбинаторной оптимизации, линейного, целочисленного и динамического программирования, а также алгоритмов на графах, в том числе приближённых алгоритмов, существенные результаты получены в [21], [49], [34], [54]. Вычислительные алгоритмы формирования верхнего нуля монотонной булевой функции, порождённой неориентированным графом, и алгоритм покрытия вершин ориентированного графа — суть комбинаторный и комбинаторно–графовый алгоритмы, на основе которых в диссертационной работе разработаны проблемно–ориентированные программные комплексы для решения исследуемых прикладных задач планирования и организации железнодорожных перевозок.

Целью работы является разработка математических моделей и вычислительных алгоритмов для решения прикладных задач планирования и организации железнодорожных перевозок.

Для достижения поставленной цели решаются следующие **задачи**:

1) разработка математической модели для решения задачи планирования железнодорожных перевозок на этапе формирования бесконфликтного набора нормативных ниток. Исследование свойств неориентированного графа кон-

фликтов и сведение исходной задачи к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом,

2) исследование свойств максимального верхнего нуля и разработка вычислительного алгоритма формирования верхнего нуля монотонной булевой функции, порождённой неориентированным графом,

3) разработка математической модели для решения задачи организации железнодорожных перевозок на этапе назначения и перемещения локомотивов без учёта ограничений на использование и техническое обслуживание локомотивов. Исследование свойств ориентированного графа совместимости заданий на перевозку и сведение исходной задачи к задаче минимального покрытия вершин ориентированного графа множеством путей,

4) исследование свойств минимального покрытия и разработка вычислительного алгоритма покрытия вершин ориентированного графа множеством максимальных по включению путей,

5) разработка и тестирование программных комплексов, реализующих алгоритм формирования верхнего нуля монотонной булевой функции, порождённой неориентированным графом, и алгоритм покрытия вершин ориентированного графа множеством максимальных по включению путей.

Научная новизна. В рамках исследования получены следующие новые результаты:

1) разработана математическая модель для решения задачи планирования железнодорожных перевозок на этапе формирования бесконфликтного набора нормативных ниток,

2) доказано утверждение об оценке числа единиц в максимальном верхнем нуле и разработан вычислительный алгоритм формирования верхнего нуля монотонной булевой функции, порождённой неориентированным графом, и получена оценка вычислительной сложности разработанного алгоритма,

3) разработана математическая модель для решения задачи организации железнодорожных перевозок на этапе назначения и перемещения локомотивов без учёта ограничений на использование и техническое обслуживание локомотивов,

4) доказано утверждение о свойствах минимального покрытия и разработан вычислительный алгоритм покрытия вершин ориентированного графа множеством максимальных по включению путей.

Практическая значимость. Разработанные в диссертационной работе вычислительные алгоритмы решения исследуемых задач лежат в основе программных комплексов для технических управляющих систем, реализация которых и внедрение в эксплуатацию позволит добиться существенного экономического эффекта в области рационального распределения ресурсов. В рамках исследования разработаны на языке Visual Basic проблемно-ориентированные программные комплексы «Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом» для решения задачи планирования железнодорожных перевозок, и «Алгоритм покрытия вершин ориентированного графа множеством максимальных по включению путей» для решения задачи организации железнодорожных перевозок.

Методы исследования. Для разработки математических моделей для решения исследуемых задач используются методы теории графов. Для разработки алгоритмов на графах, вычислительного алгоритма формирования верхнего нуля монотонной булевой функции, порождённой неориентированным графом, а также для разработки вычислительного алгоритма покрытия вершин ориентированного графа, используются методы комбинаторной оптимизации. Для разработки проблемно-ориентированных программных комплексов, реализующих алгоритмы решения исследуемых задач, и для проведения вычислительных экспериментов используются компьютерные технологии.

Достоверность полученных результатов обеспечивается использованием классического аппарата моделирования и адекватностью приложения выбранной методологии в исследуемых задачах. Реализация посредством компьютерных технологий подтверждает корректность разработанных алгоритмов, кроме того, исходные данные для вычислительных экспериментов отвечают реальным планам перевозок для существующих железнодорожных транспортных сетей и апробированы на примере оптимизации планирования железнодорожных перевозок для Московской железной дороги, имеющей наиболее развитый и сложный граф сети среди всех дорог ОАО РЖД.

Апробация работы. Основные результаты работы докладывались на следующих научных конференциях: 1) Международная научная конференция «Гагаринские чтения» (Москва, 2016, 2017), 2) Международная научная конференция «Системный анализ, управление и навигация» (Евпатория, 2016, 2017), 3) Всероссийская научная конференция «Управление большими системами»

(Самара, 2016), 4) Международная научная конференция «Математика, информатика и физика и их приложения в науке и образовании» (Москва, 2016).

Личный вклад. Автором работы сформулированы утверждение об оценке числа единиц в максимальном верхнем нуле монотонной булевой функции, порождённой неориентированным графом, и утверждение о свойстве специфического ориентированного графа, на основе которых, совместно с Гайнановым Д. Н., разработаны вычислительные алгоритмы решения исследуемых задач. Посредством программных комплексов на языке Visual Basic автором реализованы разработанные алгоритмы, проведены вычислительные эксперименты и анализ полученных результатов.

Публикации. Основные результаты по теме диссертации изложены в 11 работах, 4 из которых опубликованы в журналах, рекомендованных ВАК [9], [10], [11], [12], в том числе 2 опубликованы в журналах, цитируемых международной базой Web of Science [9], [10], 6 из которых опубликованы в тезисах докладов [13], [14], [15], [16], [18], [17], и 1 из которых — зарегистрированный комплекс программ [50].

Объем и структура работы. Диссертация состоит из введения, четырех глав и заключения. Полный объем диссертации 128 страниц текста с 11 рисунками и 2 таблицами. Список литературы содержит 71 наименование.

В **первой главе** приводятся основные понятия теории графов и свойства объектов теории графов, используемых в работе, разработаны подходы к математическому моделированию в задачах планирования и организации железнодорожных перевозок.

Решение задачи планирования для заданного множества нормативных ниток графика движения поездов состоит в формировании множества бесконфликтных наборов нормативных ниток, каждый элемент которого может служить допустимым расписанием для практической организации железнодорожных перевозок. Математической теоретико-графовой моделью для решения задачи планирования на этапе формирования множества бесконфликтных наборов нормативных ниток служит неориентированный граф конфликтов.

Вводится в рассмотрение ориентированный граф сети, в котором каждая дуга характеризует ориентированный перегон между соседними станциями, соответствующими вершинам. Для каждой дуги ориентированного графа сети заданы технические ограничения на скорость отправления и скорость прибытия, максимальную скорость движения, вес поезда, допустимый к перевозке,

время отправления из начальной станции маршрута и общее время движения по соответствующему ориентированному перегону. На основе этих параметров могут быть рассчитаны стратегии движения, отвечающие наименьшим энергозатратам на перевозку, и для каждой дуги ориентированного графа, при различных значениях заданных параметров, может быть сформировано множество энергоэффективных стратегий движения по соответствующему ориентированному перегону. Такая конструкция задаёт ориентированный мультиграф энергоэффективных стратегий движения, который не имеет петель, но пара вершин может быть связана несколькими дугами. Множество нормативных ниток графика движения поездов отвечает множеству путей в ориентированном мультиграфе энергоэффективных стратегий движения, и устроено как объединение композиций энергоэффективных стратегий, при этом последовательность дуг образует путь в графе в том и только в том случае, если параметры дуг, следующих друг за другом, не противоречат практической реализации перевозки по соответствующим ориентированным перегонам.

На множестве нормативных ниток вводится отношение конфликтности таким образом, что пара нормативных ниток (последовательность дуг ориентированного мультиграфа энергоэффективных стратегий движения) являются конфликтными в случае невозможности одновременной организации соответствующих перевозок. При этом различают однонаправленный, при следовании в одном направлении, и разнонаправленный, при следовании в противоположных направлениях, конфликты. Подмножество нормативных ниток, в котором любая пара элементов не имеет конфликтов, образует бесконфликтный набор нормативных ниток. Различные наборы образуют множество бесконфликтных наборов нормативных ниток графика движения поездов, в котором каждый элемент может служить допустимым расписанием для организации железнодорожных перевозок. Формирование такого множества является задачей комбинаторной оптимизации, которая, в рамках разработанной теоретико-графовой модели, сводится к задаче расшифровки монотонной булевой функции, порожденной неориентированным графом конфликтов. Каждая нормативная нитка соответствует единичному значению булевой переменной, и специальным образом заданная булева функция принимает нулевое значение на множестве переменных, если среди соответствующих нормативных ниток не найдётся ни одной конфликтной пары. Приоритетными для практического планирования перевозок являются наборы, содержащие наибольшее число попарно бесконфликтных

нормативных ниток. Таким образом, множество максимальных верхних нулей содержит наборы, единичные компоненты которых отвечают бесконфликтным нормативным ниткам.

Решение задачи организации железнодорожных перевозок для заданных вариантных графиков движения и перемещения поездов состоит в назначении и перемещении локомотивов для исполнения плана перевозок. Математической теоретико–графовой моделью для решения задачи организации на этапе назначения и перемещения локомотивов служит ориентированный граф совместимости заданий на перевозку.

В ориентированном графе сети заданы размеры движения в виде матрицы корреспонденций и план поездоформирования. По строкам и столбцам матрицы корреспонденций располагаются вершины ориентированного графа, и ненулевые элементы характеризуют количество поездов, необходимое к отправке для соответствующих станций. План поездоформирования соответствует множеству путей в графе, каждый из которых является допустимым для организации перевозок. Вариантный график движения поездов является подмножеством некоторого бесконфликтного набора нормативных ниток, в котором каждая нитка соответствует некоторому пути из плана поездоформирования, и для каждого элемента матрицы корреспонденций существуют нормативные нитки в достаточном количестве. Таким образом, вариантный график движения содержит по крайней мере одну нормативную нитку, соответствующую некоторому элементу из плана поездоформирования, для исполнения каждой запланированной перевозки. Вариантный график перемещений устроен аналогичным образом. Если для некоторых фиксированных станций соответствующий элемент размеров движения ненулевой, то существует некоторое количество (одно или более) перемещений в направлении, обратном заданному. Тогда вариантный график перемещений содержит по крайней мере одну нормативную нитку, соответствующую некоторому элементу из плана поездоформирования, для исполнения каждого существующего перемещения. При этом варианты графики движения и перемещений образуют бесконфликтный набор нормативных ниток, то есть любая пара ниток в объединении вариантных графиков является бесконфликтной.

Объединение вариантных графиков порождает ориентированный граф совместимости заданий на перевозку, в котором каждая нормативная нитка взаимно однозначно соответствует некоторой вершине, и пара вершин связана

дугой, если соответствующие нормативные нитки совместимы по станциям и скорости движения для некоторого действительного неотрицательного числа, характеризующего нормативное время, необходимое для подготовки локомотива назначения локомотива для исполнения следующей нитки. С целью исполнения заданного плана перевозок требуется установить соответствие между множеством локомотивов и множеством вершин графа, соответствующих нормативным ниткам из вариантного графика движения. В рамках разработанной теоретико–графовой модели для решения задачи требуется сформировать множество путей, где каждый элемент соответствует некоторому набору нормативных ниток, для исполнения которых будет назначен локомотив. Приоритетным для практической организации перевозок является исполнение плана посредством наименьшего количества локомотивов. Таким образом, покрытие вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей отвечает решению задачи организации на этапе назначения и перемещения локомотивов.

Во **второй главе** разработаны вычислительные теоретико–графовый и комбинаторный алгоритмы решения задачи планирования железнодорожных перевозок.

Теоретико–графовый алгоритм формирования множества бесконфликтных наборов нормативных ниток графика движения поездов основан на свойствах неориентированного графа конфликтов. Период планирования разбит на трехчасовые интервалы, и для каждой нормативной нитки фиксируется интервал, в котором начинается движение по соответствующей нитке. Если для некоторой последовательности интервалов определён набор ниток, то из всего множества выбираются нитки, движение по которым начинается в следующем текущем интервале, и бесконфликтные с заданным набором. При этом по определению подграф неориентированного графа конфликтов, порождённый выбранными нормативными нитками, не имеет рёбер. Ограничение на время начала движения в значительной мере снижает размерность задачи на каждом шаге, повышая тем самым качество решения — выбор для включения в бесконфликтный набор производится не из всех ниток, а только из некоторого их числа, всё же достаточно большого для эффективного планирования. Кроме того, при таком подходе выбор ниток, претендующих на включение в бесконфликтный набор, в каждые сутки периода планирования существенно расширяется, поскольку для исполнения фиксируются только некоторые поднаборы, объеди-

нение которых равномошно заданному плану перевозок на каждые сутки, в то время как нитки, не вошедшие в набор для исполнения, вновь претендуют на включение в некоторый бесконфликтный набор. Это обстоятельство обеспечивает высокую эффективность и практическую значимость алгоритма, получившего название «Бегущая волна» для оперативного планирования перевозок.

Комбинаторный алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом конфликтов, основан на свойствах максимальных верхних нулей. Вводятся в рассмотрение количественные характеристики для неориентированного графа конфликтов, отражающие число вершин в окрестности некоторой фиксированной вершины, и число ребер, связывающих попарно вершины в окрестности. Доказано, что булевы переменные, соответствующие вершинам с нулевым количеством рёбер в окрестности, гарантированно входят в некоторый набор из множества максимальных верхних нулей. На этом основан алгоритм жадного поиска. Входом алгоритма является неориентированный граф конфликтов, заданный списками смежности вершин, а выходом — некоторый двоичный набор, единичные компоненты которого соответствуют попарно бесконфликтным нормативным ниткам. На каждом шаге алгоритма фиксированной булевой переменной, соответствующей вершине с нулевым значением рёбер в окрестности, присваивается единичное значение, а всем переменным, соответствующим вершинам в окрестности, присваиваются нулевые значения. Все обозначенные переменные исключаются из рассмотрения, наряду с соответствующими вершинами, и алгоритм продолжает работу для вновь сформированного множества вершин. В диссертационной работе доказано утверждение о том, что если в результате работы алгоритма множество вершин пусто, то полученный двоичный набор является элементом множества максимальных верхних нулей, и, следовательно, отвечает некоторому решению задачи планирования на этапе формирования множества бесконфликтных наборов нормативных ниток. В случае непустого множества, всем необозначенным переменным присваиваются нулевые значения, и полученный набор является элементом множества верхних нулей, то есть соответствует некоторому бесконфликтному набору нормативных ниток, однако, не максимальному. Дальнейшее исследование продолжается посредством присвоения единичных значений переменным, соответствующим вершинам с наименьшим числом рёбер в окрестности. Доказано, что при таком подходе отклонение числа единиц в полученном наборе от числа единиц в максимальном верхнем нуле не превышает суммар-

ного количества рёбер в окрестностях вершин, соответствующих переменным с единичными значениями. Таким образом, алгоритм жадного поиска позволяет определить некоторый элемент множества максимальных верхних нулей или получить оценку числа единиц в максимальном верхнем нуле, или, в терминах исходной задачи, сформировать максимальный по включению бесконфликтный набор нормативных ниток, или оценить число ниток в максимальном бесконфликтном наборе. Оценка отклонения оказывается весьма полезной при параллельной реализации алгоритма жадного поиска и алгоритма поиска с возвратом. Последний на каждом шаге вычисляет количественные характеристики вершин, соответствующих булевым переменным, и выбирает те из них, для которых число вершин в окрестности максимально, а число рёбер, связывающих попарно вершины в окрестности, минимально. Такой подход существенно замедляет время работы алгоритма, однако либо полученный в результате набор является элементом множества максимальных верхних нулей, либо полученная оценка числа единиц в максимальном верхнем нуле довольно эффективна. Кроме того, принимая во внимание оценку, полученную посредством реализации алгоритма жадного поиска, в некоторых случаях удаётся установить точное число единиц в максимальном верхнем нуле, и, соответственно, число ниток в максимальном бесконфликтном наборе. Результаты работы алгоритмов жадного поиска и поиска с возвратом зависят от выбора начальной вершины, и реализация алгоритмов для каждого возможного выбора позволяет сформировать множество максимальных верхних нулей. С позиции внедрения алгоритмов в практическую деятельность, формирование всего множества не является необходимым, поскольку любого бесконфликтного набора нормативных ниток графика движения поездов достаточно для организации перевозок.

В **третьей главе** разработаны вычислительные теоретико–графовый и комбинаторно–графовый алгоритмы решения задачи организации железнодорожных перевозок.

Теоретико–графовый алгоритм устанавливает соответствие между множеством нормативных ниток, необходимых к исполнению, и множеством локомотивов, заданным начальными условиями доступности. Допустимое отображение удовлетворяет определённым условиям практической организации перевозок: начальные условия доступности локомотива, назначенного для исполнения некоторой нитки, совместимы со станцией и временем начала движения по нитке, кроме того, в последовательности ниток, для исполнения которых назначен

один и тот же локомотив, каждые соседние члены совместимы по станциям и времени движения, и в любой момент времени некоторый локомотив может быть назначен для исполнения только одной нитки. Вводится в рассмотрение двоичный массив, длина которого соответствует мощности множества ниток, которые могут быть использованы для исполнения перевозок, и вспомогательный алгоритм призван определить, является ли исполнение некоторой нитки актуальным в текущий момент времени. Приоритетными для исполнения являются нитки, соответствующие компоненты которых в массиве имеют нулевые значения. При таком подходе число перемещений локомотивов без нагрузки составов вагонов будет наименьшим. Алгоритм требует больших вычислительных затрат, поскольку сложность экспоненциально зависит от числа ниток на входе алгоритма, что можно частично преодолеть, установив некоторые ограничения на последовательные нитки, для исполнения которых назначен один и тот же локомотив. Рациональным представляется рассмотрение в качестве совместимых только тех ниток, для которых интервал между окончанием движения по нитке и началом движения по следующей нитке не превышает средней продолжительности ниток, необходимых для исполнения.

Комбинаторно–графовый алгоритм основан на сведении исходной задачи к задаче покрытия вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей. Доказано утверждение о специфическом свойстве матрицы смежности вершин рассматриваемого ориентированного графа, на основании которого множество путей графа может быть однозначно представлено в виде двоичной матрицы. Вершины ориентированного графа соответствуют нормативным ниткам, необходимым для исполнения, и располагаются по столбцам матрицы, и каждая строка отвечает некоторому элементу множества путей. Для сокращения количества используемых локомотивов, среди элементов множества путей выбираются максимальные по включению, поскольку при таком подходе каждый локомотив будет назначен для исполнения наибольшего количества нормативных ниток. Формирование множества максимальных путей является задачей комбинаторной оптимизации высокой размерности, для решения которой в работе разработан циклический алгоритм с ветвлением. Таким образом решение задачи организации железнодорожных перевозок сводится к решению задачи покрытия множества вершин ориентированного графа элементами множества максимальных путей. Доказано утверждение о свойстве покрытия вершин, согласно

которому покрытие вершин элементами множества максимальных путей, равно мощное минимальному покрытию, также будет минимальным. Для каждого элемента множества максимальных путей определяются количественные характеристики, называемые индексами, и строки двоичной (каждый элемент принимает значение 0 или 1) матрицы, отвечающие множеству максимальных путей, рассматриваются совместно с соответствующими индексами строк. Алгоритм покрытия реализует процесс сортировки максимальных по включению путей относительно обобщённого индекса, при этом приоритетным для включения в покрытие является путь, имеющий максимальный индекс строки, то есть число вершин, входящих в путь максимально, минимальный суммарный индекс строки, то есть число путей, содержащих нитки, входящие в фиксированный путь, минимально, и минимальный индекс нитки строки, то есть число путей, содержащих нитку, минимально по всем ниткам в строке. Нитки, входящие в выбранный путь, исключаются из рассмотрения, ввиду чего значения индексов обновляются. В результате последовательной реализации алгоритма формирования множества путей, алгоритма формирования множества максимальных путей и алгоритма покрытия вершин ориентированного графа совместимости заданий на перевозку имеем набор строк двоичной матрицы, каждая из которых соответствует некоторой последовательности нормативных ниток, необходимых для исполнения, причём каждая последовательность содержит максимальное число ниток, и количество последовательностей минимально, что отвечает некоторому решению задачи организации железнодорожных перевозок.

В четвёртой главе приводятся описания проблемно-ориентированных программных комплексов для решения задач планирования и организации железнодорожных перевозок и результаты вычислительных экспериментов.

Реализация алгоритма жадного поиска, также, как и алгоритма поиска с возвратом, позволяет получить либо точное решение, либо эффективную оценку числа единиц в максимальном верхнем нуле монотонной булевой функции, порождённой неориентированным графом. Для каждого алгоритма оценки могут отличаться, однако, в некоторых случаях, удаётся получить точную оценку. Так, для заданного неориентированного графа конфликтов, заданного списками смежности вершин, посредством реализации алгоритма жадного поиска получен некоторый набор из множества верхних нулей, и оценка числа единиц в максимальном верхнем нуле. Для того же неориентированного графа, посредством реализации алгоритма поиска с возвратом, получена усиленная оценка, и,

с учётом ранее полученного результата, доказано, что набор из множества верхних нулей является максимальным верхним нулём монотонной булевой функции, порождённой рассматриваемым неориентированным графом, и, соответственно, отвечает некоторому решению задачи планирования. Программный комплекс на языке Visual Basic «Алгоритм формирования множества максимальных верхних нулей монотонной булевой функции, порождённой неориентированным графом» для решения задачи планирования железнодорожных перевозок основан на описанном принципе действия, параллельно реализует разработанные алгоритмы.

Решение задачи покрытия вершин ориентированного графа минимальным числом максимальных по включению путей требует реализации нескольких алгоритмов. Для ориентированного графа совместимости заданий на перевозку, заданного матрицей смежности вершин, в первую очередь формируется множество строк двоичной матрицы, каждая из которых отвечает некоторому элементу множества путей. Доказано, что покрытие вершин графа множеством путей может быть найдено среди максимальных элементов, на основании чего, посредством реализации циклического алгоритма с ветвлением, формируется множество максимальных путей. Непосредственно алгоритм покрытия реализует операцию последовательной сортировки элементов множества максимальных путей относительно обобщённых индексов. Приоритетный путь включается в покрытие, и на каждом шаге вершины ориентированного графа, входящие в выбранный путь, исключаются из рассмотрения. Таким образом, после каждой итерации, значения индексов обновляются, и алгоритм продолжает работу до полного исчерпания множества вершин ориентированного графа. Полученный в результате набор максимальных путей является искомым покрытием, и отвечает некоторому решению задачи организации. Программный комплекс на языке Visual Basic «Алгоритм покрытия вершин ориентированного графа минимальным числом максимальных по включению путей» для решения задачи организации железнодорожных перевозок основан на описанном принципе действия, последовательно реализует разработанные алгоритмы.

Глава 1. Математическое моделирование в задачах планирования и организации железнодорожных перевозок

1.1. Основные используемые понятия теории графов

Методы теории графов широко используются для разработки математических моделей производственных процессов, коммуникационных сетей в части маршрутизации, в том числе, транспортных сетей, а также для разработки математической основы вычислительных систем и сетей передачи данных.

Под графом в общем смысле понимается пара

$$G = (V, E) , \quad (1.1.1)$$

где V – множество вершин, является подмножеством некоторого счётного множества, и $E \subseteq (V \times V)$ – множество рёбер, соединяющих пары вершин. Представление рассматриваемых объектов в виде графов позволяет не только эффективно исследовать структурные свойства, но и решать ряд алгоритмических задач — теория графов является универсальным средством для описания системных взаимосвязей. Разработка алгоритмов на графах является задачей повышенной комбинаторной сложности, и ключевым аспектом разработки алгоритмов является рациональное моделирование задачи.

Выбор подходящего типа графа, определённого своими фундаментальными свойствами, является важнейшим этапом в описании структуры данных. Так, для решения задачи планирования железнодорожных перевозок, в качестве теоретико–графовой модели выступает неориентированный граф конфликтов, и для решения задачи организации железнодорожных перевозок — ориентированный граф совместимости заданий на перевозку.

В неориентированном графе вида (1.1.1) рёбрами являются неупорядоченные пары вершин:

$$\forall v_i, v_j \in V : \{v_i, v_j\} \in E \implies \{v_j, v_i\} \in E ,$$

в общем случае граф называется ориентированным.

Для ориентированных графов в представленной работе принято обозначение:

$$\vec{G} = (V, E) , \quad (1.1.2)$$

где V – множество вершин, является подмножеством некоторого счётного множества, и $E \subseteq (V \times V)$ – множество дуг, каждым элементом которого является упорядоченная пара вершин.

Формальное описание неориентированного графа конфликтов приводится в Разделе 1.2.1. Вершинами при этом являются нормативные нитки графика движения поездов, а рёбрами соединены пары вершин, для которых соответствующие нормативные нитки имеют однонаправленный или разнонаправленный конфликты. Это свойство играет ключевую роль в Главе 2 для сведения исходной задачи к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом.

В ориентированном графе совместимости заданий на перевозку вершинами являются нормативные нитки из некоторого бесконфликтного набора, и пара вершин связаны дугой, если соответствующие нормативные нитки совместимы с точки зрения практической реализации перевозки. В Разделе 1.2.2 подробно описана процедура разработки теоретико–графовой модели для решения задачи организации. Ориентированный граф совместимости заданий на перевозку имеет ряд структурных особенностей, например, строение путей, лежащих в основе исследования в Главе 3.

На эффективную разработку алгоритмов существенное влияние оказывает выбор способа описания графа. Основными типами структуры данных на графах являются матрица смежности и списки смежности.

Не уменьшая общности, положим, что граф вида (1.1.1) или (1.1.2) имеет n вершин и m рёбер, то есть

$$V = \{v_1, v_2, \dots, v_n\} , |V| = n , \\ |E| = m .$$

Использование *списков смежности* для отражения связности вершин является самым распространённым способом задания данных на графах. Проверку на присутствие некоторого ребра в графе при заданных списках смежности выполнить достаточно трудно, однако, для разработки вычислительных алгоритмов этот способ имеет преимущество, ввиду возможности представления рёбер посредством массивов связанных списков. Для каждой вершины графа список

смежности имеет вид:

$$\forall v_i, v_{i_1}, v_{i_2}, \dots, v_{i_k} \in V: (v_i, v_{i_1}), (v_i, v_{i_2}), \dots, (v_i, v_{i_k}) \in E, \\ (v_i; v_{i_1}, v_{i_2}, \dots, v_{i_k}),$$

причём для неориентированного графа каждое ребро входит в структуру дважды. Так, например, для пары смежных вершин графа (1.1.1) списки смежности будут иметь вид:

$$(v_i; v_j, v_{i_1}, \dots, v_{i_k}), \\ (v_j; v_i, v_{j_1}, \dots, v_{j_k}),$$

где

$$v_i, v_j, v_{i_1}, \dots, v_{i_k}, v_{j_1}, \dots, v_{j_k} \in V: \\ \{v_i, v_j\}, \{v_i, v_{i_1}\}, \dots, \{v_i, v_{i_k}\}, \{v_j, v_{j_1}\}, \dots, \{v_j, v_{j_k}\} \in E.$$

В Главе 2 для решения задачи планирования исследуются свойства неориентированного графа конфликтов. Вводится в рассмотрение специальным образом заданная монотонная булева функция такая, что булевым переменным соответствуют вершины графа. На основании количественных характеристик вершин, разработаны приближённые комбинаторные алгоритмы формирования элементов множества максимальных верхних нулей рассматриваемой функции, причём в Разделах 2.2.2 и 4.1 для разработки и программной реализации алгоритмов неориентированный граф конфликтов задан списками смежности своих вершин.

Другим способом задания графа (1.1.1) или (1.1.2) является *матрица смежности* вида:

$$\mathcal{A}_G = \left\| \left\| a_{ij} \right\|_{[n \times n]} \right\|, \\ \mathcal{A}_G = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & \dots & a_{1(n-1)} & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \dots & \dots & a_{2(n-1)} & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \dots & \dots & a_{3(n-1)} & a_{3n} \\ \dots & \dots \\ \dots & \dots \\ a_{(n-1)1} & a_{(n-1)2} & a_{(n-1)3} & a_{(n-1)4} & \dots & \dots & 0 & a_{(n-1)n} \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & \dots & a_{n(n-1)} & a_{nn} \end{vmatrix},$$

где

$$a_{ij} = \begin{cases} 1, & \text{если } i \neq j \text{ и } (v_i, v_j) \in E, \\ 0, & \text{иначе,} \end{cases}$$

причём для неориентированного графа матрица смежности, очевидно, является симметричной.

В Разделе 3.2 для ориентированного графа совместимости заданий на перевозку построена матрица смежности, на основании чего доказано утверждение о свойстве множества путей рассматриваемого графа. Специальным образом заданное отображение позволяет представить каждый элемент из множества путей в виде строки двоичной матрицы, для которой разрабатываются эффективные алгоритмы формирования множества максимальных по включению строк и алгоритм сортировки строк относительно количественных характеристик, называемых обобщённым индексом.

Решение задачи планирования железнодорожных перевозок, в рамках теоретико-графового подхода, сводится к задаче о независимом множестве — подмножестве вершин неориентированного графа конфликтов, для которых ни одна пара вершин не связана ребром. Задача о максимальном независимом подмножестве алгоритмически идентична задаче о максимальной клике в дополнении графа конфликтов — подмножестве вершин графа, в котором каждая пара вершин связаны ребром, при том, что дополнение графа имеет то же множество вершин, и пара вершин связаны ребром в дополнении, в том и только в том случае, если в исходном графе эти вершины не смежны. На Рис. 1 граф G имеет максимальные независимые множества $\{v_1, v_4\}$, $\{v_2, v_3\}$, и его дополнение граф \bar{G} имеет максимальные клики $\{v_1, v_4\}$, $\{v_2, v_3\}$:

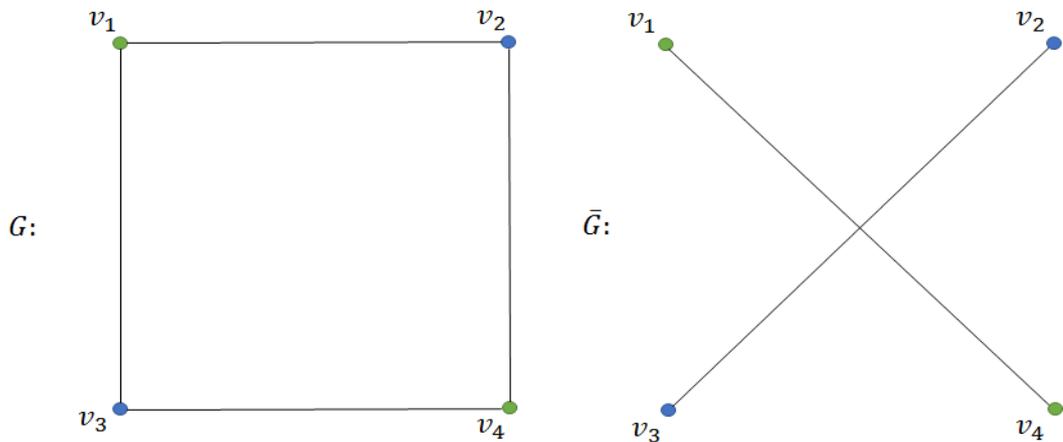


Рис. 1 Максимальные независимые множества графа и максимальные клики дополнения графа

Задачи о максимальном независимом множестве и максимальной клике графа являются классическими NP -полными, то есть слишком сложны, и для решения этих задач не существует на сегодняшний день алгоритмов с полиномиальным временем исполнения. Исчерпывающее решение таких задач предоставляет только алгоритм полного перебора с возвратом, однако эвристические алгоритмы на основе рандомизированных методов позволяют находить достаточно большие клики и независимые подмножества.

В Разделе 2.2.2 для решения задачи расшифровки монотонной булевой функции, порождённой неориентированным графом конфликтов, разработаны приближённые алгоритмы формирования элементов множества максимальных верхних нулей. По определению рассматриваемой булевой функции полученное решение соответствует некоторому независимому множеству неориентированного графа конфликтов, и, таким образом, отвечает решению задачи планирования. Для разработанного алгоритма поиска с возвратом получена оценка вычислительной сложности:

$$O(n \cdot nm + nm) = O(n^2m) ,$$

где входом является неориентированный граф вида (1.1.1).

Задача организации железнодорожных перевозок сводится к рассмотрению задачи о минимальном покрытии вершин ориентированного графа зависимости перевозок множеством максимальных по включению путей. В ориентированном графе вида (1.1.2) последовательность различных дуг и вершин, как следствие, также, различных, вида:

$$\left(v_{i_1}, (v_{i_1}, v_{i_2}), v_{i_2}, \dots, v_{i_{k-1}}, (v_{i_{k-1}}, v_{i_k}), v_{i_k} \right) : \\ v_{i_1}, v_{i_2}, \dots, v_{i_{k-1}}, v_{i_k} \in V , (v_{i_1}, v_{i_2}), \dots, (v_{i_{k-1}}, v_{i_k}) \in E ,$$

образует путь, и максимальный путь, если никакая вершина графа не может быть включена в последовательность.

Разработанный в Главе 3 алгоритм покрытия последовательно реализует алгоритм формирования множества путей графа, алгоритм формирования множества максимальных путей и алгоритм сортировки элементов множества максимальных путей. Известные рекурсивные алгоритмы обходов графов в ширину или в глубину предоставляют механизмы для посещения каждой вершины и каждой дуги графа вида (1.1.2), в рамках которых могут быть решены задачи формирования множества путей и множества максимальных путей. Разница

между алгоритмами обхода в ширину и в глубину заключается в порядке исследования вершин. На Рис. 2 в результате обхода в глубину множество путей, упорядоченное относительно начала обхода, имеет вид:

$$\begin{aligned} & (v_1, (v_1, v_2), v_2) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_4), v_4) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_5), v_5) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_5), v_5, (v_5, v_7), v_7) , \\ & (v_1, (v_1, v_3), v_3) , \\ & (v_1, (v_1, v_3), v_3, (v_3, v_6), v_6) , \\ & (v_1, (v_1, v_3), v_3, (v_3, v_8), v_8) , \end{aligned}$$

и в результате обхода того же графа в ширину множество путей, упорядоченное относительно начала обхода, имеет вид:

$$\begin{aligned} & (v_1, (v_1, v_2), v_2) , \\ & (v_1, (v_1, v_3), v_3) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_4), v_4) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_5), v_5) , \\ & (v_1, (v_1, v_3), v_3, (v_3, v_6), v_6) , \\ & (v_1, (v_1, v_3), v_3, (v_3, v_8), v_8) , \\ & (v_1, (v_1, v_2), v_2, (v_2, v_5), v_5, (v_5, v_7), v_7) . \end{aligned}$$

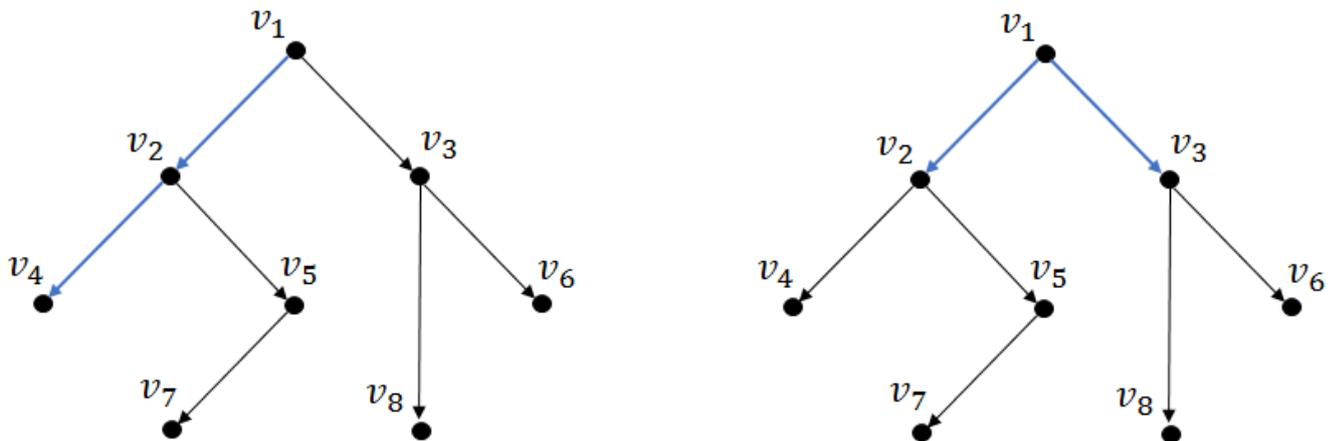


Рис. 2 Обход ориентированного графа в глубину и в ширину

Алгоритмы обходов достаточно сложны в части разработки и программной реализации, и часто используют вспомогательные графовые объекты такие, как компоненты связности и сильной связности, раскраски, что приводит к дополнительным вычислительным затратам. В некоторых случаях, для задач большой размерности, такие подходы оказываются необоснованными, ввиду чего в Главе 3 разработаны эффективные циклические алгоритмы с ветвлением для формирования множества путей и множества максимальных по включению путей, и циклический алгоритм для формирования покрытия. Входом алгоритмов является двоичная матрица, где каждая строка представляет элемент множества путей, и каждый столбец представляет вершину ориентированного графа совместимости заданий на перевозку. Решением задачи организации железнодорожных перевозок в такой постановке является покрытие всех столбцов двоичной матрицы. Результаты вычислительных экспериментов в Разделе 4.2 показывают, что в некоторых случаях разработанный алгоритм позволяет сформировать минимальное покрытие, то есть при исключении любого элемента из решения полученный набор строк двоичной матрицы перестаёт отвечать решению исходной задачи.

1.2. Теоретико–графовые модели в задачах планирования и организации железнодорожных перевозок

1.2.1. Теоретико–графовая модель для решения задачи формирования бесконфликтного набора нормативных ниток

Ориентированный мультиграф. Важный этап решения задачи планирования железнодорожных перевозок составляет формирование множества энергоэффективных нормативных ниток графика движения поездов. Энергоэффективность понимается как стратегия организации движения поездов по ориентированным перегонам железнодорожной сети, доставляющая наименьшие из возможных энергетические затраты. В рамках представленной работы не исследуется подробно вопрос расчетов энергетических затрат на организацию движения поездов по ориентированным перегонам железнодорожной сети, а также не

исследуется вопрос расчета некоторых физических параметров, используемых в постановке задачи формирования набора энергоэффективных нормативных ниток графика движения поездов.

Будем рассматривать ориентированный граф железнодорожной транспортной сети:

$$\vec{\Gamma} = (S, E) ,$$

где множество вершин $S = \{s_i : i = 1, 2, \dots, n\}$ – характеризует множество станций, и множество дуг $E \subseteq \{(s_i, s_j) : i = 1, 2, \dots, n, j = 1, 2, \dots, n, |i - j| = 1\}$ – характеризует множество ориентированных перегонов, связывающих соседние станции.

Пусть имеется линейный участок железной дороги (s_1, \dots, s_n) с начальной станцией s_1 и конечной станцией s_n , для которого выполнены тяговые расчеты (см., например, [48]).

Для каждого ориентированного перегона (s_i, s_j) , соответствующего некоторой дуге ориентированного графа сети

$$(s_i, s_j) : i = 1, 2, \dots, n, j = 1, 2, \dots, n, |i - j| = 1 ,$$

заданы:

- 1) профиль дороги $h_{i,j}$;
- 2) вес брутто поезда Q_{ij} , допустимый к перевозке на перегоне;
- 3) максимальная допустимая скорость $v_{\max}(s_i, s_j)$ движения на ориентированном перегоне (s_i, s_j) ;
- 4) скорость $\vec{v}_H(s_i, s_j)$ отправления поезда со станции s_i ;
- 5) скорость $\vec{v}_K(s_i, s_j)$ прибытия поезда на станцию s_j ;
- 6) время $t_H(s_i, s_j)$ отправления поезда со станции s_i ;
- 7) время $t_{i,j}$ движения поезда на ориентированном перегоне (s_i, s_j) .

Для каждого ориентированного перегона (s_i, s_j) при заданных 1)–7) можно выбрать график движения:

$$g_{i,j}(t) , t \in [t_H(s_i, s_j); t_H(s_i, s_j) + t_{i,j}] ,$$

как функцию расстояния, пройденного от станции s_i , при этом каждому графику $g_{i,j}(t)$ соответствуют энергозатраты на перевозку $E(g_{i,j}(\cdot))$. Тогда для организации движения по ориентированному перегону (s_i, s_j) при заданных 1)–7) будем выбирать график $g_{i,j}(t)$ таким образом, чтобы энергетические затраты

на организацию движения поезда по ориентированному перегону железнодорожной сети $E(g_{i,j}(\cdot))$ были минимальны.

В рамках настоящей работы не рассматриваются способы задания $g_{i,j}(t)$ и расчета $E(g_{i,j}(\cdot))$.

Определение 1.2.1. Энергоэффективной стратегией движения по ориентированному перегону (s_i, s_j) будем называть набор параметров

$$\vec{E}(s_i, s_j) = \left(\vec{v}_H(s_i, s_j), \vec{v}_K(s_i, s_j), t_H(s_i, s_j), t_{i,j}, g_{i,j}(\cdot) \right).$$

Для различных $\vec{v}_H(s_i, s_j)$, $\vec{v}_K(s_i, s_j)$, $t_H(s_i, s_j)$ и $t_{i,j}$ имеем различные $g_{i,j}(\cdot)$ и соответствующие $E(g_{i,j}(\cdot))$, тогда

$$\vec{E}^k(s_i, s_j) = \left(\vec{v}_H^k(s_i, s_j), \vec{v}_K^k(s_i, s_j), t_H^k(s_i, s_j), t_{i,j}, g_{i,j}^k(\cdot) \right),$$

где $k = 1, 2, \dots, N_{i,j}$, $N_{i,j}$ – количество стратегий движения по ориентированному перегону (s_i, s_j) , представляет собой k -ю энергоэффективную стратегию движения по ориентированному перегону (s_i, s_j) .

Таким образом для каждого ориентированного перегона (s_i, s_j) может быть построено множество $E(s_i, s_j)$ энергоэффективных стратегий движения

$$E(s_i, s_j) = \{ \vec{E}^k(s_i, s_j) \}, k = 1, 2, \dots, N_{i,j}.$$

На участке (s_1, \dots, s_n) множество энергоэффективных стратегий движения:

$$\mathcal{E} = \bigcup_{|i-j|=1} \{ E(s_i, s_j) : i = 1, 2, \dots, n, j = 1, 2, \dots, n \}$$

порождает ориентированный мультиграф с множеством вершин $V = \{v_i\}$, и множеством дуг $E = \{(v_i, v_j)_k\}$, $i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$, $k = 1, 2, \dots, N_{i,j}$:

$$\vec{G} = (V, E),$$

где вершина $v_i \in V$ соответствует станции $s_i \in (s_1, \dots, s_n)$, и дуга $(v_i, v_j)_k \in E$ соответствует энергоэффективной стратегии движения $\vec{E}^k(s_i, s_j) \in \mathcal{E}$. Другими словами, k -я дуга, соединяющая соседние вершины v_i и v_j , соответствующие начальной и конечной станциям ориентированного перегона (s_i, s_j) , соответствует k -й энергоэффективной стратегии движения по ориентированному перегону (s_i, s_j) . На Рис. 3 изображён фрагмент произвольного ориентированного мультиграфа энергоэффективных стратегий движения по некоторому участку железнодорожной транспортной сети.

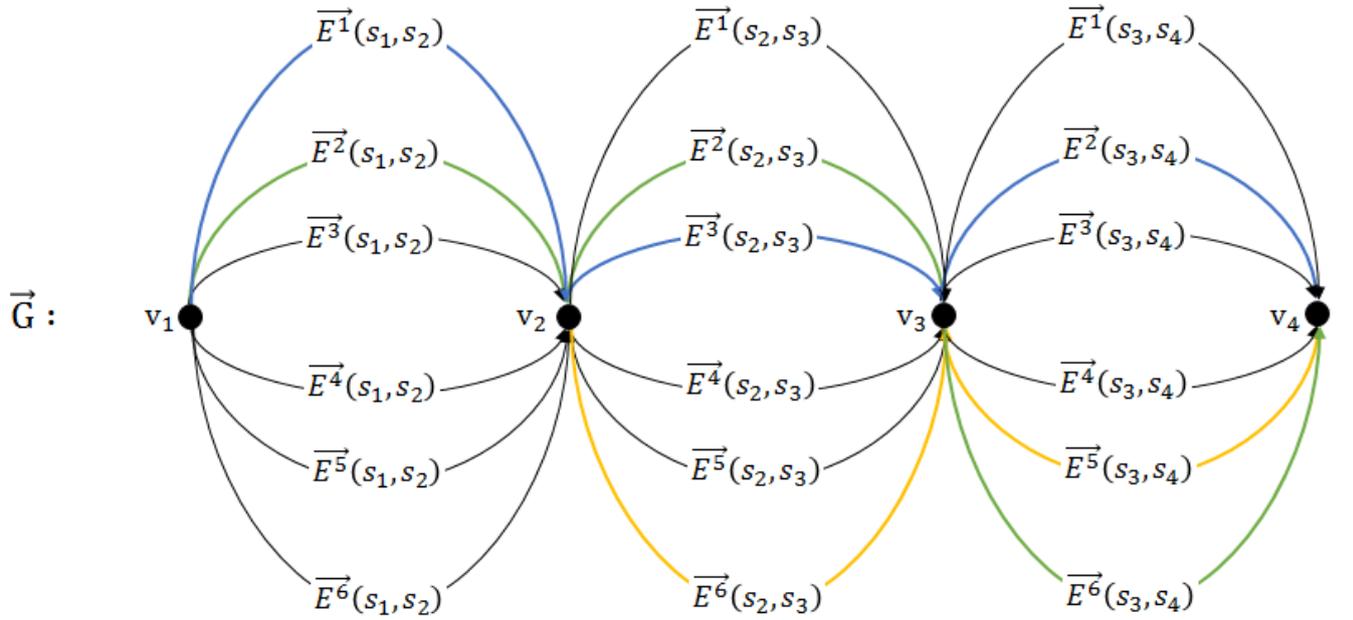


Рис. 3 Ориентированный мультиграф энергоэффективных стратегий движения

Для ориентированного мультиграфа на Рис. 3 множество вершин и дуг отвечает множеству станций и энергоэффективных стратегий движения по ориентированным перегонам, связывающим соседние станции:

$$\begin{aligned} \{v_1, v_2, v_3, v_4\} \subseteq V &\longrightarrow \{s_1, s_2, s_3, s_4\} \subseteq S, \\ \{(v_1, v_2)_k : k \in [1, 6]\} \subseteq E &\longrightarrow \left\{ \vec{E}^k(s_1, s_2) : k \in [1, 6] \right\} \subseteq E(s_1, s_2), \\ \{(v_2, v_3)_k : k \in [1, 6]\} \subseteq E &\longrightarrow \left\{ \vec{E}^k(s_2, s_3) : k \in [1, 6] \right\} \subseteq E(s_2, s_3), \\ \{(v_3, v_4)_k : k \in [1, 6]\} \subseteq E &\longrightarrow \left\{ \vec{E}^k(s_3, s_4) : k \in [1, 6] \right\} \subseteq E(s_3, s_4). \end{aligned}$$

В ориентированном мультиграфе \vec{G} для $i < j$ определим допустимые (v_i, v_j) -пути, как последовательности дуг графа таким образом, чтобы соседние дуги

$$(v_{i_1}, v_{i_2})_{k_1}, k_1 = 1, 2, \dots, N_{i-1, i}$$

и

$$(v_{i_2}, v_{i_3})_{k_2}, k_2 = 1, 2, \dots, N_{i, i+1},$$

следующие друг за другом в допустимом (v_i, v_j) -пути, были совместимы по скорости и времени. Условия совместимости выполнены, если энергоэффективные стратегии движения $\vec{E}^{k_1}(s_{i_1}, s_{i_2})$ и $\vec{E}^{k_2}(s_{i_2}, s_{i_3})$, соответствующие дугам

$(v_{i_1}, v_{i_2})_{k_1}$ и $(v_{i_2}, v_{i_3})_{k_2}$, таковы, что

$$\begin{cases} \vec{v}_H^{k_2}(s_{i_2}, s_{i_3}) = \vec{v}_K^{k_1}(s_{i_1}, s_{i_2}); \\ t_H^{k_2}(s_{i_2}, s_{i_3}) \geq t_H^{k_1}(s_{i_1}, s_{i_2}) + t_{i_1, i_2}. \end{cases}$$

Аналогично определяются допустимые (v_j, v_i) -пути.

Пусть, например, в ориентированном мультиграфе на Рис. 3 выполняются условия совместимости:

$$\begin{cases} \vec{v}_H^3(s_2, s_3) = \vec{v}_K^1(s_1, s_2), \\ t_H^3(s_2, s_3) \geq t_H^1(s_1, s_2) + t_{1,2}, \\ \vec{v}_H^2(s_3, s_4) = \vec{v}_K^3(s_2, s_3), \\ t_H^2(s_3, s_4) \geq t_H^3(s_2, s_3) + t_{2,3}, \\ \vec{v}_H^2(s_2, s_3) = \vec{v}_K^2(s_1, s_2), \\ t_H^2(s_2, s_3) \geq t_H^2(s_1, s_2) + t_{1,2}, \\ \vec{v}_H^6(s_3, s_4) = \vec{v}_K^2(s_2, s_3), \\ t_H^6(s_3, s_4) \geq t_H^2(s_2, s_3) + t_{2,3}, \\ \vec{v}_H^5(s_3, s_4) = \vec{v}_K^6(s_2, s_3), \\ t_H^5(s_3, s_4) \geq t_H^6(s_2, s_3) + t_{2,3}. \end{cases}$$

Тогда множество допустимых путей имеет вид:

$$\left\{ \left((v_1, v_2)_1, (v_2, v_3)_3, (v_3, v_4)_2 \right), \left((v_1, v_2)_2, (v_2, v_3)_2, (v_3, v_4)_6 \right), \left((v_2, v_3)_2, (v_3, v_4)_6 \right), \left((v_2, v_3)_6, (v_3, v_4)_5 \right) \right\}.$$

Определение 1.2.2. *Нормативной ниткой графика движения поезда называется любой допустимый (v_i, v_j) - или (v_j, v_i) -путь. Множество нормативных ниток графика движения поезда будем обозначать \mathcal{N} .*

Здесь график движения поезда на участке (s_1, s_n) устроен как множество композиций графиков движения по ориентированным перегонам, построенных таким образом, что дуги ориентированного мультиграфа, соответствующие энергоэффективным стратегиям, содержащим в качестве параметров эти графики, образуют допустимые (v_i, v_j) - или (v_j, v_i) -пути.

Неориентированный граф конфликтов. Планирование железнодорожных перевозок, помимо условия энергетической эффективности движения поездов по ориентированным перегонам железнодорожной сети, требует формирования множества бесконфликтных наборов нормативных ниток, каждый элемент которого представляет набор энергоэффективных нормативных ниток графика движения поездов. Бесконфликтность, с практической точки зрения, понимается как возможность реальной организации железнодорожного движения, согласно плану перевозок, содержащим в качестве отдельных элементов нормативные нитки из любого бесконфликтного набора энергоэффективных нормативных ниток графика движения поездов.

Множество нормативных ниток графика движения поездов суть множество допустимых ориентированных путей ориентированного мультиграфа энергоэффективных стратегий движения поездов по ориентированным перегонам железнодорожной сети. На множестве \mathcal{N} нормативных ниток графика введем бинарное отношение “конфликтности”.

Пусть для каждого ориентированного перегона (s_i, s_j) и нормативной нитки $n \in \mathcal{N}$ однозначно определен номер пути $W(s_i, s_j, n)$, по которому будет осуществляться движение поезда по нитке n по ориентированному перегону (s_i, s_j) .

Обозначим через d_{\min} некоторое заданное минимальное географическое расстояние, допустимое между поездами при движении по одному и тому же ориентированному перегону.

Рассматривается период планирования движения $[T_0, T]$, где T_0 и T – время начала и время окончания периода планирования соответственно.

Определение 1.2.3. Будем говорить, что две нормативные нитки n_1 и n_2 имеют однонаправленный конфликт на ориентированном перегоне (s_i, s_j) , если $W(s_i, s_j, n_1) = W(s_i, s_j, n_2)$ и они проходят его в одном и том же направлении по стратегиям $\vec{E}_{k_1}(s_i, s_j)$ и $\vec{E}_{k_2}(s_i, s_j)$ соответственно, и существует момент времени $t \in [T_0, T]$ из периода планирования, для которого

$$\left| g_{i,j}^{k_1}(t) - g_{i,j}^{k_2}(t) \right| \leq d_{\min}.$$

Аналогично определим разнонаправленный конфликт на ориентированном перегоне.

Определение 1.2.4. Будем говорить, что две нормативные нитки n_1 и n_2 имеют разнонаправленный конфликт на ориентированном перегоне (s_i, s_j) , если $W(s_i, s_j, n_1) = W(s_j, s_i, n_2)$ и одна из ниток проходит его в направлении (s_i, s_j) (для определенности, нитка n_1), а другая в направлении (s_j, s_i) (для определенности, нитка n_2) по стратегиям $\vec{E}_{k_1}(s_i, s_j)$ и $\vec{E}_{k_2}(s_j, s_i)$ соответственно, и существует момент времени $t \in [T_0, T]$ из периода планирования, для которого

$$\left| g_{i,j}^{k_1}(t) - \left(|s_{i,j}| - g_{j,i}^{k_2}(t) \right) \right| \leq d_{\min},$$

где $|s_{i,j}|$ – общая длина пути с номером $W(s_i, s_j, n_1) = W(s_j, s_i, n_2)$ на ориентированном перегоне (s_i, s_j) .

Обобщая понятия однонаправленного конфликта на ориентированном перегоне и разнонаправленного конфликта на ориентированном перегоне, определим конфликтные нормативные нитки.

Определение 1.2.5. Две нормативные нитки n_1 и n_2 называются конфликтными, если существует ориентированный перегон (s_i, s_j) , на котором они имеют однонаправленный или разнонаправленный конфликты.

Введем в рассмотрение математическую теоретико–графовую конструкцию, характеризующую структурные и комбинаторные свойства наборов энергоэффективных нормативных ниток, элементы которых представляют собой допустимые, с точки зрения практической организации грузовых железнодорожных перевозок, стратегии движения поездов по ориентированным перегонам железнодорожной сети.

Определение 1.2.6. Бинарное отношение конфликтности определяет неориентированный граф конфликтов

$$G = (\mathcal{N}, U), \tag{1.2.1}$$

где $\{n_i, n_j\} \in U$, если нормативные нитки n_i и n_j конфликтны.

Неориентированный граф конфликтов служит теоретико–графовой моделью для решения задачи планирования железнодорожных перевозок на этапе формирования множества бесконфликтных наборов энергоэффективных нормативных ниток графика движения поездов по ориентированным перегонам железнодорожной сети.

Определение 1.2.7. Любое подмножество \mathcal{N}' множества нормативных ниток \mathcal{N} такое, что порожденный подграф графа конфликтов, множеством вершин которого служит $\mathcal{N}' \subset \mathcal{N}$, пуст, т.е. не имеет ребер:

$$\langle \mathcal{N}' \rangle_G = (\mathcal{N}', \emptyset),$$

будем называть совместной системой нормативных ниток или бесконфликтным набором нормативных ниток.

Свойства несовместных систем общего вида и методы исследования несовместных систем, в том числе графовые методы и методы комбинаторной оптимизации в приложении к исследованию несовместных систем, подробно изучены в [4], [61]. Любой бесконфликтный набор нормативных ниток может служить допустимым расписанием для практической организации железнодорожных перевозок.

1.2.2. Теоретико–графовая модель для решения задачи о назначении и перемещении локомотивов

Важный этап решения задачи организации железнодорожных перевозок составляет назначение и перемещение локомотивов для исполнения фиксированного плана перевозок, представленного бесконфликтным набором нормативных ниток графика движения поездов. При этом приоритетным является исполнение плана посредством наименьшего числа локомотивов, и, ввиду необходимости технического обслуживания и существующих ограничений на использование локомотивов, особого внимания требует задача минимизации числа перемещений локомотивов без нагрузки составов. Рассмотрим теоретико–графовую конструкцию, характеризующую основные структурные и комбинаторные свойства задачи о назначении и перемещении локомотивов.

Для ориентированного графа сети

$$\vec{\Gamma} = (S, E), \quad (1.2.2)$$

где множество вершин $S = \{s_i : i = 1, 2, \dots, n\}$ соответствует множеству станций, и множество дуг $E \subseteq \{(s_i, s_j) : i = 1, 2, \dots, n, j = 1, 2, \dots, n\}$ соответствует

множеству ориентированных перегонов, связывающих соседние станции, определяют размеры движения на планируемый период времени.

Размеры движения могут быть заданы в виде матрицы корреспонденций

$$\mathcal{R} = \left\| r(s_i, s_j) \right\|, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n,$$

где $r(s_i, s_j)$ – количество поездов, необходимое к отправке из станции s_i в станцию s_j в планируемый период времени. Заметим, что матрица \mathcal{R} является сильно разреженной и, как правило, $r(s_i, s_j) \neq 0$ для станций s_i, s_j , являющихся сортировочными станциями или стыковыми станциями ориентированного графа сети $\vec{\Gamma}$ с внешними сетями.

Для организации перевозок из определенных размеров движения \mathcal{R} используют так называемый план поездоформирования

$$\mathcal{P} = \bigcup_{(s_i, s_j) \in E} p(s_i, s_j), \quad (1.2.3)$$

где последовательность вершин и дуг ориентированного графа $\vec{\Gamma} = (S, E)$ вида

$$\begin{aligned} p(s_i, s_j) &= (s_i = s_{i_1}, (s_{i_1}, s_{i_2}), s_{i_2}, \dots, s_{i_{k-1}}, (s_{i_{k-1}}, s_{i_k}), s_{i_k} = s_j) : \\ & \quad s_{i_1}, s_{i_2}, \dots, s_{i_{k-1}}, s_{i_k} \in S, \\ & \quad (s_{i_1}, s_{i_2}), \dots, (s_{i_{k-1}}, s_{i_k}) \in E, \end{aligned}$$

определяет простой ориентированный путь $p(s_i, s_j)$ в ориентированном графе сети, допустимый для выполнения перевозки из станции s_i в станцию s_j .

Множество нормативных ниток графика движения поездов, каждая из которых соответствует некоторому пути из плана поездоформирования \mathcal{P} , будем обозначать $\mathcal{N}(\mathcal{P})$.

Пусть $M_{i,j}$ – количество нормативных ниток, соответствующих множеству допустимых (v_i, v_j) -путей в ориентированном мультиграфе \vec{G} , и обозначим через

$$\mathcal{M}_{i,j} = \{n_{i,j}^m : m = 1, 2, \dots, M_{i,j}\} \subseteq \mathcal{N}(\mathcal{P})$$

множество всех нормативных ниток, соответствующих множеству

$$\mathcal{P}_{ij} = \{p_{i,j}^m : m = 1, 2, \dots, M_{i,j}\}$$

допустимых (v_i, v_j) -путей ориентированного мультиграфа энергоэффективных стратегий движения поездов.

Задача 1.2.1. Для заданных размеров движения \mathcal{R} и набора нормативных ниток графика движения поездов $\mathcal{N}(\mathcal{P})$ найти бесконфликтный набор нормативных ниток $\mathcal{N}(\mathcal{P}, \mathcal{R})$ такой, что

$$\mathcal{N}(\mathcal{P}, \mathcal{R}) \subseteq \mathcal{N}(\mathcal{P}) ,$$

$$|\mathcal{M}_{i,j} \cap \mathcal{N}(\mathcal{P}, \mathcal{R})| \geq r(s_i, s_j)$$

для всех

$$i = 1, 2, \dots, n , j = 1, 2, \dots, n : r(s_i, s_j) > 0 .$$

Подмножество $\mathcal{N}(\mathcal{P}, \mathcal{R}) \subseteq \mathcal{N}(\mathcal{P})$ определяет вариантный график движения поездов, и содержит бесконфликтные нормативные нитки, в количестве, достаточном для исполнения заданного плана перевозок, представленного матрицей корреспонденций размеров движения, посредством движения поездов по ориентированным перегонам железнодорожной сети, допустимым для организации перевозок и определенным в заданном плане поездоформирования на рассматриваемый период планирования. Физический смысл задачи 1.2.1 состоит в том, что для каждой планируемой перевозки построена нормативная нитка вариантного графика движения такая, что начальная и конечная вершины нормативной нитки соответствуют начальной и конечной станциям перевозки.

Вариантный график движения поездов лежит в основе теоретико-графовой модели для решения задачи организации железнодорожных перевозок посредством назначения и перемещения минимального числа локомотивов имеющегося парка по допустимым путям из определенного плана поездоформирования для исполнения максимального подмножества нормативных ниток из определенного плана перевозок, представленного матрицей корреспонденций размеров движения.

Рассмотрим задачу организации перевозок согласно вариантного графика $\mathcal{N}(\mathcal{P}, \mathcal{R})$ с помощью назначения и перемещения имеющегося парка локомотивов. Преобразуем вариантный график $\mathcal{N}(\mathcal{P}, \mathcal{R})$ движения поездов в план перевозок.

Для допустимого простого ориентированного (v_i, v_j) -пути p в ориентированном мультиграфе энергоэффективных стратегий движения \overrightarrow{G} , соответствующего некоторой нормативной нитке из бесконфликтного набора нормативных ниток графика движения поездов $p \in \mathcal{N}(\mathcal{P})$, определим четверку вида

$$z(p) = (s_n(p) , t_n(p) , s_k(p) , t_k(p)) ,$$

и будем называть её заданием на перевозку. Параметры s_n , t_n определяются как станция s_i начала движения и время начала движения по соответствующей энергоэффективной стратегии движения по ориентированному перегону железнодорожной сети, входящей в состав нормативной нитки $p \in \mathcal{N}(\mathcal{P})$ первой по порядку, и s_k , t_k – станция s_j окончания движения и время $(t_n + t_{i,j})$ окончания движения по соответствующей энергоэффективной стратегии движения по ориентированному перегону железнодорожной сети, входящей в состав нормативной нитки $p \in \mathcal{N}(\mathcal{P})$ последней по порядку.

Планом перевозок будем называть множество заданий на перевозку, полученное с учетом всех нормативных ниток из вариантного графика движения поездов $\mathcal{N}(\mathcal{P}, \mathcal{R})$:

$$Z = \{z(p) : p \in \mathcal{N}(\mathcal{P}, \mathcal{R})\}.$$

Введем в рассмотрение матрицу

$$\mathcal{R}^* = \left\| r^*(s_i, s_j) \right\|, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n,$$

такую, что для всех i, j , таких, что

$$r(s_i, s_j) = 0: \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, n,$$

выполняется

$$r^*(s_j, s_i) = 0.$$

Тогда, описанным для вариантного графика движения поездов $\mathcal{N}(\mathcal{P}, \mathcal{R})$ способом, определим вариантный график перемещения поездов $\mathcal{N}(\mathcal{P}, \mathcal{R}^*)$, бесконфликтный с вариантным графиком движения $\mathcal{N}(\mathcal{P}, \mathcal{R})$, т.е.

$$\langle \mathcal{N}(\mathcal{P}, \mathcal{R}) \cup \mathcal{N}(\mathcal{P}, \mathcal{R}^*) \rangle_G = (\mathcal{N}(\mathcal{P}, \mathcal{R}) \cup \mathcal{N}(\mathcal{P}, \mathcal{R}^*), \emptyset).$$

Преобразуем множество $\mathcal{N}(\mathcal{P}, \mathcal{R}^*)$ в план перемещений локомотивов, также, описанным способом:

$$D = \left\{ d(p) = (s_n(p), t_n(p), s_k(p), t_k(p)) : p \in \mathcal{N}(\mathcal{P}, \mathcal{R}^*) \right\}.$$

Аналогично множеству заданий на перевозку, полученному с учетом всех нормативных ниток из вариантного графика движения поездов $\mathcal{N}(\mathcal{P}, \mathcal{R})$, множество перемещений получено с учетом все нормативных ниток из вариантного графика перемещения поездов $\mathcal{N}(\mathcal{P}, \mathcal{R}^*)$.

Упорядочим множество $(Z \cup D)$ в лексикографическом порядке относительно $t_{\text{н}}(p)$, $t_{\text{к}}(p)$ и переобозначим:

$$Z \cup D = \left\{ v_i = (s_{\text{н}}(v_i), t_{\text{н}}(v_i), s_{\text{к}}(v_i), t_{\text{к}}(v_i)) \right\}, \quad (1.2.4)$$

где $i = 1, 2, \dots, |Z \cup D|$, и $|Z \cup D|$ — число всевозможных перемещений локомотивов по ориентированным перегонам железнодорожной сети, включая $v \in Z$ — задания на перевозку, соответствующие нормативным ниткам из определенных размеров движения, и $v \in D$ — допустимые перемещения локомотивов без нагрузки составов.

Множество $(Z \cup D)$ всевозможных перемещений локомотивов по ориентированным перегонам железнодорожной сети порождает ориентированный граф $\vec{\mathbb{G}}$ совместимости заданий на перевозку такой, что

$$\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E}), \quad (1.2.5)$$

$$(v_i, v_j) \in \mathbb{E} \iff \begin{cases} s_{\text{к}}(v_i) = s_{\text{н}}(v_j), \\ t_{\text{к}}(v_i) \leq t_{\text{н}}(v_j) + \Delta, \end{cases}$$

где действительное неотрицательное число $\Delta \geq 0$ характеризует нормативное время, необходимое для подготовки локомотива после исполнения некоторой нитки к исполнению следующей.

Ориентированный граф совместимости заданий на перевозку служит теоретико-графовой моделью для решения задачи организации железнодорожных перевозок, суть которой состоит в назначении и перемещении локомотивов имеющегося парка сети, заданных начальными условиями доступности для назначения, для исполнения максимального подмножества нормативных ниток из определенного плана перевозок, заданного бесконфликтным набором нормативных ниток, посредством минимального числа используемых локомотивов, при минимальном числе перемещений локомотивов без нагрузки составов. Свойства ориентированного графа совместимости заданий на перевозку положены в основу эффективных вычислительных алгоритмов решения задачи о назначении и перемещении локомотивов.

Глава 2. Задача планирования железнодорожных перевозок на этапе формирования бесконфликтного набора нормативных ниток

2.1. Постановка

С практической точки зрения задача планирования состоит в формировании допустимого расписания для исполнения заданного плана перевозок. В рамках теоретико–графовой модели неориентированный граф конфликтов является основным объектом исследования для решения задачи планирования. Как было показано в Подразделе 1.2.1 любое независимое подмножество вершин неориентированного графа конфликтов может служить допустимым расписанием, то есть является бесконфликтным набором нормативных ниток графика движения поездов.

В Подразделе 2.2.1 рассматривается теоретико–множественный подход к решению задачи планирования, основанный на свойствах неориентированного графа конфликтов и разработан алгоритм формирования множества бесконфликтных наборов нормативных ниток. На искомое решение задачи, кроме условия бесконфликтности ниток, устанавливаются ограничения практического характера, связанные с объёмом необходимых перевозок, что особенно важно в условиях оперативного планирования.

В Подразделе 2.2.2 рассматривается другой подход, в рамках которого исходная задача сводится к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом конфликтов.

Для множества нормативных ниток графика движения поездов, построенного с учётом плана поездоформирования (1.2.3), соответствующий граф конфликтов (1.2.1) имеет вид

$$G = (\mathcal{N}(\mathcal{P}), U):$$

$$\mathcal{N}(\mathcal{P}) = \{n_1, n_2, \dots, n_n\},$$

и отвечает простому графу, заданному на упорядоченном множестве вершин:

$$G = (V(G), \mathcal{E}(G)): \tag{2.1.1}$$

$$V(G) = \{v_1, v_2, \dots, v_n\}, \mathcal{E}(G) = \{e_1, e_2, \dots, e_p\}.$$

Для графа вида (2.1.1) введём обозначения:

– $G\langle U \rangle : U \subset V(G)$, $G\langle U \rangle \subseteq G$ – индуцированный подграф графа G на подмножестве U вершин графа,

– $\mathcal{N}(v : v \in V(G)) \subset V(G)$ – множество вершин, смежных вершине v в графе G , или окрестность вершины v в графе,

– $\binom{U}{2} : U \subseteq V(G)$ – семейство неупорядоченных 2-подмножеств подмножества U вершин графа,

– $[n] = \{1, 2, \dots, n\}$ – конечная последовательность натуральных чисел,

– $|\cdot|$ – мощность некоторого множества,

– $\text{supp}(x) = \{i \in [n] : x_i = 1\}$, $x = (x_1, x_2, \dots, x_n) \in \{0, 1\}^n$ – носитель некоторого двоичного набора, содержит номера единичных компонент.

Рассмотрим монотонную булеву функцию (МБФ)

$$f_G : \{0, 1\}^n \longrightarrow \{0, 1\} , \quad (2.1.2)$$

множество единиц которой задано следующим образом:

$$T(f_G) = f_G^{-1}(1) ,$$

$$f_G(x) = 1 \iff \left| \mathcal{E}(G) \cap \binom{\{v_i \in V(G) : i \in \text{supp}(x)\}}{2} \right| \geq 1 , \quad (2.1.3)$$

иными словами, мы полагаем $f_G(x) = 1$ тогда и только тогда, когда индуцированный подграф

$$G\langle \{v_i \in V(G) : i \in \text{supp}(x)\} \rangle$$

имеет по меньшей мере одно ребро.

Другая монотонная булева функция

$$g_G : \{0, 1\}^n \rightarrow \{0, 1\} , \quad (2.1.4)$$

естественным образом сопоставляемая графу G , определяется множеством своих нулей так:

$$F(g_G) = g_G^{-1}(0) ,$$

$$g_G(x) = 0 \iff G\langle \{v_i \in V(G) : i \in \text{supp}(x)\} \rangle \text{ — полный} , \quad (2.1.5)$$

причём к полным графам также относят пустой граф и изолированные вершины.

Пусть $X \subseteq \{0, 1\}^n$ — множество двоичных наборов длины n . На множестве X задано отношение частичного порядка:

$$\begin{aligned} & \forall x, x' \in \{0, 1\}^n : \\ & x = (x_1, \dots, x_n) , x' = (x'_1, \dots, x'_n) , \\ & x < x' \iff \exists j \in [n] : \forall i \neq j , i \in [n] \begin{cases} x_j < x'_j , \\ x_i \leq x'_i . \end{cases} \end{aligned}$$

Обозначим множество всех максимальных элементов частично упорядоченного множества $(X, <)$ и множество элементов множества X с максимальным числом единиц в наборе соответственно через

$$\max_{\subseteq} X , \max_{|\cdot|} X .$$

Двоичный набор $x \in \{0, 1\}^n$ называется *верхним нулем* некоторой монотонной булевой функции

$$\varphi : \{0, 1\}^n \rightarrow \{0, 1\} ,$$

если

$$\begin{aligned} & \forall x' \in \{0, 1\}^n : x < x' \\ & \begin{cases} \varphi(x) = 0 , \\ \varphi(x') = 1 , \end{cases} \end{aligned}$$

двойственным образом, двоичный набор $x \in \{0, 1\}^n$ называется *нижней единицей* монотонной булевой функции, если

$$\begin{aligned} & \forall x' \in \{0, 1\}^n : x > x' \\ & \begin{cases} \varphi(x) = 1 , \\ \varphi(x') = 0 . \end{cases} \end{aligned}$$

В этих терминах множество верхних нулей монотонной булевой функции соответствует множеству максимальных элементов, на каждом из которых функция принимает нулевое значение, то есть, для функций вида (2.1.2) и (2.1.4) множества верхних нулей будут иметь вид:

$$\max_{\subseteq} F(f_G) , \max_{\subseteq} F(g_G) ,$$

соответственно.

Верхний нуль $x \in \{0,1\}^n$ некоторой монотонной булевой функции φ называется *максимальным*, если число единичных компонент в нём максимальное среди всех элементов семейства верхних нулей:

$$|\text{supp}(x)| = \max \left\{ \left| \text{supp}(x') : x' \in \max_{\subseteq} F(\varphi) \right| \right\},$$

то есть носитель набора имеет максимальную мощность среди всех носителей верхних нулей.

Теоретико-графовой конструкцией, связывающей между собой МБФ (2.1.2) и (2.1.4), является дополнительный граф. Для графа вида (2.1.1) дополнительный граф будет иметь вид:

$$\bar{G} = \left(V(G), \mathcal{E}(\bar{G}) = (V_2^{(G)}) \setminus \mathcal{E}(G) \right),$$

таким образом, непосредственно из определений (2.1.3) и (2.1.5) следуют тождества

$$f_G = g_{\bar{G}}, f_{\bar{G}} = g_G.$$

Из определения графа конфликтов следует, что нули функции f_G определяют бесконфликтные наборы нормативных ниток графика движения поездов. На практике более существенными являются бесконфликтные наборы нормативных ниток, содержащие наибольшее возможное число элементов, что в постановке булевых функций соответствует максимальным верхним нулям функции f_G . Таким образом, задача планирования железнодорожных перевозок может быть решена в постановке задачи формирования множества максимальных верхних нулей монотонной булевой функции.

Задача 2.1.1. Для функции f_G вида (2.1.2) найти множество $\max_{\subseteq} F(f_G)$ её верхних нулей.

Задача 2.1.2. Для функции f_G найти её максимальные верхние нули, то есть наборы $x \in \max_{\subseteq} F(f_G)$ с максимальным количеством $|\text{supp}(x)|$ единичных компонент.

В терминах теории графов решения Задачи 2.1.1 и Задачи 2.1.2 соответствуют максимальному по размеру и максимальному по включению независимым подмножествам заданного графа. Доказательство NP -полноты задачи о независимом подмножестве основано на сведении к классической NP -полной задаче выполнимости и подробно представлено в [20], [34].

В Подразделе 2.2.2 разрабатываются приближённые алгоритмы формирования верхнего нуля и максимального верхнего нуля монотонной булевой функции, порождённой неориентированным графом, каждый из которых отвечает некоторому решению задачи планирования железнодорожных перевозок.

2.2. Решение

Рассмотрим подробнее способы решения задачи планирования на этапе формирования множества неконфликтных наборов нормативных ниток графика движения поездов.

2.2.1. Алгоритм «Бегущая волна»

Теоретико–множественный алгоритм решения задачи формирования множества неконфликтных наборов нормативных ниток графика движения поездов основан на свойствах неориентированного графа конфликтов и структурных свойствах множества нормативных ниток графика движения поездов.

Пусть задан рассматриваемый период планирования K дней.

Обозначим через \mathcal{N}_i множество нормативных ниток графика движения поездов, которые имеют начало в i -е сутки, тогда множество нормативных ниток, необходимых к исполнению в планируемый период времени,

$$\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 \cup \dots \cup \mathcal{N}_K .$$

По определению неориентированный граф конфликтов (1.2.1) на рассматриваемый период планирования имеет вид

$$G = (\mathcal{N}, U) ,$$

где вершины n_i и n_j неориентированного графа конфликтов, соответствующие ниткам из множества нормативных ниток графика движения поездов, соединены ребром $\{n_i, n_j\} \in U$ в том и только в том случае, когда нормативные нитки n_i и n_j конфликтны.

Пусть каждые сутки в рассматриваемом периоде планирования разбиты на 8 трехчасовых интервалов I_1, I_2, \dots, I_8 , и определён план перевозок, заданный посредством множества нормативных ниток графика движения поездов, исполнение которых характеризуется по дням и направлениям:

$$(\text{vol}_1^{A \rightarrow B}, \text{vol}_1^{B \rightarrow A}, \dots, \text{vol}_K^{A \rightarrow B}, \text{vol}_K^{B \rightarrow A}),$$

где $\text{vol}_i^{A \rightarrow B}$ и $\text{vol}_i^{B \rightarrow A}$ — количество нормативных ниток, необходимых к исполнению в i -е сутки, начало движения по которым соответствует пункту A , и окончание движения по которым соответствует пункту B , и соответственно количество нормативных ниток, необходимых к исполнению в i -е сутки, начало движения по которым соответствует пункту B , и окончание движения по которым соответствует пункту A .

Если движение по некоторой нормативной нитке $n \in \mathcal{N}$ начинается в интервале $I \in \{I_1, \dots, I_{8K}\}$, то будем записывать этот факт в виде

$$\text{init}(n) = I,$$

тогда для набора $\mathcal{N}' \subseteq \mathcal{N}$ определим

$$\text{init}(\mathcal{N}') = \bigcup_{n \in \mathcal{N}'} \text{init}(n).$$

Определим алгоритм «Бегущая волна», и приведем формальное описание схемы реализации алгоритма «Бегущая волна» для формирования множества бесконфликтных наборов нормативных ниток графика движения поездов.

Пусть для заданной начальной последовательности интервалов

$$(I_1, \dots, I_q)$$

задан некоторый бесконфликтный набор нормативных ниток $\mathcal{N}' \subseteq \mathcal{N}$.

Алгоритм \mathcal{A} для набора \mathcal{N}' такого, что

$$\text{init}(\mathcal{N}') \subseteq I_1 \cup \dots \cup I_q,$$

определяет некоторый поднабор нормативных ниток из множества нормативных ниток графика движения поездов $\Delta\mathcal{N} \subseteq \mathcal{N}$ такой, что все нормативные нитки, входящие в поднабор, имеют начало в определенные сутки из периода планирования:

$$\text{init}(\Delta\mathcal{N}) = I_{q+1},$$

и соответствующий неориентированный граф конфликтов имеет вид:

$$G\langle \mathcal{N}' \cup \Delta\mathcal{N} \rangle = (\mathcal{N}' \cup \Delta\mathcal{N}, \emptyset),$$

то есть не имеет рёбер, и все нормативные нитки, соответствующие вершинам неориентированного графа конфликтов, составляют бесконфликтный набор нормативных ниток графика движения поездов.

Алгоритм \mathcal{B} для полученного набора нормативных ниток $\Delta\mathcal{N}$ фиксирует некоторым образом поднабор $\Delta\mathcal{N}' \subseteq \Delta\mathcal{N}$, и набор нормативных ниток $\Delta\mathcal{N}'$ подлежит реализации, то есть для каждой нормативной нитки из набора нормативных ниток $\Delta\mathcal{N}'$ должен быть назначен некоторый доступный локомотив из локомотивного парка сети. Здесь каждые 8 трехчасовых интервалов составляют очередные сутки периода планирования.

Полученный набор нормативных ниток графика движения поездов $\Delta\mathcal{N}'$, такой что

$$\text{init}(\Delta\mathcal{N}') = I_{q+1},$$

будем обозначать через

$$\Delta\mathcal{N}'(I_{q+1}).$$

Для некоторого поднабора нормативных ниток графика движения поездов обозначим через

$$\begin{aligned} (\mathcal{N}^*)_{A \rightarrow B}, (\mathcal{N}^*)_{B \rightarrow A} : \\ \mathcal{N}^* \subseteq \mathcal{N}, \end{aligned}$$

множество всех нормативных ниток из набора \mathcal{N}^* , начало движения по которым соответствует пункту A , и окончание движения по которым соответствует пункту B , и соответственно множество всех нормативных ниток из набора нормативных ниток графика движения поездов \mathcal{N}^* , начало которых соответствует пункту B , и окончание движения по которым соответствует пункту A .

Реализация алгоритма «Бегущая волна» организована таким образом, что для интервалов

$$I_{q+1}, I_{q+2}, \dots, I_{q+8}$$

очередных i -х суток, где $q = 8(i - 1)$, $i = 1 \dots, K$, необходимо выполняются условия

$$\left| (\Delta\mathcal{N}'(I_{q+1}) \cup \dots \cup \Delta\mathcal{N}'(I_{q+8}))_{A \rightarrow B} \right| = \text{vol}_i^{A \rightarrow B}, \quad (2.2.1)$$

$$|(\Delta\mathcal{N}'(I_{q+1}) \cup \dots \cup \Delta\mathcal{N}'(I_{q+8}))_{B \rightarrow A}| = \text{vol}_i^{B \rightarrow A}, \quad (2.2.2)$$

$$\begin{aligned} G\langle \Delta\mathcal{N}'(I_1) \cup \dots \cup \Delta\mathcal{N}'(I_{q+8}) \rangle = \\ = (\Delta\mathcal{N}'(I_1) \cup \dots \cup \Delta\mathcal{N}'(I_{q+8}), \emptyset). \end{aligned} \quad (2.2.3)$$

Выполнение условий (2.2.1)–(2.2.3) обеспечивает формирование набора нормативных ниток графика движения поездов, мощность которого отвечает количеству нормативных ниток, необходимых к исполнению в текущие сутки из периода планирования, согласно заданному плану перевозок, и бесконфликтного с набором нормативных ниток, реализация которого актуальна в текущие сутки из периода планирования.

Схема алгоритма «Бегущая волна» для заданного набора интервалов

$$(I_1, I_2, \dots, I_{8K}),$$

где K – число суток в рассматриваемом периоде планирования, для формирования очередного набора нормативных ниток графика движения поездов, необходимых к исполнению, бесконфликтного с актуальным реализуемым набором нормативных ниток графика движения поездов:

1) Начальный набор интервалов пуст, $q = 0$, $\mathcal{N}' = \emptyset$

Пока $q < 8K$ **выполнять**

2) Применяя алгоритм \mathcal{A} , получаем $\Delta\mathcal{N}$ такой, что $\text{init}(\Delta\mathcal{N}) = I_{q+1}$

3) Применяя алгоритм \mathcal{B} получаем $\Delta\mathcal{N}' \subseteq \Delta\mathcal{N}$

4) $\mathcal{N}' \leftarrow \mathcal{N}' \cup \Delta\mathcal{N}'$, $q \leftarrow q + 1$

Конец условия

Выбранное для разработанного теоретико-множественного алгоритма название «Бегущая волна» объясняется, главным образом, тем, что реализация алгоритма создает наиболее комфортные условия, приближенные к оперативной организации железнодорожных перевозок, в связи с тем, что перед выбором набора нормативных ниток $\Delta\mathcal{N}'$ из множества нормативных ниток $\Delta\mathcal{N}$ в распоряжении алгоритма имеется набор нормативных ниток $\Delta\mathcal{N}$, существенно больший, чем набор $\Delta\mathcal{N}'$, нормативные нитки которого необходимо требуют назначения на доступные локомотивы парка сети. Такой подход обеспечивает при выборе набора нормативных ниток $\Delta\mathcal{N}'$ практическое использование полноценного набора нормативных ниток \mathcal{N} графика движения поездов. Но после

того, как выбор набора $\Delta\mathcal{N}'$ сделан, набор нормативных ниток графика движения поездов подлежит реализации и, поскольку вновь сформированный набор нормативных ниток графика движения поездов существенно меньше, чем набор нормативных ниток графика движения поездов \mathcal{N} , то фиксация набора $\Delta\mathcal{N}'$ несущественно отражается на доступности нормативных ниток графика движения поездов для очередного текущего интервала периода планирования. С точки зрения практической реализации алгоритма и внедрения в технологическую практику, несущественное сужение множества нормативных ниток, доступных для выбора в качестве элементов формируемого бесконфликтного набора нормативных ниток, необходимых к исполнению, представляет особый интерес.

2.2.2. Алгоритм расшифровки монотонной булевой функции

В Разделе 2.1 было показано, что задача планирования железнодорожных перевозок может быть решена в постановке задачи формирования множества максимальных верхних нулей монотонной булевой функции, порождённой неориентированным графом конфликтов. При этом каждый элемент множества максимальных верхних нулей отвечает некоторому бесконфликтному набору нормативных ниток графика движения поездов, то есть может служить допустимым расписанием для практической организации перевозок. Для графов вида (2.1.1) рассмотрим Задачи 2.1.1 и 2.1.2 из Раздела 2.1 более подробно.

Утверждение 2.2.1. Пусть $v_i \in V(G)$ — вершина графа $G = (V(G), \mathcal{E}(G))$ такая, что для её окрестности $\mathcal{N}(v_i)$ индуцированный подграф $G\langle\mathcal{N}(v_i)\rangle$ графа G является полным. Тогда существует максимальный верхний нуль $x' \in \max_{|\cdot|} \max_{\subseteq} F(f_G)$ функции f_G такой, что $x'_i = 1$.

Доказательство. Рассмотрим произвольный максимальный верхний нуль функции f_G

$$x \in \max_{|\cdot|} \max_{\subseteq} F(f_G),$$

и свяжем с ним множество индексов

$$I = \{s \in [n] : v_s \in \mathcal{N}(v_i)\}.$$

Поскольку, рассматриваемый набор по предположению является максимальным верхним нулём функции, то

$$\exists j \in I \cup \{i\}: x_j = 1 ,$$

так как в противном случае можно указать набор, больший чем рассматриваемый набор:

$$x' \in \{0, 1\}^n: \forall s \in [n] \setminus \{i\} \begin{cases} x'_i = 1 , \\ x'_s = x_s , \end{cases}$$

тогда, по определению функции f_G имеем:

$$f_G(x) = 0 , x_s = 0 \forall s \in I \implies f_G(x') = 0 ,$$

и, таким образом, получено строгое включение

$$\text{supp}(x') \not\supseteq \text{supp}(x) : f_G(x') = f_G(x) = 0 ,$$

что противоречит условию максимальности рассматриваемого набора.

Рассмотрим два возможных варианта. Если $x_i = 1$, то доказательство завершается.

Пусть $x_i = 0$ и $x_s = 1$ для некоторого индекса $s \in I$, тогда для набора x можно указать набор

$$x' \in \{0, 1\}^n: \begin{cases} x'_j = x_j \forall j \in [n] \setminus \{i, s\} , \\ x'_i = 1 , \\ x'_s = 0 , \end{cases}$$

причём, в силу условия полноты индуцированного подграфа $G\langle \mathcal{N}(v_i) \rangle$, имеем

$$x' \in \max_{|\cdot|} \max_{\subseteq} F(f_G) : \text{supp}(x') = \text{supp}(x) ,$$

и, таким образом, для полученного максимального верхнего нуля выполняется

$$x'_i = 1 ,$$

что и требовалось доказать. □

Определение 2.2.1. Для целого числа $k \in [n - 1]$, вершину $v \in V(G)$ графа $G = (V(G), \mathcal{E}(G))$ назовем k -вершиной, если $|\mathcal{N}(v)| = k$ и индуцированный подграф $G\langle \mathcal{N}(v) \rangle$ графа G является полным.

Определение 2.2.2. Для целых чисел $k \in [n - 1]$ и $m \in [n - 1]$, вершину $v \in V(G)$ графа $G = (V(G), \mathcal{E}(G))$ назовем (k, m) -вершиной, если $k = |\mathcal{N}(v)|$ и

$$m = C_k^2 - \left| \mathcal{E}(G) \cap \binom{\mathcal{N}(v)}{2} \right|.$$

В этих терминах, любая вершина $v \in V(G)$ графа $G = (V(G), \mathcal{E}(G))$ такая, что v — (k, m) -вершина, является k -вершиной при $m = 0$.

На основе Утверждения 2.2.1 может быть построен эффективный рекурсивный алгоритм для решения Задачи 2.1.2, который завершает свою работу либо построением максимального верхнего нуля функции f_G , либо сведением Задачи 2.1.2 для функции f_G к новой Задаче 2.1.2 для функции $f_{G'}$, где $G' \subset G$, то есть снижением размерности решаемой задачи.

Для вершины $v \in V_0$, $V_0 \subseteq V$ обозначим через $\mathcal{N}(v, V_0) \subset V_0$ окрестность вершины v в индуцированном подграфе $G\langle V_0 \rangle$.

Алгоритм 1: Алгоритм $A(G, V_0)$ генерации максимального верхнего нуля $x = (x_1, \dots, x_n) \in \{0, 1\}^n$ функции f_G .

Входные данные: G, V_0

Выходные данные: V_0, x

$x_i = 0$ для всех $v_i \in V_0$

для $v_i \in V_0$ **выполнять**

если v_i — $|\mathcal{N}(v_i, V_0)|$ -вершина в подграфе $G\langle V_0 \rangle$ **то**

$x_i \leftarrow 1$

$x_j \leftarrow 0$ для всех $v_j \in \mathcal{N}(v_i, V_0)$

$V_0 \leftarrow V_0 \setminus (\{v_i\} \cup \mathcal{N}(v_i, V_0))$

$A(G, V_0)$

конец условия

конец цикла

Если по окончании работы Алгоритма 1 получим $V_0 = \emptyset$, то полученный набор $x \in \{0, 1\}^n$, согласно Утверждению 2.2.1, является максимальным верхним нулем функции f_G .

Если же по окончании работы Алгоритма 1 имеем $V_0 \neq \emptyset$, то для всех вершин графа $G\langle V \setminus V_0 \rangle$ определены значения некоторых компонент x_i такие, что существует максимальный верхний нуль x' функции f_G с точно такими

же значениями для этих компонент, то есть $x'_i = x_i$. При этом достигнуто снижение размерности задачи с $|V|$ до $|V_0|$.

С точки зрения практической организации перевозок часто оказывается достаточным формирование некоторого бесконфликтного набора нормативных ниток графика движения поездов, который содержит нитки в достаточном количестве для организации заданного плана перевозок. В этой связи актуальным представляется формирование некоторого набора из множества верхних нулей. Дальнейшее исследование задачи формирования множества максимальных верхних нулей может быть продолжено посредством реализации приближённых алгоритмов формирования множества верхних нулей.

Жадный поиск. Один из подходов для решения задачи формирования множества верхних нулей монотонной булевой функции, порождённой неориентированным графом конфликтов, основан на алгоритме жадного поиска. При этом входными данными для задачи является множество вершин графа, полученное в результате реализации Алгоритма 1 из Подраздела 2.2.2.

Лемма 2.2.1. Пусть заданы два графа $G_1 = (V, \mathcal{E}(G_1))$ и $G_2 = (V, \mathcal{E}(G_2))$ с одним и тем же множеством вершин V , и

$$\mathcal{E}(G_1) \subseteq \mathcal{E}(G_2) .$$

Тогда

$$\max_{|\cdot|} \max_{\subseteq} F(f_{G_2}) \subseteq \max_{\subseteq} F(f_{G_2}) \subseteq F(f_{G_2}) \subseteq F(f_{G_1}) .$$

Доказательство. Очевидно, что

$$\max_{|\cdot|} \max_{\subseteq} F(f_{G_2}) \subseteq \max_{\subseteq} F(f_{G_2}) \subseteq F(f_{G_2}) ,$$

где $\max_{|\cdot|} \max_{\subseteq} F(f_{G_2})$ – множество максимальных верхних нулей МБФ f_{G_2} .

Рассмотрим произвольный набор $x \in \{0, 1\}^n$ такой, что $x \in F(f_{G_2})$. По определению множества нулей МБФ f_{G_2} имеем:

$$|\mathcal{E}(G_2) \cap (\{v_i : i \in \text{supp}(x)\})| = 0 .$$

По условию Леммы, $\mathcal{E}(G_1) \subseteq \mathcal{E}(G_2)$ и $V(G_1) = V(G_2)$, следовательно,

$$|\mathcal{E}(G_1) \cap (\{v_i : i \in \text{supp}(x)\})| = 0 , \forall x \in \{0, 1\}^n ,$$

значит, по определению множества нулей МБФ f_{G_1} , получаем

$$x \in F(f_{G_1}) . \quad (2.2.4)$$

Тогда для любых наборов $x \in \{0, 1\}^n$ таких, что $x \in F(f_{G_2})$, выполнено включение (2.2.4), то есть

$$F(f_{G_2}) \subseteq F(f_{G_1}) ,$$

что и требовалось доказать. □

Следует отметить, что в условиях Леммы 2.2.1 не выполняется включение для множества верхних нулей:

$$\max_{\subseteq} F(f_{G_2}) \not\subseteq \max_{\subseteq} F(f_{G_1}) . \quad (2.2.5)$$

Для доказательства (2.2.5) можно рассмотреть, например, два графа, заданных одним и тем же множеством вершин, таких что:

$$\begin{aligned} G_1 &= (V(G_1), \mathcal{E}(G_1)) = (V, \emptyset) , \\ G_2 &= (V(G_2), \mathcal{E}(G_2)) = (V, \binom{V}{2}) , \end{aligned}$$

очевидно удовлетворяющих условию Леммы 2.2.1

$$\begin{aligned} \mathcal{E}(G_1) &\subseteq \mathcal{E}(G_2) , \\ V(G_1) &= V(G_2) . \end{aligned}$$

Граф G_1 не имеет рёбер, поэтому множество верхних нулей функции f_{G_1} состоит из единственного набора

$$x = (1, 1, \dots, 1) .$$

Граф G_2 – полный, значит множество верхних нулей функции f_{G_2} имеет вид:

$$\begin{aligned} x^1 &= (1, 0, \dots, 0) , \\ x^2 &= (0, 1, \dots, 0) , \\ &\vdots \\ x^n &= (0, 0, \dots, 1) . \end{aligned}$$

Любой набор $x \in \max_{\subseteq} F(f_{G_2})$ является нулем функции f_{G_1} , однако, ни один такой набор не является её верхним нулём, то есть

$$\max_{\subseteq} F(f_{G_2}) \subseteq F(f_{G_1}), \quad \max_{\subseteq} F(f_{G_2}) \not\subseteq \max_{\subseteq} F(f_{G_1}),$$

что является заключением Леммы 2.2.1 и подтверждает (2.2.5).

Введём обозначение для числа единиц в максимальном верхнем нуле монотонной булевой функции:

$$\max_0 f_G = |\text{supp}(x)| : x \in \max_{|\cdot|} \max_{\subseteq} F(f_G).$$

Следствие 2.2.1. Пусть заданы графы $G_1 = (V, \mathcal{E}_1)$ и $G_2 = (V, \mathcal{E}_2)$ такие, что $\mathcal{E}_1 \subseteq \mathcal{E}_2$. Тогда

$$\max_0 f_{G_1} \geq \max_0 f_{G_2}.$$

Доказательство. Рассмотрим произвольный элемент множества максимальных верхних нулей $x \in \max_{|\cdot|} \max_{\subseteq} F(f_{G_2})$. Согласно Лемме 2.2.1, для заданных графов имеем $x \in F(f_{G_1})$.

По определению максимального верхнего нуля монотонной булевой функции

$$\forall x \in F(f_{G_1}) \exists x' \in \max_{|\cdot|} \max_{\subseteq} F(f_{G_1}) : x' \geq x,$$

и тогда имеем

$$\max_0 f_{G_1} = |\text{supp}(x')| \geq |\text{supp}(x)| = \max_0 f_{G_2},$$

что и требовалось доказать. \square

Утверждение 2.2.2. Пусть задан граф $G = (V(G), \mathcal{E}(G))$, и пусть вершины v_i и v_j в графе G не смежны. Тогда

$$\max_0 f_G \geq \max_0 f_{G \cup \{\{v_i, v_j\}\}} \geq \max_0 f_G - 1. \quad (2.2.6)$$

Доказательство. Непосредственно из Следствия 3.2.1 верно первое неравенство

$$\max_0 f_G \geq \max_0 f_{G \cup \{\{v_i, v_j\}\}},$$

и для доказательства второго неравенства

$$\max_0 f_{G \cup \{\{v_i, v_j\}\}} \geq \max_0 f_G - 1$$

рассмотрим произвольный верхний нуль функции f_G вида

$$x = (x_1, \dots, x_n) .$$

Случай 1.

Предположим, что $x_i = 0$ и $x_j = 0$. Тогда рассматриваемый набор x также является нулем функции $f_{G \cup \{v_i, v_j\}}$ и, согласно Лемме 2.2.1, является максимальным верхним нулем, потому что в противном случае мы бы получили, опять же, согласно Лемме 2.2.1, что

$$\exists x' \in \max_{\subseteq} \max_{|\cdot|} F \left(f_{G \cup \{v_i, v_j\}} \right) : x' > x, x'_i \geq x_i ,$$

и, по определению максимального верхнего нуля монотонной булевой функции, выполняется неравенство

$$|\text{supp}(x')| > |\text{supp}(x)| ,$$

что противоречит условию максимальности рассматриваемого набора.

Следовательно, для случая 1 утверждение доказано:

$$\max_0 f_G = \max_0 f_{G \cup \{v_i, v_j\}} \geq \max_0 f_G - 1 .$$

Случай 2.

Для определенности, пусть $x_i = 1$, $x_j = 0$.

Тогда при добавлении ребра $\{v_i, v_j\}$, набор x снова является нулем функции $f_{G \cup \{v_i, v_j\}}$ и, как показано выше, также и максимальным верхним нулем функции.

Случай 3.

Пусть $x_i = 1$, $x_j = 1$.

Тогда при добавлении ребра $\{v_i, v_j\}$ получаем, что набор x не является нулем функции $f_{G \cup \{v_i, v_j\}}$. В этом случае можем указать набор

$$x' : \begin{cases} x'_s = x_s \quad \forall s \in [n] \setminus \{i\} , \\ x'_i = 0 . \end{cases}$$

Полученный набор x' будет нулем функции $f_{G \cup \{v_i, v_j\}}$, при этом, по построению, имеем:

$$|\text{supp}(x')| = |\text{supp}(x)| - 1 ,$$

и по определению максимального верхнего нуля функции, имеем:

$$\max_0 f_{G \cup \{\{v_i, v_j\}\}} \geq |\text{supp}(x')| = |\text{supp}(x)| - 1 = \max_0 f_G - 1 ,$$

что и требовалось доказать. \square

Следствие 2.2.2. Пусть заданы граф $G = (V(G), \mathcal{E}(G))$, и семейство попарно различных ребер, не являющихся ребрами графа:

$$\{e_1, e_2, \dots, e_t\} \subset \binom{V(G)}{2} \setminus \mathcal{E}(G) .$$

Тогда

$$\max_0 f_{G \cup \{e_1, e_2, \dots, e_t\}} \geq \max_0 f_G - t .$$

Доказательство. Применяя t раз Утверждение 2.2.2 к графу G , убеждаемся в справедливости следствия. \square

На основе Утверждения 2.2.2 можно модифицировать Алгоритм 1 из Подраздела 2.2.1 таким образом, что работа алгоритма будет продолжаться до полного исчерпания вершин в остающемся множестве V_0 , и при этом будет найден нуль x функции f_G , для которого одновременно будет вычислена величина

$$\max_0 f_G - |\text{supp}(x)|$$

для оценки отклонения количества единиц в полученном наборе x от количества единиц в максимальном верхнем нуле функции.

Алгоритм 2: Алгоритм $A_m(G, V_0)$.

Входные данные: $G, V_0, m \in [n]$

Выходные данные: V_0, Ind, x

$x_i = 0$ для всех $v_i \in V_0$

$\text{Ind} = 0$

для $v_i \in V_0$ выполнять

если $v_i - (|\mathcal{N}(v_i, V_0)|, m)$ -вершина в подграфе $G \langle V_0 \rangle = G'$ то

$x_i \leftarrow 1$

$x_j \leftarrow 0$ для всех $v_j \in \mathcal{N}(v_i, V_0)$

$V_0 \leftarrow V_0 \setminus (\{v_i\} \dot{\cup} \mathcal{N}(v_i, V_0))$

$\text{Ind} \leftarrow 1$

конец цикла

В Алгоритме 2 для данного значения m и для каждой вершины исходного множества V_0 осуществляется последовательная проверка, является ли рассматриваемая вершина $(|\mathcal{N}(v_i, V_0)|, m)$ -вершиной. Если таких вершин нет, то никаких действий не производится и множество V_0 на выходе алгоритма совпадает с входным множеством V , признак $\text{Ind} = 0$, двоичный набор x не определен. В случае, когда такая вершина v_i будет найдена, выходное множество V_0 получается из входного множества V_0 посредством «удаления» из него вершины v_i и ее окрестности, $\text{Ind} = 1$ и соответствующая компонента x_i набора x принимает значение 1.

Алгоритм 3: Алгоритм $B(G, V_0)$.

Входные данные: G, V_0

Выходные данные: $x \in \max_{\subseteq} F(f_G)$

пока $V_0 \neq \emptyset$ **выполнять**

$\text{Ind} = 1$

пока $(\text{Ind} = 1) \ \& \ V_0 \neq \emptyset$ **выполнять**

$m \leftarrow 0$

$A_m(G, V_0)$

$\text{Ind} \leftarrow \text{Ind}(A_m(G, V_0))$

конец цикла

пока $(\text{Ind} = 0) \ \& \ V_0 \neq \emptyset$ **выполнять**

$m \leftarrow m + 1$

$A_m(G, V_0)$

$\text{Ind} \leftarrow \text{Ind}(A_m(G, V_0))$

конец цикла

конец цикла

В ходе реализации Алгоритма 3 происходит непрерывное обращение к Алгоритму 2, в результате выполнения которого формируется набор x . Полученный набор из множества нулей функции f_G и является результатом работы Алгоритма 3.

Утверждение 2.2.3. Пусть v_i - (k, m) -вершина графа $G = (V(G), \mathcal{E}(G))$, $v_i \in V(G)$. Тогда существует набор $x' \in \max_{\subseteq} F(f_G)$ такой, что $x'_i = 1$, и

выполняется неравенство

$$|\text{supp}(x')| \geq \max_0 f_G - m .$$

Доказательство. Пусть, по определению (k, m) -вершины графа, для некоторой вершины v_i имеем

$$\{e_1, e_2, \dots, e_m\} = (\mathcal{N}_2^{(v_i)}) \setminus (\mathcal{E}(G) \cap (\mathcal{N}_2^{(v_i)})) ,$$

тогда вершина v_i является k -вершиной в графе

$$G_1 \langle \mathcal{N}(v_i) \rangle = (V(G) , \mathcal{E}(G) \cup \{e_1, e_2, \dots, e_m\}) ,$$

полученном добавлением рёбер в окрестности вершины v_i до полного индуцированного подграфа.

Согласно Утверждению 2.2.1, имеем для G_1 :

$$\exists x : x_i = 1 , x \in \max_{|\cdot|} \max_{\subseteq} F(f_{G_1}) ,$$

и, согласно Следствию 3.2.2 из Утверждения 2.2.2, имеем также для G_1 :

$$\max_0 f_{G_1} \geq \max_0 f_G - m .$$

Поскольку $x \in \max_{|\cdot|} \max_{\subseteq} F(f_{G_1})$, то

$$|\text{supp}(x)| \geq \max_0 f_G - m .$$

По определению верхнего нуля функции, существует набор $x' \in F(f_{G_1})$ такой, что $x' \geq x$ и, следовательно,

$$|\text{supp}(x')| \geq |\text{supp}(x)| \geq \max_0 f_G - m ,$$

что и требовалось доказать. □

В Алгоритме 1 поиск k -вершины, в очередном цикле работы, ведется до первой найденной k -вершины. Такой подход минимизирует число действий в текущем цикле работы алгоритма, но, возможно, дает не лучшее решение в случае обращения к Алгоритму 3 в части оценки отклонения числа единиц в полученном наборе от числа единиц в максимальном верхнем нуле рассматриваемой монотонной булевой функции. Возможно, лучшая оценка может быть определена посредством реализации алгоритма поиска с возвратом, входными данными для которого является не множество вершин графа, полученное в результате Алгоритма 1, а множество вершин исходного неориентированного графа конфликтов.

Поиск с возвратом. Алгоритм поиска с возвратом реализует на каждом шаге выбор вершины неориентированного графа, количественные характеристики которой являются наилучшими среди всех элементов текущего множества вершин. Такой подход предполагает большее количество действий для каждого текущего цикла, однако, можно ожидать более точное приближение к максимальному верхнему нулю.

Алгоритм 4:

Входные данные: $G, m = 0$

Выходные данные: $x \in \max_{\subseteq} F(f_G), m$ [оценка отклонения числа единиц в полученном наборе от $\max_0 f_G$]

пока $V_0 \neq \emptyset$ **выполнять**

[для всех вершин $v_i \in V_0 \neq \emptyset$ вычислить параметры k_i, m_i такие, что v_i является (k_i, m_i) -вершиной в графе $G\langle V_0 \rangle$, в множестве V_0 выделить подмножество $V'_0 \subseteq V$ с минимальными значениями параметра m_i . Среди выделенных вершин V'_0 выделить вершину $v_{i_0} \in V'_0$ с максимальным значением параметра k_i]

$x_{i_0} \leftarrow 1$

$m \leftarrow m + m_{i_0}$

$V_0 \leftarrow V_0 \setminus \{ \{v_i\} \cup \mathcal{N}(v_i, V_0) \}$

конец цикла

Для набора $x \in \max_{\subseteq} F(f_G)$, полученного посредством реализации Алгоритма 4, справедлива оценка точности решения:

$$\max_0 f_G - |\text{supp}(x)| \leq m .$$

Приведем оценку вычислительной сложности Алгоритма 4.

Для каждой вершины v_i из текущего множества V_0 необходимо вычислить количество вершин в окрестности $\mathcal{N}(v_i, V_0)$ и количество рёбер, недостающих в окрестности $\mathcal{N}(v_i, V_0)$, чтобы индуцированный подграф $G\langle \mathcal{N}(v_i, V_0) \rangle$ стал полным. Следуя определенному принципу, удаляем вершины $v_i \cup \mathcal{N}(v_i, V_0)$ и рёбра $e_i \in G\langle \{v_i\} \cup \mathcal{N}(v_i, V_0) \rangle$ до полного исчерпания вершин в текущем множестве V_0 . Для входных данных заданной размерности

$$V(G) = \{v_1, \dots, v_n\} ,$$

$$\mathcal{E}(G) = \{e_1, \dots, e_p\} ,$$

получаем следующую оценку.

Всего за время работы Алгоритма 4 осуществляется не более n итераций, в каждой из которых требуется не более $O(np)$ действий для вычисления параметров k и t , и не более $O(p)$ действий для операции исключения вершины и её окрестности из текущего графа. Таким образом, Алгоритм 4 имеет вычислительную сложность:

$$O(n \cdot np + np) = O(n^2p) .$$

В некоторых случаях посредством параллельной реализации Алгоритма 3 и Алгоритма 4 может быть получена точная оценка числа единиц в максимальном верхнем нуле рассматриваемой монотонной булевой функции. Результаты разработанных алгоритмов для формирования элементов множества максимальных верхних нулей и множества верхних нулей зависят от выбора вершины, начиная с которой формируются искомые наборы. Реализация алгоритмов для каждого возможного выбора позволяет сформировать множество всех максимальных верхних нулей или множество всех верхних нулей, и оценка отклонения числа единиц в полученных наборах от числа единиц в максимальном верхнем нуле позволяет определить, имеет ли смысл дальнейший поиск, или набор, в случае принадлежности множеству верхних нулей, отвечает некоторому рациональному решению задачи планирования железнодорожных перевозок.

Разработанный в Главе 4 проблемно-ориентированный программный комплекс для решения задачи планирования железнодорожных перевозок основан на параллельной реализации разработанных алгоритмов. В Подразделе 4.1 приводятся результаты реализации разработанного программного комплекса, где для монотонной булевой функции, порождённой заданным неориентированным графом конфликтов, получена точная оценка числа единиц в максимальном верхнем нуле.

Глава 3. Задача организации железнодорожных перевозок на этапе назначения и перемещения локомотивов

Важным этапом организации железнодорожных перевозок является задача о назначении и перемещении локомотивов, заданных начальными условиями доступности, для исполнения плана перевозок. При этом приоритетным является исполнение плана перевозок посредством наименьшего числа назначений локомотивов и наименьшего числа перемещений локомотивов без нагрузки составов вагонов. Ввиду этих ограничений формальная постановка задачи существенно усложняется, кроме того, в некоторых случаях исполнение заданного плана перевозок посредством заданного множества локомотивов невозможно, и в этом случае задача сводится к исполнению наибольшего по мощности подмножества перевозок. В Разделе 3.1 в рамках теоретико–множественного подхода решению задачи отвечает некоторое допустимое отображение, и в рамках теоретико–графового подхода исходная задача сводится к рассмотрению задачи о покрытии вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей.

3.1. Постановка

3.1.1. Теоретико–множественный подход

Постановка задачи о назначении и перемещении локомотивов для исполнения заданного плана перевозок в рамках теоретико–множественного подхода обращается к Алгоритму «Бегущая волна» из Подраздела 2.2.1. В результате работы алгоритма получен некоторый бесконфликтный набор нормативных ниток графика движения поездов, для исполнения которого необходимо произвести назначение локомотивов парка, доступных для эксплуатации.

Рассматриваемый период планирования состоит из конечного числа суток, каждые из которых разбиты на трёхчасовые интервалы, и период планирования

может быть представлен в виде:

$$I_1 \cup I_2 \cup \dots \cup I_{8K-1} \cup I_{8K} ,$$

и для любого момента времени из периода планирования выполняется включение:

$$t \in I_1 \cup \dots \cup I_{8K} .$$

Обозначим через $\Delta\mathcal{N}(\mathcal{A}, \mathcal{B})$ результат работы Алгоритма «Бегущая волна» при заданных результатах работы алгоритмов \mathcal{A} и \mathcal{B} :

$$\Delta\mathcal{N}(\mathcal{A}, \mathcal{B}) = (\Delta\mathcal{N}'(I_1) , \Delta\mathcal{N}'(I_2) , \dots , \Delta\mathcal{N}'(I_{8K})) , \quad (3.1.1)$$

Набор $\Delta\mathcal{N}(\mathcal{A}, \mathcal{B})$ как последовательность формирует задание для назначения и перемещения локомотивов, или задание на перевозку.

В Разделе 1.2.1 нормативная нитка графика определена как допустимый путь в ориентированном мультиграфе энергоэффективных стратегий движения, при этом, как следствие из определения 1.2.2, каждая нормативная нитка характеризуется станцией и временем начала и окончания движения, которые соответствуют вершинам и параметрам дуг, входящих в состав пути в графе первой и последней по порядку:

$$n: (s_n(n) , t_n(n) , s_k(n) , t_k(n)) .$$

Формально понятие задания на перевозку как набора параметров для каждой нормативной нитки приводится в Разделе 1.2.2 при описании теоретико-графовой модели для решения задачи организации железнодорожных перевозок — ориентированного графа совместимости заданий на перевозку.

Как и ранее в Разделе 2.2.1 для некоторой нитки, начало движение по которой приходится на определённый интервал времени из периода планирования, принято обозначение:

$$\text{init}(n) = I: n \in \Delta\mathcal{N}(\mathcal{A}, \mathcal{B}) , I \in \{I_1, \dots, I_{8K}\} .$$

Для любого момента времени из периода планирования и некоторого подмножества нормативных ниток обозначим множество нормативных ниток, движение по которым актуально в рассматриваемый момент времени:

$$\forall t \in I_1 \cup \dots \cup I_{8K} , \\ \mathcal{N}^*(t) = \{n \in \mathcal{N}^*: t_n(n) \leq t \leq t_k(n)\} : \mathcal{N}^* \subseteq \Delta\mathcal{N}(\mathcal{A}, \mathcal{B}) ,$$

и для каждой нормативной нитки обозначим протяжённость маршрута следования через $d(n)$.

Пусть задано множество локомотивов, и множество локомотивов, приписанных к депо A и к депо B :

$$\begin{aligned} \mathcal{L} &= \{L_i : i = 1, 2, \dots, l\} , \\ \mathcal{L} &= \{L_i : i = 1, 2, \dots, l\} = D_A \dot{\cup} D_B . \end{aligned}$$

Важным понятием для дальнейшего рассмотрения является понятие технического обслуживания локомотива (ТО). Для каждого локомотива существует некоторое конечное число видов технического обслуживания. При этом за наблюдаемый период времени для каждого локомотива может быть произведено некоторое число ТО каждого вида, и каждое следующее по порядку ТО каждого вида для каждого локомотива характеризуется временем начала (ТОН) и временем окончания (ТОК):

$$\forall L_i, i = 1, \dots, l : \begin{cases} \text{ТОН}_{i,j}^k, k = 1, \dots, K_{i,j}, j = 1, \dots, p, \\ \text{ТОК}_{i,j}^k, k = 1, \dots, K_{i,j}, j = 1, \dots, p, \end{cases}$$

где j — вид ТО, и k — порядковый номер ТО j -го вида для i -го локомотива. В этих обозначениях для каждого локомотива порядковый номер ТО определённого вида задаётся парой

$$\text{ТО}_{i,j}^k = (\text{ТОН}_{i,j}^k, \text{ТОК}_{i,j}^k) .$$

При эксплуатации локомотивов должны соблюдаться ограничения на выполнение технического обслуживания каждого вида — пробег локомотива и время, прошедшее после крайнего ТО данного вида, не должны превосходить заданных значений. В любой момент времени из периода планирования для каждого локомотива определены допустимый пробег dist и время time до очередного ТО каждого вида, в течении которого локомотив может быть использован для организации перевозок:

$$\forall L_i, i = 1, \dots, l : \begin{cases} \text{dist}(\text{ТО}_{i,j}^k, t), k = 1, \dots, K_{i,j}, j = 1, \dots, p, \\ \text{time}(\text{ТО}_{i,j}^k, t), k = 1, \dots, K_{i,j}, j = 1, \dots, p. \end{cases}$$

При этом для каждого локомотива определены нормативные значения допустимого пробега локомотива после ТО каждого вида, и максимального времени,

допустимого для эксплуатации локомотива в период между последовательными ТО каждого вида:

$$\forall L_i, \text{ТО}_{i,j}, i = 1, \dots, l, j = 1, \dots, p: \begin{cases} \text{dist}_{i,j}^{\max}, \\ \text{time}_{i,j}^{\max}. \end{cases}$$

Кроме того, для каждого локомотива, доступного для назначения для исполнения перевозки, определена станция, в которой локомотив находится в момент времени из периода планирования:

$$\forall L_i, i = 1, \dots, l: \text{pos}(L_i, t).$$

Таким образом, в рамках теоретико-множественного подхода, можно привести формальную постановку задачи о назначении и перемещении локомотивов для исполнения задания на перевозку.

Задача 3.1.1. Пусть заданы время, начиная с которого каждый локомотив впервые доступен для планирования, и начальные условия доступности для каждого локомотива:

$$\begin{cases} t_i^0 \in I_1 \cup I_2 \cup \dots \cup I_{8K-1} \cup I_{8K}, i = 1, \dots, p, \\ \text{pos}(L_i, t_i^0) \in S: i = 1, \dots, l, \\ \text{time}(\text{ТО}_{i,j}^1, t_i^0): i = 1, \dots, l, j = 1, \dots, p, \\ \text{dist}(\text{ТО}_{i,j}^1, t_i^0): i = 1, \dots, l, j = 1, \dots, p. \end{cases}$$

Для заданного процесса «Бегущая волна» вида (3.1.1) необходимо произвести назначение ниток на локомотивы

$$f: \{ \Delta \mathcal{N}'(I_1) \cup \dots \cup \Delta \mathcal{N}'(I_{8K}) \} \longrightarrow 2^{\{L_i: i=1, \dots, l\}}.$$

таким образом, что

$$\left| f(\{ \Delta \mathcal{N}'(I_1) \cup \dots \cup \Delta \mathcal{N}'(I_{8K}) \}) \right| - \text{минимально},$$

где

$$f(\Delta \mathcal{N}(\mathcal{A}, \mathcal{B})) = \bigcup_{n \in \Delta \mathcal{N}(\mathcal{A}, \mathcal{B})} f(n),$$

и для отображения

$$h_f: \mathcal{L} \longrightarrow 2^{\Delta \mathcal{N}'(I_1) \cup \dots \cup \Delta \mathcal{N}'(I_{8K})},$$

определённого для каждого отображения f , выполняется

$$|h_f(L_i)(t)| \leq 1, \quad (3.1.2)$$

где

$$h_f(L_i) = \{n \in \Delta\mathcal{N}(\mathcal{A}, \mathcal{B}) : L_i \in f(n)\},$$

и существует семейство ТО вида

$$\text{ТО} = \{\text{ТО}_{i,j}^k = (\text{ТОН}_{i,j}^k, \text{ТОК}_{i,j}^k) : i = 1, \dots, l, j = 1, \dots, p, k = 1, \dots, K_{i,j}\},$$

такое, что для каждого локомотива и каждого вида ТО выполняются условия:

$$\left\{ \begin{array}{l} \forall i = 1, \dots, l, j = 1, \dots, p, \\ \text{pos}(L_i, \text{ТОН}_{i,j}^k) \in S : k = 1, \dots, K_{i,j}; \\ \text{dist}(\text{ТО}_{i,j}^{k+1}, \text{ТОК}_{i,j}^k) = \text{dist}_{i,j}^{\max} : k = 1, \dots, K_{i,j} - 1; \\ \sum \{d(n) : n \in h_f(L_i), \text{init}(n) \subseteq [t_i^0, \text{ТОН}_{i,j}^1]\} \leq \\ \leq \text{dist}(\text{ТО}_{i,j}^1, t_i^0); \\ \sum \{d(n) : n \in h_f(L_i), \text{init}(n) \subseteq [\text{ТОК}_{i,j}^k, \text{ТОН}_{i,j}^{k+1}]\} \leq \\ \leq \text{dist}_{i,j}^{\max} : k = 1, \dots, K_{i,j} - 1; \\ \text{ТОН}_{i,j}^1 \leq t_i^0 + \text{time}(\text{ТО}_{i,j}^1, t_i^0); \\ \text{ТОН}_{i,j}^{k+1} \leq \text{ТОК}_{i,j}^k + \text{time}_{i,j}^{\max} : k = 1, \dots, K_{i,j} - 1; \\ [\text{ТОН}_{i,j}^k, \text{ТОК}_{i,j}^k] \cap [t_{\text{H}}(n), t_{\text{K}}(n)] = \emptyset : \\ k = 1, \dots, K_{i,j}, n \in h_f(L_i). \end{array} \right. \quad (3.1.3)$$

Выполнение условия (3.1.2) обеспечивает гарантию того, что в любой момент времени каждый локомотив может быть назначен для исполнения не более одной нормативной нитки.

В Разделе 3.2 разработан алгоритм для решения задачи о назначении и перемещении локомотивов в теоретико-множественной постановке для заданного плана перевозок и локомотивного парка сети для линейного участка дороги без учёта ограничений на использование и техническое обслуживание локомотивов.

3.1.2. Теоретико–графовый подход

Ориентированный граф совместимости заданий на перевозку является теоретико–графовой моделью для решения задачи организации на этапе назначения и перемещения локомотивов. В Разделе 1.2.2 приводится описание разработанной модели, на основании которой, в рамках теоретико–графового подхода, задача о назначении и перемещении локомотивов сводится к рассмотрению задачи покрытия вершин ориентированного графа.

Пусть задан ориентированный граф железнодорожной транспортной сети вида (1.2.2). Рассмотрим упорядоченное множество всевозможных перемещений локомотивов вида (1.2.4), элементы которого заданы наборами параметров и соответствуют нормативным ниткам из вариантного графика движения (задания на перевозку) и вариантного графика перемещения (перемещения локомотивов без нагрузки составов вагонов) поездов:

$$(Z \cup D) = \{v_i = (s_{\text{н}}(v_i), t_{\text{н}}(v_i), s_{\text{к}}(v_i), t_{\text{к}}(v_i))\}.$$

Для любого момента времени из периода планирования обозначим множество перемещений, соответствующих нормативным ниткам, движение по которым актуально в рассматриваемый момент времени:

$$(Z \cup D)(t) = \{v_i : t_{\text{н}}(v_i) \leq t \leq t_{\text{к}}(v_i)\}.$$

Множество локомотивов задано начальными условиями доступности локомотивов для назначения посредством указания станции, в которой локомотив находится в момент начала планирования, и временем, начиная с которого каждый локомотив впервые доступен для назначения:

$$\mathcal{L} = \{L_i : i = 1, 2, \dots, l\},$$

$$\forall L_i \in \mathcal{L} : \begin{cases} s^0(L_i) \in S = \{s_1, \dots, s_n\}, \\ t^0(L_i), \end{cases}$$

при этом будем считать, что перевозки, осуществление которых локомотивами имеющегося парка с заданными начальными условиями невозможно, выполняются «фантомным» локомотивом L_0 .

Введём в рассмотрение отображение множества всевозможных перемещений на множество локомотивов, доступных для назначения

$$f: (Z \cup D) \longrightarrow 2^{\mathcal{L} \cup \{L_0\}}, \quad (3.1.4)$$

которое будем называть допустимым, если в любой момент времени, аналогично условию (3.1.2) в постановке задачи в теоретико-множественном подходе, каждый локомотив может быть назначен для исполнения не более одной нитки:

$$|h_f(L_i)(t)| \leq 1, \quad i = 1, \dots, l. \quad (3.1.5)$$

В связи с (3.1.5) для каждого локомотива множество нормативных ниток, для исполнения которых назначен рассматриваемый локомотив, может быть линейно упорядочено:

$$h_f(L_i) = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}, \quad k = |h_f(L_i)|.$$

Тогда в допустимом отображении вида (3.1.4) для каждого локомотива и множества перемещений, для исполнения которых назначен рассматриваемый локомотив, то есть

$$\forall L_i \in f(Z \cup D) : h_f(L_i) = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\},$$

необходимо выполняются условия совместимости перемещений вида:

$$\begin{cases} s_H(v_{i_1}) = s^0(L_i), \\ t^0(L_i) \leq t_H(v_{i_1}), \\ s_K(v_{i_1}) = s_H(v_{i_2}), \dots, s_K(v_{i_{k-1}}) = s_H(v_{i_k}), \\ t_K(v_{i_1}) \leq t_H(v_{i_2}) - \Delta, \dots, t_K(v_{i_{k-1}}) \leq t_H(v_{i_k}) - \Delta, \end{cases} \quad (3.1.6)$$

где действительное неотрицательное число $\Delta \geq 0$ характеризует нормативное время, необходимое для подготовки локомотива после исполнения некоторой нитки к исполнению следующей.

Ниже под $Arg \min$ ($Arg \max$) будут пониматься множества всех объектов, на которых достигается минимум (максимум) соответствующей функции, в отличие от $\arg \min$ ($\arg \max$), которые традиционно обозначают лишь один некоторый объект, обладающий соответствующим свойством.

Рассмотрим цепочку включений,

$$F \supseteq F_1 \supseteq F_2 \supseteq F_3 \supseteq F_4,$$

где

$$\begin{aligned}
 F &= \{f: (Z \cup D) \rightarrow 2^{\mathcal{L} \cup \{L_0\}}\}, \\
 F_1 &\text{ — множество допустимых отображений,} \\
 F_2 &= \mathit{Arg} \max_{f \in F_1} |h_f(\mathcal{L}) \cap Z|, \\
 F_3 &= \mathit{Arg} \min_{f \in F_2} |f(Z \cup D)|, \\
 F_4 &= \mathit{Arg} \min_{f \in F_3} |h_f(\mathcal{L}) \cap D|.
 \end{aligned}$$

В терминах допустимого отображения и рассмотренной цепочки включений постановка задача о назначении и перемещении локомотивов может быть сформулирована следующим образом.

Задача 3.1.2. *Для заданных $(Z \cup D)$ и множества локомотивов \mathcal{L} с начальными условиями доступности, найти отображение $f \in F_4$.*

Рассмотрим далее ориентированный граф совместимости заданий на перевозку вида (1.2.5), порождённый множеством всевозможных перемещений локомотивов. Множество простых ориентированных путей вида

$$p = (v_{i_1}, (v_{i_1}, v_{i_2}), v_{i_2}, \dots, v_{i_{k-1}}, (v_{i_{k-1}}, v_{i_k}), v_{i_k}), \quad (3.1.7)$$

где $v_{i_k} \in (Z \cup D)$, $(v_{i_{k-1}}, v_{i_k}) \in \mathbb{E}$, обозначим через

$$\mathbb{P} = \{p_i: i = 1, \dots, N\},$$

а начало некоторого простого ориентированного пути $p \in \mathbb{P}$ обозначим

$$\mathit{first}(p) = (s_H(v_{i_1}), t_H(v_{i_1}), s_K(v_{i_1}), t_K(v_{i_1})).$$

Сформулируем задачу о покрытии вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей и покажем, что такое покрытие отвечает некоторому решению задачи о назначении и перемещении локомотивов.

Задача 3.1.3. *В ориентированном графе вида (1.2.5) найти подмножество простых ориентированных путей, равномоцное некоторому подмножеству попарно различных локомотивов:*

$$\mathbb{P}' \subseteq \mathbb{P}: \exists \mathcal{L}' \subseteq \mathcal{L}, |\mathcal{L}'| = |\mathbb{P}'|,$$

для которого существует взаимно однозначное соответствие

$$g: \mathbb{P}' \rightarrow \mathcal{L}' ,$$

такое что для любого элемента выполняются условия совместимости вида:

$$\forall p' \in \mathbb{P}': \begin{cases} s_n(\text{first}(p')) = s^0(g(p')) , \\ t_n(\text{first}(p')) \geq t^0(g(p')) , \end{cases}$$

и которое содержит максимально возможное число заданий на перевозку и при этом минимально возможное число путей и минимально возможное число перемещений локомотивов без нагрузки составов вагонов.

Обозначим множество вершин ориентированного графа совместимости заданий на перевозку, входящих в состав некоторого простого ориентированного пути через

$$\text{vert}(p) = \{v: v \in p\} , p \in \mathbb{P} .$$

Обозначим для каждого локомотива множество всех простых ориентированных путей, начало которых «совместимо» с начальными условиями доступности через

$$\mathbb{P}(L_i) \subseteq \mathbb{P} , L_i \in \mathcal{L} ,$$

$$\forall p(L_i) \in \mathbb{P}(L_i): \begin{cases} s_n(\text{first}(p(L_i))) = s^0(L_i) , \\ t_n(\text{first}(p(L_i))) \geq t^0(L_i) , \end{cases} \quad (3.1.8)$$

и при этом множество

$$\mathbb{P}(L_0) ,$$

по определению, включает все простые ориентированные пути, содержащие вершины, соответствующие перемещениям, выполнение которых невозможно посредством имеющегося парка локомотивов.

Пусть задано отображение

$$\omega: \mathcal{L}' \rightarrow \bigcup_{L_i \in \mathcal{L}'} \mathbb{P}(L_i) , \mathcal{L}' \subseteq \mathcal{L} ,$$

такое что

$$\forall L_i \in \mathcal{L}': \omega(L_i) \in \mathbb{P}(L_i) , \quad (3.1.9)$$

и обозначим через Ω_1 множество всех пар, в которых отображение ω удовлетворяет условию (3.1.9), вида

$$(\mathcal{L}', \omega: \mathcal{L}' \rightarrow \bigcup_{L \in \mathcal{L}} \mathbb{P}(L)) . \quad (3.1.10)$$

Рассмотрим цепочку включений:

$$\Omega_1 \supseteq \Omega_2 \supseteq \Omega_3 \supseteq \Omega_4:$$

$$\Omega_2 = \text{Arg} \max_{(\mathcal{L}', \omega) \in \Omega_1} \left| Z \cap \bigcup_{L \in \mathcal{L}'} \text{vert}(p(L)) \right| ,$$

$$\Omega_3 = \text{Arg} \min_{(\mathcal{L}', \omega) \in \Omega_2} |\mathcal{L}'| ,$$

$$\Omega_4 = \text{Arg} \min_{(\mathcal{L}', \omega) \in \Omega_3} \left| D \cap \bigcup_{L \in \mathcal{L}'} \text{vert}(p(L)) \right| ,$$

и отображение, которое каждой паре вида (3.1.10) ставит в соответствие отображение вида (3.1.4):

$$\gamma: \Omega_1 \longrightarrow F ,$$

таким образом, что

$$\begin{aligned} f(v_i) = L_j &\iff v_i \in \text{vert}(p(L_j)) , \\ f(v_i) = L_0 &\iff v_i \notin \bigcup_{L_j \in \mathcal{L}'} \text{vert}(p(L_j)) . \end{aligned}$$

Покажем, что прикладная задача назначения и перемещения локомотивов в постановке вида Задачи 3.1.2 может быть сведена к теоретико-графовой задаче покрытия вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей.

Утверждение 3.1.1. *Отображение γ устанавливает взаимно однозначное соответствие между следующими парами множеств:*

$$\Omega_1 \text{ и } F_1 , \quad (3.1.11)$$

$$\Omega_2 \text{ и } F_2 , \quad (3.1.12)$$

$$\Omega_3 \text{ и } F_3 , \quad (3.1.13)$$

$$\Omega_4 \text{ и } F_4 . \quad (3.1.14)$$

Доказательство. Докажем сначала (3.1.11).

По определению множества $\mathbb{P}(L_i) \subseteq \mathbb{P}$, для каждого простого ориентированного пути $p(L_i) \in \mathbb{P}(L_i)$ выполняются условия (3.1.8), то есть для отображения f частично выполняются условия (3.1.6) допустимого отображения.

По построению имеем:

$$f(v_{i_1}) = f(v_{i_2}) = L_j \iff v_{i_1}, v_{i_2} \in \text{vert}(p(L_j)),$$

и, в то же время, по определению ориентированного графа совместимости заданий на перевозку,

$$v_{i_1}, v_{i_2} \in \text{vert}(p(L_j)) \iff t_{\text{н}}(v_{i_2}) \geq t_{\text{к}}(v_{i_1}) + \Delta,$$

или

$$v_{i_1}, v_{i_2} \in \text{vert}(p(L_j)) \iff t_{\text{н}}(v_{i_1}) \geq t_{\text{к}}(v_{i_2}) + \Delta,$$

откуда следует, что условия (3.1.6) допустимого отображения полностью выполняются.

С другой стороны, для допустимого отображения $f \in F_1$ можно построить пару $(\mathcal{L}', \omega: \mathcal{L}' \rightarrow \bigcup_{L_i \in \mathcal{L}'} \mathbb{P}(L_i))$ таким образом, что

$$f(Z \cup D) \setminus \{L_0\} = \mathcal{L}',$$

и

$$\forall L_i \in \mathcal{L}': h_f(L_i) = \text{vert}(p(L_i)), \quad (3.1.15)$$

и, таким образом, установлена двойственная взаимосвязь между допустимыми отображениями $f \in F_1$ и парами $(\mathcal{L}', \omega: \mathcal{L}' \rightarrow \bigcup_{L_i \in \mathcal{L}'} \mathbb{P}(L_i)) \in \Omega_1$.

Далее, из (3.1.15) следует, что

$$h_f(\mathcal{L}') = \bigcup_{L_i \in \mathcal{L}'} \text{vert}(p(L_i)), \quad (3.1.16)$$

и из (3.1.16) следует, что

$$\gamma(\Omega_2) = F_2 \text{ и } \gamma^{-1}(F_2) = \Omega_2.$$

Для отображения γ число локомотивов в паре $(\mathcal{L}', \omega: \mathcal{L}' \rightarrow \bigcup_{L_i \in \mathcal{L}'} \mathbb{P}(L_i))$ равно числу локомотивов $|f(Z \cup D) \setminus \{L_0\}|$, из чего следует, что

$$\gamma(\Omega_3) = F_3 \text{ и } \gamma^{-1}(F_3) = \Omega_3,$$

Наконец, из соотношения (3.1.16) снова получаем, что

$$\gamma(\Omega_4) = F_4 \text{ и } \gamma^{-1}(F_4) = \Omega_4 ,$$

что и требовалось доказать. \square

Таким образом покрытие вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей отвечает некоторому решению задачи о назначении и перемещении локомотивов.

Утверждение 3.1.2. Любая пара $\left(\mathcal{L}', \omega: \mathcal{L}' \rightarrow \bigcup_{L_i \in \mathcal{L}'} \mathbb{P}(L_i) \right) \in \Omega_4$ является решением Задачи 3.1.3 и взаимно однозначно соответствует некоторому решению $f \in F_4$ Задачи 3.1.2.

Доказательство. Утверждение является непосредственным следствием Утверждения 3.1.1. \square

В Разделе 3.2 для решения задачи организации железнодорожных перевозок на этапе назначения и перемещения локомотивов без учёта ограничений на использование и техническое обслуживание локомотивов разработаны алгоритм формирования множества путей ориентированного графа, алгоритм формирования множества максимальных по включению путей и алгоритм покрытия вершин ориентированного графа множеством максимальных по включению путей. В Разделе 4.2 приводится описание проблемно-ориентированного программного комплекса, реализующего последовательно разработанные алгоритмы.

3.2. Решение

Разработка программного и аппаратного обеспечения систем управления процессами планирования и организации железнодорожных перевозок требует разработки эффективных вычислительных алгоритмов.

3.2.1. Алгоритм назначения и перемещения локомотивов

Рассмотрим алгоритм решения Задачи 3.1.1 из Раздела 3.1 для линейного участка дороги без учёта ограничений на использование и техническое обслуживание локомотивов, в предположении, что локомотивы $\mathcal{L} = \{L_i: i = 1, \dots, l\}$ локализованы на станциях A и B в начале периода планирования, имеются в необходимом количестве и доступны далее на всем периоде планирования. В этих условиях необходимо исполнить план перевозок с использованием наименьшего количества локомотивов в реализации плана перевозок.

Пусть заданы:

1) Период планирования, состоящий из конечного числа суток, каждые из которых разбиты на трёхчасовые интервалы:

$$I_1 \cup I_2 \cup \dots \cup I_q, q = 8K,$$

2) Бесконфликтный набор нормативных ниток, полученный посредством реализации Алгоритма «Бегущая волна»:

$$\Delta\mathcal{N}(\mathcal{A}, \mathcal{B}) = (\Delta\mathcal{N}'(I_1), \dots, \Delta\mathcal{N}'(I_q)),$$

3) Множество локомотивов, доступных для назначения, распределенных на станциях A и B :

$$\{L_i: i = 1, \dots, l\}.$$

Множество $\Delta\mathcal{N}(\mathcal{A}, \mathcal{B})$ может быть представлено в виде последовательности нормативных ниток, заданных наборами параметров:

$$\mathcal{N} = \left(\mathbf{n}_1 = (s_{\text{н}}(\mathbf{n}_1), (t_{\text{н}}(\mathbf{n}_1), 1), s_{\text{к}}(\mathbf{n}_1), (t_{\text{к}}(\mathbf{n}_1), 1)), \dots, \right. \\ \left. \mathbf{n}_p = (s_{\text{н}}(\mathbf{n}_p), (t_{\text{н}}(\mathbf{n}_p), r), s_{\text{к}}(\mathbf{n}_p), (t_{\text{к}}(\mathbf{n}_1), r)) \right),$$

где $s_{\text{н}}(\mathbf{n}_i)$ и $s_{\text{к}}(\mathbf{n}_i)$ – станция начала и окончания движения по нитке \mathbf{n}_i , $p = |\Delta\mathcal{N}(\mathcal{A}, \mathcal{B})|$, и нитки упорядочены по возрастанию в лексикографическом порядке относительно порядкового номера суток r , $r = 1, \dots, K$, и относительно времени начала движения по нитке \mathbf{n}_i в соответствующих сутках.

Для нормативных ниток из бесконфликтного набора, начало движения по которым приходится на определённый интервал из периода планирования,

принято обозначение:

$$\forall n_i \in \mathcal{N}: \text{init}(n_i) = I, I \in I_1 \cup \dots \cup I_q.$$

Для $r \in \{1, \dots, K\}$ введём в рассмотрение ориентированный граф зависимостей ниток такой, что:

$$\Gamma = (\{n_i: i = 1, \dots, p\}, E), \quad (3.2.1)$$

$$(n_i, n_j) \in E \iff \begin{cases} s_n(n_j) = s_k(n_i), \\ t_n(n_j) \geq t_k(n_i) + \Delta \\ \left\lceil \frac{\text{init}(n_j)}{8} \right\rceil - \left\lceil \frac{\text{init}(n_i)}{8} \right\rceil \leq r, \end{cases}$$

где $\Delta \geq 0$ – нормативное время, необходимое для подготовки локомотива для исполнения нитки n_j после исполнения нитки n_i , и обозначение $\lceil \cdot \rceil$ определяет целое число, не меньшее действительного числа, указанного в скобках.

Массив mark длины p используется для пометки ниток из \mathcal{N} . Нитку n_i будем называть непомеченной, когда для ее выполнения еще не назначен локомотив и $\text{mark}(n_i) = 0$.

Потребуется следующие обозначения:

$$\Gamma_I(n_i) = \{n_j \in \mathcal{N}: (n_i, n_j) \in E, \text{init}(n_j) = I\},$$

$$\Gamma_I^0(n_i) = \{n_j \in \mathcal{N}: (n_i, n_j) \in E, \text{mark}(n_j) = 0, \text{init}(n_j) = I\},$$

Переменная NL используется для номера локомотива.

Алгоритм С (n_i, n_j): Алгоритм поиска ближайшей к нитке n_i непомеченной нитки n_j .

Если $\{k \in [m, q]: \text{init}(n_i) = I_m, m = 1, \dots, q, \Gamma_{I_k}^0(n_i) \neq \emptyset\} = \emptyset$ **то**

$$\mathcal{C}(n_i, n_j) = -1$$

нитка n_j не определена

Конец Алгоритма

иначе

$$k_0 \leftarrow \min \{k \in [m, q]: \text{init}(n_i) = I_m, m = 1, \dots, q, \Gamma_{I_k}^0(n_i) \neq \emptyset\}$$

$$j_0 \leftarrow \min \{j = 1, \dots, q: n_j \in \Gamma_{I_{k_0}}^0(n_i)\}$$

$$n_j = n_{j_0}$$

$$\mathcal{C}(n_i, n_j) = 1$$

Конец Алгоритма

Алгоритм D: Алгоритм назначения и перемещения локомотивов.

1) $NL = 1$

$\text{mark}(n_i) = 0 \forall i = 1, \dots, p$

$P = \emptyset$ [множество ниток, назначенных на текущий локомотив L_{NL}]

2) **Если** $\{i = 1, \dots, p: \text{mark}(n_i) = 0\} = \emptyset$ **то**

Конец Алгоритма

иначе

$i_0 = \min \{i = 1, \dots, p: \text{mark}(n_i) = 0\}$

$P \leftarrow P \cup \{n_{i_0}\}$

$\text{mark}(n_{i_0}) = 1$

3) **Если** $\Gamma_{I_k}(n_{i_0}) = \emptyset \forall k \in [m, q]: \text{init}(n_{i_0}) = I_m, m = 1, \dots, q$ **то**

[завершается сбор ниток, назначенных на локомотив L_{NL} и производится назначение ниток из P на локомотив L_{NL}]

$P \leftarrow \emptyset$

$NL = NL + 1$

Идти к 2)

иначе

$k_0 = \min \{k \in [m, q]: \text{init}(n_{i_0}) = I_m, m = 1, \dots, q, \Gamma_{I_k}(n_{i_0}) \neq \emptyset\}$

Если $\Gamma_{I_{k_0}}^0(n_{i_0}) \neq \emptyset$ **то**

$j_0 = \min \{j = 1, \dots, p: n_j \in \Gamma_{I_{k_0}}^0(n_{i_0})\}$

$P \leftarrow P \cup \{n_{j_0}\}$

$i_0 \leftarrow j_0$

$\text{mark}(n_{j_0}) = 1$

Идти к 3)

иначе

Выполнить $C(n_{i_0}, n_j)$

Выполнить $C(n_i, n_j)$ для всех $n_i \in \Gamma_{I_{k_0}}(n_{i_0})$

Если $\begin{cases} C(n_{i_0}, n_j) = -1 \\ C(n_i, n_j) = -1 \text{ для всех } n_i \in \Gamma_{I_k}(n_{i_0}) \end{cases}$ **то**

[завершается сбор ниток, назначенных на локомотив L_{NL} и производится назначение ниток из P на локомотив L_{NL}]

$P \leftarrow \emptyset$

$NL = NL + 1$

Идти к 2)

иначе

[среди найденных ниток взять первую с минимальным номером j_0 в порядке ее появления]

Если n_{j_0} из $C(n_{i_0}, n_j)$ **то**

$$P \leftarrow P \cup \{n_{j_0}\}$$

$$i_0 \leftarrow j_0$$

$$\text{mark}(n_{j_0}) = 1$$

Идти к 3)

иначе

$$P \leftarrow P \cup \{n_i\} \cup \{n_{j_0}\}$$

$$i_0 \leftarrow j_0$$

$$\text{mark}(n_{j_0}) = 1$$

Идти к 3)

В практических задачах в результате работы Алгоритма «Бегущая волна» получается множество $\Delta\mathcal{N}(\mathcal{A}, \mathcal{B})$, число элементов которого достигает в типичных случаях величин порядка нескольких сотен ниток в сутки.

Оценим вычислительную сложность Алгоритма \mathcal{D} назначения и перемещения локомотивов. В процессе работы алгоритма производится пометка непомеченных изначально вершин графа зависимостей ниток. В 2) и в 3) алгоритма осуществляется пометка очередной вершины графа, причём общее число действий в 2) не превосходит

$$O(\text{const} \cdot p) = O(p),$$

где const — некоторая константа, и p — число вершин ориентированного графа зависимостей ниток.

Пусть n_{i_0} — вершина графа Γ , $\Gamma(n_{i_0})$ — окрестность первого порядка вершины n_{i_0} и $\Gamma\Gamma(n_{i_0})$ — окрестность второго порядка вершины n_{i_0} . В 3) для заданной вершины n_{i_0} просматривается подмножество множества $\{n_{i_0} \cup \Gamma(n_{i_0}) \cup \Gamma\Gamma(n_{i_0})\}$ вершин графа Γ , для построения которого просматриваются все дуги индуцированного подграфа $\Gamma\langle\{n_{i_0}\} \cup \Gamma(n_{i_0}) \cup \Gamma\Gamma(n_{i_0})\rangle$.

Обозначим наибольшее возможное число дуг в семействе индуцированных подграфов через

$$M(\Gamma) = \max \{E(n_i) : i = 1, \dots, p\},$$

где $E(n_i)$ — число дуг в индуцированном подграфе $\Gamma\langle\{n_i\} \cup \Gamma(n_i) \cup \Gamma\Gamma(n_i)\rangle$.

Тогда вычислительная сложность Алгоритма \mathcal{D} назначения и перемещения локомотивов не превосходит

$$O(p \cdot M(\Gamma)) = O(p \cdot |E|),$$

где $|E|$ – число дуг ориентированного графа зависимостей ниток.

Замечание 3.2.1. В определении ориентированного графа зависимостей ниток участвует параметр $r = 1, \dots, K$. Обозначим через Γ^i ориентированный граф, определенный при $r = i$. Тогда из определения следует справедливость цепочки включений:

$$\Gamma^1 \subseteq \dots \subseteq \Gamma^K.$$

Решение Задачи 3.1.1 может быть тем лучше, чем больше значение r , однако при этом растет объем вычислений алгоритма назначения и перемещения локомотивов. Одним из естественных компромиссов между сложностью задачи и качеством получаемого решения может быть значение r , которое зависит от средней протяженности ниток, его можно рассчитать следующим образом:

$$r_{rec} = \left\lceil \frac{\sum_{i=1}^p |n_i|}{24p} \right\rceil,$$

где $|n_i| = (t_k(n_i) - t_n(n_i))$.

В задачах организации железнодорожных перевозок большой размерности для исполнения нормативных ниток зачастую используют несколько локомотивов, эксплуатация которых допустима в некоторых пределах обслуживания. Для каждого участка железнодорожной сети линейной структуры разработанный алгоритм имеет особую практическую ценность и может быть реализован для решения задачи о назначении и перемещении локомотивов. В Разделе 4.2 приводятся результаты работы алгоритма для заданного бесконфликтного набора нормативных ниток графика движения поездов.

3.2.2. Алгоритм покрытия

В рамках теоретико–графового подхода задача организации железнодорожных перевозок на этапе назначения и перемещении локомотивов в Разделе 3.2 была сведена к задаче покрытия вершин ориентированного графа совместимости заданий на перевозку.

Рассмотрим ориентированный граф совместимости заданий на перевозку вида (1.2.5). Специфическая структура ориентированного графа совместимости заданий на перевозку позволяет выявить полезные свойства множества путей графа, на основе которых разработаны эффективные вычислительные алгоритмы для решения задачи о покрытии вершин графа минимальным числом максимальных по включению путей, которое отвечает некоторому решению задачи организации железнодорожных перевозок.

Утверждение 3.2.1. Пусть для ориентированного графа совместимости заданий на перевозку

$$\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E}), |Z \cup D| = n,$$

построена матрица смежности вершин вида

$$A_{\vec{\mathbb{G}}} = \|a_{ij}\|_{[n \times n]}.$$

Тогда

$$a_{ij} = 0: i \geq j, i = 1, 2, \dots, n, j = 1, 2, \dots, n,$$

и

$$A_{\vec{\mathbb{G}}} = \begin{vmatrix} 0 & a_{12} & a_{13} & a_{14} & \dots & \dots & a_{1(n-1)} & a_{1n} \\ 0 & 0 & a_{23} & a_{24} & \dots & \dots & a_{2(n-1)} & a_{2n} \\ 0 & 0 & 0 & a_{34} & \dots & \dots & a_{3(n-1)} & a_{3n} \\ \dots & \dots \\ \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & a_{(n-1)n} \\ 0 & 0 & 0 & 0 & \dots & \dots & 0 & 0 \end{vmatrix}.$$

Доказательство. По определению матрицы смежности вершин ориентированного графа

$$a_{ij} = \begin{cases} 1, & \text{если } (v_i, v_j) \in \mathbb{E}, \\ 0, & \text{в противном случае.} \end{cases}$$

В то же время, по определению ориентированного графа совместимости заданий на перевозку $\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E})$, для любой пары вершин

$$(v_i, v_j) \in \mathbb{E} \iff \begin{cases} s_{\text{к}}(v_i) = s_{\text{н}}(v_j) , \\ t_{\text{к}}(v_i) + \Delta \leq t_{\text{н}}(v_j) , \end{cases}$$

где $v_i, v_j \in (Z \cup D)$ – нормативные нитки из бесконфликтного набора нормативных ниток, взаимно однозначно соответствующие вершинам графа совместимости заданий на перевозку.

Но, по определению, бесконфликтный набор нормативных ниток $(Z \cup D)$ взаимно однозначно соответствует множеству вершин ориентированного графа совместимости заданий на перевозку $\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E})$ и упорядочен лексикографически. Значит, из

$$t_{\text{н}}(v_i) < t_{\text{к}}(v_i) < t_{\text{к}}(v_i) + \Delta \leq t_{\text{н}}(v_j)$$

следует, что

$$i \leq j \text{ для всех } i, j: (v_i, v_j) \in \mathbb{E} ,$$

и

$$a_{ij} = 0 \text{ для всех } i, j: i \geq j ,$$

что и требовалось доказать. □

Следствие 3.2.1. *В ориентированном графе совместимости заданий на перевозку для любого простого ориентированного пути вида (3.1.7) последовательность*

$$(i_1, i_2, \dots, i_k)$$

номеров вершин графа, входящих в путь возрастает.

Доказательство. Доказательство следует из Утверждения 3.2.1. Поскольку номера столбцов и строк матрицы смежности $\mathcal{A}_{\vec{\mathbb{G}}} = \left\| a_{ij} \right\|_{[n \times n]}$ соответствуют номерам i, j соответствующих вершин графа, и выполняется

$$(v_i, v_j) \in \mathbb{E} \iff i > j ,$$

то из

$$(v_{i_1}, v_{i_2}) \in \mathbb{E} , (v_{i_2}, v_{i_3}) \in \mathbb{E} , \dots , (v_{i_{k-1}}, v_{i_k}) \in \mathbb{E}$$

следует

$$i_1 > i_2, i_2 > i_3, \dots, i_{k-1} > i_k,$$

что и требовалось доказать. \square

Задание структуры на графе посредством матрицы смежности вершин с точки зрения программной реализации алгоритмов на графах является не самым рациональным подходом. На основании Утверждения 3.2.1 множество путей ориентированного графа совместимости заданий на перевозку может быть представлено в виде двоичной матрицы, что существенно упрощает разработку, и такое представление лежит в основе программной реализации алгоритмов для решения задачи организации железнодорожных перевозок.

Следствие 3.2.2. *В ориентированном графе совместимости заданий на перевозку существует единственный путь p с множеством вершин $vert(p)$, или, другими словами, если p и p' – простые ориентированные пути в графе такие, что*

$$vert(p) = vert(p'),$$

то

$$p = p'.$$

Доказательство. Рассмотрим простой ориентированный путь вида

$$p: vert(p) = \{v_{i_1}, v_{i_2}, \dots, v_{i_k}\}, v_{i_k} \in (Z \cup D),$$

и простой ориентированный путь вида

$$p': vert(p') = \{v_{j_1}, v_{j_2}, \dots, v_{j_k}\}, v_{j_k} \in (Z \cup D).$$

Согласно Следствию 3.2.1, последовательности

$$(i_1, i_2, \dots, i_k)$$

и

$$(j_1, j_2, \dots, j_k)$$

возрастающие. Тогда, в силу того, что для конечного множества натуральных чисел существует единственная возрастающая последовательность, имеем

$$(i_1, i_2, \dots, i_k) = (j_1, j_2, \dots, j_k).$$

По определению (3.1.7) простого ориентированного пути соседние вершины в последовательности вершин и дуг соединены дугой, следовательно

$$p = (v_{i_1}, (v_{i_1}, v_{i_2}), v_{i_2}, \dots, v_{i_{k-1}}, (v_{i_{k-1}}, v_{i_k}), v_{i_k}),$$

и

$$\begin{aligned} p' &= (v_{j_1}, (v_{j_1}, v_{j_2}), v_{j_2}, \dots, v_{j_{k-1}}, (v_{j_{k-1}}, v_{j_k}), v_{j_k}) = \\ &= (v_{i_1}, (v_{i_1}, v_{i_2}), v_{i_2}, \dots, v_{i_{k-1}}, (v_{i_{k-1}}, v_{i_k}), v_{i_k}). \end{aligned}$$

Таким образом

$$p = p',$$

что и требовалось доказать. □

Пусть \mathbb{P} – множество всех простых ориентированных путей ориентированного графа совместимости заданий на перевозку, и $X \subseteq \{0, 1\}^n$ – множество двоичных наборов длины n .

Пусть также задано отображение

$$\chi: \mathbb{P} \rightarrow X$$

такое, что для каждого простого ориентированного пути $p \in \mathbb{P}$ из множества путей вида

$$p = (v_{i_1}, (v_{i_1}, v_{i_2}), v_{i_2}, \dots, v_{i_{k-1}}, (v_{i_{k-1}}, v_{i_k}), v_{i_k})$$

построен двоичный набор вида

$$\chi(p) = (\alpha_1, \alpha_2, \dots, \alpha_n),$$

где

$$\alpha_i = 1 \Rightarrow v_i \in p.$$

Следствие 3.2.3. *Отображение χ является однозначным и множество \mathbb{P} может быть представлено в виде двоичной матрицы, содержащей $n = |Z \cup D|$ столбцов, каждая строка которой является двоичным набором $\chi(p)$, соответствующим некоторому простому ориентированному пути $p \in \mathbb{P}$ из множества путей.*

Доказательство. Однозначность отображения χ следует из доказательства Следствия 3.2.2.

Рассмотрим матрицу

$$M = \left\| \alpha_{ij} \right\|_{[q \times n]},$$

где $n = |Z \cup D|$, $q = |\mathbb{P}|$.

Записав последовательно в строках двоичной матрицы M двоичные наборы $\chi(p)$ для всех $p \in \mathbb{P}$ получим двоичную матрицу, в которой все строки попарно различны, ввиду однозначности отображения χ , и для каждой строки матрицы

$$(\alpha_{1j}, \dots, \alpha_{nj})$$

существует единственный путь

$$\chi^{-1}(\alpha_{1j}, \dots, \alpha_{nj}) \in \mathbb{P}$$

в графе $\vec{G} = ((Z \cup D), \mathbb{E})$, причём

$$\{\chi^{-1}(\alpha_{1j}, \dots, \alpha_{nj}) : j = 1, 2, \dots, q\} = \mathbb{P},$$

что и требовалось доказать. □

Посредством реализации алгоритмов обхода ориентированных графов в ширину и в глубину может быть сформировано множество путей, однако, алгоритмов с полиномиальным временем исполнения на сегодняшний день не существует. Приведём циклический алгоритм с ветвлением для формирования множества путей ориентированного графа совместимости заданий на перевозку, в основе которого лежат принципы алгоритмов обхода.

Для любого простого ориентированного пути $p \in \mathbb{P}$ в ориентированном графе совместимости заданий на перевозку, представленного в виде последовательности вершин (такое представление возможно и однозначно, согласно Следствию 3.2.3)

$$p = (v_{i_1}, v_{i_2}, \dots, v_{i_{k-1}}, v_{i_k}), v_{i_k} \in (Z \cup D),$$

обозначим вершину, входящую в последовательность вершин последней по порядку, как

$$last(p) = v_i, \text{ где } i > j \text{ для всех } j: v_j \in vert(p);$$

и для любого простого ориентированного пути $p \in \mathbb{P}$ в ориентированном графе совместимости заданий на перевозку обозначим окрестность вершины $last(p)$

в ориентированном графе как

$$\mathcal{N}(\text{last}(p)) = \{v_i : (\text{last}(p), v_i) \in \mathbb{E}\}.$$

Алгоритм 5: Алгоритм построения множества всех простых ориентированных путей ориентированного графа совместимости заданий на перевозку.

Входные данные: $\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E})$

Выходные данные: \mathbb{P}

1: $i = 1$;

$$\mathbb{P}_1 = \{(v_1), (v_2), \dots, (v_n)\};$$

$$\mathbb{P} = \emptyset;$$

2: $\text{Ind} = 0$;

$$i = i + 1;$$

$$\mathbb{P}_i = \emptyset;$$

3: **для каждого** $p \in \mathbb{P}_{i-1}$ **выполнять**

4: **если** $\mathcal{N}(\text{last}(p)) = \emptyset$ [окрестность вершины, входящей в путь последней]
то

5: продолжать ;

6: **иначе**

7: $\text{Ind} \leftarrow \text{Ind} + 1$;

$$\mathbb{P}_i \leftarrow \mathbb{P}_i \cup \{p \cup \{v_i\} : v_i \in \mathcal{N}(\text{last}(p))\};$$

 продолжать ;

конец цикла

8: **если** $\text{Ind} \neq 0$ **то**

9: $\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{P}_i$;

 идти к 2 ;

10: **иначе**

11: $\mathbb{P} \leftarrow \mathbb{P} \cup \mathbb{P}_1$;

 конец алгоритма ;

12: **конец цикла**

Множество путей ориентированного графа совместимости заданий на перевозку, полученное посредством реализации Алгоритма 5, лежит в основе приближённого алгоритма для решения задачи покрытия вершин графа минимальным числом максимальных по включению путей. Непосредственно из контекста

следует закономерная необходимость в рассмотрении множества максимальных путей в ориентированном графе совместимости заданий на перевозку.

Введем на множестве \mathbb{P} отношение частичного порядка « $<$ » следующим образом:

$$p_1 < p_2 \iff \text{vert}(p_1) \subset \text{vert}(p_2) ;$$

и обозначим через

$$\mathbb{P}_{\max}$$

семейство всех максимальных элементов частично упорядоченного множества $(\mathbb{P}, <)$.

Рассмотрим $\vec{G} = ((Z \cup D), \mathbb{E})$, $|Z \cup D| = n$ – ориентированный граф совместимости заданий на перевозку, и $\mathbb{P} = \{p_i, i = 1, 2, \dots, q\}$ – множество всех простых ориентированных путей графа \vec{G} .

Пусть $C \subseteq \mathbb{P}$ – покрытие множества вершин графа \vec{G} набором k попарно различных простых ориентированных путей:

$$\bigcup \{\text{vert}(p) : p \in C\} = Z \cup D ,$$

и

$$|C| = k .$$

Утверждение 3.2.2. *Для любого покрытия C множества вершин графа совместимости заданий на перевозку существует покрытие C' , такое что*

$$C' \subset \mathbb{P}_{\max}$$

и

$$|C| \geq |C'| .$$

Доказательство. В частично упорядоченном множестве $(\mathbb{P}, <)$ для каждого элемента $p \in \mathbb{P}$ имеем либо

$$p \in \mathbb{P}_{\max} ,$$

либо среди элементов множества $\mathbb{P} = \{p_i, i = 1, 2, \dots, q\}$, найдется простой ориентированный путь

$$p' \in \mathbb{P} : p < p' .$$

Таким образом, для любого простого ориентированного пути $p_i \in C$, $i \in \{1, k\}$, среди элементов множества \mathbb{P} найдется простой ориентированный путь

$$p'_{s_i} \in \mathbb{P} : p_i \leq p'_{s_i} ,$$

и, следовательно,

$$p'_{s_i} \in \mathbb{P}_{\max} .$$

Из определения частично упорядоченного множества $(\mathbb{P}, <)$ имеем

$$Z \cup D = \bigcup \{vert(p_i) , i = 1, 2, \dots, k\} \subseteq \bigcup \{vert(p'_{s_i}) , i = 1, 2, \dots, k\} \subseteq Z \cup D ,$$

то есть множество

$$C' = \{p'_{s_i} , i = 1, 2, \dots, k\}$$

является покрытием множества вершин ориентированного графа совместимости заданий на перевозку.

Поскольку все элементы набора C попарно различны по условию, а элементы набора C' не обязательно попарно различны, то

$$|C'| \leq k = |C| ,$$

и

$$|C| \geq |C'| ,$$

что и требовалось доказать. □

Введем обозначение \mathbf{C} для множества всех покрытий множества вершин ориентированного графа совместимости заданий на перевозку наборами простых ориентированных путей из множества \mathbb{P} . Пусть

$$m_1 = \min\{|C| , C \in \mathbf{C}\} .$$

Введем также обозначение \mathbf{C}' для множества всех покрытий множества вершин ориентированного графа совместимости заданий на перевозку наборами максимальных простых ориентированных путей из множества \mathbb{P}_{\max} . Пусть

$$m_2 = \min\{|C'| , C' \in \mathbf{C}'\} .$$

Покрытие $C \in \mathbf{C}$ такое, что

$$|C| = m_1$$

будем называть минимальным покрытием вершин графа набором путей из множества \mathbb{P} . Аналогично, минимальным покрытием вершин графа набором максимальных путей из множества \mathbb{P}_{\max} будем называть покрытие $C' \in \mathbf{C}'$ такое, что

$$|C'| = m_2 .$$

Следующее утверждение о свойстве покрытия позволяет ограничиться множеством максимальных путей для решения задачи покрытия вершин ориентированного графа, что существенно снижает размерность задачи покрытия, и, соответственно, задачи организации железнодорожных перевозок, без потери качества решения в части количества локомотивов, задействованных для исполнения заданного плана перевозок.

Следствие 3.2.4. *В ориентированном графе совместимости заданий на перевозку мощность m_1 минимального покрытия вершин графа набором простых ориентированных путей из множества \mathbb{P} равна мощности m_2 минимального покрытия вершин графа набором максимальных путей из множества \mathbb{P}_{\max} , то есть*

$$m_1 = m_2 .$$

Доказательство. Пусть m_1 – мощность минимального покрытия $C \in \mathbf{C}$ множества вершин ориентированного графа совместимости заданий на перевозку набором попарно различных простых ориентированных путей из множества \mathbb{P} .

Из Утверждения 3.2.2 следует, что существует покрытие $C' \in \mathbf{C}'$ множества вершин ориентированного графа совместимости заданий на перевозку набором максимальных простых ориентированных путей из множества \mathbb{P}_{\max} такое, что

$$|C'| \leq |C| .$$

Пусть m_2 – мощность покрытия $C' \in \mathbf{C}'$. Тогда

$$m_2 \leq m_1 .$$

По определению частично упорядоченного множества $(\mathbb{P}, <)$ имеем

$$\mathbb{P}_{\max} \subseteq \mathbb{P} ,$$

и, следовательно, можем заключить, что покрытие $C' \in \mathbf{C}'$ является также покрытием множества вершин ориентированного графа совместимости заданий на перевозку набором простых ориентированных путей из множества \mathbb{P} , и $C' \in \mathbf{C}$.

По определению минимального покрытия множества вершин ориентированного графа совместимости заданий на перевозку набором попарно различных простых ориентированных путей из множества \mathbb{P} имеем:

$$m_1 \leq m_2 ,$$

и, окончательно получаем

$$m_1 = m_2 ,$$

что и требовалось доказать. □

Алгоритм 6: Алгоритм построения множества максимальных простых ориентированных путей ориентированного графа совместимости заданий на перевозку.

Входные данные: $\vec{G} = ((Z \cup D), E), \mathbb{P}$

Выходные данные: \mathbb{P}_{\max}

- 1: $\mathbb{P}_{\max} = \emptyset$
 - 2: **для каждого** $p_1 \in \mathbb{P}$ **выполнять**
 - 3: $\text{Ind}_{\max} = 1$ [индикатор максимальности пути]
 - 4: **для каждого** $p_2 \in \mathbb{P}$ **выполнять**
 - 5: **если** $\text{vert}(p_1) \subset \text{vert}(p_2)$ **то**
 - 6: $\text{Ind}_{\max} = 0$
 конец цикла
 - 7: **иначе**
 - 8: продолжать
 - 9: **если** $\text{Ind}_{\max} = 1$ [путь p_1 оказался максимальным] **то**
 - 10: $\mathbb{P}_{\max} \leftarrow \mathbb{P}_{\max} \cup \{p_1\}$
 - 11: **иначе**
 - 12: продолжать
- конец цикла**
- конец цикла**

Согласно Следствию 3.2.4 любое покрытие вершин множеством максимальных путей, равномощное минимальному покрытию, является минимальным. Таким образом, решение задачи о покрытии вершин ориентированного графа совместимости заданий на перевозку может быть найдено среди элементов множества максимальных путей, полученного посредством реализации Алгоритма 6.

Пусть заданы граф совместимости заданий на перевозку $\vec{G} = ((Z \cup D), E)$ и семейство \mathbb{P}_{\max} максимальных простых ориентированных путей графа.

Введем в рассмотрение для каждого ориентированного пути $p \in \mathbb{P}_{\max}$ несколько индикаторов. Пусть задано некоторое подмножество $V' \subseteq (Z \cup D)$

множества вершин ориентированного графа совместимости заданий на перевозку $\vec{\mathbb{G}} = ((Z \cup D), \mathbb{E})$. Тогда:

$$ind_1(v) = |\{p \in \mathbb{P}_{\max} : v \in vert(p)\}|, \quad (3.2.2)$$

$$ind_2(p, V') = |V' \cap vert(p)|, \quad (3.2.3)$$

$$ind_3(p, V') = \sum \{ind_1(v) : v \in \{V' \cap vert(p)\}\}, \quad (3.2.4)$$

$$ind_4(p, V') = \min \{ind_1(v) : v \in \{V' \cap vert(p)\}\}. \quad (3.2.5)$$

Обозначим

$$Ind(p, V') = (ind_2(p, V'), ind_3(p, V'), ind_4(p, V')),$$

и будем называть $Ind(p, V')$ обобщенным индексом пути $p \in \mathbb{P}_{\max}$ на подмножестве $V' \subseteq (Z \cup D)$ вершин ориентированного графа совместимости заданий на перевозку.

Рассмотрим семейство

$$S(V') = \{(Ind(p, V'), p) : p \in \mathbb{P}_{\max}, V' \cup vert(p) \neq \emptyset\}.$$

Введём в рассмотрение операцию

$$\text{sort}(S(V'))$$

последовательной сортировки элементов семейства $S(V')$ относительно обобщенного индекса пути $Ind(p, V')$ такой, что элементы семейства сначала упорядочиваются по убыванию индекса ind_2 , затем по возрастанию индекса ind_3 , и затем по возрастанию индекса ind_4 . Первый элемент в упорядоченном семействе $S(V')$ обозначим

$$\text{prime}(\text{sort}(S(V'))).$$

Алгоритм 7: Алгоритм $\mathcal{S}(Ind)$ покрытия вершин графа совместимости заданий на перевозку множеством максимальных простых ориентированных путей.

Входные данные: $(Z \cup D), S(Z \cup D)$

Выходные данные: $C \in \mathbf{C}'$ [покрытие вершин максимальными путями]

1: $V' = (Z \cup D)$;

$C = \emptyset$

2: пока $V' \neq \emptyset$ выполнять

3: $p = \mathit{prime}(\mathit{sort}(S(V')))$
 $V' \leftarrow V' \setminus \mathit{vert}(p)$
 $C \leftarrow C \cup \{p\}$

конец цикла

Для заданного множества путей, последовательная реализация разработанных алгоритмов позволяет сформировать покрытие вершин ориентированного графа совместимости заданий на перевозку множеством максимальных по включению путей, причем, согласно Утверждению 3.2.4, такое покрытие будет минимальным по включению, если оно равномощно некоторому минимальному покрытию, что отвечает решению задачи организации железнодорожных перевозок.

В Разделе 4.2 разработан проблемно–ориентированный программный комплекс для решения задачи организации железнодорожных перевозок, последовательно реализующий Алгоритм 5, Алгоритм 6 и Алгоритм 7, и приводятся результаты реализации программного комплекса для заданного бесконфликтного набора нормативных ниток графика движения поездов, и показано, что полученное посредством реализации программного комплекса покрытие вершин графа набором максимальных по включению путей является минимальным.

Глава 4. Проблемно–ориентированные программные комплексы

4.1. Программный комплекс для решения задачи формирования бесконфликтного набора нормативных ниток

Для решения задачи планирования железнодорожных перевозок на этапе формирования множества бесконфликтных наборов нормативных ниток графика движения поездов в Главе 2 разработаны эффективные вычислительные алгоритмы. В рамках теоретико–графового подхода исходная задача сводится к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом конфликтов. В Главе 1 было показано, что решение практической задачи планирования отвечает решению известной NP–полной задаче о максимальном независимом подмножестве. В этой связи разработка эффективных вычислительных алгоритмов формирования множества максимальных верхних нулей, каждый элемент которого соответствует некоторому бесконфликтному набору нормативных ниток, кроме практической значимости в рамках настоящего исследования, представляет самостоятельный интерес как решение классической задачи комбинаторной оптимизации большой размерности.

Проблемно–ориентированный программный комплекс «Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом» разработан на языке Visual Basic и основан на параллельной реализации группы Алгоритмов 1 и 3 и Алгоритма 4 из Главы 2, причём внутри группы Алгоритмы 1 и 3 реализуются последовательно.

Рассмотрим подробнее принципы работы алгоритмов, лежащих в основе программного комплекса для решения задачи планирования, для принятых в Разделе 2.1 обозначений.

На Рис. 4 приводится блок–схема алгоритма формирования множества максимальных верхних нулей. Входными данными для алгоритма формирования максимального верхнего нуля является неориентированный граф конфликтов, заданный списками смежности своих вершин:

$$G = (V_0, E), |V_0| = n,$$

где, по определению, каждой вершине $v_i \in V_0$ соответствует нитка из заданного набора \mathcal{N} нормативных ниток, и вершины связаны ребром только в том случае, если соответствующие нормативные нитки имеют однонаправленный или разнонаправленный конфликт. На первом шаге полагаем, что всем вершинам графа соответствуют нулевые компоненты двоичного набора длины n , что в терминах задачи планирования означает, что бесконфликтный набор нормативных ниток пуст:

$$x = (x_1 = 0, x_2 = 0 \dots, x_n = 0) \implies \Delta\mathcal{N} = \emptyset,$$

где

$$\begin{aligned} \Delta\mathcal{N} &= \{n_1, \dots, n_k\} \subseteq \mathcal{N}: \\ n_i &\longleftrightarrow v_i, v_i \in V_0. \end{aligned}$$

В каждом цикле среди текущего множества вершин графа конфликтов осуществляется поиск вершины, окрестность которой порождает полный индуцированный подграф. Другими словами, в индуцированном подграфе, порождённом текущим множеством вершин графа конфликтов, осуществляется поиск ближайшей k -вершины, где

$$k = |\mathcal{N}(v_i, V_0)|, v_i \in V_0.$$

Как только подходящая вершина найдена, соответствующей компоненте двоичного набора присваивается единичное значение и всем компонентам, соответствующим вершинам в окрестности выбранной вершины присваиваются нулевые значения:

$$\begin{aligned} x &= (x_1 = 0, x_2 = 0 \dots, x_i = 1, \dots, x_n = 0): \\ G\langle \mathcal{N}(v_i, V_0) \rangle &= \left(\mathcal{N}(v_i, V_0), \left(\mathcal{N}^{(v_i, V_0)} \right) \right). \end{aligned}$$

При этом формируется текущий бесконфликтный набор нормативных ниток вида:

$$\Delta\mathcal{N} = \{v_i\},$$

и вершины, для которых определены значения соответствующих компонент двоичного набора, исключаются из множества вершин графа конфликтов, то есть

$$V_0 \longleftarrow V_0 \setminus \{v_i \cup \mathcal{N}(v_i, V_0)\}.$$

Алгоритм продолжает работу до тех пор, пока среди текущего множества вершин могут быть найдены претенденты, соответствующие нормативным ниткам из бесконфликтного набора, то есть такие вершины, для которых выполняется условие полноты индуцированного подграфа, порождённого окрестностью.

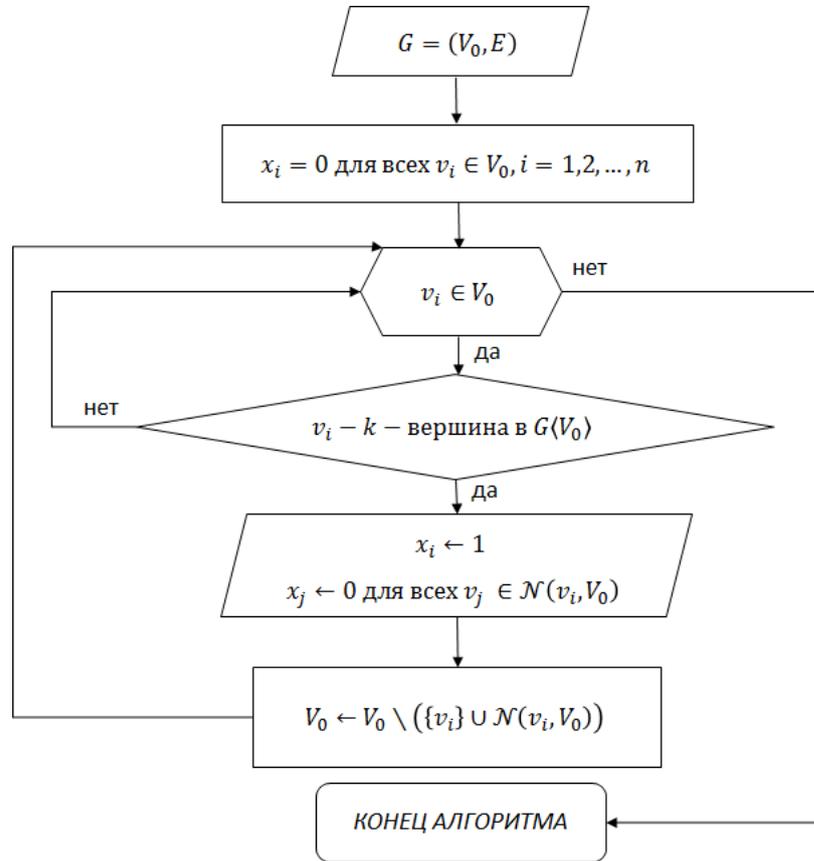


Рис. 4 Блок-схема $A(G, V_0)$

Если при этом текущее множество вершин графа конфликтов пусто, то полученный двоичный набор является элементом множества максимальных верхних нулей монотонной булевой функции, порождённой неориентированным графом конфликтов:

$$x = (x_1, x_2, \dots, x_n) \in \max_{\subseteq} \max_{|\cdot|} F_{f_G}, \quad (4.1.1)$$

и соответствующий бесконфликтный набор нормативных ниток:

$$\Delta \mathcal{N} = \{v_i : x_i = 1\} \quad (4.1.2)$$

отвечает некоторому решению задачи планирования и может служить допустимым расписанием для практической организации железнодорожных перевозок.

Если текущее множество вершин не пусто, то формируется набор вида (4.1.1) по правилу:

$$V_0 \neq \emptyset \implies x' = (x'_1, x'_2, \dots, x'_n) : \quad (4.1.3)$$

$$x'_i = \begin{cases} x'_i, & \text{если } v_i \in V_0, \\ x_i, & \text{иначе,} \end{cases}$$

который является элементом множества максимальных верхних нулей. Таким образом, максимальный верхний нуль функции не определён, однако получен набор, для которого определены некоторые компоненты. При этом размерность задачи снижается, и исследование продолжается на множестве вершин, существенно меньшем по мощности, по сравнению с множеством вершин исходного неориентированного графа конфликтов.

Множество вершин, полученное посредством реализации Алгоритма $A(G, V_0)$ порождает индуцированный подграф, который является входными данными для алгоритма формирования множества верхних нулей монотонной булевой функции, порождённой исходным неориентированным графом конфликтов. Такой подход лежит в основе алгоритма жадного поиска, чем и обусловлена последовательная реализация алгоритмов группы в разработанном программном комплексе для решения задачи планирования железнодорожных перевозок.

На Рис. 5 приводится блок-схема вспомогательного Алгоритма $A_m(G, V_0)$ поиска ближайшей вершины, окрестность которой порождает индуцированный подграф такой, что до полноты в подграфе недостаёт некоторого заданного числа рёбер. В части определения значений компонент двоичного набора и в части формирования текущего множества вершин алгоритм повторяет процедуры Алгоритма $A(G, V_0)$ соответственно.

Индикатор вспомогательного алгоритма характеризует редукцию исходного множества вершин – принимает нулевое значение, если множество вершин, полученное посредством реализации алгоритма совпадает с исходным. При этом все компоненты полученного двоичного набора сохраняют нулевые значения, и бесконфликтному набору нормативных ниток отвечает пустое множество вершин. Важной особенностью вспомогательного алгоритма является необходимое условие не пустоты входного множества вершин, что, в рамках последовательной реализации группы алгоритмов в разработанном программном комплексе, означает, что обращение к вспомогательному алгоритму осуществляется лишь

в том случае, если посредством реализации алгоритма $A(G, V_0)$ сформирован набор вида (4.1.3) и набор из множества максимальных верхних нулей функции не определён.

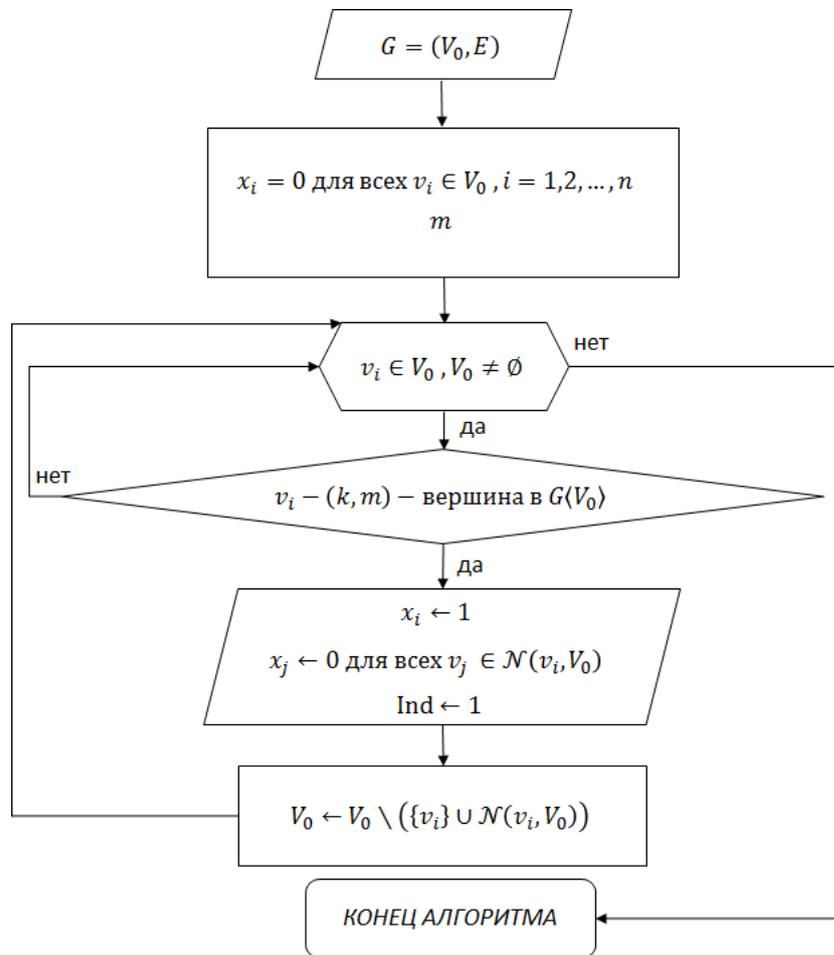


Рис. 5 Блок-схема $A_m(G, V_0)$

Алгоритм $B(G, V_0)$ для формирования множества верхних нулей на каждом шаге обращается к вспомогательному алгоритму. На Рис. 6 приводится блок-схема Алгоритма $B(G, V_0)$. Работа алгоритма продолжается до полного исчерпания множества вершин заданного графа конфликтов. При этом на первом шаге значение m полагается равным нулю, и, таким образом, алгоритм обращается к Алгоритму $A(G, V_0)$, то есть, в случае пустоты множества вершин, полученного посредством реализации вспомогательного алгоритма, формируется набор из множества верхних нулей функции, который является элементом множества максимальных верхних нулей вида (4.1.1), и соответствующее множество вершин графа конфликтов отвечает бесконфликтному набору нормативных ниток графика движения поездов. С точки зрения практической

организации железнодорожных перевозок, для бесконфликтного набора нормативных ниток оказывается достаточным включение некоторого, не всегда наибольшего, подмножества ниток, в количестве, достаточном для исполнения заданного плана перевозок. В постановке задачи расшифровки монотонной булевой функции, порождённой неориентированным графом конфликтов, такому решению отвечают элементы множества верхних нулей. В этой связи разработанные приближённые алгоритмы, лежащие в основе программного комплекса для решения задачи планирования, представляются весьма актуальными.

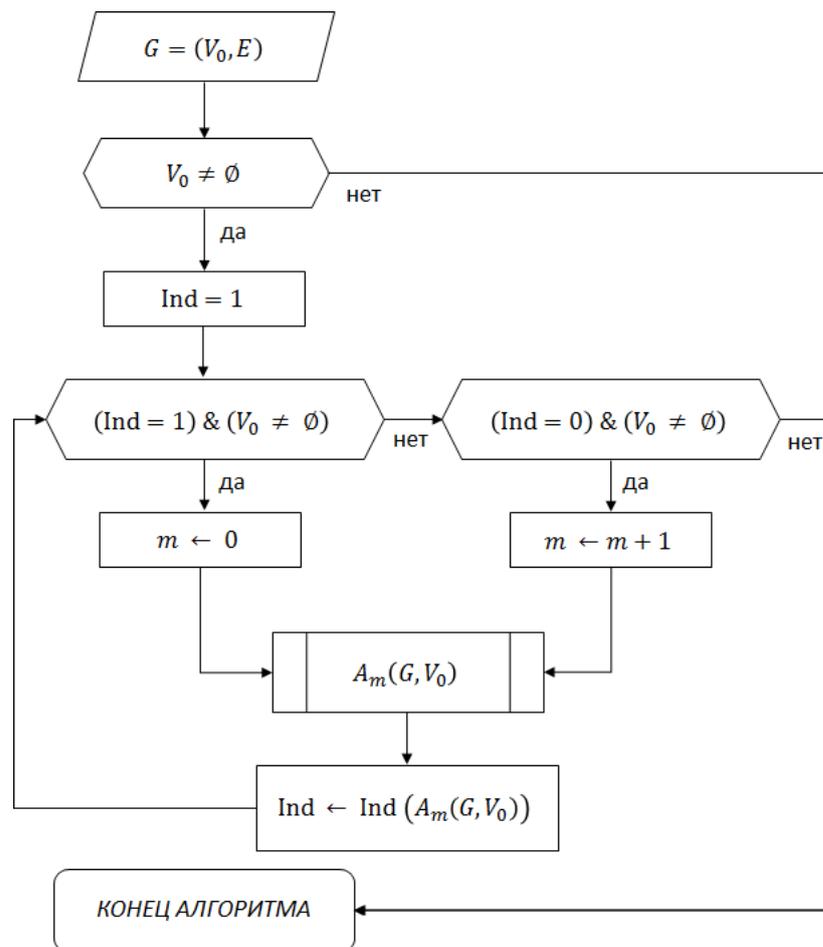


Рис. 6 Блок-схема алгоритма $B(G, V_0)$

Показательным, в части результатов реализации вспомогательного алгоритма, является значение индикатора. Если индикатор принимает нулевое значение, что соответствует непустому текущему множеству вершин, отличному от исходного, то значение m увеличивается на единицу и алгоритм формирования множества верхних нулей начинает очередной цикл работы. В этом случае полученный набор вида (4.1.3) является элементом множества верхних нулей

монотонной булевой функции, порождённой неориентированным графом конфликтом:

$$x = (x_1, x_2, \dots, x_n) \in \max_{\subseteq} F(f_G), \quad (4.1.4)$$

и соответствующее множество вершин графа отвечает бесконфликтному набору нормативных ниток графика движения поездов вида (4.1.2), то есть решению задачи планирования, однако, может быть, не самому лучшему в части количества ниток в наборе.

Алгоритм жадного поиска на каждом шаге выбирает ближайшую подходящую вершину из текущего множества вершин графа, и при этом общее количество итераций сокращается. Алгоритм поиска с возвратом на каждом шаге осуществляет вычисление числовых характеристик для каждой вершины текущего множества. Среди всех вершин выбирается подмножество с наибольшим значением характеристики k , из элементов которого выбирается ближайшая вершина с наименьшим значением характеристики m :

$$\begin{aligned} x_{i_0} &\leftarrow 1: v_{i_0} - (k_{i_0}, m_{i_0})\text{-вершина в графе } G\langle V_0 \rangle, \\ x_j &\leftarrow 0: v_j \in \mathcal{N}(v_{i_0}, V_0), \end{aligned}$$

где

$$\begin{aligned} k_{i_0} &= \max \{k_i: v_i - (k_i, m_i)\text{-вершина в графе } G\langle V_0 \rangle\}, \\ m_{i_0} &= \min \{m_i: v_i - (k_i, m_i)\text{-вершина в графе } G\langle V_0 \rangle\}. \end{aligned}$$

Для полученного решения полезно оценить отклонение числа единиц в наборе от числа единиц в максимальном верхнем нуле функции. В Подразделе 2.2.2 было показано, что в результате реализации алгоритма поиска с возвратом для числа единиц в полученном наборе вида 4.1.4 справедлива оценка отклонения от числа единиц в максимальном верхнем нуле функции:

$$\max_0 f_G - |\text{supp}(x)| \leq m,$$

где

$$m = \sum_{v_i: x_i = 1} m_i.$$

В терминах задачи планирования железнодорожных перевозок такая оценка показывает, насколько меньше нормативных ниток содержится в полученном бесконфликтном наборе, по сравнению с наибольшим возможным количеством.

Таким образом, посредством параллельной реализации разработанных алгоритмов формируется элемент множества максимальных верхних нулей или элемент множества верхних нулей монотонной булевой функции, порождённой неориентированным графом конфликтов, который соответствует некоторому решению задачи планирования железнодорожных перевозок. При этом результат во многом зависит от выбора начальной вершины графа, начиная с которой осуществляется поиск претендента на включение в бесконфликтный набор, и, посредством реализации алгоритмов для каждого возможного выбора, может быть сформировано множество всех максимальных верхних нулей или множество всех верхних нулей функции.

На Рис. 7 Приводится блок-схема проблемно-ориентированного программного комплекса «Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом» для решения задачи планирования железнодорожных перевозок на этапе формирования множества бесконфликтных наборов нормативных ниток графика движения поездов.

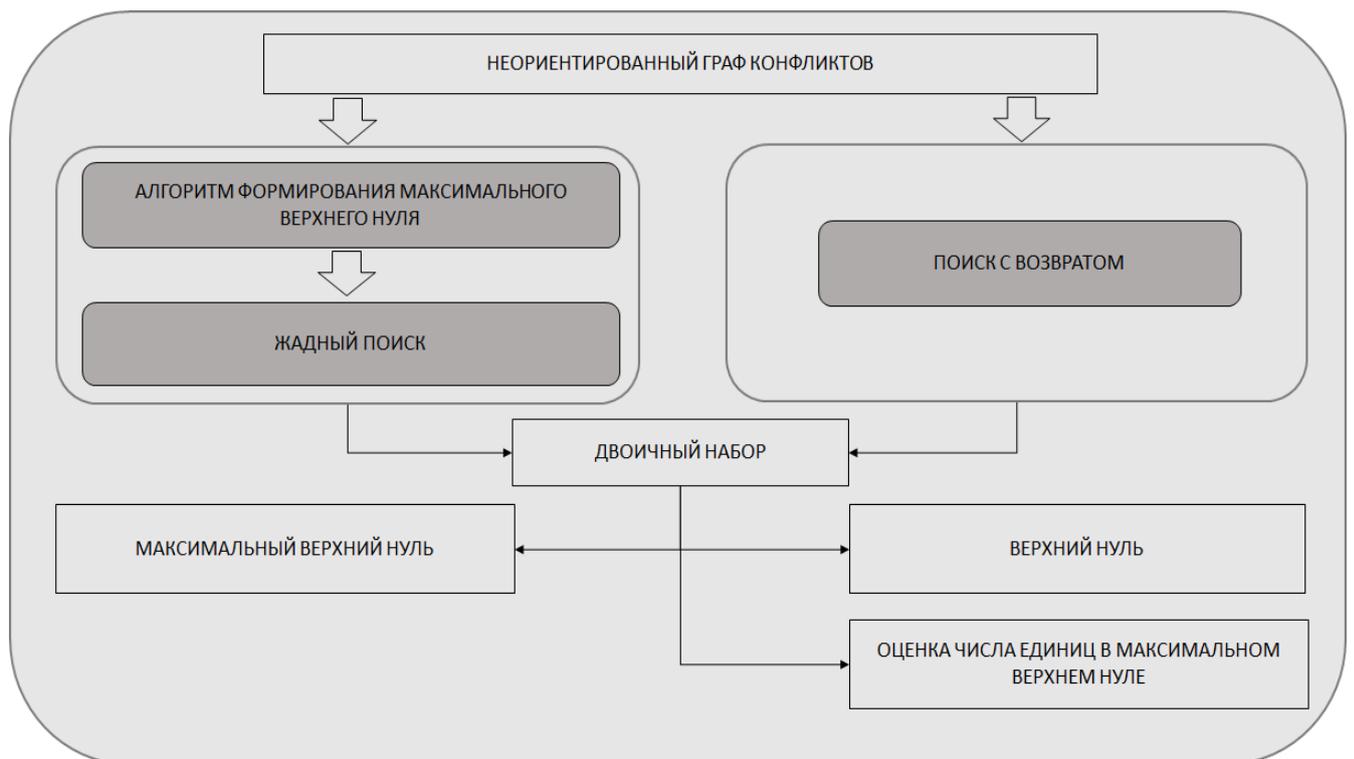


Рис. 7 Программный комплекс «Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом»

Посредством реализации программного комплекса для ряда практических задач были достигнуты масштабные редукции размерности и точные оценки

отклонения мощности полученного решения (количество нормативных ниток в бесконфликтном наборе) от мощности наибольшего бесконфликтного набора. Приведём результаты реализации программного комплекса для решения задачи планирования для заданного неориентированного графа конфликтов. При этом последовательно приведём результаты реализации каждого из параллельных подходов и покажем, что для рассматриваемой задачи посредством реализации программного комплекса получена точная оценка отклонения числа нормативных ниток в выходном наборе от числа нормативных ниток в наибольшем бесконфликтном наборе.

На вход программного комплекса поступает неориентированный граф конфликтов, вершинам которого соответствуют нормативные нитки графика движения поездов,

$$G = (V_0 = \{v_1, v_2, \dots, v_{22}\}, E), \quad (4.1.5)$$

заданный списками смежности своих вершин:

$$\begin{aligned} \mathcal{N}(v_1) &= \{v_2, v_3\}, \\ \mathcal{N}(v_2) &= \{v_1, v_3\}, \\ \mathcal{N}(v_3) &= \{v_1, v_2, v_4, v_9\}, \\ \mathcal{N}(v_4) &= \{v_3, v_5, v_6, v_{11}\}, \\ \mathcal{N}(v_5) &= \{v_4, v_6\}, \\ \mathcal{N}(v_6) &= \{v_4, v_5, v_7, v_{12}, v_{10}\}, \\ \mathcal{N}(v_7) &= \{v_6, v_8\}, \\ \mathcal{N}(v_8) &= \{v_7, v_{12}, v_{16}, v_{17}\}, \\ \mathcal{N}(v_9) &= \{v_3, v_{11}, v_{13}\}, \\ \mathcal{N}(v_{10}) &= \{v_6, v_{12}, v_{11}, v_{14}, v_{15}\}, \\ \mathcal{N}(v_{11}) &= \{v_4, v_{10}, v_9, v_{14}\}, \\ \mathcal{N}(v_{12}) &= \{v_6, v_8, v_{10}, v_{16}\}, \\ \mathcal{N}(v_{13}) &= \{v_9, v_{14}\}, \\ \mathcal{N}(v_{14}) &= \{v_{10}, v_{11}, v_{13}, v_{15}\}, \\ \mathcal{N}(v_{15}) &= \{v_{10}, v_{14}, v_{16}, v_{20}, v_{21}\}, \\ \mathcal{N}(v_{16}) &= \{v_8, v_{12}, v_{15}, v_{17}, v_{19}\}, \\ \mathcal{N}(v_{17}) &= \{v_8, v_{16}, v_{18}, v_{19}\}, \end{aligned}$$

$$\begin{aligned}\mathcal{N}(v_{18}) &= \{v_{17}, v_{19}\}, \\ \mathcal{N}(v_{19}) &= \{v_{16}, v_{17}, v_{18}, v_{20}, v_{21}, v_{22}\}, \\ \mathcal{N}(v_{20}) &= \{v_{15}, v_{19}, v_{21}, v_{22}\}, \\ \mathcal{N}(v_{21}) &= \{v_{15}, v_{19}, v_{20}, v_{22}\}, \\ \mathcal{N}(v_{22}) &= \{v_{19}, v_{20}, v_{21}\}.\end{aligned}$$

В группе алгоритмов для заданного графа сначала реализуется Алгоритм $A(V_0, E)$. Для каждой вершины выполняется проверка условия полноты индуцированного подграфа, порождённого окрестностью вершины, то есть, является ли вершина k -вершиной в подграфе, порождённом текущим множеством вершин.

Результаты работы Алгоритма $A(G, V_0)$:

1. $x_i = 0$ для всех $i = 1, \dots, n$
2. v_1 — 2-вершина в $G\langle V_0 \rangle$

$$\begin{aligned}x_1 &\longleftarrow 1 \\ x_2 &\longleftarrow 0 \\ x_3 &\longleftarrow 0 \\ V_0 &\longleftarrow V_0 \setminus \{v_1, v_2, v_3\}\end{aligned}$$
3. v_4 — не 3-вершина в $G\langle V_0 \rangle$
4. v_5 2-вершина в $G\langle V_0 \rangle$

$$\begin{aligned}x_5 &\longleftarrow 1 \\ x_4 &\longleftarrow 0 \\ x_6 &\longleftarrow 0 \\ V_0 &\longleftarrow V_0 \setminus \{v_4, v_5, v_6\}\end{aligned}$$
5. v_7 — 1-вершина в $G\langle V_0 \rangle$

$$\begin{aligned}x_7 &\longleftarrow 1 \\ x_8 &\longleftarrow 0 \\ V_0 &\longleftarrow V_0 \setminus \{v_7, v_8\}\end{aligned}$$
6. v_9 — не 2-вершина в $G\langle V_0 \rangle$
7. v_{10} — не 4-вершина в $G\langle V_0 \rangle$
8. v_{11} — не 3-вершина в $G\langle V_0 \rangle$
9. v_{12} — не 2-вершина в $G\langle V_0 \rangle$
10. v_{13} — не 2-вершина в $G\langle V_0 \rangle$
11. v_{14} — не 4-вершина в $G\langle V_0 \rangle$

12. v_{15} — не 5-вершина в $G\langle V_0 \rangle$
13. v_{16} — не 4-вершина в $G\langle V_0 \rangle$
14. v_{17} — не 3-вершина в $G\langle V_0 \rangle$
15. v_{18} — 2-вершина в $G\langle V_0 \rangle$
 $x_{18} \longleftarrow 1$
 $x_{17} \longleftarrow 0$
 $x_{19} \longleftarrow 0$
 $V_0 \longleftarrow V_0 \setminus \{v_{17}, v_{18}, v_{19}\}$
16. v_9 — не 2-вершина в $G\langle V_0 \rangle$
17. v_{10} — не 4-вершина в $G\langle V_0 \rangle$
18. v_{11} — не 3-вершина в $G\langle V_0 \rangle$
19. v_{12} — не 2-вершина в $G\langle V_0 \rangle$
20. v_{13} — не 2-вершина в $G\langle V_0 \rangle$
21. v_{14} — не 4-вершина в $G\langle V_0 \rangle$
22. v_{15} — не 5-вершина в $G\langle V_0 \rangle$
23. v_{16} — не 2-вершина в $G\langle V_0 \rangle$
24. v_{20} — не 3-вершина в $G\langle V_0 \rangle$
25. v_{21} — не 3-вершина в $G\langle V_0 \rangle$
26. v_{22} — 2-вершина в $G\langle V_0 \rangle$
 $x_{22} \longleftarrow 1$
 $x_{20} \longleftarrow 0$
 $x_{21} \longleftarrow 0$
 $V_0 \longleftarrow V_0 \setminus \{v_{20}, v_{21}, v_{22}\}$

Полученный двоичный набор является элементом множества нулей монотонной булевой функции, порождённой неориентированным графом конфликтов и имеет вид:

$$x = (1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1) ,$$

$$x \in F(f_G) .$$

При этом формируется набор из множества максимальных верхних нулей функции, для которого определены некоторые компоненты:

$$x' = (1, 0, 0, 0, 1, 0, 1, 0, x'_9, x'_{10}, x'_{11}, x'_{12}, x'_{13}, x'_{14}, x'_{15}, x'_{16}, 0, 1, 0, 0, 0, 1) ,$$

$$x' \in \max_{|\cdot|} \max_{\subseteq} F(f_G) .$$

Текущее множество вершин графа на выходе алгоритма не пусто и имеет вид:

$$V_0 = \{v_9, v_{10}, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}, v_{16}\}, \quad (4.1.6)$$

и, таким образом, имеет место реализация алгоритма жадного поиска для формирования набора из множества верхних нулей функции, входными данными для которого служит граф, порождённый множеством вершин вида (4.1.6).

Результаты работы Алгоритма $B(G, V_0)$:

1. $V_0 \neq \emptyset$

$$\text{Ind} = 1$$

2. $(\text{Ind} = 1) \ \& \ (V_0 \neq \emptyset)$

$$m \leftarrow 0$$

$A_0(G, V_0)$:

$$x_i = 0 \text{ для всех } i = 1, \dots, n$$

$$v_9 \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{10} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{11} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{12} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{13} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{14} \text{ — не 3-вершина в } G\langle V_0 \rangle$$

$$v_{15} \text{ — не 3-вершина в } G\langle V_0 \rangle$$

$$v_{16} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$V_0 \leftarrow V_0$$

$$\text{Ind} \leftarrow 0$$

3. $(\text{Ind} = 0) \ \& \ (V_0 \neq \emptyset)$

$$m \leftarrow m + 1$$

$A_1(G, V_0)$:

$$x_i = 0 \text{ для всех } i = 1, \dots, n$$

$$v_9 \text{ — } (2, 1)\text{-вершина в } G\langle V_0 \rangle$$

$$v_9 \leftarrow 1$$

$$v_{11} \leftarrow 0$$

$$v_{13} \leftarrow 0$$

$$V_0 \leftarrow V_0 \setminus \{v_9, v_{11}, v_{13}\}$$

$$\text{Ind} \leftarrow 1$$

4. $(\text{Ind} = 1) \ \& \ (V_0 \neq \emptyset)$

$$m \longleftarrow 0$$

$A_0(G, V_0)$:

$$x_i = 0 \text{ для всех } i = 1, \dots, n$$

$$v_{10} \text{ — не 3-вершина в } G\langle V_0 \rangle$$

$$v_{12} \text{ — не 2-вершина в } G\langle V_0 \rangle$$

$$v_{14} \text{ — 2-вершина в } G\langle V_0 \rangle$$

$$x_{14} \longleftarrow 1$$

$$x_{10} \longleftarrow 0$$

$$x_{15} \longleftarrow 0$$

$$V_0 \longleftarrow V_0 \setminus \{v_{10}, v_{14}, v_{15}\}$$

$$\text{Ind} \longleftarrow 1$$

5. $(\text{Ind} = 1) \ \& \ (V_0 \neq \emptyset)$

$$m \longleftarrow 0$$

$A_0(G, V_0)$:

$$x_i = 0 \text{ для всех } i = 1, \dots, n$$

$$v_{12} \text{ — 1-вершина в } G\langle V_0 \rangle$$

$$x_{12} \longleftarrow 1$$

$$x_{16} \longleftarrow 0$$

$$V_0 \longleftarrow V_0 \setminus \{v_{12}, v_{16}\}$$

$$\text{Ind} \longleftarrow 1$$

6. $V_0 = \emptyset$

Полученный двоичный набор является элементом множества верхних нулей функции, порождённой исходным графом конфликтов:

$$x' = (1, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1) ,$$

$$x' \in \max_{\subseteq} F(f_G) ,$$

и элементом множества максимальных верхних нулей функции, порождённой графом, полученным из исходного посредством добавления одного ребра, то есть

$$x' \in \max_{|\cdot|} \max_{\subseteq} F(f_{G \cup \{v_{11}, v_{13}\}}) .$$

Таким образом, может быть получена оценка числа единиц в максимальном верхнем нуле функции, порождённой исходным графом конфликтов:

$$\max_0 f_G \leq \max_0 f_{G \cup \{v_{11}, v_{13}\}} + 1 = |\text{supp}(x')| + 1 = 9. \quad (4.1.7)$$

В Таблице 1 представлены результаты последовательной реализации алгоритмов в группе программного комплекса.

Таблица 1

m	0	0	0	0	0	0	1	0	0		x'
Ind	1	1	1	1	1	1	0	1	1		
v_1	1	0	0	0	0	0	0	0	0		1
v_2	1	0	0	0	0	0	0	0	0		0
v_3	1	0	0	0	0	0	0	0	0		0
v_4	1	1	0	0	0	0	0	0	0		0
v_5	1	1	0	0	0	0	0	0	0		1
v_6	1	1	0	0	0	0	0	0	0		0
v_7	1	1	1	0	0	0	0	0	0		1
v_8	1	1	1	0	0	0	0	0	0		0
v_9	1	1	1	1	1	1	0	0	0		1
v_{10}	1	1	1	1	1	1	1	0	0		0
v_{11}	1	1	1	1	1	1	0	0	0		0
v_{12}	1	1	1	1	1	1	1	1	0		1
v_{13}	1	1	1	1	1	1	0	0	0		0
v_{14}	1	1	1	1	1	1	1	0	0		1
v_{15}	1	1	1	1	1	1	1	0	0		0
v_{16}	1	1	1	1	1	1	1	1	0		0
v_{17}	1	1	1	1	0	0	0	0	0		0
v_{18}	1	1	1	1	0	0	0	0	0		1
v_{19}	1	1	1	1	0	0	0	0	0		0
v_{20}	1	1	1	1	1	0	0	0	0		0
v_{21}	1	1	1	1	1	0	0	0	0		0
v_{22}	1	1	1	1	1	0	0	0	0		1

Столбцы соответствуют текущему состоянию множества вершин графа, из которого последовательно исключаются вершины, окрестности которых по-

рождают полный индуцированный подграф на текущем множестве вершин. Соответствующим компонентам двоичного набора, согласно алгоритму, присваиваются единичные и нулевые значения, в результате чего формируется элемент множества верхних нулей монотонной булевой функции, порождённой исходным неориентированным графом. Если при этом индикатор алгоритма не принимает нулевое значение ни на каком шаге, то полученный двоичный набор принадлежит множеству максимальных верхних нулей.

В Таблице 2 представлены результаты реализации параллельного алгоритма программного комплекса: каждый столбец представляет очередную итерацию алгоритма поиска с возвратом, и значения в ячейках таблицы отвечают количественным характеристикам соответствующих вершин (количество вершин в окрестности и количество рёбер, недостающих до полного индуцированного подграфа, порождённого окрестностью) на текущем множестве вершин. Согласно алгоритму, на каждом шаге для вершин с наибольшим значением k и наименьшим значением m соответствующим компонентам двоичного набора присваиваются единичные значения, а всем компонентам соответствующим вершинам в окрестности присваиваются нулевые значения.

Посредством реализации алгоритма поиска с возвратом получен набор из множества верхних нулей монотонной булевой функции, порождённой исходным графом, вида:

$$x' = (1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1) , \quad (4.1.8)$$

$$x' \in \max_{\subseteq} F(f_G) ,$$

для которого также справедлива оценка числа единиц в максимальном верхнем нуле, то есть

$$\max_0 f_G \leq |\text{supp}(x')| + 1 = 10 .$$

Принимая во внимание оценку (4.1.7) можно заключить, что в максимальном верхнем нуле монотонной булевой функции, порождённой неориентированным графом конфликтов (4.1.5), девять компонент имеют единичные значения. Структура разработанного программного комплекса, в рамках которого для заданного графа реализуются различные подходы к расшифровке порождённой булевой функции, позволяет заключить, что двоичный набор (4.1.8), полученный посредством реализации алгоритма поиска с возвратом, в части количества единичных компонент отвечает решению с максимальным числом

В практике решения задач планирования железнодорожных перевозок, довольно часто оказывается достаточным формирование бесконфликтного набора нормативных ниток графика движения поездов некоторой, не максимальной, мощности. Такому решению отвечает некоторый элемент множества верхних нулей монотонной булевой функции, порождённой соответствующим неориентированным графом конфликтов, и, таким образом, представляется актуальным внедрение разработанного программного комплекса в технологическую практику планирования железнодорожных перевозок, поскольку его реализация либо позволяет сформировать набор из множества максимальных верхних нулей, либо получить верхний нуль и оценку числа единиц в максимальном верхнем нуле. На основе полученной оценки может быть построен прогноз качества решения — чем меньше отклонение числа единиц в полученном наборе от числа единиц в максимальном верхнем нуле функции, тем точнее полученное решение, и, можно полагать, что, с практической точки зрения, решение не уступает наилучшему.

4.2. Программный комплекс для решения задачи о назначении и перемещении локомотивов

Для решения задачи организации железнодорожных перевозок на этапе назначения и перемещения локомотивов в Главе 3 разработаны теоретико-множественный алгоритм и алгоритм покрытия вершин ориентированного графа. В основе проблемно-ориентированного программного комплекса для решения задачи организации лежит алгоритм покрытия вершин ориентированного графа совместимости заданий на перевозку минимальным числом максимальных по включению путей. В ходе работы программного комплекса для заданного графа совместимости заданий на перевозку последовательно реализуются алгоритм формирования множества путей, алгоритм формирования множества максимальных путей и алгоритм сортировки путей для покрытия.

Приведём сначала результаты реализации алгоритма назначения и перемещения локомотивов для заданных периода планирования и множества локомотивов, распределённых по депо.

Пусть период планирования состоит из трёх суток, каждые из которых разбиты на трёхчасовые интервалы:

$$I_1, I_2, I_3, \dots, I_{23}, I_{24},$$

и локомотивы в количестве десяти штук распределены по депо A и B равномерно, то есть

$$\mathcal{L} = \{L_1, \dots, L_{10}\} = D_A \cup D_B,$$

и доступны для назначения в любое время на протяжении неограниченного срока в течении периода планирования. В таких условиях ограничения на использование и необходимость технического обслуживания локомотивов не учитываются.

В бесконфликтном наборе нормативных ниток каждая нитка задана набором параметров, характеризующих станции начала и окончания движения, и время начала и окончания движения, причём время представлено парой абсолютного значения и номера суток в периоде планирования:

$$\Delta\mathcal{N} = \left\{ n = (s_H(n), (t_H(n), r), s_K(n), (t_K(n), r)) \right\}, \quad (4.2.1)$$

$$s_H, s_K \in \{A, B\}, r = 1, 2, 3.$$

Представим бесконфликтный набор в виде последовательности нормативных ниток, упорядоченной лексикографически относительно времени начала движения:

$$\begin{aligned} n_1 &= (B, (1,1), A, (6,1)), \\ n_2 &= (A, (1,1), B, (10,1)), \\ n_3 &= (B, (2,1), A, (10,1)), \\ n_4 &= (A, (3,1), B, (13,1)), \\ n_5 &= (B, (7,1), A, (13,1)), \\ n_6 &= (A, (7,1), B, (16,1)), \\ n_7 &= (A, (8,1), B, (19,1)), \\ n_8 &= (A, (11,1), B, (21,1)), \end{aligned}$$

$$\begin{aligned} n_9 &= \left(B, (11,1), A, (21,1) \right), \\ n_{10} &= \left(A, (12,1), B, (24,1) \right), \end{aligned}$$

$$n_{10 \cdot (r-1) + i} = \left(s_H(n_i), (t_H(n_i), r), s_K(n_i), (t_K(n_i), r) \right) : i = 1, \dots, 10, r = 2, 3.$$

В реализации теоретико-множественного алгоритма назначения и перемещения локомотивов важным объектом исследования является ориентированный граф зависимостей ниток. Вершины графа (3.2.1) однозначно соответствуют нормативным ниткам из бесконфликтного набора и дугами соединены вершины, соответствующие нормативные нитки для которых, при заданных нормативном времени и рекомендованном значении параметра, удовлетворяют условиям совместимости.

Положим нормативное время, необходимое для подготовки локомотива после исполнения некоторой нитки, и рекомендованное значение параметра для графа зависимостей ниток соответственно:

$$\begin{aligned} \Delta &= 1, \\ r_{rec} &= \left\lfloor \frac{\sum_{i=1}^{30} |n_i|}{24 \cdot 30} \right\rfloor = 1. \end{aligned}$$

Таким образом, для заданных периода планирования, множества локомотивов, бесконфликтного набора нормативных ниток графика движения поездов, нормативного времени и рекомендованного параметра, ориентированный граф зависимостей ниток (3.2.1) может быть задан списками смежности своих вершин:

$$\begin{aligned} (n_1 : n_6, n_7, n_8, n_{10}, n_{12}, n_{14}, n_{16}, n_{17}, n_{18}, n_{20}) &, \\ (n_2 : n_9, n_{11}, n_{13}, n_{15}, n_{19}) &, \\ (n_3 : n_8, n_{10}, n_{12}, n_{14}, n_{16}, n_{17}, n_{18}, n_{20}) &, \\ (n_4 : n_{11}, n_{13}, n_{15}, n_{19}) &, \\ (n_5 : n_{12}, n_{14}, n_{16}, n_{17}, n_{18}, n_{20}) &, \\ (n_6 : n_{11}, n_{13}, n_{15}, n_{19}) &, \\ (n_7 : n_{11}, n_{13}, n_{15}, n_{19}) &, \\ (n_8 : n_{11}, n_{13}, n_{15}, n_{19}) &, \end{aligned}$$

$$\begin{aligned}
& (n_9 : n_{12}, n_{14}, n_{16}, n_{17}, n_{18}, n_{20}) , \\
& (n_{10} : n_{11}, n_{13}, n_{15}, n_{19}) , \\
& (n_{11} : n_{16}, n_{17}, n_{18}, n_{20}, n_{22}, n_{24}, n_{26}, n_{27}, n_{28}, n_{30}) , \\
& (n_{12} : n_{19}, n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{13} : n_{18}, n_{20}, n_{22}, n_{24}, n_{26}, n_{27}, n_{28}, n_{30}) , \\
& (n_{14} : n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{15} : n_{22}, n_{24}, n_{26}, n_{27}, n_{28}, n_{30}) , \\
& (n_{16} : n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{17} : n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{18} : n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{19} : n_{22}, n_{24}, n_{26}, n_{27}, n_{28}, n_{30}) , \\
& (n_{20} : n_{21}, n_{23}, n_{25}, n_{29}) , \\
& (n_{21} : n_{26}, n_{27}, n_{28}, n_{30}) , \\
& (n_{22} : n_{29}) , \\
& (n_{23} : n_{28}, n_{30}) .
\end{aligned}$$

На каждом шаге алгоритм назначения и перемещения локомотивов выбирает вершину ориентированного графа зависимостей ниток, соответствующую нормативной нитке, для исполнения которой назначается некоторый локомотив. В окрестности первого порядка выбранной вершины осуществляется поиск нитки, для исполнения которой может быть назначен тот же локомотив. Если в окрестности первого порядка подходящей нитки не найдётся, поиск продолжается в окрестности второго и последующих порядков, и, в случае прохождения всего графа и отсутствия искомым ниток, шаг алгоритма завершается. Сформированная в результате последовательность вершин графа соответствует набору нормативных ниток, исполнение которых осуществляется одним локомотивом. Выбирается новый локомотив и начинается очередной шаг алгоритма до тех пор, пока для всех нормативных ниток из бесконфликтного набора не будет назначен локомотив.

Результат работы алгоритма назначения и перемещения локомотивов:

$$1. 1) NL = 1$$

$$\text{mark}(n_i) = 0 \text{ для всех } i = 1, \dots, 30$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_1]

$$2) i_0 = 1$$

$$P \longleftarrow n_1$$

$$\text{mark}(n_1) = 1$$

$$3) \Gamma_{I_k}(n_1) \neq \emptyset$$

$$k_0 = 3$$

$$\Gamma_{I_3}^0(n_1) \neq \emptyset$$

$$j_0 = 6$$

$$P \longleftarrow P \cup n_6$$

$$i_0 = 6$$

$$\text{mark}(n_6) = 1$$

$$3) \Gamma_{I_k}(n_6) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_6) \neq \emptyset$$

$$j_0 = 11$$

$$P \longleftarrow P \cup n_{11}$$

$$i_0 = 11$$

$$\text{mark}(n_{11}) = 1$$

$$3) \Gamma_{I_k}(n_{11}) \neq \emptyset$$

$$k_0 = 11$$

$$\Gamma_{I_{11}}^0(n_{11}) \neq \emptyset$$

$$j_0 = 16$$

$$P \longleftarrow P \cup n_{16}$$

$$i_0 = 16$$

$$\text{mark}(n_{16}) = 1$$

$$3) \Gamma_{I_k}(n_{16}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{16}) \neq \emptyset$$

$$j_0 = 21$$

$$P \longleftarrow P \cup n_{21}$$

$$i_0 = 21$$

$$\text{mark}(n_{21}) = 1$$

$$3) \Gamma_{I_k}(n_{21}) \neq \emptyset$$

$$k_0 = 19$$

$$\Gamma_{I_{19}}^0(n_{21}) \neq \emptyset$$

$$j_0 = 26$$

$$P \longleftarrow P \cup n_{26}$$

$$i_0 = 26$$

$$\text{mark}(n_{26}) = 1$$

$$3) \Gamma_{I_k}(n_{26}) = \emptyset$$

$L_1 \longleftarrow n_1 \cup n_6 \cup n_{11} \cup n_{16} \cup n_{21} \cup n_{26}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_1]

$$NL = 2$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_2]

$$2. 2) i_0 = 2$$

$$P \longleftarrow n_2$$

$$\text{mark}(n_2) = 1$$

$$3) \Gamma_{I_k}(n_2) \neq \emptyset$$

$$k_0 = 4$$

$$\Gamma_{I_4}^0(n_2) \neq \emptyset$$

$$j_0 = 9$$

$$P \longleftarrow P \cup n_9$$

$$i_0 = 3$$

$$\text{mark}(n_9) = 1$$

$$3) \Gamma_{I_k}(n_9) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_9) \neq \emptyset$$

$$j_0 = 12$$

$$P \longleftarrow P \cup n_{12}$$

$$i_0 = 12$$

$$\text{mark}(n_{12}) = 1$$

$$3) \Gamma_{I_k}(n_{12}) \neq \emptyset$$

$$k_0 = 12$$

$$\Gamma_{I_{12}}^0(n_{12}) \neq \emptyset$$

$$j_0 = 19$$

$$P \longleftarrow P \cup n_{19}$$

$$i_0 = 19$$

$$\text{mark}(n_{19}) = 1$$

$$3) \Gamma_{I_k}(\mathbf{n}_{19}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(\mathbf{n}_{19}) \neq \emptyset$$

$$j_0 = 22$$

$$P \longleftarrow P \cup \mathbf{n}_{22}$$

$$i_0 = 22$$

$$\text{mark}(\mathbf{n}_{22}) = 1$$

$$3) \Gamma_{I_k}(\mathbf{n}_{22}) \neq \emptyset$$

$$k_0 = 20$$

$$\Gamma_{I_{20}}^0(\mathbf{n}_{22}) \neq \emptyset$$

$$j_0 = 29$$

$$P \longleftarrow P \cup \mathbf{n}_{29}$$

$$i_0 = 29$$

$$\text{mark}(\mathbf{n}_{29}) = 1$$

$$3) \Gamma_{I_k}(\mathbf{n}_{29}) = \emptyset$$

$L_2 \longleftarrow \mathbf{n}_2 \cup \mathbf{n}_9 \cup \mathbf{n}_{12} \cup \mathbf{n}_{19} \cup \mathbf{n}_{22} \cup \mathbf{n}_{29}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_2]

$$NL = 3$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_3]

$$3. 2) i_0 = 3$$

$$P \longleftarrow \mathbf{n}_3$$

$$\text{mark}(\mathbf{n}_3) = 1$$

$$3) \Gamma_{I_k}(\mathbf{n}_3) \neq \emptyset$$

$$k_0 = 4$$

$$\Gamma_{I_4}^0(\mathbf{n}_3) \neq \emptyset$$

$$j_0 = 8$$

$$P \longleftarrow P \cup \mathbf{n}_8$$

$$i_0 = 8$$

$$\text{mark}(\mathbf{n}_8) = 1$$

$$3) \Gamma_{I_k}(\mathbf{n}_8) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(\mathbf{n}_8) \neq \emptyset$$

$$j_0 = 13$$

$$P \longleftarrow P \cup \mathbf{n}_{13}$$

$$i_0 = 13$$

$$\text{mark}(n_{13}) = 1$$

$$3) \Gamma_{I_k}(n_{13}) \neq \emptyset$$

$$k_0 = 12$$

$$\Gamma_{I_{12}}^0(n_{13}) \neq \emptyset$$

$$j_0 = 18$$

$$P \longleftarrow P \cup n_{18}$$

$$i_0 = 18$$

$$\text{mark}(n_{18}) = 1$$

$$3) \Gamma_{I_k}(n_{18}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{18}) \neq \emptyset$$

$$j_0 = 23$$

$$P \longleftarrow P \cup n_{23}$$

$$i_0 = 23$$

$$\text{mark}(n_{23}) = 1$$

$$3) \Gamma_{I_k}(n_{23}) \neq \emptyset$$

$$k_0 = 20$$

$$\Gamma_{I_{20}}^0(n_{23}) \neq \emptyset$$

$$j_0 = 28$$

$$P \longleftarrow P \cup n_{28}$$

$$i_0 = 28$$

$$\text{mark}(n_{28}) = 1$$

$$3) \Gamma_{I_k}(n_{28}) = \emptyset$$

$L_3 \longleftarrow n_3 \cup n_8 \cup n_{13} \cup n_{18} \cup n_{23} \cup n_{28}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_3]

$$NL = 4$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_4]

$$4. 2) i_0 = 4$$

$$P \longleftarrow n_4$$

$$\text{mark}(n_4) = 1$$

$$3) \Gamma_{I_k}(n_4) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_4) = \emptyset$$

$$\mathcal{C}(n_4, n_{15}) = 1$$

$$\mathcal{C}(n_{11}, n_{17}) = 1$$

$$\mathcal{C}(n_{13}, n_{20}) = 1$$

$$k_0 = 11$$

$$j_0 = 15$$

$$P \longleftarrow P \cup n_{15}$$

$$i_0 = 15$$

$$\text{mark}(n_{15}) = 1$$

$$3) \Gamma_{I_k}(n_{15}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{15}) \neq \emptyset$$

$$j_0 = 24$$

$$P \longleftarrow P \cup n_{15}$$

$$i_0 = 24$$

$$\text{mark}(n_{24}) = 1$$

$$3) \Gamma_{I_k}(n_{24}) = \emptyset$$

$L_4 \longleftarrow n_4 \cup n_{15} \cup n_{24}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_4]

$$NL = 5$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_5]

$$5. 2) i_0 = 5$$

$$P \longleftarrow n_5$$

$$\text{mark}(n_5) = 1$$

$$3) \Gamma_{I_k}(n_5) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_5) \neq \emptyset$$

$$j_0 = 14$$

$$P \longleftarrow P \cup n_{14}$$

$$i_0 = 14$$

$$\text{mark}(n_{14}) = 1$$

$$3) \Gamma_{I_k}(n_{14}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{14}) = \emptyset$$

$$\mathcal{C}(n_{14}, n_{25}) = 1$$

$$\mathcal{C}(n_{21}, n_{27}) = 1$$

$$\mathcal{C}(n_{23}, n_{30}) = 1$$

$$k_0 = 19$$

$$j_0 = 25$$

$$P \longleftarrow P \cup n_{25}$$

$$i_0 = 25$$

$$\text{mark}(n_{25}) = 1$$

$$3) \Gamma_{I_k}(n_{25}) = \emptyset$$

$L_5 \longleftarrow n_5 \cup n_{14} \cup n_{25}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_5]

$$NL = 6$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_6]

$$6. 2) i_0 = 7$$

$$P \longleftarrow n_7$$

$$\text{mark}(n_7) = 1$$

$$3) \Gamma_{I_k}(n_7) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_7) = \emptyset$$

$$\mathcal{C}(n_7, n_j) = -1 \text{ для всех } j = 1, \dots, 30$$

$$\mathcal{C}(n_{11}, n_{17}) = 1$$

$$\mathcal{C}(n_{13}, n_{20}) = 1$$

$$k_0 = 11$$

$$j_0 = 17$$

$$P \longleftarrow P \cup n_{11} \cup n_{17}$$

$$i_0 = 17$$

$$\text{mark}(n_{17}) = 1$$

$$3) \Gamma_{I_k}(n_{17}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{17}) = \emptyset$$

$$\mathcal{C}(n_{17}, n_j) = -1 \text{ для всех } j = 1, \dots, 30$$

$$\mathcal{C}(n_{21}, n_{27}) = 1$$

$$\mathcal{C}(n_{23}, n_{30}) = 1$$

$$k_0 = 19$$

$$j_0 = 27$$

$$P \longleftarrow P \cup n_{21} \cup n_{27}$$

$$i_0 = 27$$

$$\text{mark}(n_{27}) = 1$$

$$3) \Gamma_{I_k}(n_{27}) = \emptyset$$

$L_6 \longleftarrow n_7 \cup n_{11} \cup n_{17} \cup n_{21} \cup n_{27}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_6]

$$NL = 7$$

$P \longleftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_7]

$$7. 2) i_0 = 10$$

$$P \longleftarrow n_{10}$$

$$\text{mark}(n_{10}) = 1$$

$$3) \Gamma_{I_k}(n_{10}) \neq \emptyset$$

$$k_0 = 9$$

$$\Gamma_{I_9}^0(n_{10}) = \emptyset$$

$$\mathcal{C}(n_{10}, n_j) = -1 \text{ для всех } j = 1, \dots, 30$$

$$\mathcal{C}(n_{11}, n_{20}) = 1$$

$$\mathcal{C}(n_{13}, n_{20}) = 1$$

$$k_0 = 12$$

$$j_0 = 20$$

$$P \longleftarrow P \cup n_{11} \cup n_{20}$$

$$i_0 = 20$$

$$\text{mark}(n_{20}) = 1$$

$$3) \Gamma_{I_k}(n_{20}) \neq \emptyset$$

$$k_0 = 17$$

$$\Gamma_{I_{17}}^0(n_{20}) = \emptyset$$

$$\mathcal{C}(n_{20}, n_j) = -1 \text{ для всех } j = 1, \dots, 30$$

$$\mathcal{C}(n_{21}, n_{30}) = 1$$

$$\mathcal{C}(n_{23}, n_{30}) = 1$$

$$k_0 = 20$$

$$j_0 = 30$$

$$P \longleftarrow P \cup n_{21} \cup n_{30}$$

$$i_0 = 30$$

$$\text{mark}(n_{30}) = 1$$

$$3) \Gamma_{I_k}(n_{30}) = \emptyset$$

$L_7 \leftarrow n_{10} \cup n_{11} \cup n_{20} \cup n_{21} \cup n_{30}$ [завершается сбор ниток, для исполнения которых назначен локомотив L_7]

$$NL = 8$$

$P \leftarrow \emptyset$ [множество ниток, для исполнения которых назначен локомотив L_8]

8. 2) $\text{mark}(n_i) = 1$ для всех $i = 1, \dots, 30$

Конец алгоритма

Таким образом, план перевозок, заданный бесконфликтным набором нормативных ниток $\Delta\mathcal{N}$, может быть исполнен посредством локомотивов L_1, \dots, L_7 .

В рамках теоретико-графового подхода, разработанного в Разделе 3.2, задача организации железнодорожных перевозок сводится к задаче покрытия вершин ориентированного графа совместимости заданий на перевозку. Проблемно-ориентированный программный комплекс «Алгоритм покрытия вершин ориентированного графа минимальным числом максимальных по включению путей» для решения задачи организации разработан на языке Visual Basic и основан на последовательной реализации Алгоритма 5, Алгоритма 6 и Алгоритма 7 из Главы 3. Рассмотрим подробнее принципы реализации разработанного программного комплекса.

На Рис. 8 приводится блок-схема циклического Алгоритма 5 с ветвлением для формирования множества путей ориентированного графа. На вход алгоритма поступает ориентированный граф совместимости заданий на перевозку вида:

$$\vec{\mathbb{G}} = \left((Z \cup D), \mathbb{E} \right),$$

где множество вершин взаимно однозначно соответствует множеству всевозможных перемещений. В Разделе 3.2 приняты обозначения

$$\mathbb{P} = \left\{ p = (v_{i_1}, (v_{i_1}, v_{i_2}), \dots, (v_{i_{k-1}}, v_{i_k}), v_{i_k}) : \right. \\ \left. v_{i_1}, \dots, v_{i_k} \in (Z \cup D), (v_{i-1}, v_i) \in \mathbb{E} \right\},$$

$$\text{last}(p) \in (Z \cup D).$$

соответственно для множества путей и для вершины, входящей в последовательность вершин и дуг последней по порядку.

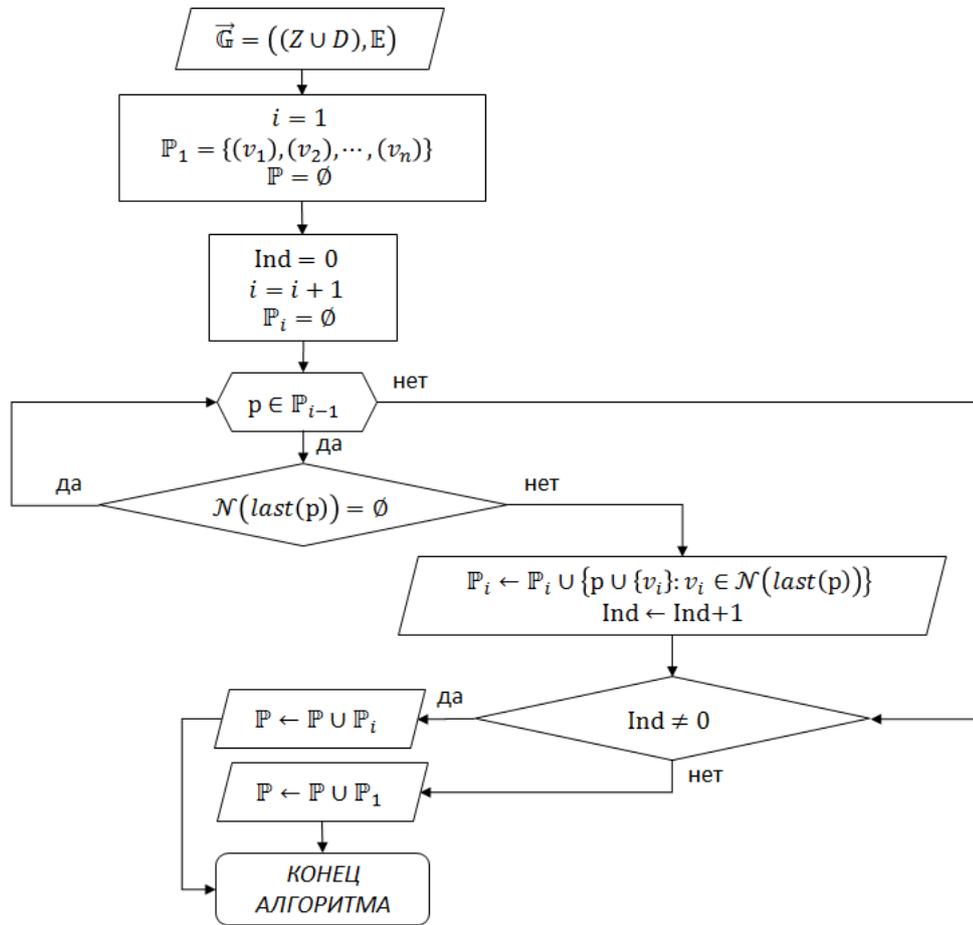


Рис. 8 Алгоритм формирования множества путей ориентированного графа

Начальное множество содержит одноэлементные пути, каждый из которых соответствует некоторой вершине, и на каждом шаге для каждого элемента из текущего множества путей выполняется поиск инцидентных вершин для вершины, входящей в рассматриваемый путь последней по порядку. Каждая смежная вершина, в совокупности с рассматриваемым элементом из текущего множества путей, образует путь в ориентированном графе. Алгоритм продолжает работу до тех пор, пока в текущем множестве путей для каждого элемента не будут исчерпаны все смежные вершины в графе.

Полученное множество путей графа совместимости заданий на перевозку является входным для алгоритма формирования множества максимальных путей, в этой связи в программном комплексе реализуется принцип последовательного исполнения разработанных алгоритмов. На Рис. 9 приводится блок-схема циклического Алгоритма 6 с ветвлением для формирования множества максимальных путей.

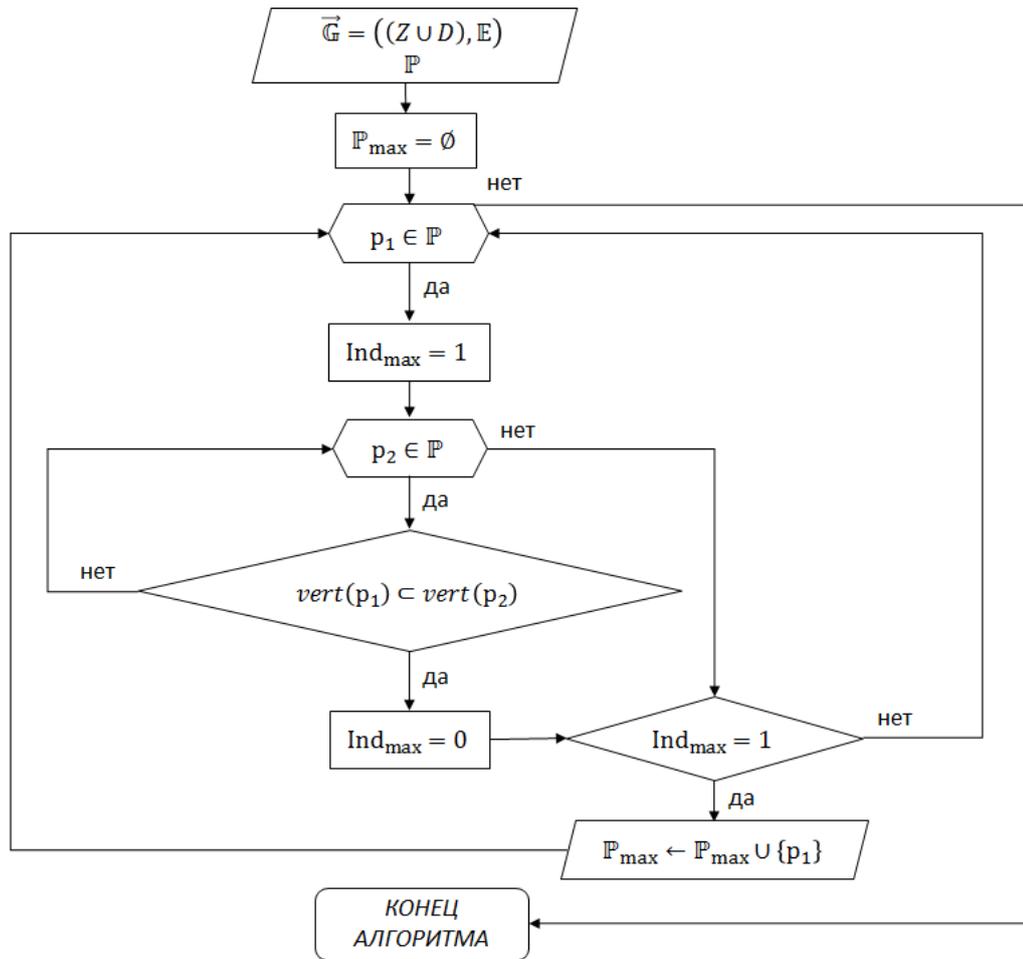


Рис. 9 Алгоритм формирования множества максимальных по включению путей ориентированного графа

Для каждого фиксированного элемента из множества путей графа вводится переменная индикатора максимальности, значение которой принимается равным единице, если множество вершин, входящих в последовательность фиксированного пути, включает множество вершин, входящих в последовательность некоторого рассматриваемого пути:

$$vert(p_1) \subset vert(p_2) .$$

На основании Утверждения 3.2.2 и Следствия 3.2.4 из Раздела 3.2 можно полагать, что покрытие вершин ориентированного графа множеством максимальных путей, равномощное некоторому минимальному покрытию, также является минимальным. Таким образом, полученное в результате реализации алгоритма, множество максимальных путей является входным для алгоритма покрытия. Для каждого элемента множества максимальных по включению путей в рассмотрение семейство индексов вида (3.2.2)–(3.2.5) на подмно-

жестве вершин:

$$p: \left\{ ind_1(v), ind_2(p, V'), ind_3(p, V'), ind_4(p, V') \right\},$$

где

$$p \in \mathbb{P}_{\max}, v \in (Z \cup D), V' \subseteq (Z \cup D),$$

и обобщённый индекс

$$Ind(p, V') = \left(ind_2(p, V'), ind_3(p, V'), ind_4(p, V') \right).$$

На Рис. 10 приводится блок-схема циклического Алгоритма 7 для сортировки множества максимальных по включению путей относительно семейства обобщённых индексов его элементов.

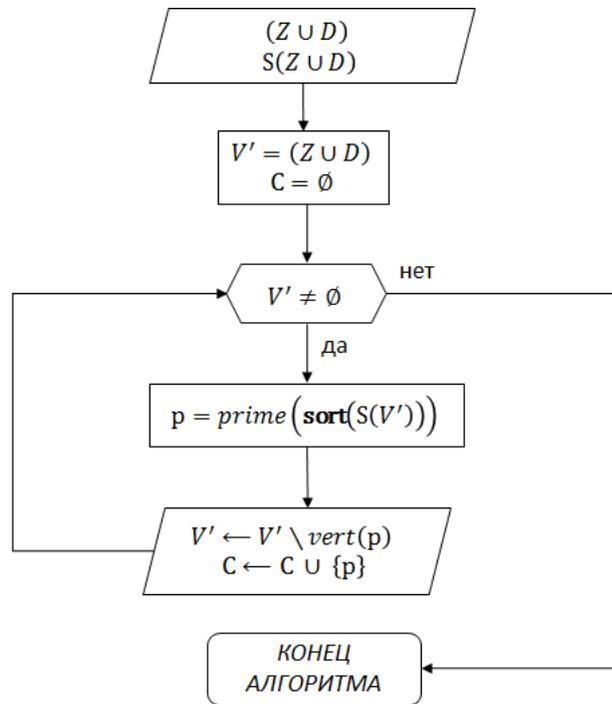


Рис. 10 Алгоритм сортировки множества максимальных по включению путей

Каждый максимальный по включению путь рассматривается в совокупности с соответствующим обобщённым индексом на текущем подмножестве вершин:

$$S(V') = \left\{ (Ind(p, V'), p) : V' \cup vert(p) \neq \emptyset \right\},$$

причём на входе алгоритма текущее множество вершин отвечает множеству вершин графа совместимости заданий на перевозку, то есть включает вершины, соответствующие всем перемещениям локомотивов из множества всевозможных перемещений.

В результате операции сортировки формируется первый в упорядоченном семействе элемент, обозначаемый как

$$prime\left(\text{sort}(S(V'))\right),$$

который выбирается для покрытия вершин графа совместимости заданий на перевозку. На каждом шаге алгоритма все вершины, входящие в выбранный для покрытия элемент, исключаются из текущего подмножества, и алгоритм продолжает работу до полного исчерпания множества вершин.

На Рис. 11 приводится блок-схема проблемно-ориентированного программного комплекса «Алгоритм покрытия вершин ориентированного графа минимальным числом максимальных по включению путей» для решения задачи организации железнодорожных перевозок.

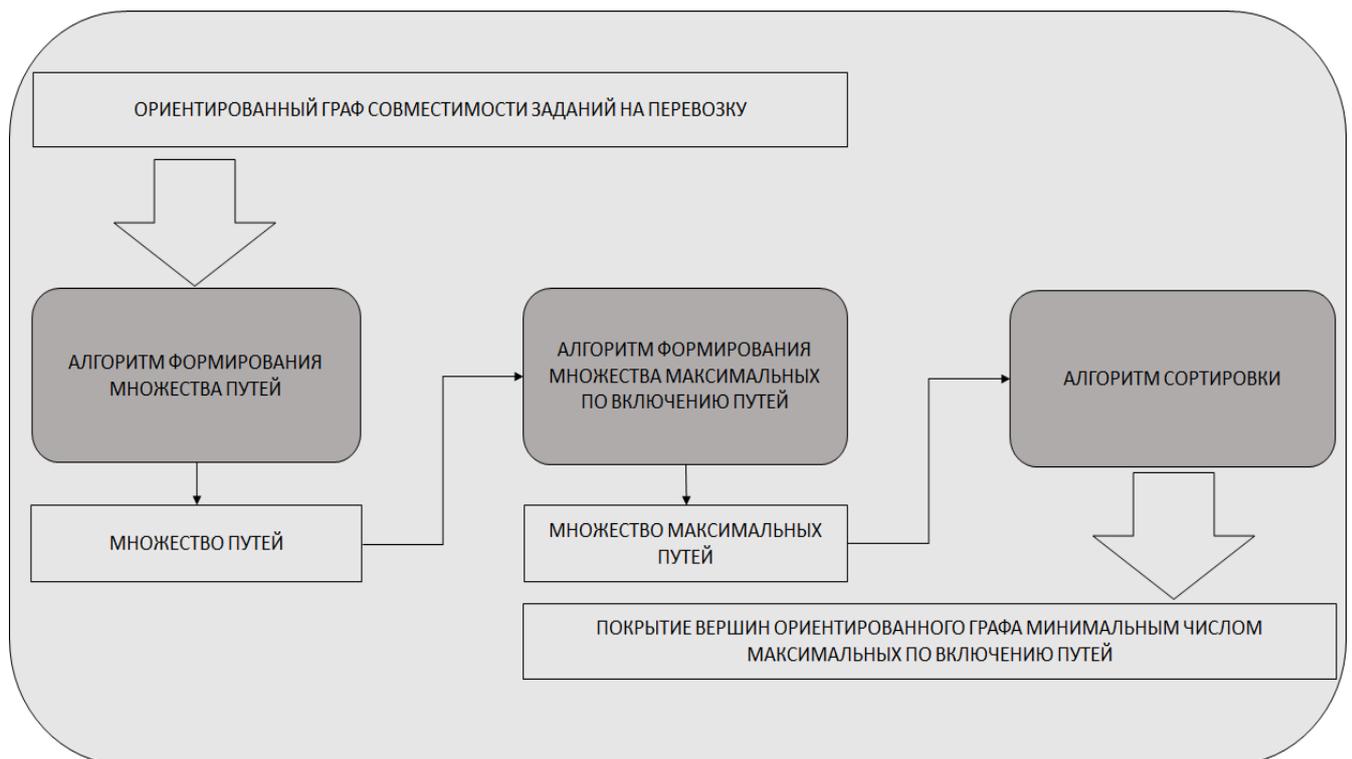


Рис. 11 Программный комплекс «Алгоритм покрытия вершин ориентированного графа минимальным числом максимальных по включению путей»

Приведём результаты реализации программного комплекса для решения задачи организации для заданных периода планирования $K = 3$ и нормативного времени, необходимого для подготовки локомотива после исполнения неко-

торой нитки $\Delta = 1$ и множества всевозможных перемещений локомотивов вида (4.2.1), то есть

$$\Delta \mathcal{N} = \mathcal{N}(\mathcal{P}, \mathcal{R}) = \mathcal{N}(\mathcal{P}, \mathcal{R}^*) .$$

Соответствующий ориентированный граф совместимости заданий на перевозку

$$\begin{aligned} \vec{\mathbb{G}} &= \left((Z \cup D), \mathbb{E} \right) , \\ (Z \cup D) &= \mathcal{N}(\mathcal{P}, \mathcal{R}) \cup \mathcal{N}(\mathcal{P}, \mathcal{R}^*) , \end{aligned}$$

может быть задан списками смежности своих вершин:

$$(v_1 : v_6, v_7, v_8, v_{10}, v_{12}, v_{14}, v_{16}, v_{17}, v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_2 : v_9, v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_3 : v_8, v_{10}, v_{12}, v_{14}, v_{16}, v_{17}, v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_4 : v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_5 : v_{12}, v_{14}, v_{16}, v_{17}, v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_6 : v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_7 : v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_8 : v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_9 : v_{12}, v_{14}, v_{16}, v_{17}, v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{10} : v_{11}, v_{13}, v_{15}, v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{11} : v_{16}, v_{17}, v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{12} : v_{19}, v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{13} : v_{18}, v_{20}, v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{14} : v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{15} : v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{16} : v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{17} : v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{18} : v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{19} : v_{22}, v_{24}, v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{20} : v_{21}, v_{23}, v_{25}, v_{29}) ;$$

$$(v_{21} : v_{26}, v_{27}, v_{28}, v_{30}) ;$$

$$(v_{22} : v_{29}) ;$$

$$(v_{23} : v_{28}, v_{30}) ;$$

$$(v_{24}) ;$$

$$(v_{25}) ;$$

$$(v_{26}) ;$$

$$(v_{27}) ;$$

$$(v_{28}) ;$$

$$(v_{29}) ;$$

$$(v_{30}) ;$$

На вход программного комплекса поступает ориентированный граф совместимости заданий на перевозку, и, действуя последовательно по алгоритмам комплекса, формируется множество максимальных по включению путей, и для каждого элемента формируется обобщенный индекс пути на множестве вершин графа. Семейство максимальных по включению путей и соответствующих значений обобщенного индекса поступает на вход алгоритма сортировки.

В результате работы алгоритма сортировки получены результаты:

$$1. 1) V' = (Z \cup D) ;$$

$$C = \emptyset ;$$

$$2) V' = V :$$

$$3) p_1 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) = \\ = \left((6, 550, 53) , (v_2, v_9, v_{12}, v_{19}, v_{22}, v_{29}) \right) ;$$

$$V' \leftarrow V' \setminus \{v_2, v_9, v_{12}, v_{19}, v_{22}, v_{29}\} ;$$

$$C \leftarrow \{p_1\} ;$$

$$2. 2) V' = V \setminus \{v_2, v_9, v_{12}, v_{19}, v_{22}, v_{29}\} :$$

$$3) p_2 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) = \\ = \left((6, 1325, 190) , (v_3, v_8, v_{13}, v_{18}, v_{23}, v_{28}) \right) ;$$

$$V' \leftarrow V' \setminus \{v_3, v_8, v_{13}, v_{18}, v_{23}, v_{28}\} ;$$

$$C \leftarrow \{p_1, p_2\} ;$$

3. 2) $V' = V \setminus \{v_2, v_3, v_8, v_9, v_{12}, v_{13}, v_{18}, v_{19}, v_{22}, v_{23}, v_{28}, v_{29}\}$;
 3) $p_3 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) =$
 $= \left((6, 1535, 102), (v_1, v_7, v_{11}, v_{17}, v_{21}, v_{27}) \right)$;
 $V' \leftarrow V' \setminus \{v_1, v_7, v_{11}, v_{17}, v_{21}, v_{27}\}$;
 $C \leftarrow \{p_1, p_2, p_3\}$;
4. 2) $V' = \{v_4, v_5, v_6, v_{10}, v_{14}, v_{15}, v_{16}, v_{20}, v_{24}, v_{25}, v_{26}, v_{30}\}$;
 3) $p_4 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) =$
 $= \left((3, 180, 40), (v_5, v_{14}, v_{25}) \right)$;
 $V' \leftarrow V' \setminus \{v_5, v_{14}, v_{25}\}$;
 $C \leftarrow \{p_1, p_2, p_3, p_4\}$;
5. 2) $V' = \{v_4, v_6, v_{10}, v_{15}, v_{16}, v_{20}, v_{24}, v_{26}, v_{30}\}$;
 3) $p_5 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) =$
 $= \left((3, 202, 50), (v_4, v_{15}, v_{24}) \right)$;
 $V' \leftarrow V' \setminus \{v_4, v_{15}, v_{24}\}$;
 $C \leftarrow \{p_1, p_2, p_3, p_4, p_5\}$;
6. 2) $V' = \{v_6, v_{10}, v_{16}, v_{20}, v_{26}, v_{30}\}$;
 3) $p_6 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) =$
 $= \left((3, 411, 120), (v_1, v_6, v_{11}, v_{16}, v_{21}, v_{26}) \right)$;
 $V' \leftarrow V' \setminus \{v_6, v_{16}, v_{26}\}$;
 $C \leftarrow \{p_1, p_2, p_3, p_4, p_5, p_6\}$;
7. 2) $V' = \{v_{10}, v_{20}, v_{30}\}$;
 3) $p_7 = \mathit{prime}\left(\mathit{sort}(S(V'))\right) =$
 $= \left((3, 709, 213), (v_1, v_{10}, v_{11}, v_{20}, v_{21}, v_{30}) \right)$;
 $V' \leftarrow V' \setminus \{v_{10}, v_{20}, v_{30}\}$;
 $C \leftarrow \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$;
8. 2) $V' = \emptyset$;
 Конец алгоритма

Полученный посредством реализации программного комплекса набор

$$C = \left\{ (v_2, v_9, v_{12}, v_{19}, v_{22}, v_{29}), (v_3, v_8, v_{13}, v_{18}, v_{23}, v_{28}), (v_1, v_6, v_{11}, v_{17}, v_{21}, v_{26}), \right. \\ \left. (v_5, v_{14}, v_{25}), (v_4, v_{15}, v_{24}), (v_1, v_7, v_{11}, v_{16}, v_{21}, v_{27}), (v_1, v_{10}, v_{11}, v_{20}, v_{21}, v_{30}) \right\}$$

является покрытием множества вершин ориентированного графа совместимости заданий на перевозку элементами множества максимальных по включению

путей. Кроме того, набор является минимальным, поскольку каждый элемент содержит уникальную вершину. Другими словами, набор несократим, так как удаление любого пути приведет к тому, что оставшийся поднабор не будет покрывать уникальную вершину из удаленного пути. Множество уникальных вершин, например, таково:

$$v_2, v_3, v_7, v_5, v_4, v_6, v_{10},$$

в порядке перечисления путей в наборе.

В приложении к практической задаче организации железнодорожных перевозок найденное решение отвечает назначению и перемещению локомотивов для исполнения нормативных ниток:

$$L_1: n_2, n_9, n_{12}, n_{19}, n_{22}, n_{29},$$

$$L_2: n_3, n_8, n_{13}, n_{18}, n_{23}, n_{28},$$

$$L_3: n_1, n_6, n_{11}, n_{17}, n_{21}, n_{26},$$

$$L_4: n_5, n_{14}, n_{25},$$

$$L_5: n_4, n_{15}, n_{24},$$

$$L_6: n_1, n_7, n_{11}, n_{16}, n_{21}, n_{27},$$

$$L_7: n_1, n_{10}, n_{11}, n_{20}, n_{21}, n_{30},$$

при этом повторное назначение локомотивов для исполнения нормативных ниток n_1, n_{11}, n_{21} организовано с целью подачи локомотива на соответствующую станцию.

В практике решения задач организации железнодорожных перевозок для дорог сложной структуры рассматривают участки обслуживания локомотивов, называемые плечами обслуживания, которые включают депо, станции технического обслуживания и множество путей, допустимых для движения некоторого множества локомотивов. Для каждого участка обслуживания может быть получено решение, что позволяет высоко оценить практическую ценность разработанного программного комплекса для решения задачи организации на этапе назначения и перемещения локомотивов.

Заключение

В рамках исследования были рассмотрены задача планирования и задача организации железнодорожных перевозок. Практическая значимость рассмотренных задач обусловлена, в первую очередь, стратегическим значением железнодорожного транспорта. Кроме того, посредством методов математического моделирования, задачи сводятся к классическим задачам теории графов и комбинаторной оптимизации, в связи с чем исследование представляет самостоятельный интерес как общий подход к решению рассматриваемого класса задач.

Для решения задачи планирования железнодорожных перевозок на этапе формирования множества бесконфликтных наборов нормативных ниток графика движения поездов разработана теоретико-графовая модель — неориентированный граф конфликтов. На основании структурных свойств неориентированного графа конфликтов исходная задача планирования железнодорожных перевозок сводится к задаче расшифровки монотонной булевой функции, порождённой неориентированным графом.

Для решения задачи организации железнодорожных перевозок на этапе назначения и перемещения локомотивов разработана теоретико-графовая модель — ориентированный граф совместимости заданий на перевозку. На основании структурных свойств ориентированного графа совместимости заданий на перевозку исходная задача организации железнодорожных перевозок сводится к задаче покрытия вершин ориентированного графа минимальным числом максимальных по включению путей.

В рамках теоретико-графового подхода разработан вычислительный алгоритм «Бегущая волна» для решения задачи планирования, эффективность которого особенно актуальна в условиях оперативного планирования. Для решения задачи расшифровки разработаны приближённые комбинаторные алгоритмы формирования множества максимальных верхних нулей монотонной булевой функции, порождённой неориентированным графом конфликтов, причём решение, полученное посредством реализации разработанных алгоритмов, отвечает некоторому решению практической задачи планирования. С точки зрения практического планирования перевозок достаточным может оказаться некоторое решение, возможно, не максимальное. В этой связи, на основании утвержде-

ния об оценке отклонения решения, приближённые алгоритмы имеют высокую практическую ценность.

В рамках теоретико–множественного подхода к решению задачи организации железнодорожных перевозок разработан вычислительный алгоритм назначения и перемещения локомотивов. Для решения задачи покрытия разработаны циклические алгоритмы с ветвлением для формирования множества путей и множества максимальных по включению путей, а также циклический алгоритм сортировки множества максимальных по включению путей. Последовательная реализация разработанных алгоритмов позволяет получить решение задачи покрытия, которое отвечает некоторому решению практической задачи организации. Ввиду особенностей строения ориентированного графа совместности заданий на перевозку, на основании утверждения о свойстве минимального покрытия, может быть получено достаточно эффективное решение в части количества используемых локомотивов на этапе назначения и перемещения.

Разработанные приближённые комбинаторные алгоритмы формирования множества максимальных верхних нулей и множества верхних нулей лежат в основе проблемно–ориентированного программного комплекса «Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом» для решения задачи планирования железнодорожных перевозок. Реализация программного комплекса позволяет получить точное или приближённое решение, включая оценку отклонения для приближённого решения.

Разработанные циклические алгоритмы с ветвлением для формирования множества путей, множества максимальных путей и сортировки множества максимальных путей лежат в основе проблемно–ориентированного программного комплекса «Алгоритм покрытия вершин ориентированного графа минимальным числом максимальных по включению путей» для решения задачи организации железнодорожных перевозок. Реализация программного комплекса позволяет получить решение для линейного участка дороги, или любого участка обслуживания некоторого множества локомотивов, причём, в некоторых случаях, решение несократимо, то есть является минимальным.

Внедрение разработанных программных комплексов в технологическую практику планирования и организации железнодорожных перевозок может позволить существенно повысить эффективность эксплуатации экономических и технических ресурсов, что отвечает современным тенденциям развития железнодорожной транспортной инфраструктуры.

Литература

- [1] Азанов В.М., Буянов М.В., Гайнанов Д.Н., Иванов С.В. Алгоритмическое и программное обеспечение для назначения локомотивов с целью перевозки грузовых составов // Вестн. ЮУрГУ. Сер. Матем. моделирование и программирование. 2016. № 9. С. 73–85.
- [2] Белый О.В., Кокурин И.М. Организация грузовых железнодорожных перевозок: пути оптимизации // Транспорт российской Федерации. Журнал о науке, практике, экономике. 2011. Т. 35. № 4.
- [3] Берцун В.Н. Математическое моделирование на графах. Часть 2: Томск: Изд-во Том. ун-та, 2013.
- [4] Гайнанов Д.Н. Комбинаторная геометрия и графы в анализе несовместных систем и распознавании образов. М.: Изд-во Наука, 2014.
- [5] Гайнанов, Д.Н. Об одном критерии оптимальности алгоритма расшифровки монотонных булевых функций // Журнал вычислительной математики и математической физики. 1984. Т. 24. № 8. С. 1250–1257.
- [6] Гайнанов Д.Н. Алгоритмы на графах, порождаемых противоречивыми системами условий, и их применение в задачах управления качеством // Дис. канд. техн. наук. Свердловск, 1981.
- [7] Гайнанов Д.Н. О связности графов некоторых классов систем независимости // Исследования по теории выпуклых множеств и графов. Свердловск: УНЦ АН СССР. 1987. С. 16–23.
- [8] Гайнанов Д.Н., Новокшенов В.Ю., Тягунов Л.И. О графах, порождаемых несовместными системами линейных неравенств // Математические заметки. 1983. Т. 33. № 2. С. 293–300.
- [9] Гайнанов Д.Н., Рассказова В.А. Алгоритм расшифровки монотонных булевых функций, порождённых неориентированными графами // Вестник ЮУрГУ. 2016. Т. 9. № 3. С. 17–30.

- [10] Гайнанов Д.Н., Коныгин А.В., Рассказова В.А. Математическое моделирование грузовых железнодорожных перевозок методами теории графов и комбинаторной оптимизации // Автомат. и телемех. 2016. № 11. С. 60–79.
- [11] Гайнанов Д.Н., Рассказова В.А. Математическое моделирование в задаче оптимального назначения и перемещения локомотивов методами теории графов и комбинаторной оптимизации // Труды МАИ. 2017. № 92.
- [12] Гайнанов Д.Н., Рассказова В.А. Алгоритм покрытия вершин ориентированного графа множеством ориентированных путей в задаче оптимального назначения и перемещения локомотивов // Вестник информационных и компьютерных технологий. 2017. № 5. С. 51–56.
- [13] Гайнанов Д. Н., Рассказова В. А. Теоретико–графовый алгоритм решения задачи о назначении и перемещении локомотивов // Сборник тезисов докладов XLII международной научной конференции «Гагаринские чтения» – 12–15 апреля 2016 г., Москва. – М.: Изд-во МАИ, 2016. – Т. 1, С. 203–204.
- [14] Гайнанов Д. Н., Рассказова В. А. Алгоритм вершинного покрытия для минимизации холостого хода в задаче назначения и перемещения локомотивов // Сборник тезисов докладов XXI международной научной конференции «Системный анализ, управление и навигация», 3–10 июля 2016 г., Евпатория. – М.: Изд-во МАИ, 2016. – С. 133–134.
- [15] Гайнанов Д. Н., Рассказова В. А. Покрытие вершин графа в задаче о назначении локомотивов // Сборник тезисов докладов всероссийской научной конференции «Управление большими системами», 5–9 сентября 2016 г., Самара. – М.: ИПУ РАН, 2016. – С. 312.
- [16] Гайнанов Д. Н., Рассказова В. А. Покрытие вершин графа в задаче оптимального назначения и перемещения локомотивов // Сборник тезисов докладов международной научной конференции «Математика, информатика и физика и их приложения в науке и образовании», 12–15 декабря 2016 г., Москва. – М.: Московский технологический университет (МИРЭА), 2016. – С. 83–85.
- [17] Гайнанов Д. Н., Рассказова В. А. Математическое моделирование в задаче планирования железнодорожных перевозок // Сборник тезисов докладов

дов XLIII международной научной конференции «Гагаринские чтения» – 18–20 апреля 2017 г., Москва. – М.: Изд-во МАИ, 2017. С. 703–704.

- [18] Гайнанов Д. Н., Рассказова В. А. Покрытие вершин ориентированного графа в задаче о назначении и перемещении локомотивов // Тезисы докладов международной научной конференции «Системный анализ, управление и навигация», июль 2017 г., Евпатория. –М.: Изд-во МАИ, 2017, принята заявка на участие в конференции.
- [19] Гоманков Ф.С. Технология и организация перевозок на железнодорожном транспорте: учебник для вузов. М.: Транспорт, 1994.
- [20] Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. М.: Изд-во Мир, 1982.
- [21] Дасгупта С., Пападимитриу Х., Вазирани У. Алгоритмы. М.: Издательство МЦНМО, 2014.
- [22] Дистель Р. Теория графов. Новосибирск: Изд-во ин-та математики им. С.Л. Соболева СО РАН, 2002.
- [23] Ерёмин И. И. Линейная оптимизация и системы линейных неравенств. Прикладная математика и информатика. М.: Издательский центр Академия, 2007.
- [24] Ерёмин И. И. Противоречивые модели оптимального планирования. Свердловск: Экономико–математическая библиотека, 1988.
- [25] Ерёмин И.И. Противоречивые модели экономики. Свердловск: Средне–Уральское книжное издательство, 1986.
- [26] Ерёмин И.И. Теория линейной оптимизации. Екатеринбург: Экономико–математическая литература, 1999.
- [27] Ерёмин И.И., Мазуров Вл.Д., Астафьев Н.Н. Несобственные задачи линейного и выпуклого программирования. М.: Наука, 1983.
- [28] Ефименко Ю.И. Общий курс железных дорог. М.: Изд-во Академия, 2005.

- [29] Иванов С. В., Кибзун А. И., Наумов А. В. Двухуровневая задача оптимизации деятельности железнодорожного транспортного узла // УБС. 2012. № 38. С. 140–160.
- [30] Иванов С. В., Кибзун А. И., Осокин А. В. Оптимизационная стохастическая модель назначения локомотивов для перевозки грузовых составов // Автомат. и телемех. 2016. № 11. С. 80–95.
- [31] Ивахненко А.Г. Моделирование сложных систем по экспериментальным данным. М.: Радио и Связь, 1986.
- [32] Кобко, Л.И. Комплексный комбинаторный метод построения расписания работы рабочих мест первичных производственных систем // Труды МАИ. 2001. № 3.
- [33] Кормен Т., Лейзерсон Ч., Ривест Р. и др. Алгоритмы: построение и анализ. 2-е изд. М.: Изд-во Вильямс, 2005.
- [34] КORTE Б., Фиген Й. Комбинаторная оптимизация. Теория и алгоритмы. М.: Издательство МЦНМО, 2015.
- [35] Коршунов, А.Д. Монотонные булевы функции / А.Д. Коршунов // Успехи математических наук. – 2003. – Т. 58, № 5 (535). – С. 89–162.
- [36] Лазарев А.А. Оценки абсолютной погрешности и схема приближенного решения задач теории расписаний // Журн. вычислит. матем. и мат. физики. 2009. Т. 49. № 2. С. 14–34.
- [37] Лазарев А.А., Гафаров Е.Р. Преобразование сетевого графика задач теории расписаний с ограничениями предшествования // ДАН. 2008. Т. 424. № 2. С. 7–9.
- [38] Лазарев А.А., Мусатова Е.Г., Гафаров Е.Р., Кварацхелия А.Г. Теория расписаний. Задачи железнодорожного планирования. М.: ИПУ РАН, 2012.
- [39] Мазуров Вл.Д. Метод комитетов в задачах оптимизации и классификации. М.: Наука, 1990.
- [40] Мазуров Вл.Д. О комитете системы выпуклых неравенств // Труды ИСМ. М.: МГУ. 1996. № 14. С. 41.

- [41] Мазуров Вл.Д. О построении комитета системы выпуклых неравенств // Кибернетика. 1967. № 2. С. 56–59.
- [42] Мазуров Вл.Д., Хачай М.Ю. Комитетные конструкции // Известия Уральского гос. ун-та, 14. Сер. матем. и мех. 1999. Вып. 2. С. 77–108.
- [43] Мазуров Вл.Д., Хачай М.Ю. Комитеты систем линейных неравенств // Автоматика и телемеханика. 2004. № 2. С. 43–54.
- [44] Мазуров Вл.Д., Хачай М.Ю., Рыбин А.И. Комитетные конструкции для решения задач выбора, диагностики и прогнозирования // Труды института математики и механики УрО РАН. 2002. Т. 8. № 1. С. 66–102.
- [45] Мазуров Вл.Д., Казанцев В.С., Белецкий И.Г., Кривоногов А.И., Смирнов А.И. Вопросы обоснования и применения комитетных алгоритмов распознавания // Распознавание, классификация, прогноз. Математические методы и их применение. М.: Наука. 1988. Вып. 1. С. 114–148.
- [46] Малинина, Н.Л. Противоречия в свойствах двух основных типов сетевых моделей и пути их разрешения // Труды МАИ. 2010. № 37.
- [47] Орлов А. И. Графы при моделировании процессов управления промышленными предприятиями // Управление большими системами. № 30. С. 62–75
- [48] Осипов С.И., Осипов С.С. Основы тяги поездов. М.: УМК МПС, 2000.
- [49] Пападимитриу Х., Страйглиц К. Комбинаторная оптимизация. Алгоритмы и сложность. Издательство Мир, 1982.
- [50] Рассказова, В.А. Алгоритм расшифровки монотонной булевой функции, порождённой неориентированным графом. Программа для ЭВМ.
- [51] Сапоженко, А.А. Проблема Дедекинда и метод граничных функционалов // А.А. Сапоженко. – Москва: Физматлит, 2009.
- [52] Севастьянов С.В. Геометрические методы и эффективные алгоритмы в теории расписаний // Дис. док. физ.-мат. наук. Новосибирск. 2000.
- [53] Сотсков Ю.Н., Танаев В.С., Струсевиц В.А. Теория расписаний. Многостадийные системы. М.: Наука, 1989.

- [54] Скиена С. Алгоритмы. Руководство по разработке. Санкт-Петербург: БХВ–Петербург, 2014.
- [55] Шепитько Т.В. , Гасанов А.И. , Бучкин В.А. Математические модели и методы инженерных расчетов на ЭВМ: Учебное пособие. М.: МИИТ, 2004.
- [56] Akimova E.N., Gainanov D.N., Golubev O.A., et al. The Problem of Scheduling for the Linear Section of a Single-track Railway with Independent Edges Orientations// 1st Ural Workshop on Parallel, Distributed, and Cloud Computing for Young Scientists. CEUR Workshop Proc. 2015. V. 1513. P. 130–136.
- [57] Bioch, J.C. Minimum self-dualdecompositions of positive dual-minor Boolean functions / J.C. Bioch, T. Ibaraki, K. Makino // Discrete Applied Mathematics. – 1999. – Vol. 96–97. – P. 307–326.
- [58] Boros, E. Polynomial time recognition of 2-monotonic positive Boolean functions given by an oracle / E. Boros, P. Hammer, T. Ibaraki, K. Kawakami // SIAM J.Comput. – 1997. – № 26. – P. 93–109.
- [59] Burdett O., Kozan E. A Disjunctive Graph Model and Framework for Constructing New Train Schedules// Eur. J. Oper. Res. 2010. V. 200. P. 85–98.
- [60] Domingo, C. Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries / C. Domingo, N. Mishra, L. Pitt // Machine Learning. – 1999. – № 37(1). – P. 89–110.
- [61] Gainanov Damir N. Graphs for Pattern Recognition. Infeasible Systems of Linear Inequalities. De Gruyter, 2016.
- [62] Gholami O., Sotskov Y.N. Scheduling Algorithm with Controllable Train Speeds and Departure Times to Decrease the Total Train Tardiness// Int. J. Industrial Engineering Computations. 2014. V. 5. P. 281–294.
- [63] Gholami O., Sotskov Y.N., Werner F. Job-shop Problems with Objectives Appropriate to Train Scheduling in a Single-track Railway// Proc. 2 Int. Conf. on Simulation and Modeling Methodologies, Technologies and Applications. Roma, Italy. 2012. P. 425–430.

- [64] Gholami O., Sotskov Y.N. Mixed Graph Model and Algorithms for Parallel-machine Job-shop Scheduling Problems // Int. J. Production Research. 2015. V. 8. P. 1–16.
- [65] Lusby R., Ryan D. Railway Track Allocation. Models and Methods // Oper. Res. Spektrum. 2011. V. 33. P. 843–883.
- [66] Makino, K. A fast and simple algorithm for identifying 2-monotonic positive Boolean functions / K. Makino, T. Ibaraki // Journal of Algorithms. – 1998. – № 26(2). – P. 291–305.
- [67] Makino, K. The maximum latency and identification of positive boolean functions / K. Makino, T. Ibaraki // SIAM J.Comput. – 1997. – № 26. – P. 1363–1383.
- [68] Torvik, V.I. Guided inference of nested monotone boolean functions / V.I. Torvik, E. Triantaphyllou // Information Sciences. – 2003. – № 151 (SUPPL). – P. 171–200.
- [69] Torvik V.I., Triantaphyllou E. Inference of monotone Boolean functions // in Floudas C.A., Pardalos P.M. (eds) Encyclopedia of optimization. 2nd ed. New York, 2009, pp. 1591–1598.
- [70] Triantaphyllou, E. Data mining and knowledge discovery via logic-based methods. Theory, algorithms and applications / E. Triantaphyllou. – New York: Springer optimization and its application, 2010.
- [71] Valiant, L. A theory of the learnable / L. Valiant // Commun. ACM. – 1984. – № 27(11). – P. 1134–1142.