В. С. ТАНАЕВ
В. В. ШКУРБА

Введение в теорию расписаний

ЭКОНОМИКО-МАТЕМАТИЧЕСКАЯ
БИБЛИОТЕКА

В. С. ТАНАЕВ,
В. В. ШКУРБА

Введение в теорию расписаний

Под редакцией
Д. Б. ЮДИНА

ИЗДАТЕЛЬСТВО «НАУКА»
ГЛАВНАЯ РЕДАКЦИЯ
ФИЗИКО-МАТЕМАТИЧЕСКОЙ ЛИТЕРАТУРЫ
Москва 1975

518
Т 18
УДК 519.95

Введение в теорию расписаний. В. С. Танаев, В. В. Шкурба (серия «Экономико-математическая библиотека»), Главная редакция физико-математической литературы изд-ва «Наука», М., 1975.

В монографии предпринята попытка в сжатой и вместе с тем доступной для широкого читателя форме отразить современное состояние теории расписаний. Основное внимание уделяется рассмотрению задач оптимального упорядочения комплекса взаимосвязанных операций во времени. Приводятся описания точных и приближенных методов их решения.

Монография предназначена для студентов и преподавателей вузов математических специальностей, специалистов в области управления, инженеров и практиков, сталкивающихся с задачами дискретной оптимизации вообще, оптимального моделирования дискретных систем и календарного планирования процессов в частности.

Илл. 34, таблиц 28, библ. названий 428.

Вячеслав Сергеевич Танаев,
Виктор Васильевич Шкурба

ВВЕДЕНИЕ В ТЕОРИЮ РАСПИСАНИЙ

М., 1975 г., 256 стр. с илл.
Редактор *Н. П. Рябенькая*
Техн. редактор *Н. В. Кошелева* Корректор *В. П. Сорокина*

Сдано в набор 23/XII 1974 г. Подписано к печати 4/IV 1975 г.
Бумага $84\times108^1/_{32}$, тип. № 1. Физ. печ. л. 8. Условн. печ. л. 13,44. Уч.-изд. л. 14,24.
Тираж 9800 экз. Т-06545. Цена книги 1 р. 06 к. Заказ № 1501.

Издательство «Наука»
Главная редакция физико-математической литературы
117071, Москва, В-71, Ленинский проспект, 15

2-типография изд-ва «Наука», Москва, Шубинский пер., 10.

$Т\frac{20204-063}{053(02)-75}$ 81-74

ОГЛАВЛЕНИЕ

Предисловие редактора 5
Предисловие авторов 7

Глава 1. **Введение** 9
§ 1. Предмет теории расписаний 9
§ 2. Классификация задач 12
§ 3. Формы представления расписаний 17
§ 4. Общая характеристика методов 23
§ 5. Библиографическая справка 26

Глава 2. **Элементы комбинаторного анализа** 27
§ 1. Множества, отношения, отображения, графы . 27
§ 2. Упорядоченность 31
§ 3. Смешанные (дизъюнктивные) графы 40
§ 4. Перестановки. Задачи упорядочения. Перестановочный прием 47
§ 5. Оптимизация функций, рекуррентно заданных на множестве перестановок 50
§ 6. Библиографическая справка 59

Глава 3. **Детерминированные системы обслуживания с одним прибором** 61
§ 1. Предварительные замечания 61
§ 2. Интервалы очередности 64
§ 3. Директивные сроки 73
§ 4. Задача коммивояжера 88
§ 5. Взаимосвязанные требования 100
§ 6. Древовидно упорядоченные требования 103
§ 7. Общий случай 110
§ 8. Библиографическая справка 116

Глава 4. **Параллельные приборы** 119
§ 1. Прерывания 119
§ 2. Обслуживание в заданные сроки 128
§ 3. Некоторые задачи упорядочения 133
§ 4. Библиографическая справка 143

Глава 5. **Детерминированные системы обслуживания с двумя последовательными приборами** 145
§ 1. Последовательное обслуживание 145
§ 2. Параллельно-последовательное обслуживание . . 156
§ 3. Библиографическая справка 165

Глава 6. **M последовательных приборов. Одинаковые маршруты** . . . 166
§ 1. Общие замечания . . . 166
§ 2. Конструктивный подход . . . 169
§ 3. Элиминация . . . 174
§ 4. Вырожденные случаи . . . 180
§ 5. Библиографическая справка . . . 184

Глава 7. **M последовательных приборов. Различные маршруты** . . . 186
§ 1. Сетевое представление . . . 186
§ 2. Генераторы допустимых расписаний . . . 192
§ 3. Случайный поиск с обучением . . . 198
§ 4. Линейные модели . . . 207
§ 5. Библиографическая справка . . . 209

Глава 8. **Многооператорные процессы обслуживания** . . . 211
§ 1. Процессы с неограниченным числом операторов переноса . . . 211
§ 2. Процессы с ограниченным числом операторов переноса . . . 222
§ 3. Библиографическая справка . . . 232

Цитированная литература . . . 233

ПРЕДИСЛОВИЕ РЕДАКТОРА

Многие задачи планирования и управления требуют упорядочения во времени фиксированной системы ресурсов для выполнения определенной совокупности работ. От выбора постановки и качества решения таких задач существенно зависит рациональная организация работ и эффективность производства.

Начиная с пятидесятых годов, задачи календарного планирования и оперативного управления привлекают внимание специалистов по исследованию операций. В связи с большим разнообразием анализируемых ситуаций исследования группировались по различным характерным признакам и проводились в рамках различных научных дисциплин. В теории сетевого планирования основное внимание уделялось распределению времени и материальных ресурсов при выполнении заданного комплекса работ. В теории расписаний рассматривались неделимые виды ресурсов (станки, машины) и такие виды работ, как операции по обработке и транспортировке некоторых деталей, изделий, продуктов. В теории массового обслуживания рассматривались задачи назначения приоритетов в обслуживании поступающих заявок некоторыми устройствами, приборами и т. п.

Формальные модели, отвечающие разнообразным по постановке и содержанию задачам календарного планирования и оперативного управления, обнаруживают определенное сходство. Для их анализа могут быть использованы и однотипные математические методы.

В настоящее время наблюдается тенденция к формированию единой научной дисциплины, в рамках которой сосредоточилось бы изучение различных по приложениям, но единых по структуре моделей задач календарного планирования и оперативного управления.

Предлагаемая вниманию читателей монография В. С. Танаева (Минск) и В. В. Шкурбы (Киев) является, вероятно, наиболее полным изложением теории расписаний в ее классическом понимании. Авторы монографии предлагают использовать термин «теория расписаний» в качестве названия указанной единой дисциплины. В этом смысле предметом монографии является детальное рассмотрение важного ее раздела — задач упорядочения.

По простоте и доступности изложения монография рассчитана на читателей, желающих получить первоначальное представление о предмете. По полноте же и оригинальности приводимых результатов она представляет определенный интерес и для специалистов по теории расписаний.

При подготовке монографии авторы стремились к тому, чтобы у читателя сложилось достаточно полное представление о месте теории расписаний в общей теории управления большими системами, об особенностях решаемых задач, о том арсенале разнообразных приемов и методов, использование которых позволяет, по крайней мере, надеяться на возможность построения рациональных расписаний в реальных ситуациях.

В монографии представлены не все интересные результаты и перспективные направления развития теории расписаний. Однако обширный список литературы и библиографические справки, сопровождающие каждую главу, позволяют читателю ориентироваться в современном состоянии исследований.

Можно надеяться, что читатель, не нашедший в монографии непосредственного ответа на свой вопрос, сумеет во всяком случае использовать полученную информацию для составления расписания самостоятельной подготовки к построению и анализу модели интересующей его задачи.

Д. Б. Юдин

ПРЕДИСЛОВИЕ АВТОРОВ

В рамках теории расписаний строятся и анализируются математические модели специфических ситуаций, постоянно возникающих при календарном планировании различных видов человеческой деятельности; создаются формальные методы принятия наилучших решений в этих ситуациях; вырабатываются практические рекомендации по улучшению качества планирования и управления.

Среди задач теории расписаний особое место занимают задачи упорядочения, рассмотрению которых, в основном, и посвящена эта книга. В ней отсутствуют обычно приводимые в такого рода работах сведения из теории массового обслуживания, сетевого планирования, математического программирования и некоторых других смежных областей, которые достаточно полно представлены в существующей литературе.

В первой главе книги предпринята попытка определить предмет теории расписаний, классифицировать задачи, кратко охарактеризовать методы их решения. Вторая глава носит вспомогательный характер и содержит широко используемые в дальнейшем сведения из комбинаторного анализа. Последующие две главы посвящены рассмотрению так называемых одностадийных задач теории расписаний. В пятой главе рассматриваются задачи построения оптимальных (по быстродействию) расписаний для двух приборов при различных допущениях относительно рарактера обслуживания требований. В шестой и седьмой главах анализируются системы, содержащие произвольное число последовательных приборов. Наконец, восьмая глава посвящена описанию систем с операторами переноса.

Все главы книги, а подчас и отдельные параграфы в значительной степени самостоятельны. Однако выбранный порядок глав обеспечивает определенную логическую последовательность изложения материала. Каждая глава сопровождается краткой библиографической справкой.

Авторы считают своим приятным долгом поблагодарить В. С. Гордона, Г. М. Левина, Е. В. Левнера и Ю. Н. Сотскова за ряд полезных замечаний и советов, сделанных по рукописи и способствовавших существенному улучшению ее качества. Большую помощь в оформлении рукописи оказали С. Н. Воронкова и С. И. Крючкова.

Авторы выражают искреннюю признательность профессору Д. Б. Юдину, взявшему на себя нелегкий труд редактирования этой книги.

Все замечания и пожелания, присланные читателями в адрес книги, будут восприняты авторами с благодарностью.

В. С. Танаев, В. В. Шкурба

ГЛАВА 1
ВВЕДЕНИЕ

§ 1. Предмет теории расписаний

1.1. Расширение масштабов современного производства, усложнение уровня проводимых мероприятий, необходимость координации деятельности больших коллективов людей существенно усложнили функции организационного управления.

В различных областях целенаправленной человеческой деятельности, в сложных, зачастую противоречивых условиях приходится принимать решения, нередко связанные с судьбами людей и большими материальными затратами. Принимаемые решения всегда направлены на достижение некоторых целей и реализуются в рамках некоторой системы ограничений, обусловленных конкретными обстоятельствами проведения мероприятия.

Как правило, одни и те же цели могут быть достигнуты различным образом, с различными затратами труда и материальных ресурсов. Выбрать наиболее экономичный и целесообразный путь, принять обоснованное, наиболее правильное решение — далеко не простая задача и для своего решения требует привлечения современных научных методов.

Эти методы, которые принято объединять под общим названием «*исследование операций*», практически сформировались в последнее тридцатилетие. В настоящее время исследование операций — достаточно мощный инструмент количественного анализа сложных целенаправленных процессов, протекающих в различных сферах человеческой деятельности. Методы исследования операций способствуют выработке рациональных, обоснованных решений по управлению этими процессами.

1.2. При операционном подходе к поиску наиболее эффективных путей достижения цели осуществляется построение *математической модели*, содержащей описание этой цели и отражающей условия проведения операции.

На этом этапе тщательно анализируется содержание операции, определяются необходимые действия, выявляются условия их выполнения, оцениваются разнообразные ограничения и т. п. Методология исследования операций позволяет систематизировать всю эту работу, придать ей большую целенаправленность и определенность.

Оценка и сравнение эффективности возможных способов действий при достижении поставленной цели проводятся на основании построенной модели. Эта же модель позволяет принять и наилучшее в рассматриваемой ситуации решение. В большинстве случаев выработка такого решения требует привлечения соответствующих математических методов оптимизации и сопряжена с большим объемом вычислений.

Анализируемые в рамках исследований операций модели являются разумным компромиссом между двумя естественными, но противоречивыми тенденциями. С одной стороны, желательно, чтобы модель возможно полнее отражала реальные процессы, с другой — она должна быть настолько простой, чтобы можно было получить искомые результаты за практически приемлемое время. Постоянное развитие математических методов и вычислительной техники позволяет расширить круг анализируемых моделей и тем самым расширить сферу практического применения методов исследования операций.

1.3. С использованием операционного подхода оказалось возможным разработать эффективную методику управления транспортными перевозками, решить ряд задач распределения ресурсов и размещения производительных сил, выработать наилучшие стратегии управления запасами и т. д.

В середине пятидесятых годов начались интенсивные исследования по построению и анализу *моделей календарного планирования* и разработке методов принятия плановых решений с использованием этих моделей. Среди причин, вызвавших появление этого нового направления в исследовании операций, в первую очередь следует отметить все возрастающую необходимость планирования и управления сложными разработками, включающими большое число взаимосвязанных работ, требующих многочисленных исполнителей и значительных материальных затрат.

Все отчетливее осознавалось, что качество функционирования современного производства во многом определяется качеством решений, принимаемых на этапах календарного планирования и оперативного управления. Наряду с улучшением качества плановых решений необходимо было также сократить сроки их выработки, повысить оперативность и гибкость управления.

Временная увязка всего множества действий, сопряженных с достижением заданной цели, уже сама по себе достаточно сложная задача. Если же речь идет о построении наилучшего календарного плана, да еще в кратчайший срок, то сложность этой задачи неизмеримо возрастает. Традиционные методы календарного планирования все меньше отвечали запросам практики. В этих условиях применение математических методов и вычислительной техники открывало определенные перспективы и позволяло надеяться на успешное преодоление возникших затруднений.

1.4. Круг вопросов, связанных с построением наилучших календарных планов (расписаний), особенно с разработкой математических методов получения решений с использованием соответствующих моделей, изучается в рамках *теории расписаний*.

Эта теория использует характерный для исследования операций модельный подход к анализу реальных процессов. Изучаемые в теории расписаний модели отражают специфические ситуации, возникающие при календарном планировании различных видов целенаправленной человеческой деятельности.

Разнообразие моделей, степень их общности и универсальности постепенно увеличиваются, охватывая все более широкую сферу возможных приложений — календарное планирование производства, транспорта, военных операций, обучения, информационно-вычислительных процессов и т. п. По мере усложнения моделей усложняются и методы принятия плановых решений с использованием этих моделей. В свою очередь совершенствование методов, поиск новых подходов открывает перспективы новым приложениям.

Следует отметить, что уже на начальной стадии развития теории расписаний стало ясно, что задачи оптимального календарного планирования исключительно трудоемки. Для их решения пришлось привлечь обширный арсенал средств современной прикладной математики, провести обстоятельные экспериментальные исследования, выдвинуть и проанализировать различные гипотезы, разработать новые подходы и методы. Тем не менее значительная доля полученных результатов носит негативный характер и скорее выявляет сложность проблемы, чем намечает конструктивные пути ее решения.

1.5. При построении математических моделей календарного планирования, как, впрочем, и при операционном подходе к анализу иных видов целенаправленной деятельности, серьезные затруднения вызывает недостаточная информированность исследователя об анализируемом процессе.

В этой книге рассматриваются простейшие случаи принятия плановых решений в *условиях полной определенности*, с четким знанием целей и ограничений, с фиксированными известными значениями неконтролируемых факторов.

Наряду с такого рода ситуациями в теории расписаний рассматриваются ситуации, в которых решения приходится принимать в условиях риска или неопределенности. Так, при календарном планировании военных операций, работы рыбодобывающих судов, сельскохозяйственных работ и т. п. в лучшем случае известны распределения вероятностей многих факторов, существенно влияющих на формирование календарных планов.

§ 2. Классификация задач

2.1. Целенаправленную деятельность всегда можно рассматривать как некоторый, протекающий во времени процесс, результатом которого является достижение поставленных целей. Как правило, этот процесс включает определенную совокупность действий, мероприятий, работ, которые реализуются в некоторой существующей или специально создаваемой системе.

Выполняемые действия, мероприятия, работы не являются независимыми. Существуют определенные взаимосвязи, как между работами рассматриваемой совокупности, так и между этими работами и работами, реализуемыми вне рассматриваемой совокупности. Так, сборка некоторого узла технического устройства может быть проведена лишь после завершения изготовления всех составляющих его компонент. Возведение фундамента строящегося здания невозможно без наличия кирпича и цемента, изготовление и транспортировка которых может и не входить в рассматриваемую совокупность работ по строительству здания.

Наличие такого рода взаимосвязей между работами налагает определенные ограничения на возможное протекание соответствующего целенаправленного процесса во времени. Ту или иную работу оказывается возможным осуществить только при наступлении вполне определенной ситуации в процессе выполнения работ рассматриваемой совокупности.

Существенные ограничения налагает и та конкретная система, в которой этот процесс реализуется.

Во многих случаях систему можно и целесообразно представить в виде некоторой совокупности определенным образом взаимосвязанных подсистем — блоков, устройств, приборов. На вход каждого блока поступает некоторое множество заявок, заказов, требований на выполнение определенного вида работ. В результате выполнения этих работ появляются новые заказы, заявки, требования на выполнение иных видов работ, поступающие на вход некоторых других блоков и т. д., пока не будет выполнена совокупность всех реализуемых в данной системе работ. Поскольку процесс выполнения работ протекает в реальном времени, то перед отдельно выбранным блоком может образоваться очередь требований на выполнение работ.

2.2. Следует отметить, что как деление всего множества работ на отдельные «работы», так и представление системы в виде набора взаимосвязанных «блоков» могут быть проведены с различной степенью детализации и зависят от конкретно анализируемой ситуации.

Рассмотрим, например, некоторый завод, выполняющий отдельные заказы. В зависимости от конкретно анализируемой ситуации в качестве совокупности работ может быть выбрана совокупность заказов. Завод же можно представить одним блоком, на вход которого поступают требования на выполнение этих заказов. Вместе с тем заказы можно детализировать до отдельных видов работ, а завод рассматривать как совокупность блоков — производственных участков, на каждом из которых выполняется определенный вид работ.

Аналогично, вычислительную систему можно рассматривать либо как единый блок, либо как совокупность взаимосвязанных блоков, выделяя, например, арифметические и запоминающие устройства, устройства ввода-вывода информации и т. д. Соответственно работами будут служить либо программы, либо отдельные виды действий, предусмотренные данными программами.

При планировании учебного процесса могут быть выделены такие «блоки» как преподаватели, осуществляющие определенные работы по обучению студентов, учебные группы студентов, воспринимающие определенную информацию, и т. д. При планировании работы транспорта могут быть выделены «блоки» — дороги, «блоки» — локомотивы и т. д.

Степень детализации работ и уровень «структуризации» системы обычно согласуются между собой и с теми задачами, которые предполагается решать.

2.3. При реализации данной совокупности работ в данной или проектируемой системе необходимо закрепить определенные работы за определенными блоками, согласовать длительности выполнения работ и установить порядок выполнения их во времени.

Простейшим и наиболее изученным классом задач теории расписаний являются *задачи упорядочения.* В этих задачах распределение работ по блокам и длительности их выполнения предполагаются заданными. Необходимо указать наиболее эффективную стратегию управления очередями требований на выполнение работ каждым блоком.

В *задачах согласования* основное внимание уделяется выбору длительностей работ при заданном их распределении по блокам.

Наконец, в *задачах распределения* предполагается, что один и тот же блок может выполнять различные работы или наборы работ. Необходимо указать в некотором смысле наилучшее распределение работ по блокам.

Во всех перечисленных классах задач теории расписаний совокупность работ и структура реализующей их системы предполагаются заданными и неизменными.

В последнее время наметилось новое направление в теории расписаний, связанное с изучением влияния структуры системы на временные характеристики протекающих в ней процессов. Необходимость в изучении задач такого рода вызвана в первую очередь развитием вычислительной техники, появившейся возможностью создавать различные по составу и конфигурации вычислительные системы.

Приведенная классификация задач теории расписаний, как, впрочем, и любая другая классификация, в значительной степени условна и скорее подчеркивает некоторые характерные особенности той или иной задачи, чем лишает ее всех остальных черт, возможно присущих другим классам задач. Многие авторы, например, к задачам теории расписаний относят только задачи упорядочения.

Задачи согласования объединяют термином «сетевое планирование». Распределительные задачи также группируют в отдельную, рассматриваемую вне теории расписаний совокупность задач.

Приведем примеры некоторых типичных задач теории расписаний.

2.4. Наиболее известной в теории расписаний задачей упорядочения является задача построения оптимального (в том или ином смысле) расписания процесса обработки «деталей» некоторой совокупностью «машин».

Процесс обработки каждой детали включает последовательное выполнение некоторого множества операций. Каждая операция выполняется некоторой машиной. Длительность выполнения каждой операции предполагается известной. Она может быть как детерминированной, так и случайной величиной.

Каждая машина одновременно может выполнять не более одной операции.

При построении наилучшего, скажем по быстродействию, расписания процесса обработки всех деталей необходимо решить, в какой именно последовательности должны выполняться операции каждой машиной.

В рассматриваемом случае множество «работ» — это множество всех операций. Машины представляют собой «блоки» системы, реализующей заданную совокупность работ. Взаимосвязи между работами задаются посредством указания последовательностей их выполнения при обработке каждой детали.

Если одна и та же операция может быть выполнена не на одной машине, а на любой из некоторой совокупности машин, то наряду с упорядочением операций необходимо провести и распределение их по машинам.

2.5. Классической задачей согласования является задача определения длительностей выполнения работ заданной совокупности с целью минимизации суммарной стоимости работ при заданном общем времени выполнения всех работ совокупности.

Каждая работа характеризуется длительностью и стоимостью ее выполнения. Предполагается, что длительность выполнения работы может изменяться в заданных пределах. Стоимость выполнения работы зависит от длительности. Обычно предполагается, что эта зависимость описывается выпуклой, чаще всего, линейной функцией.

Работы рассматриваемой совокупности не являются независимыми. Для каждой работы известно множество работ, после выполнения которых может быть начато выполнение данной работы.

Необходимо выбрать длительность каждой работы (в заданных пределах) таким образом, чтобы общее время выполнения всех работ не превосходило заданной величины, а суммарная стоимость их выполнения была наименьшей. Нас может интересовать также уменьшение общего времени выполнения всех работ при заданной суммарной стоимости их выполнения.

Эта задача существенно усложняется, если приходится принимать во внимание обычную на практике ограниченность других видов ресурсов, отличных от временных и финансовых.

2.6. Среди многочисленных распределительных задач теории расписаний отметим одну из простейших — так называемую задачу балансирования сборочной линии.

В этой задаче необходимо распределить некоторую совокупность работ по рабочим местам, расположенным в линию. Для каждой работы известна длительность ее выполнения и множество работ, после завершения которых может быть начато выполнение данной работы.

С учетом заданных отношений предшествования работы распределяются таким образом, чтобы суммарное время выполнения работ на каждом рабочем месте не превосходило заданную величину (так называемого времени цикла) и число рабочих мест было минимально.

Широко известны также задачи закрепления преподавателей за группами студентов, распределения учебных помещений, распределения программ в многопроцессорных вычислительных системах и т. п.

2.7. При построении операционных моделей календарного планирования в реальных ситуациях редко удается получить «чистую» задачу упорядочения, согласования или распределения. Как правило, принятие наилучших плановых решений сопряжено с рассмотрением экстремальных задач, в которых элементы упорядочения, согласования и распределения в значительной мере «переплетаются».

Тем не менее выделение и анализ «чистых» задач представляют интерес как в непосредственно прикладном аспекте, так и в плане разработки эффективных общих методов оптимального календарного планирования.

В этой книге основное внимание уделяется задачам упорядочения. С задачами согласования и распределения заинтересованный читатель может ознакомиться по многочисленным монографиям, посвященным исследованию операций и сетевому планированию.

§ 3. Формы представления расписаний

В расписаниях для рассматриваемой совокупности действий, операций, работ указываются моменты их начала, а подчас моменты окончания или даже каких-то промежуточных состояний их выполнения во времени. При необходимости указываются также те блоки системы, которые реализуют выполнение соответствующих действий, операций, работ.

3.1. На практике предложены и используются различные способы представления расписаний. Наиболее наглядны из них *графические представления* — графики Ганта, ленточные графики, планировочные графики, циклограммы, оперограммы, учетные графики и т. п.

На рис. 1.3.1 приведен образец *графика Ганта.* Этот график отражает загрузку рабочих мест № 1 и № 2 (блоков) при обработке некоторых деталей (партий деталей). Обработка каждой детали сводится к выполнению ряда технологических операций. Каждая операция на рисунке представляется отрезком, по длине равным продолжительности выполнения операции в выбранном масштабе времени. Моменты начала и конца выполнения

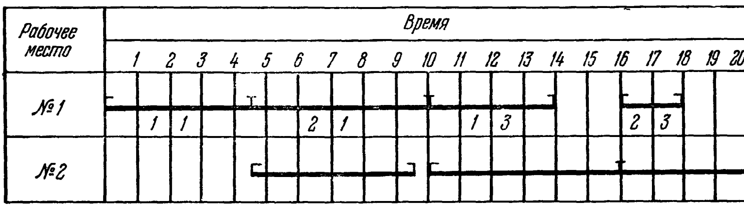

Рис. 1.3.1. График Ганта.

операций обозначаются отметками ┌ и ┐ соответственно. Под отрезками обычно указывают какие-то характеристики операции — например, принятые номер детали (партии деталей), номер операции, размер партии и т. п.

В *хрономограмме*, представленной на рис. 1.3.2, основное внимание обращается на последовательность и взаимоувязку во времени выполнения комплекса работ безотносительно к характеру использования ресурсов.

Весьма специфично в диспетчерских службах железных дорог графически представляются *расписания движения поездов* (рис. 1.3.3). Расписание занимает часть координатной плоскости, где на оси абсцисс откладывается время (от 0 до 24 час.), а на ось ординат наносятся точки, соответствующие станциям рассматриваемой железнодорожной ветви. Расписание движения (формирования, расформирования, стоянки) того или иного состава задается линией, которая совмещает пространственное и временное расположение состава в каждый момент времени.

3.2. В последние годы широкое распространение получили особые формы представления календарных планов в виде так называемых *стрелочных диаграмм* или *сетевых графиков*. Такие формы чаще всего используются для представления календарных планов выполнения сложных разработок, индивидуальных заказов особой важности, проектирования или строительства уникальных объектов большим количеством соисполнителей.

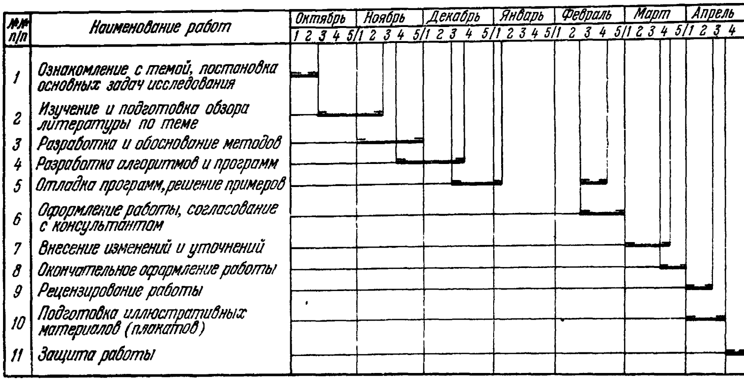

Рис. 1.3.2. Хрономограмма выполнения курсовой (дипломной) работы.

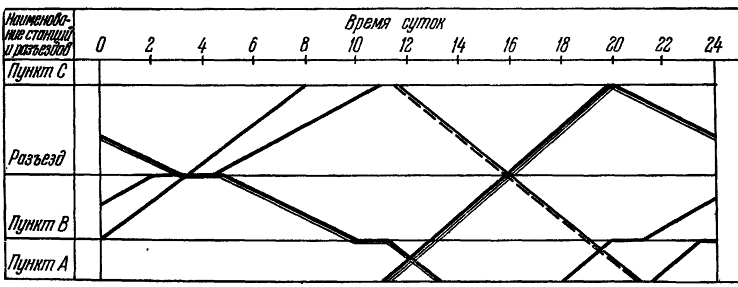

Рис. 1.3.3. График движения поездов.

Для таких разработок, проектов, строек особо важно представить взаимоувязку во времени выполнения отдельных работ или их комплексов. Этим целям служат сетевые графики.

Образец сетевого графика в так называемой форме «работы-вершины» представлен на рис. 1.3.4. Вершинами сетевого графика обозначены отдельные работы, на которые расщепляется общий комплекс работ, дуги представляют информацию о взаимозависимости очередности выполнения работ.

Логическая взаимозависимость очередности выполнения работ, представленная сетевым графиком, облегчает прежде всего распараллеливание комплекса разработок, распределение работ между исполнителями.

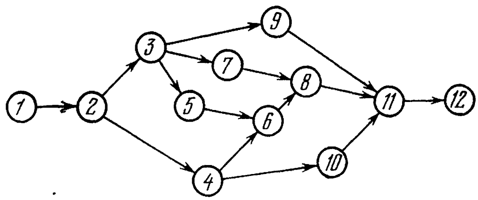

Рис. 1.3.4. Сетевой график разработки и внедрения задачи (программного комплекса) в АСУ. Работы: *1* — предварительное определение перечня и структуры выдаваемых документов, информационных массивов и характера их использования; *2* — разработка общей схемы решения задачи, утверждение перечня и форм выдаваемых документов, выдача задания на программирование, корректировку базовых массивов и первичных документов и т. п.; *3* — определение структур данных и способов кодирования информации; *4* — обеспечение формирования первичных данных; *5* — обеспечение формирования нормативных массивов; *6* — обеспечение формирования базовых массивов; *7* — разработка программного обеспечения; *8* — отладка программ; *9* — техническое обеспечение решения задачи; *10* — организационное обеспечение решения задачи; *11* — опытно-промышленная проверка; *12* — корректировка по результатам проверки.

Масштабное представление продолжительности и сроков выполнения работ, столь наглядное в графиках Ганта, используется и в сетевых графиках в так называемой форме «вершины-события».

В сетевых графиках в форме «вершины-события» работы отображаются дугами, а вершины отделяют работы друг от друга — их принято отождествлять с некоторыми событиями (например, приемкой выполнения работ). Дуги, исходящие из некоторой вершины, соответствуют работам, которые могут быть начаты только после того, как совершилось событие, представленное этой вершиной (т. е. закончены все работы, соответствующие стрелкам, входящим в эту вершину). На рис. 1.3.5 дано такое представление сетевого графика (идентичного графику рис. 1.3.4) в форме «вершины-события».

Из графика в форме «вершины-работы» можно всегда получить (очевидным преобразованием) график в форме «вершины-события». Понятно, что осуществим всегда и обратный переход. Неудобством второго представления является то, что для отражения очередности выполнения работ приходится вводить так называемые «фиктивные

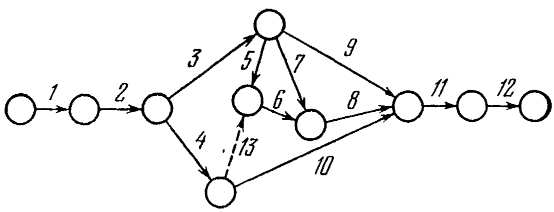

Рис. 1.3.5. Представление сетевого графика в форме «вершины-события».

работы» (работа *13* на рис. 1.3.5). Фиктивные работы вводят, в частности, для устранения случаев двух или нескольких параллельных дуг, выходящих из одной и той же и входящих в одну и ту же вершину (особенно если «приемка» каждой из работ может происходить отдельно).

3.3. Как ни наглядны бывают формы графического представления расписаний, все они обычно сопровождаются и некоторыми таблицами с численными характеристиками графиков. Во многих случаях *табличное представление* является общепринятым, в достаточной мере наглядным и не требующим дополнительных пояснений — расписание учебных занятий, футбольный календарь, расписание работы учреждений и т. п.

Для удобства пользования данные, характеризующие календарный план (особенно в случае их подготовки с использованием ЭВМ), выдают в нескольких разрезах. Так, отдельно группируется информация о динамике обработки каждой детали, о загрузке каждого рабочего места и т. п.

Табл. 1.3.1 представляет пример календарного плана работы производственного участка, выданного на буквенно-цифровую печать ЭВМ.

Таблица 1.3.1

Сменное задание

Цех 1 Число 3
Смена 2 Месяц 1

Номер				Время запуска партии	Длит. операции	Размер партии
станка	детали	операции	партии			
6	18 010 123	2	2	20.12	2.36	555
7	17 102 153	3	1	17.00	6.00	1170
8	17 102 151	3	3	17.00	6.00	1180
9	17 102 151	3	2	16.00	4.12	836
9	17 070 142	5	2	20.12	2.36	1650
19	18 230 118	3	1	22.30	0.18	33
20	17 102 150	7	1	17.12	2.24	2170
21	17 102 150	5	2	16.00	2.06	1850

3.4. В теории расписаний широко используется специфическая форма представления расписаний посредством задания *вектор-функций времени*. Эта форма представления расписаний, в основном, и используется в данной книге.

Так, для представления расписания рис. 1.3.1 достаточно задать вектор-функцию $s(t) = \{s_1(t), s_2(t)\}$, компоненты которой описывают загрузку рабочих мест № 1 и № 2 во времени. Если $s_1(t') = 0$, то в момент времени $t = t'$ рабочее место № 1 не загружено, на нем не выполняется ни одна из операций рассматриваемой совокупности. Аналогично, если $s_2(t') = 0$, то в момент времени $t = t'$ не загружено рабочее место № 2. Если $s_1(t') = k$, то в момент времени $t = t'$ на рабочем месте № 1 выполняется операция с номером k. Если $s_2(t') = l$, то на рабочем месте № 2 в момент времени $t = t'$ выполняется операция с номером l.

Например, $s_1(2) = 11$, $s_1(12) = 13$, $s_1(15) = 0$ и т. д.

Выбор той или иной формы представления расписаний определяется теми конкретными условиями, в которых возникает необходимость в их составлении и рассмотрении.

§ 4. Общая характеристика методов

4.1. В теории расписаний используется большинство известных в настоящее время идей и методов принятия наилучших решений посредством построения и анализа соответствующих операционных моделей. Прогресс в развитии методов решения экстремальных задач, методов статистических испытаний, эвристического программирования и т. д. в той или иной мере оказывал влияние и на развитие методов построения оптимальных расписаний. В рамках теории расписаний в свою очередь разрабатывались приемы и методы, получившие затем определенное распространение в иных разделах исследования операций.

Интересно отметить, что совершенствование методов теории расписаний протекало в условиях относительной стабильности анализируемых моделей. На протяжении двух десятилетий предлагались различные методы решения, по существу, одних и тех же классических сетевых задач, задач упорядочения операций обработки деталей машинами, весьма немногочисленных распределительных задач. Это явление ни в коем случае не следует объяснять отсутствием иных моделей. Модели теории расписаний непрерывно развиваются, совершенствуются, обретают определенную общность, полнее отражают конкретные ситуации календарного планирования. Объяснение скорее следует искать в том, что уже на стадии решения простейших по постановке задач возникли серьезные затруднения. Кроме того, хорошо зарекомендовавшие себя методы решения этих задач, как правило, допускают естественное распространение и обобщение на более сложные реальные задачи.

4.2. Становление теории расписаний как самостоятельного направления прикладной математики происходило в период, когда в достаточной мере был развит аппарат линейного программирования, исследованы многие линейные модели, получены обнадеживающие результаты в практическом использовании этих моделей и методов.

Совершенно естественно, что *методы линейного программирования* получили определенное распространение и в календарном планировании различных видов человеческой деятельности, особенно для решения задач временного согласования и распределения планируемых работ и ресурсов. Наряду с применением общих методов решения линейных задач начали разрабатываться и специальные методы, ориентированные на решения различных классов задач календарного планирования.

Первые же результаты по созданию эффективных методов решения линейных задач с условием *целочисленности* переменных значительно расширили круг анализируемых в теории расписаний линейных моделей. Появились определенные возможности учитывать комбинаторный характер плановых решений, неделимость отдельных видов работ и ресурсов и т. п.

Использование линейных моделей для описания реальных ситуаций даже в тех немногих случаях, когда это возможно, приводит к необходимости решения задач линейного программирования большого размера. В то же время специфические особенности этих задач позволяют предложить эффективные *декомпозиционные* подходы к их решению.

Нелинейные модели в теории расписаний получили относительно небольшое распространение, что связано, вероятно, с общим недостаточным развитием аппарата нелинейного, особенно невыпуклого программирования.

Аналогичная ситуация имеет место и с *вероятностными моделями*, хотя в направлении их анализа и предпринимаются весьма серьезные усилия.

4.3. Достаточно развитыми методами теории расписаний являются методы, основанные на идеях *последовательного конструирования, анализа и отсеивания вариантов* расписаний. Предложены многочисленные правила доминирования, позволяющие «отбраковывать» целые группы расписаний по отдельным построенным их фрагментам. Разработаны вычислительные схемы, в которых проводится систематическая оценка перспективности, предпочтительности одних расписаний относительно других конструируемых расписаний. Такие схемы, в частности, могут включать эвристические элементы (использования разнообразных функций предпочтения), некоторые способы обучения в сочетании с элементами статистического моделирования и т. п.

Для решения многих задач теории расписаний может быть непосредственно использован аппарат *динамического программирования.*

В простейших случаях оптимальные расписания могут быть построены в результате весьма простых рассуждений относительно изменения характеристик расписания при некоторых элементарных его преобразованиях. Совокупность такого рода приемов составляет основу так называемых *комбинаторных методов* теории расписаний. В результате их использования получены наиболее эффектные алгоритмы решения некоторых, правда, весьма немногих задач теории расписаний.

В *графических методах* прослеживается стремление использовать те же приемы локального «воздействия» на расписание, но с привлечением таких изобразительных средств, как графики, диаграммы, схемы.

Определенное распространение в теории расписаний получил и *теоретико-игровой подход*, особенно к анализу ситуаций с элементами неопределенности.

4.4. Среди приближенных методов теории расписаний в первую очередь следует отметить обширную группу *эвристических методов.* В этих методах в той или иной степени отражается опыт, накопленный в результате многократного построения расписаний в практически однотипных условиях. Область поиска наилучшего расписания при этом искусственно сужается, подчас до одного конкретного расписания. Эвристические методы нашли широкое применение при решении практических задач и позволяют получить относительно неплохие расписания при сравнительно небольшом объеме необходимых вычислений.

Несколько более ограниченное распространение получили различные вариации *метода статистических испытаний.* Этот метод позволяет построить расписание, близкое к искомому, в результате анализа некоторой совокупности случайным образом выбираемых расписаний. Обычно в вычислительных схемах предусматриваются возможности обучения, адаптации этих схем к решаемым классам задач.

Разнообразные *локальные методы* существенно используют понятие близости одного расписания к другому. В результате оказывается возможным введение понятия окрестности расписания как совокупности расписаний, наиболее близких к рассматриваемому. Локальные методы позволяют получить одно или несколько локально оптимальных расписаний с последующим выбором наилучшего из них.

Полезным приемом в теории расписаний оказалось исследование анализируемых процессов *в асимптотике*. При этом «гипертрофируются» те или иные особенности рассматриваемой ситуации. Предполагается, например, что процесс протекает бесконечно долго, что возможна сколь угодно частая смена состояний этого процесса и т. п. Получаемая в результате информация используется для построения близких к оптимальным расписаний протекания реальных процессов.

Необходимо отметить, что при разработке приближенных методов решения задач теории расписаний весьма нетривиальными являются вопросы оценки близости построенного расписания к искомому оптимальному.

§ 5. Библиографическая справка

Заметное влияние на выбор определения предмета теории расписаний и классификацию задач оказали работы Р. Беллмана [236], Р. Акофа и М. Сасиени [4]. Общее представление о задачах сетевого планирования и распределительных задачах можно получить по упомянутой монографии Р. Акофа и М. Сасиени и монографии А. Кофмана и Г. Дебазея [85].

Многие вопросы теории расписаний детально рассматриваются в монографиях В. В. Шкурбы, Т. П. Подчасовой, А. Н. Пшичук и Л. П. Тур [211], А. И. Семенова и В. М. Португала [156], Р. Конвея, В. Максвелла и Л. Миллера [254], Х. Мюллера-Мербаха [354], А. Л. Лурье [111], А. А. Воронова [39], Н. П. Бусленко [31], С. Ашура [220], С. А. Думлера [63], А. А. Корбута и Ю. Ю. Финкельштейна [83], Н. Б. Мироносецкого [118].

Достаточно полное описание методов решения задач упорядочения содержится в обзорных статьях Р. Л. Сиссона [405, 406], В. Я. Бурдюка и В. В. Шкурбы [26], В. Н. Буркова и С. Е. Ловецкого [28, 29], С. Е. Осколковой и И. О. Осколкова [130], П. Меллора [345]. Следует отметить также обзорные статьи [65, 117, 203, 205, 213, 227, 260, 271, 307, 389, 391].

ГЛАВА 2

ЭЛЕМЕНТЫ КОМБИНАТОРНОГО АНАЛИЗА

В большинстве ситуаций, расематриваемых в данной книге, построение оптимальных расписаний сопряжено с решением некоторых комбинаторных задач, сформулированных относительно заданных неупорядоченных или частично упорядоченных множеств. Эти задачи, как правило, состоят в отыскании наилучших «доупорядочений» заданных множеств либо наилучших разбиений их на подмножества.

В этой главе в весьма сжатой форме приводятся некоторые, необходимые в дальнейшем, сведения из теории графов, теории упорядоченных множеств, теории оптимизации функций на множествах перестановок.

§ 1. Множества, отношения, отображения, графы

1.1. Понятие множества широко используется не только в математике, но и других науках. Интуитивно под *множеством* принято понимать совокупность, объединение некоторых элементов, предметов, объектов. В различных науках обычно фиксируются некоторые множества предметов, свойства которых исследуются в данной науке. Например, в алгебре рассматриваются множества уравнений и многочленов, в геометрии — множества точек, прямых, плоскостей, в календарном планировании — множество работ, в антропологии — множество людей и т. д.

Понятие *отношения* является таким же первичным, как и понятие множества. Говоря, что элементы или множества находятся в некотором отношении, подразумевают установление между ними некоторой определенной связи, зависимости.

Примерами отношений являются отношения равенства, принадлежности (множеству), входимости (подмножества в множество).

Пожалуй, основными изучаемыми в математике отношениями являются отображения. *Отображение* Γ множества X во множество Y есть такое отношение, когда каждому элементу $x \in X$ ставится в соответствие множество элементов $\Gamma x \subseteq Y$. В этих обозначениях X — область определения отображения, Y — область его значений, Γx — образ элемента x. Отображение Γ^{-1}, обратное отображению Γ, определяется как такое отношение, когда каждому $y \in Y$ ставится в соответствие множество $\Gamma^{-1}y$ тех $x \in X$, для которых $y \in \Gamma x$, $\Gamma^{-1}y$ — прообраз y.

Если Γx для каждого x состоит из единственного элемента $y \in Y$, то отображение обычно называют оператором.

Операцией называют отображение подмножеств множества A в элементы множества A. Так, бинарная операция, или бинарная композиция, есть оператор, отображающий пары (a, b) элементов из A в элементы множества A.

Примерами бинарных операций могут служить арифметические операции, теоретико-множественные операции такие, как объединение и пересечение множеств.

1.2. Удобной и наглядной формой представления множеств с определенными на них отношениями являются графы.

Графом обычно принято называть следующую конструкцию. Заданы n точек, произвольно расположенных на плоскости. От некоторых точек (не обязательно от каждой) к некоторым другим точкам проведены стрелки. Можно считать, что стрелки определяют некоторое отображение точек в другие точки из числа заданных.

В общем случае считается, что задан граф, если заданы непустое множество X и отображение Γ множества X в X.

Граф мы будем обозначать через $G = (X, \Gamma)$.

Очевидно, любое так определенное конечное (со сравнительно небольшим количеством элементов) множество с заданным на нем отношением допускает приведенное выше геометрическое представление. Это представление (в некотором фрагменте) удобно бывает и для бесконечных множеств.

Частным (вырожденным) случаем графа является нуль-граф: нуль-граф состоит из изолированных элементов, иными словами, для любого x в нуль-графе $\Gamma x = \phi$.

В отличие от этого в полном графе для любого x множество $\Gamma x = X$ (полным называется и граф, для которого $\Gamma x = X \setminus \{x\}$).

В соответствии с графическим представлением принято называть элементы множества X точками или вершинами (узлами) графа́, а пары $u = (x, y)$ такие, что $x \in X$, $y \in \Gamma x$,— дугами графа. Множество всех дуг в графе будем обозначать через $\vec{U}$.

В дуге $u = (x, y)$ вершины x и y — граничные вершины, при этом x — начало дуги, y — конец дуги; говорят также, что дуга исходит из x и заходит в y. Дуга (x, y) — стрелка — имеет ориентацию, она направлена от x к y. Дуга (x, x) называется петлей.

Часто в задачах в отношении между образом и прообразом ориентация нас не интересует, важно лишь то, что вершины x и y соединены дугой. Будем говорить в этом случае, что вершины x и y соединены ребром $[x, y]$ (графически это соответствует тому, что точки соединяются линиями, а не стрелками). Множество ребер мы будем обозначать через U.

Графы с дугами называются *ориентированными*, графы с ребрами — *неориентированными*.

В рассмотрениях часто используется представление графа перечислением множества X его вершин и множества $\vec{U}$ дуг или множества U ребер.

Будем обозначать через $G = (X, \vec{U})$ ориентированные графы, через $G = (X, U)$ неориентированные графы, обозначение $G = (X, \Gamma)$ сохраним, в основном, для утверждений, справедливых для ориентированных и неориентированных графов.

Граф $G^1 = (X^1, \Gamma^1)$ будем называть *частичным графом* графа $G^2 = (X^2, \Gamma^2)$, если $X^1 = X^2 = X$ и для каждого $x \in X$ множество $\Gamma^1 x \subseteq \Gamma^2 x$ (множество дуг — или ребер — G^2 содержит множество дуг — или ребер — G^1).

Граф $G^1 = (X^1, \Gamma^1)$ называется *подграфом* графа $G^2 = (X^2, \Gamma^2)$, если $X^1 \subset X^2$ и $\Gamma^1 x = \Gamma^2 x \cap X^1$ (любая дуга — или ребро — (x, y) из G^2 содержится в графе G^1, если $x \in X^1$ и $y \in X^1$).

1.3. Понятие пути, контура, цепи, цикла, связности есть понятия некоторых отношений, порождаемых в X отношением Γ.

Дуги (ребра) u и v смежны, если они различны и имеют одну общую граничную вершину. Вершины x и y смежны, если они различны и соединены ребром или дугой.

Путем в графе $G = (X, \vec{U})$ называется последовательность дуг $u_1, u_2, \ldots, u_m$, когда конец каждой предшествующей дуги совпадает с началом следующей; иными словами, когда последовательность дуг имеет вид

$$(x_1, x_2), (x_2, x_3), \ldots, (x_{n-1}, x_n). \tag{1.1}$$

Здесь x_1 — начало пути, x_n — конец пути.

Контур — это путь, у которого $x_1 = x_n$; элементарный контур не содержит других совпадающих вершин, кроме x_1 и x_n.

Понятиям пути и контура в неориентированном графе соответствуют понятия цепи и цикла.

Граф $G = (X, U)$ называется *связным*, если любые две его вершины можно соединить цепью.

Компонентой связности графа G называется любой его подграф, который наряду с некоторой вершиной x содержит и все вершины, с которыми x в G соединена цепью. Связный граф состоит из одной компоненты связности. Различные компоненты связности графа образуют разбиение множества X.

Понятие связности распространяется и на ориентированные графы $G = (X, \vec{U})$. Для этого достаточно все дуги $(x, y) \in \vec{U}$ заменить ребрами $[x, y]$ и воспользоваться приведенными определениями.

Локальной степенью вершины графа называется число ребер (дуг), инцидентных этой вершине. Если граф ориентированный, то число дуг, исходящих (заходящих) из некоторой вершины, называется полустепенью исхода (захода) этой вершины.

1.4. Среди графов особое место занимают так называемые деревья и прадеревья.

Деревом называется связный неориентированный граф, имеющий не менее двух вершин и не содержащий циклов. В дереве любые две вершины связаны единственной цепью.

Деревьями часто называются и ориентированные графы, если они удовлетворяют перечисленным условиям без учета ориентации их дуг.

Связный ориентированный граф без контуров называется *прадеревом с корнем* x (x — вершина графа), если в вершину x не заходит ни одна дуга, а в каждую из остальных вершин заходит ровно по одной дуге.

Если ориентированный граф не содержит петель и контуров, то все его вершины можно распределить по *рангам* (слоям). При этом к первому рангу относят все те вершины исходного графа, полустепень захода которых равна нулю. Удаляя эти вершины вместе с инцидентными им дугами, получаем некоторый подграф. Если этот подграф не пуст, то все его вершины, полустепень захода которых равна нулю, относят ко второму рангу, удаляя их вместе с инцидентными им дугами. Процедура повторяется до тех пор, пока все вершины исходного графа не будут распределены по рангам.

§ 2. Упорядоченность

В этом параграфе приводятся необходимые сведения о свойствах упорядоченных множеств. Все рассматриваемые множества предполагаются конечными. Поскольку природа элементов этих множеств для нас безразлична, не нарушая общности, будем рассматривать множества натуральных чисел.

2.1. Пусть $N = \{1, 2, \ldots, n\}$. Бинарное отношение $\rightarrow$ на множестве N называется *отношением строгого порядка* (или *строгим порядком*), если оно антирефлексивно, т. е. ни для какого $i \in N$ не выполнено $i \rightarrow i$, и транзитивно, т. е. если $i \rightarrow j$ и $j \rightarrow k$, то $i \rightarrow k$. Отсюда следует, что это отношение антисимметрично, т. е. если выполнено $i \rightarrow j$, то невозможно $j \rightarrow i$.

Если $i \rightarrow j$, то говорят, что элемент i предшествует элементу j; если, кроме того, не существует k такого, что $i \rightarrow k$ и $k \rightarrow j$, то элемент i непосредственно предшествует элементу j. Элементы i и j называются несравнимыми, если не имеет место ни одно из соотношений $i \rightarrow j$ и $j \rightarrow i$.

Отношение строгого порядка $\rightarrow$ называется совершенным строгим порядком, если для любых несовпадающих i и j имеет место $i \rightarrow j$ или $j \rightarrow i$.

Множество N с заданным на нем отношением строгого порядка $\rightarrow$ называется *упорядоченным множеством*. Если отношение $\to$ — совершенный строгий порядок, то N — линейно упорядоченное множество. В противном случае N — частично упорядоченное множество.

Элемент $i \in N$ называется минимальным (максимальным), если не существует $j \in N$ такого, что $j \to i$ (соответственно $i \to j$). Остальные элементы называются промежуточными. Если N — линейно упорядоченное множество, то оно содержит ровно один максимальный, один минимальный и $n-2$ промежуточных элементов.

Упорядоченное множество N допускает простую графическую интерпретацию. Ориентированный граф $(N, \vec{U})$ называется *графом отношения строгого порядка* $\to$, заданного на множестве N, если $(i, j) \in \vec{U}$ в том и только в том случае, когда $i \to j$. Граф, очевидно, не содержит петель и контуров и является транзитивно замкнутым. Если отношение $\to$ — совершенный строгий порядок, то любая пара вершин графа соединена дугой.

Более распространенным является представление упорядоченного множества в виде графа $(N, \vec{U}_p)$ отношения, называемого редукцией отношения строгого порядка. Граф $(N, \vec{U}_p)$ отличается от графа $(N, \vec{U})$ тем, что он не содержит дуги (i, j) всякий раз, когда существует путь из i в j, отличный от дуги (i, j). Иными словами, дуга $(i, j) \in \vec{U}_p$ тогда и только тогда, когда i непосредственно предшествует j. Известно, что транзитивное замыкание редукции строгого порядка на конечном множестве совпадает с исходным порядком. Редукция является минимальным отношением, позволяющим восстановить исходное отношение строгого порядка в результате транзитивного замыкания. Граф $(N, \vec{U}_p)$, как правило, содержит значительно меньше дуг по сравнению с графом $(N, \vec{U})$.

В ряде случаев упорядоченное множество представляется неориентированным графом (N, U) при следующем дополнительном соглашении: если i непосредственно предшествует j, то вершина j на рисунке изображается выше вершины i и эти вершины соединяются ребром.

На рис. 2.2.1, *а* изображен граф $(N, \vec{U})$ строгого порядка $\to$, заданного на $N = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ условиями $4 \to 3$, $7 \to 9$, $3 \to 2$, $1 \to 3$, $2 \to 5$, $6 \to 7$, $6 \to 8$.

На рис. 2.2.1, *б* и рис. 2.2.1, *в* приведены графы $(N, \vec{U}_p)$ и (N, U).

2.2. Пусть на множестве N задано отношение строгого порядка $\rightarrow$. Подмножество $M \subseteq N$ называется совершенным, если отношение $\rightarrow$ на M — совершенный строгий порядок. Линейно упорядоченное множество M называется также *цепью*.

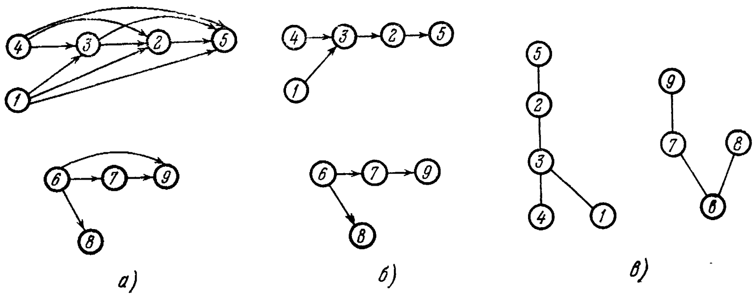

Рис. 2.2.1.

Если на любом подмножестве $M' \subseteq N$, содержащем совершенное множество M, отношение $\rightarrow$ уже не является совершенным строгим порядком, то множество M называется максимальным совершенным. Нетрудно убедиться, что для любого элемента $i \in N$ существует максимальное совершенное подмножество M множества N, содержащее этот элемент. Линейно упорядоченное множество M в этом случае называется максимальной цепью, содержащей элемент i. В частности, максимальная цепь может состоять из единственного элемента i.

В любом упорядоченном множестве могут быть выделены попарно непересекающиеся цепи, содержащие в совокупности все элементы этого множества. Множество таких цепей называется *D-разложением* упорядоченного множества.

Теорема 2.1. *Минимальное число цепей в D-разложении упорядоченного множества N равно максимальному числу s попарно несравнимых элементов этого множества.*

Доказательство. Если $s = 1$, то N — линейно упорядоченное множество и искомое D-разложение содержит одну цепь. Пусть утверждение справедливо для всех

$s \leqslant k-1$ и максимальное число попарно несравнимых элементов множества N равно k. Обозначим эти элементы через $i_1, i_2, \ldots, i_k$. Каждый элемент i_ν, $\nu = \overline{1, k}$, является некоторой цепью. Пусть вообще $C_1, C_2, \ldots, C_k$ — некоторые непересекающиеся цепи элементов множества N и $C = \bigcup\limits_{\nu=\overline{1,k}} C_\nu$.

Покажем, что если $N \diagdown C \neq \phi$ и $a \in N \diagdown C$, то существует D-разложение множества $C \bigcup \{a\}$, содержащее k цепей.

Пусть U_ν — множество всех тех элементов C_ν, которые предшествуют a; L_ν — множество всех тех элементов C_ν, которым предшествует a; P_ν — множество всех элементов C_ν, не сравнимых с a. Обозначим $U = \bigcup\limits_{\nu=\overline{1,k}} U_\nu$, $L = \bigcup\limits_{\nu=\overline{1,k}} L_\nu$, $P = \bigcup\limits_{\nu=\overline{1,k}} P_\nu$. Очевидно, $P \bigcup U \bigcup L = C$.

Теорема будет доказана, если удастся показать, что существуют такие $1 \leqslant m \leqslant k$ и $1 \leqslant l \leqslant k$, что множество $C' = C \diagdown \{U_m \bigcup L_l\}$ содержит не более $k-1$ попарно несравнимых элементов. Действительно, в этом случае минимальное D-разложение множества C', по предположению, будет содержать не более $k-1$ цепей, а D-разложение множества $C \bigcup \{a\}$ будет дополнительно содержать цепь $U_m \bigcup \{a\} \bigcup L_l$.

Предположим, что для любого $1 \leqslant \nu \leqslant k$ множество $A_\nu = P \bigcup U \diagdown U_\nu$ содержит k попарно несравнимых элементов $Q_\nu = \{q_1^\nu, q_2^\nu, \ldots, q_k^\nu\}$. Пусть эти элементы пронумерованы таким образом, что $q_i^\nu \in C_i$. Поскольку $q_\nu^\nu \notin U_\nu$, то $q_\nu^\nu \in P_\nu$. Следовательно, множество $Q = \bigcup\limits_{\nu=\overline{1,k}} Q_\nu$ содержит, по крайней мере, один элемент, принадлежащий C_ν и не сравнимый с a, для всех $\nu = \overline{1, k}$. Выберем среди всех элементов Q, принадлежащих C_ν и не сравнимых с a, максимальный элемент b_ν, $\nu = \overline{1, k}$. Покажем, что элементы $b_1, b_2, \ldots, b_k$ попарно несравнимы. Действительно, если $b_\alpha \rightarrow b_\beta$ и $b_\beta \in Q_r$, то $q_\alpha^r \rightarrow q_\beta^r$, что противоречит условию несравнимости элементов множества Q_r. Таким образом, элементы $b_1, b_2, \ldots, b_k$ попарно несравнимы и не сравнимы с элементом a. Однако, по предположению, множество N содержит не более k

попарно несравнимых элементов. Следовательно, существует такое $1 \leqslant m \leqslant k$, что множество A_m содержит не более $k-1$ попарно несравнимых элементов.

Аналогично можно показать, что существует такое $1 \leqslant l \leqslant k$, что множество $B_l = P \bigcup L \setminus L_l$ содержит не более $k-1$ попарно несравнимых элементов.

Любое подмножество T попарно несравнимых элементов множества $C' = A_m \bigcup B_l$ принадлежит A_m или (и) B_l. Действительно, если бы $x, y \in T$, $x \in U \setminus U_m$ и $y \in L \setminus L_l$, то $x \to a \to y$ и, следовательно, $x \to y$, что противоречит условию несравнимости этих элементов. Следовательно, и множество T, содержащее максимальное число попарно несравнимых элементов C', также принадлежит A_m или (и) B_l, и число этих элементов не превосходит $k-1$. Теорема доказана.

2.3. Для нахождения D-разложения с минимальным числом цепей (так же как и для нахождения максимального числа попарно несравнимых элементов) упорядоченного множества можно использовать методы линейного программирования.

Пусть $\{C_1, C_2, \ldots, C_l\}$ — некоторое D-разложение упорядоченного множества N и цепь C_ν имеет вид $i_1 \to \to i_2 \to \ldots \to i_r$. Сопоставим этому разложению набор значений $(n+1)^2$ переменных x_{ij}, $i, j = \overline{0, n}$, полагая $x_{00} = n - l$; $x_{0j} = 1$, если j является минимальным элементом некоторой цепи разложения (в частности, $x_{0i_1} = 1$); $x_{j0} = 1$, если j является максимальным элементом некоторой цепи разложения (в частности, $x_{i_r 0} = 1$); $x_{ij} = 1$, если в некоторой цепи разложения элемент i непосредственно предшествует j (в частности, $x_{i_k i_{k+1}} = 1$, $k = \overline{1, r-1}$); значения остальных переменных полагаем равными нулю.

Для того чтобы набору значений переменных x_{ij} соответствовало некоторое D-разложение, необходимо, чтобы выполнялись ограничения

$$\sum_{j=0}^{n} x_{0j} = \sum_{j=0}^{n} x_{j0} = n, \tag{2.1}$$

$$\sum_{j=0}^{n} x_{ij} = \sum_{j=0}^{n} x_{ji} = 1, \quad i = \overline{1, n}, \tag{2.2}$$

$$x_{ij} \geqslant 0, \quad i, j = \overline{0, n}. \tag{2.3}$$

Если, кроме того, потребовать, чтобы переменные x_{ij}, за возможным исключением x_{00}, принимали только значения 0 и 1, причем каждый раз, когда i не предшествует j в N, значение $x_{ij} = 0$, то оказывается возможным установить взаимно однозначное соответствие между множеством всех D-разложений и множеством наборов значений переменных x_{ij}.

Введем в рассмотрение матрицу $\| c_{ij} \|$, $i, j = \overline{0, n}$, с элементами $c_{00} = 1$, $c_{0j} = c_{j0} = 0$ для всех $j = \overline{1, n}$, $c_{ij} = 0$, если $i \to j$ в N, и $c_{ij} = -\infty$ в противном случае для всех $i, j = \overline{1, n}$. Рассмотрим задачу максимизации линейной формы

$$\sum_{i,\, j=0}^{n} c_{ij} x_{ij} \tag{2.4}$$

при ограничениях (2.1) — (2.3). Эта задача является частным случаем транспортной задачи линейного программирования, у которой существует целочисленное решение. Вместе с тем всякий раз, когда $c_{ij} = -\infty$, естественно полагать $x_{ij} = 0$.

Следовательно, задача построения D-разложения с наименьшим числом цепей сводится к транспортной задаче, для решения которой известны весьма эффективные методы.

2.4. Для разработки алгоритмов построения минимизированных D-разложений желательно располагать критериями, которые позволяли бы установить, существуют или нет D-разложения с минимальным числом цепей, содержащие в качестве составляющей данную цепь C. При этом естественно рассматривать максимальные цепи C.

Пусть s — максимальное число попарно несравнимых элементов множества N с заданным на нем порядком $\to$, а $C = (i_1, i_2, \ldots, i_r)$ — некоторая максимальная цепь этого множества.

Если множество $N \setminus C$ содержит не более $s - 1$ попарно несравнимых элементов, то в силу теоремы 2.1 указанное разложение, очевидно, существует.

В ряде случаев более удобным оказывается иной подход.

«Уплотним» цепь C введением новых элементов, отличных от элементов множества N. Если в цепи C элемент i непосредственно предшествует элементу j, то введем новый элемент k и упорядочим их так, чтобы $i \to k \to j$. Эта операция проводится для всех пар (i, j) элементов исходной цепи C. В результате получаем уплотненную цепь $\tilde{C} = (i_1, k_1, i_2, k_2, i_3, \ldots, k_{r-1}, i_r)$ и расширенное упорядоченное множество $\tilde{N} = N \cup \tilde{C}$. Обозначим максимальное число попарно несравнимых элементов множества $\tilde{N}$ через $\tilde{s}$. Очевидно, $\tilde{s} \geqslant s$.

Теорема 2.2. *Для того чтобы D-разложение с минимальным числом цепей содержало данную максимальную цепь C, необходимо и достаточно, чтобы $\tilde{s} = s$.*

Доказательство. Пусть $\{C_1, C_2, \ldots, C_s\}$ — минимальное по числу цепей D-разложение множества N, содержащее цепь C. Заменяя в этом разложении цепь C уплотненной цепью $\tilde{C}$, получаем D-разложение множества $\tilde{N}$, содержащее s цепей. В силу теоремы 2.1 значение $\tilde{s} \leqslant s$ и, следовательно, $\tilde{s} = s$.

Пусть $\tilde{s} = s$. Покажем, что множество $\tilde{N} \setminus \tilde{C}$, или что то же $N \setminus C$, содержит не более $s - 1$ попарно несравнимых элементов. Предположим, что оно содержит s попарно несравнимых элементов, и обозначим множество этих элементов через H. Если $j \in \tilde{C}$, то элемент j сравним хотя бы с одним элементом множества H. Поскольку $\tilde{C}$ — максимальная цепь, то не существует $h \in H$ такого, что $h \to i_1$ или $i_r \to h$. Следовательно, существуют $j', j'' \in C$ такие, что $k_1 \to j' \to h_1$ и $h_2 \to j'' \to k_{r-1}$, где $h_1 h_2$ — некоторые элементы множества H. Среди j' и j'', удовлетворяющих указанным условиям, выберем соответственно максимальный и минимальный элементы.

Если $j'' \to j'$, или $j'' = j'$, то $h_2 \neq h_1$ и $h_2 \to h_1$, что противоречит условию несравнимости элементов множества H.

Если $j' \to j''$, то по построению существует элемент $k_l \in \tilde{C} \setminus C$, удовлетворяющий условию $j' \to k_l \to j''$ и не сравнимый ни с одним элементом множества H. Это противоречит условию, что множество $\tilde{N}$ содержит не более s попарно несравнимых элементов.

Следовательно, множество $N \setminus C$ содержит не более $s - 1$ попарно несравнимых элементов, и исходное множество N может быть разложено на s цепей, одной из которых является цепь C. Теорема доказана.

2.5. Наряду с разложением на цепи в теории расписаний рассматриваются разнообразные «доупорядочения» частично упорядоченных множеств.

Пусть $(N, \rightarrow)$ — упорядоченное множество N с заданным на нем отношением строгого порядка $\rightarrow$. Будем говорить, что $(N, \rightarrow)$ *порождает* $(N, \Rightarrow)$-упорядоченное множество N с заданным на нем отношением строгого порядка $\Rightarrow$, если из условия $i \rightarrow j$ следует $i \Rightarrow j$. В частности, множество $(N, \rightarrow)$ порождает само себя.

В практическом отношении желательно располагать способом построения всех линейно упорядоченных множеств, порождаемых данным частично упорядоченным множеством, а также возможностями оценки или непосредственного подсчета их числа. Иными словами, нас будут интересовать вопросы, связанные с построением перестановок элементов множества N, удовлетворяющих заданному на N отношению строгого порядка $\rightarrow$. Перестановка $(i_1, i_2, \ldots, i_n)$ элементов множества N называется *допустимой* относительно заданного на N отношения строгого порядка $\rightarrow$, если из условия $i_\nu \rightarrow i_\mu$ следует $\nu < \mu$.

Процедура построения допустимой перестановки может быть описана следующим образом. Пусть отмечено $l \geqslant 0$ элементов множества N, элемент i не отмечен и не существует другого неотмеченного элемента j такого, что $j \rightarrow i$. Выберем элемент i в качестве $(l+1)$-го элемента искомой перестановки. Отметим элемент i. Повторяя эту процедуру n раз, получаем, очевидно, допустимую относительно $\rightarrow$ перестановку элементов множества N. Если на каждом шаге принимать все альтернативные решения, то можно построить все допустимые перестановки. Обозначим число таких перестановок через p.

Множество N с учетом заданного на нем отношения строгого порядка $\rightarrow$ можно представить, вообще говоря, в виде объединения нескольких, для определенности $m+1$, попарно непересекающихся подмножеств N_0, $N_1, \ldots, N_m$. Множество N_0 содержит все те элементы $i \in N$, для которых выполняется условие: не существует $j \in N$ такого, что $i \rightarrow j$ или $j \rightarrow i$. Множество N_k, $k = \overline{1, m}$, содержит все те элементы $i \in N$, для которых выполняется условие: существует $j \in N_k$ такое, что $i \to j$ или $j \to i$, и не существует таких $i \in N_k$ и $j \in N \setminus N_k$, что $i \to j$ или $j \to i$. Графически множество N_0 представляет собой совокупность изолированных вершин графа отношения строгого порядка, а множества N_k, $k = \overline{1, m}$,— компоненты связности этого графа, каждая из которых содержит более одной вершины.

Обозначим через p_k множество перестановок элементов множества N_k, удовлетворяющих заданному на нем отношению строгого порядка. Очевидно, $p_0 = n_0!$. Здесь n_k — число элементов множества N_k, $k = \overline{0, n}$.

Пусть r_j, $j \in N_k$,— число элементов $i \in N_k$ таких, что $i \to j$; R_k — множество различных значений r_j; $l_k(r)$ и $L_k(r)$ — число элементов $j \in N_k$, для которых $r_j = r$ и $r_j > r$ соответственно.

Теорема 2.3. *Число допустимых перестановок элементов упорядоченного множества N равно*

$$p = n! \prod_{k=\overline{0, m}} \frac{p_k}{n_k!}, \quad где \quad p_k \leqslant \prod_{r \in R_k} A_{n_k - r - L_k(r)}^{l_k(r)}. \qquad (2.5)$$

Доказательство. Если $m = 0$, то $p = p_0 = n!$. Пусть $m > 0$. Допустимую относительно заданного отношения строгого порядка перестановку элементов множества N_m можно разместить на n возможных линейно упорядоченных позициях $C_n^{n_m}$ способами. Число различных допустимых перестановок элементов множества N_m равно p_m. Предполагая справедливость первой формулы при всех $k < m$, имеем

$$p = C_n^{n_m} \cdot p_m (n - n_m)! \prod_{k=\overline{0, m-1}} \frac{p_k}{n_k!} = n! \prod_{k=\overline{0, m}} \frac{p_k}{n_k!}. \qquad (2.6)$$

Пусть, для определенности, $R_k = \{r_1 < r_2 < \ldots < r_\mu\}$. Среди n_k линейно упорядоченных позиций элементы множества N_k с $r_j = r_\mu$ не могут занимать первых r_μ позиций. Таким образом, число возможных размещений этих элементов не превосходит величины $A_{n_k - r_\mu}^{l_k(r_\mu)}$. Если элементы с $r_j = r_\mu$ размещены, то, поскольку элементы с $r_j = r_{\mu-1}$ не могут занимать первых $r_{\mu-1}$ позиций, число возможных размещений элементов с $r_j = r_{\mu-1}$ не превосходит величины $A_{n_k-r_\mu-r_{\mu-1}}^{l_k(r_{\mu-1})} = A_{n_k-r_{\mu-1}-L_k(r_{\mu-1})}^{l_k(r_{\mu-1})}$. Следовательно, число возможных размещений элементов множества N_k с $r_j = r_{\mu-1}$ и $r_j = r_\mu$ не превосходит величины $\prod\limits_{r=r_{\mu-1},\, r_\mu} A_{n_k-r-L_n(r)}^{l_k(r)}$. Повторяя конечное число раз эти рассуждения, убеждаемся в справедливости второй формулы. Теорема доказана.

Для вычисления значения p_k целесообразно использовать рекуррентный метод, основанный на следующем утверждении, справедливость которого непосредственно следует из описанного выше способа построения допустимых перестановок.

Пусть $\underline{N}_k$ $(\overline{N}_k)$ — множество минимальных (максимальных) элементов множества N_k. Обозначим через $p_k(i)$ число допустимых перестановок элементов множества $N_k \setminus \{i\}$. Тогда $p_k = \sum\limits_{i \in \underline{N}_k} p_k(i) = \sum\limits_{i \in \overline{N}_k} p_k(i)$. Применяя эту формулу к исходному множеству N_k, затем к каждому из множеств, образуемых из N_k удалением одного из минимальных (максимальных) элементов и т. д., получаем в итоге значение p_k. При этом естественно использовать первую часть теоремы 2.3.

§ 3. Смешанные (дизъюнктивные) графы

Смешанные графы получили широкое распространение для описания ситуаций, в которых отношение порядка необходимо распространить не на все, а на отдельные пары ранее несравнимых элементов частично упорядоченных множеств.

3.1. Пусть $(N, \to)$ — конечное частично упорядоченное множество N с заданным на нем строгим порядком $\to$, а B — некоторая совокупность пар несравнимых элементов этого множества.

Множество $(N, \Rightarrow)$ порождается множеством $(N, \to)$, если из условия $i \to j$ следует $i \Rightarrow j$. Среди всех множеств, порождаемых множеством $(N, \to)$, выделим множества $(N, \Rightarrow)$, удовлетворяющие условию: если пара $\{i, j\} \in B$, то $i \Rightarrow j$ или $j \Rightarrow i$. Обозначим совокупность таких множеств через $\tilde{K}$. Будем говорить, что множество $(N, \Rightarrow)$

порождается множеством $(N, \rightarrow)$ *по совокупности* B, если оно принадлежит $\tilde{K}$ и не порождается никаким другим множеством из $\tilde{K}$. Совокупность этих множеств обозначим через K. В частности, если B — совокупность всех пар несравнимых элементов множества $(N, \rightarrow)$, то K — совокупность всех линейно упорядоченных множеств, порождаемых множеством $(N, \rightarrow)$.

Аналогичным образом можно ввести понятие множества, порождаемого неупорядоченным множеством N по совокупности B.

Описанная ситуация графически может быть представлена следующим образом. Пусть $(N, \vec{U})$ — граф отношения $\rightarrow$. Соединим вершины i и j этого графа (неориентированным) ребром $[i, j]$ всякий раз, когда пара $\{i, j\} \in B$. Заметим, что в этом случае граф $(N, \vec{U})$ не содержит дуги, соединяющей вершины i и j. В результате получаем *смешанный* граф $G = (N, \vec{U}, U)$, содержащий множество вершин N, множество дуг $\vec{U}$ и множество ребер U.

Заменим в этом графе каждое ребро $[i, j] \in U$ одной из дуг (i, j) или (j, i). Если полученный ориентированный граф не содержит контуров, то, используя операцию транзитивного замыкания, получаем граф $(N, \vec{\vec{U}})$ искомого отношения $\Rightarrow$.

Множество $(N, \Rightarrow)$ принадлежит совокупности K. Действительно, множество $(N, \Rightarrow)$ порождается множеством $(N, \rightarrow)$, поскольку граф $(N, \vec{U})$ является частичным подграфом графа $(N, \vec{\vec{U}})$. Для всех пар $\{i, j\} \in B$ имеет место $i \Rightarrow j$ или $j \Rightarrow i$, поскольку граф $(N, \vec{\vec{U}})$ содержит одну из дуг (i, j) или (j, i). Множество $(N, \Rightarrow)$, наконец, не порождается ни одним из множеств совокупности K, отличным от $(N, \Rightarrow)$, так как граф $(N, \vec{\vec{U}})$ не содержит «лишних» дуг.

С другой стороны, каждому множеству из K соответствует некоторый ориентированный граф без контуров, полученный из графа G в результате замены каждого его ребра дугой.

Таким образом, существует взаимно однозначное соответствие между элементами множества K и бесконтурными графами, порождаемыми смешанным графом G.

Смешанный граф G иногда называют *дизъюнктивным* графом. Это название непосредственно связано с операцией замены ребра дугой, имеющей два возможных взаимно исключающих исхода.

3.2. Пусть $G = (N, \vec{U}, U)$ — некоторый смешанный граф (без петель, кратных дуг и ребер). Обозначим через $\tilde{K}_G$ множество всех ориентированных графов, порождаемых данным смешанным графом в результате замены всех его ребер дугами, а через K_G — множество тех графов из $\tilde{K}_G$, которые не содержат контуров.

Очевидно, $|\tilde{K}_G| = 2^{|U|}$ и $|K_G| \neq 0$ тогда и только тогда, когда граф $(N, \vec{U})$ не содержит контуров.

Построение графов множества K_G не вызывает особых затруднений.

Пусть $G^{(l)}$ — смешанный граф, у которого $l \geqslant 0$ вершин отмечено и существует неотмеченная вершина i, в которую не заходит ни одной дуги, исходящей из неотмеченной вершины. Отметим вершину i и в случае, если в $G^{(l)}$ есть неориентированные ребра, инцидентные этой вершине, заменим их исходящими из нее дугами. Полученный в результате граф обозначим через $G^{(l+1)}$.

Нетрудно убедиться, что любой элемент множества K_G может быть получен из исходного графа $G^{(0)} = G$ конечным числом указанных преобразований.

Многие задачи теории расписаний формулируются как задачи нахождения во множестве K_G графа или графов, обладающих определенными экстремальными свойствами. При организации процедур поиска таких графов целесообразно располагать информацией о количественных характеристиках и свойствах множества K_G.

3.3. В качестве верхней оценки числа $\rho(G) = |K_G|$ элементов K_G может быть выбрано число перестановок, допустимых относительно заданного на N графом G отношения строгого порядка $\rightarrow$. При этом непосредственно используется теорема 2.3 § 2 гл. 2.

Способ вычисления нижней оценки величины $\rho(G)$ основан на систематическом использовании следующего, в достаточной мере очевидного приема.

Пусть $\mathfrak{N} = \{N_1, N_2, \ldots, N_R\}$ — упорядоченное разбиение множества N на попарно непересекающиеся непустые подмножества, удовлетворяющие условию: если

$i \in N_p$ и $j \in N_q$, то $p < q$ для любой дуги $(i, j) \in \vec{U}$. В качестве $\mathfrak{N}$ может быть выбрано, например, разбиение множества N по рангам относительно графа $(N, \vec{U})$ или разбиение N по числу вершин, из которых существует путь в данную вершину в графе $(N, \vec{U})$.

Обозначим через $G_l = (N_l, U_l)$ подграф графа G с множеством вершин N_l, а через $G_l^k = (N_l^k, U_l^k)$ — компоненты связности этого подграфа, $l = \overline{1, h}$, $k = \overline{1, s_l}$. Очевидно, G_l, а следовательно, и G_l^k не содержат дуг множества $\vec{U}$. Разбиение $\mathfrak{N}$ и граф G порождают граф $G^* = (N, \vec{U}^*, U^*)$, получаемый из G в результате замены ребер $[i, j] \in U$ дугами (i, j) всякий раз, когда $i \in N_p$, $j \in N_q$ и $p < q$. Граф $(N, \vec{U}^*)$ не содержит контуров.

Для любого связного неориентированного графа $(\tilde{N}, \tilde{U})$ имеет место соотношение

$$\rho(\tilde{N}, \tilde{U}) \geqslant 2^{|\tilde{N}|-1}. \tag{3.1}$$

Действительно, максимальное дерево в таком графе содержит $|\tilde{N}| - 1$ ребер. Поскольку дерево не содержит циклов, то число способов бесконтурной ориентации его ребер равно правой части неравенства (3.1). Каждой ориентации ребер дерева соответствует, по крайней мере, одна бесконтурная ориентация ребер исходного графа.

Таким образом,

$$\log_2 \rho(G^*) \geqslant \sum_{k,\, l} (|N_l^k| - 1). \tag{3.2}$$

Среди дуг множества $\vec{U}^* \setminus \vec{U}$ выделим дуги (i, j), для каждой из которых в графе $(N, \vec{U}^*)$ не существует пути из i в j, отличного от дуги (i, j). Обозначим множество всех таких дуг через $\vec{U}_T^*$. Построим $|\vec{U}_T^*|$ графов: $G_1^* = (N, \vec{U}_1^*, U^*)$, $G_2^* = (N, \vec{U}_2^*, U^*)$, ..., $G_{|\vec{U}_T^*|}^* = (N, \vec{U}_{|\vec{U}_T^*|}^*, U^*)$, отличающихся от G^* ориентацией одной из дуг множества $\vec{U}_T^*$. Графы $(N, \vec{U}_k^*)$, $k = 1, 2, \ldots, |\vec{U}_T^*|$, не содержат контуров по построению и

$$\rho(G) \geqslant \rho(G^*) + \sum_{r=1}^{|\vec{U}_T^*|} \rho(G_r^*). \tag{3.3}$$

Если в качестве исходного графа G выбирать последовательно графы $G_1^*, G_2^*, \ldots, G^*_{|\vec{U}_T^*|}$, то описанная процедура позволяет получить нижние оценки значений $\rho(G_r^*)$. При этом получаем соотношения, аналогичные (3.3), и производные графы с меньшим числом неориентированных ребер, для которых в свою очередь может быть применена описанная процедура.

Пусть дуга $(i, j) \in \vec{U}_T^*$, $i \in N_p^s$, $j \in N_q^t$, $p < q$, и G_r^* — граф, полученный из G^* заменой этой дуги дугой (j, i). Нетрудно видеть, что

$$\log_2 \rho(G_r^*) \geqslant \sum_{l \leqslant p,\, k} (|N_l^k| - 1) + \\ + \sum_{l \geqslant q,\, k} (|N_l^k| - 1) - |N_p^s| - |N_q^t| + 2. \tag{3.4}$$

Соотношение (3.4) позволяет ограничить итерационный процесс получения нижних оценок значения $\rho(G)$ любым заранее выбранным числом шагов.

3.4. Для вычисления точного значения величины $\rho(G)$ заменим в G каждое ребро $[i, j] \in U$ парой дуг (i, j) и (j, i) и выделим в полученном ориентированном графе все элементарные контуры, длина которых больше двух, $\mu_1, \mu_2, \ldots, \mu_\lambda$ (список 1). Пусть $\tilde{\mu}_1$ — контур с наименьшим числом ребер, содержащий ребра, наиболее часто встречающиеся в контурах списка 1. Заменим в G каждое ребро, принадлежащее $\tilde{\mu}_1$, дугой. Если $\tilde{\mu}_1$ содержит v_1 ребер, то в результате получим 2^{v_1} различных графов $G_k = (N, \vec{U}_k, U_k)$, $k = 1, 2, \ldots, 2^{v_1}$. Пусть, для определенности, контур $\tilde{\mu}_1$ содержится в графе $G_{2^{v_1}}$. Число графов множества $\tilde{K}_G$, содержащих контур $\tilde{\mu}_1$, равно, очевидно, $2^{|U_{2^{v_1}}|}$.

Отметим графы G_k, $k = 1, 2, \ldots, 2^{v_1} - 1$. Для каждого из этих графов построим список контуров, удаляя из списка 1 соответствующие контуры. Таким образом, на первом шаге оказывается рассмотренным исходный граф G, отмеченными — графы $G_1, G_2, \ldots, G_{2^{v_1}-1}$ и отобранным — граф $G_{2^{v_1}}$. На втором шаге аналогичным образом рассматривается граф G_1. В результате этот граф оказывается рассмотренным, некоторые графы $G_{2^{v_1}+1}, \ldots$ $\ldots, G_{2^{v_1+v_2}-1}$ — отмеченными и некоторый граф $G_{2^{v_1+v_2}}$ — отобранным и содержащим контур $\widetilde{\mu}_2$. После выполнения m шагов имеем

$$\rho(G) = 2^{|U|} - \sum_{i=1}^{m} 2^{|U_{2^{w_i}}|} - \sum_{q \in Q} [2^{|U_q|} - \rho(G_q)], \qquad (3.5)$$

где $w_i = \sum_{j=1}^{i} v_j$; Q — множество индексов отмеченных, но не рассмотренных до m-го шага включительно графов; $\rho(G_q)$ — число бесконтурных графов, порождаемых графом G_q в результате замены всех его ребер дугами.

При реализации этого метода в ряде случаев оказывается целесообразным выделять контуры в графах последовательно по мере выполнения шагов алгоритма.

3.5. Известно, что любая из двух перестановок элементов множества N может быть получена из другой конечным числом транспозиций. Аналогичная ситуация имеет место и в случае, когда перестановки являются допустимыми относительно заданного на N строгого порядка.

Пусть $G' = (N, \vec{U}')$ и $G'' = (N, \vec{U}'')$ — различные графы из K_G. Обозначим через $\varepsilon(G', G'')$ — число ребер $[i, j] \in U$ таких, что $(i, j) \in \vec{U}'$, а $(j, i) \in \vec{U}''$. Пусть $E(G', G'')$ — наименьшая из абсолютных величин разностей рангов вершин i и j, вычисленных для всех указанных пар $\{i, j\}$ относительно графа G'.

Теорема 3.1. *Если $G', G'' \in K_G$ и $\varepsilon(G', G'') = m$, то существует такая последовательность графов $G' = G_0, G_1, \ldots, G_m = G''$ из K_G, что $\varepsilon(G_{l-1}, G_l) = 1$ и $E(G_{l-1}, G_l) = E(G_{l-1}, G'')$, $l = \overline{1, m}$. Если множество $U = \phi$, то существует последовательность графов $G' = G_0, G_1, \ldots, G_r = G''$ из K_G такая, что $\varepsilon(G_{l-1}, G_l) = E(G_{l-1}, G_l) = 1$, $l = \overline{1, r}$.*

Доказательство. Пусть граф G' содержит дугу (a, b), граф G'' — дугу (b, a), граф G_1 получен из графа G' в результате замены дуги (a, b) дугой (b, a), причем $E(G', G_1) = E(G', G'')$. Для доказательства первой части теоремы достаточно показать, что $G_1 \in K_G$. Допустим от противного, что существует контур в графе G_1, а следовательно, не содержащий дуги (a, b) путь из a в b в графе G'. Поскольку абсолютные величины разности рангов вершин, инцидентных каждой дуге этого пути, меньше чем $E(G', G'')$, то в графе G'' существует тот же путь из a в b и, следовательно, G'' не принадлежит K_G. Получаем противоречие.

Для доказательства второй части теоремы достаточно рассмотреть случай, когда $\varepsilon(G', G'') = 1$. Пусть графы G' и G'' отличаются ориентацией ребра $[a, b]$, причем G' содержит дугу (a, b), а граф G'' — дугу (b, a). Возможны два случая: или граф G' содержит вершину $c \neq b$, полустепень исхода которой равна нулю, или он не содержит такой вершины.

В первом случае предположим, что теорема справедлива при числе вершин, равном $n \geqslant 3$, и докажем ее справедливость при числе вершин, равном $n + 1$. При $n = 2$ теорема очевидна.

Удалим из G' и G'' вершину c вместе с инцидентными ей дугами. Обозначим полученные частичные подграфы соответственно через $\overline{G}'$ и $\overline{G}''$. По предположению существует последовательность бесконтурных графов $\overline{G}' = \overline{G}_0, \overline{G}_1, \ldots$ $\ldots, \overline{G}_r = \overline{G}''$ таких, что $\varepsilon(\overline{G}_{l-1}, \overline{G}_l) = E(\overline{G}_{l-1}, \overline{G}_l) = 1$, $l = \overline{1, r}$. Присоединим вершину c вместе с инцидентными ей дугами к каждому графу этой последовательности. Полученная последовательность является искомой.

Во втором случае единственной вершиной, полустепень исхода которой равна нулю, является вершина b. Выберем в G' вершину d, которая связана с b дугой и для которой $R(b) - R(d) = 1$, где $R(x)$ — ранг вершины x в G'. Будем рассматривать нетривиальный случай, когда $R(b) - R(a) > 1$ и, следовательно, $d \neq a$. Обозначим через $\widetilde{G}'$ и $\widetilde{G}''$ графы, отличающиеся соответственно от G' и G'' только ориентацией ребра $[b, d]$. Нетрудно видеть, что $\widetilde{G}'$ и $\widetilde{G}''$ не содержат контуров, отличаются ориентацией ребра $[a, b]$ и содержат вершину $d \neq b$, полустепень исхода которой равна нулю. Следовательно, к этим графам применима доказанная часть теоремы. Аналогичное заключение может быть сделано относительно бесконтурных графов $\tilde{G}''$ и G''. Действительно, они отличаются ориентацией ребра $[b, d]$, и поскольку в них не существует пути, соединяющего вершины a и d, то эти графы содержат вершину $e \neq d$, полустепень исхода которой равна нулю. Теорема доказана.

§ 4. Перестановки. Задачи упорядочения. Перестановочный прием

4.1. Понятие *перестановки* хорошо знакомо — расположение элементов некоторого множества в любом порядке (без повторений) представляет собой перестановку элементов этого множества.

Перестановка обычно представляется последовательностью элементов (чисел) из $N = \{1, 2, \ldots, n\}$. Символической записью этой конструкции является $\pi = (i_1, i_2, \ldots$ $\ldots, i_n)$ или $\pi = ([1], [2], \ldots, [n])$, где i_k или $[k]$ — элемент из N, занимающий в последовательности π k-е слева место.

Перестановка может быть представлена таблицей вида $\{(i, j)\}$, отражающей тот факт, что в перестановке элемент $i \in N$ занимает j-е место.

Определенное распространение получило представление перестановки в виде матрицы $\| x_{ij} \|$ размерности $n \times n$, элементами которой являются числа 0 или 1. Значение $x_{ij} = 1$ при этом соответствует записи (i, j). В этой матрице сумма элементов, стоящих в каждой строке и каждом столбце, равна 1.

Под *задачей упорядочения* понимается, как правило, задача построения одной или нескольких перестановок, удовлетворяющих определенным ограничениям и доставляющих экстремум некоторой функции или функциям, определенным на рассматриваемом множестве перестановок.

4.2. В этом параграфе мы рассмотрим сравнительно простые задачи упорядочения, решение которых может быть получено с использованием так называемого *перестановочного приема*. Суть этого приема заключается в предварительном исследовании влияния взаимного расположения соседних элементов перестановки на значение оптимизируемой функции. В ряде случаев это позволяет резко сузить множество возможных решений.

Последнее характерно для ситуаций, в которых оптимизируемой функции $f(\pi)$, заданной на множестве всех перестановок элементов множества N, оказывается возможным сопоставить некоторую функцию $\omega(i)$, $i \in N$, обладающую следующим свойством. Если $\pi = (i_1, i_2, \ldots, i_k, i_{k+1}, \ldots, i_n)$ и $\omega(i_k) \geqslant \omega(i_{k+1})$, то $f(\pi) \leqslant f(\pi')$, где π' отличается от π только транспозицией элементов i_k и i_{k+1}.

Для построения перестановки π^*, которой соответствует наименьшее значение $f(\pi)$, в этом случае достаточно упорядочить элементы N в порядке невозрастания значений $\omega(i)$, $i \in N$.

Действительно, в противном случае осуществляя транспозицию элементов i_k и i_{k+1}, всякий раз, когда $\omega(i_{k+1}) > \omega(i_k)$, приходим к перестановке π^*, и $f(\pi^*)$ не превосходит значения, соответствующего исходной перестановке.

4.3. Пусть даны два n-мерных вектора $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ и $\beta = (\beta_1, \beta_2, \ldots, \beta_n)$, компоненты которых — действительные числа. Обозначим через $\pi = (i_1, i_2, \ldots, i_n)$ некоторую перестановку элементов множества $N = \{1, 2, \ldots, n\}$. Определим

$$f_1(\pi) = \sum_{k=1}^{n} \alpha_k \beta_{i_k}. \tag{4.1}$$

Требуется построить перестановку π_1^* элементов множества N, которой соответствует наименьшее значение $f_1(\pi)$.

Не нарушая общности, будем предполагать, что компоненты векторов α и β пронумерованы так, что $\alpha_1 \leqslant \alpha_2 \leqslant \ldots \leqslant \alpha_n$.

Рассмотрим перестановку π', отличающуюся от π транспозицией элементов i_k и i_{k+1}. Имеем

$$f_1(\pi') - f_1(\pi) = (\alpha_k - \alpha_{k+1})(\beta_{i_{k+1}} - \beta_{i_k}). \tag{4.2}$$

Если $\beta_{i_k} \geqslant \beta_{i_{k+1}}$, то эта разность неотрицательна.

Таким образом, упорядочивая числа β_l в порядке их невозрастания, получаем искомую перестановку

$\pi_1^* = (i_1^*, i_2^*, \ldots, i_n^*)$. Здесь $\beta_{i_k^*} \geqslant \beta_{i_{k+1}^*}$ для всех $k = \overline{1, n-1}$.

Теорема 4.1. *Функция $f_1(\pi)$ достигает наименьшего значения тогда и только тогда, когда меньшим α_k соответствуют большие β_{i_k}.*

4.4. Рассмотрим задачу построения последовательности π_2^*, которой соответствует наименьшее значение функции

$$f_2(\pi) = \max_{1 \leqslant u \leqslant n} \left(\sum_{k=1}^{u} \alpha_{i_k} + \beta_{i_u} \right). \tag{4.3}$$

Прежде всего найдем достаточные условия, при которых $f_2(\pi) \leqslant f_2(\pi')$, где π' отличается от π транспозицией элементов, занимающих k-ю и $(k+1)$-ю позиции, т. е. если $\pi = (i_1, i_2, \ldots, i_{k-1}, i_k, i_{k+1}, i_{k+2}, \ldots, i_n)$, то $\pi' = (i_1, \ldots, i_{k-1}, i_{k+1}, i_k, i_{k+2}, \ldots, i_n)$. Имеем

$$f_2(\pi) = \max \left[\max_{\substack{1 \leqslant u \leqslant k-1 \\ k+2 \leqslant u \leqslant n}} \left(\sum_{j=1}^{u} \alpha_{i_j} + \beta_{i_u} \right), \sum_{j=1}^{k} \alpha_{i_j} + \beta_{i_k}, \sum_{j=1}^{k+1} \alpha_{i_j} + \beta_{i_{k+1}} \right]. \tag{4.4}$$

Аналогично,

$$f_2(\pi') = \max \left[\max_{\substack{1 \leqslant u \leqslant k-1 \\ k+2 \leqslant u \leqslant n}} \left(\sum_{j=1}^{u} \alpha_{i_j} + \beta_{i_u} \right), \sum_{j=1}^{k-1} \alpha_{i_j} + \alpha_{i_{k+1}} + \beta_{i_{k+1}}, \sum_{j=1}^{k+1} \alpha_{i_j} + \beta_{i_k} \right]. \tag{4.5}$$

Следовательно, для того чтобы $f_2(\pi) \leqslant f_2(\pi')$, достаточно, чтобы

$$\min(\alpha_{i_k} + \beta_{i_k}, \beta_{i_{k+1}}) \leqslant \min(\alpha_{i_{k+1}} + \beta_{i_{k+1}}, \beta_{i_k}). \tag{4.6}$$

В свою очередь для того чтобы имело место соотношение (4.6), достаточно, чтобы

$$\operatorname{sign}(\alpha_{i_k}) [M - \min(\alpha_{i_k} + \beta_{i_k}, \beta_{i_k})] \leqslant \\ \leqslant \operatorname{sign}(\alpha_{i_{k+1}}) [M - \min(\alpha_{i_{k+1}} + \beta_{i_{k+1}}, \beta_{i_{k+1}})], \tag{4.7}$$

где M — достаточно большое число (значение $M > \max\limits_{1\leqslant l\leqslant n} \min(\alpha_l + \beta_l, \beta_l)$).

Действительно, могут представиться следующие случаи:

1) $\alpha_{i_k} \leqslant 0$, $\alpha_{i_{k+1}} \geqslant 0$. В этом случае $\alpha_{i_k} + \beta_{i_k} \leqslant \beta_{i_k}$ и $\beta_{i_{k+1}} \leqslant \alpha_{i_{k+1}} + \beta_{i_{k+1}}$;

2) $\alpha_{i_k} < 0$, $\alpha_{i_{k+1}} < 0$. По условию

$$\min(\alpha_{i_k} + \beta_{i_k}, \beta_{i_k}) \leqslant \min(\alpha_{i_{k+1}} + \beta_{i_{k+1}}, \beta_{i_{k+1}}),$$

откуда $\alpha_{i_k} + \beta_{i_k} \leqslant \min(\beta_{i_k}, \beta_{i_{k+1}}, \alpha_{i_{k+1}} + \beta_{i_{k+1}})$;

3) $\alpha_{i_k} > 0$, $\alpha_{i_{k+1}} > 0$. По условию

$$\min(\alpha_{i_k} + \beta_{i_k}, \beta_{i_k}) \geqslant \min(\alpha_{i_{k+1}} + \beta_{i_{k+1}}, \beta_{i_{k+1}}),$$

откуда $\beta_{i_{k+1}} \leqslant \min(\alpha_{i_k} + \beta_{i_k}, \beta_{i_k}, \alpha_{i_{k+1}} + \beta_{i_{k+1}})$.

Следовательно, в любом случае имеет место соотношение (4.6). Таким образом, справедлива

Теорема 4.2. *Для того чтобы перестановке* $\pi = (i_1, i_2, \ldots, i_n)$ *соответствовало наименьшее значение функции* $f_2(\pi)$, *достаточно, чтобы соотношение* (4.7) *выполнялось для всех* $k = \overline{1, n-1}$.

Для нахождения оптимальной перестановки $\overset{*}{\pi}_2$ чисел множества N каждому из $i \in N$ следует приписать «вес» (приоритет)

$$\omega(i) = \operatorname{sign}(\alpha_i)\,[M - \min(\alpha_i + \beta_i, \beta_i)] \qquad (4.8)$$

и упорядочить эти числа в порядке неубывания «весов».

Класс оптимальных перестановок, вообще говоря, можно расширить, если непосредственно воспользоваться соотношением (4.6).

Это соотношение, как будет показано в § 1 гл. 5, обладает свойством транзитивности. Условие же (4.7) является достаточным условием выполнимости соотношения (4.6).

§ 5. Оптимизация функций, рекуррентно заданных на множестве перестановок

Многие задачи упорядочения естественным образом могут быть сформулированы как задачи нахождения перестановок, которым соответствуют наименьшие значения функций, в рекуррентном задании которых содержатся некоторые характеристики соответствующих перестановок. В этом параграфе мы рассмотрим одну из ситуаций такого рода, получившую широкое распространение в теории расписаний.

5.1. Пусть P — некоторое множество (не обязательно всех) перестановок $\pi = (i_1, i_2, \ldots, i_n)$ элементов множества $N = \{1, 2, \ldots, n\}$.

Под *st-окрестностью* k-го по порядку элемента в перестановке π (обозначение $\Omega_{st}(\pi, k)$) будем понимать упорядоченный набор $\langle Q, \sigma_s, \sigma_t \rangle$, где $\sigma_s = (i_{\max(1, k-s+1)}, \ldots \ldots, i_{k-1}, i_k)$ и $\sigma_t = (i_{k+1}, i_{k+2}, \ldots, i_{\min(n,k+t)})$— перестановки длины $\min(k, s)$ и $\min(n - k, t)$ соответственно и Q — множество (неупорядоченное) элементов $\{i_1, i_2, \ldots, i_{k-s}\}$. Если $k \leqslant s$, то $Q = \phi$. Предполагается, что $s \geqslant 0$ и $t \geqslant 0$.

Требуется найти перестановку $\pi^* \in P$, которой соответствует наименьшее значение функции $F(\pi, n)$, заданной на P рекуррентным соотношением

$$F(\pi, k) = \Phi\,[F(\pi, k-1), \Omega_{st}(\pi, k)], \tag{5.1}$$

где $F(\pi, 0)$ — известная не зависящая от π величина, а функция, стоящая в правой части этого соотношения, неубывающая по $F(\pi, k-1)$.

В этих терминах могут быть сформулированы весьма разнородные экстремальные задачи. Рассмотрим некоторые из них.

5.2. Примеры. 1. Требуется найти кратчайший путь из начальной вершины 1 некоторого ориентированного графа $(N, \vec{U})$ в конечную вершину n, проходящий через остальные вершины ровно по одному разу. Длина дуги $(i, j) \in \vec{U}$ равна $l_{ij} > 0$. Если $(i, j) \notin \vec{U}$, то будем полагать $l_{ij} = \infty$. Для каждой вершины $k \in N$ известно множество вершин M_k, которые должны предшествовать при обходе этой вершине. В частности, $M_1 = \phi$, $M_n = \{1, 2, \ldots, n-1\}$.

Последовательность обхода вершин естественно задавать перестановкой вида $\pi = (i_1 = 1, i_2, \ldots, i_{n-1}, i_n = n)$. Длина пути при этом равна

$$F(\pi, n) = \sum_{k=1}^{n-1} l_{i_k i_{k+1}}. \tag{5.2}$$

Положим $\Omega_{2,\,0}(\pi,\ k) = \langle Q,\ \sigma_2,\ \phi \rangle$, где $Q = \{i_1, i_2, \ldots, i_{k-2}\}$, $\sigma_2 = (i_{k-1}, i_k)$ при $k \geqslant 2$.

Имеем

$$F(\pi, k) = \Phi(F(\pi, k-1), \Omega_{2,0}(\pi, k)) = \begin{cases} F(\pi, k-1) + l_{i_{k-1} i_k}, & \text{если } M_{i_k} \subseteq \{Q, i_{k-1}\}, \\ \infty & \text{в противном случае,} \end{cases} \tag{5.3}$$

где $F(\pi, 1) = 0$.

Таким образом, задача сводится к нахождению среди всех перестановок вида $\pi = (i_1 = 1, i_2, \ldots, i_{n-1}, i_n = n)$ перестановки, которой соответствует наименьшее значение функции $F(\pi, k)$, заданной рекуррентным соотношением (5.3). В рассматриваемом случае $s = 2$, $t = 0$.

2. Процесс обработки объекта в технологической машине последовательного действия можно представить как последовательное прохождение объектом некоторого фиксированного множества N операций, возможный порядок прохождения которых регламентируется рядом технологических факторов. Время выполнения каждой операции состоит из основного и вспомогательного. Вообще говоря, основное время зависит от множества ранее выполненных операций, а вспомогательное — как от множества ранее выполненных операций, так и от последовательности их выполнения.

В зависимости от требуемой точности решения задачи и характера операций при оценке вспомогательного времени на каждую операцию достаточно учесть влияние множества всех ранее выполненных операций и последовательность выполнения не более l операций, непосредственно предшествующих данной.

В этом случае задача выбора последовательности выполнения операций, минимизирующей общее время на обработку, сводится к построению перестановки $\pi_n = (i_1, \ldots, i_k, \ldots, i_n)$ элементов множества N, которой соответствует наименьшее значение функции $F(\pi, n)$, где

$$F(\pi, k) = F(\pi, k-1) + T[\Omega_{l+1,\,0}(\pi, k)]. \tag{5.4}$$

Здесь $T[\langle Q, \sigma_{l+1}, \phi \rangle]$ — время выполнения операции i_k при условии, что ей предшествуют операции множества $Q = \{i_1, i_2, \ldots, i_{k-1}\}$ и упорядочение l операций непосредственно предшествующих операции i_k, описывается перестановкой σ_l. В этом случае $s = l + 1$, $t = 0$.

3. Такт работы m-позиционной технологической машины состоит из вспомогательного и рабочего хода. Первый заключается в одновременной переадресации объектов, находящихся в позициях $j = 1, 2, \ldots, m - 1$, в позиции $j + 1$, поступлении в первую позицию нового обрабатываемого объекта и выдачи с m-й позиции обработанного объекта; второй — в обработке объектов, находящихся в соответствующих позициях.

Продолжительность рабочего хода определяется длительностью лимитирующей в данном такте операции; продолжительность вспомогательного хода предполагается постоянной и равной $t_{\text{вс}}$.

Общее время $F(\pi, n)$ обработки заданного множества объектов $N = \{1, 2, \ldots, n\}$ при некоторой фиксированной последовательности $\pi = (i_1, i_2, \ldots, i_n)$ их поступления в машину и заданных временах t_i^j обработки i-го объекта в j-й позиции определяется следующими соотношениями:

$$F(\pi, 1) = \sum_{r=1}^{m} \max(t_{i_1}^{r}, t_{i_2}^{r-1}, \ldots, t_{i_r}^{1}) + m t_{\text{вс}}, \tag{5.5}$$

$$F(\pi, k) = F(\pi, k-1) + T[\Omega_{1, m-1}(\pi, k)] + t_{\text{вс}}. \tag{5.6}$$

Здесь $T[\Omega_{1, m-1}(\pi, k)] = \max\left(t_{i_k}^{m}, t_{i_{k+1}}^{m-1}, \ldots, t_{i_{\min(m+k-1, n)}}^{\max(m+k-n, 1)}\right)$ и, следовательно, $s = 1$, $t = m - 1$.

Задача заключается в выборе такой перестановки элементов множества N, допустимой относительно заданного на N частичного порядка, которой соответствует наименьшее значение функции $F(\pi, n)$, определенной рекуррентными соотношениями (5.5), (5.6).

4. Надежность (вероятность безотказной работы) устройства, состоящего из n узлов, при условии, что выход последних из строя не зависит друг от друга и выход каждого влечет выход из строя всего устройства, определяется величиной $p = \prod_{i=\overline{1, n}} p_i$, где p_i — надежность i-го узла.

Для повышения надежности i-го узла можно воспользоваться одним из заданных объемов средств $v_i^1, v_i^2, \ldots, v_i^{m_i}$, каждый из которых повышает надежность узла до величины p_i^j, $j = \overline{1, m_i}$.

Задачу распределения заданного объема средств ν для обеспечения максимальной надежности устройства в целом можно представить как задачу нахождения последовательности $\pi = (\nu_1^{j_1}, \nu_2^{j_2}, \ldots, \nu_n^{j_n})$, $1 \leqslant j_k \leqslant m_k$, которой соответствует наименьшее значение функции $F(\pi, n)$, где

$$F(\pi, k) = \Phi\,[F(\pi, k-1),\ \Omega_{1,0}(\pi, k)] = \\ = \begin{cases} F(\pi, k-1)\cdot p_k^{j_k}, & \text{если } \sum\limits_{i=\overline{1,k}} \nu_i^{j_i} \leqslant \nu, \\ \infty & \text{в противном случае.} \end{cases} \tag{5.7}$$

При этом предполагается, что $F(\pi, 0) = -1$. В данном случае $s = 1$, $t = 0$.

5.3. Для нахождения оптимальной перестановки $\pi^* \in P$ воспользуемся приемами, характерными для метода последовательного конструирования вариантов. По-видимому, при оптимизации функций, подобных (5.1), указанные приемы оказываются наиболее эффективными, позволяя получить решение в результате перебора числа вариантов, существенно меньшего по сравнению с непосредственным перебором всех возможных вариантов.

В дальнейшем мы будем полагать $s \geqslant 1$ и рассматривать только нормализованные множества P. Множество P перестановок элементов множества N называется *нормализованным*, если оно вместе с перестановками $\pi_1 = (i_1^1, i_2^1, \ldots, i_n^1)$ и $\pi_2 = (i_1^2, i_2^2, \ldots, i_n^2)$ такими, что $\Omega_{s-1,t}(\pi_1, k) = \Omega_{s-1,t}(\pi_2, k)$, содержит и перестановку $\pi = (i_1^1, \ldots, i_k^1, i_{k+1}^2, \ldots, i_n^2)$.

Нетрудно видеть, что множество всех перестановок элементов множества N является нормализованным так же, как и множество всех перестановок, допустимых относительно заданного на N строгого порядка при любых s и t.

В частности, множество перестановок, графически представленных на рис. 2.5.1, является нормализованным при любых значениях s и t.

Пусть $\hat{\pi} \in P$. Обозначим $\Omega(k) = \Omega_{s-1,t}(\hat{\pi}, k)$. Пусть P' — множество всех перестановок $\pi \in P$, для которых

$\Omega_{s-1,t}(\pi,k) = \Omega(k)$. Обозначим через $\tilde{F}[\Omega(k)]$ наименьшее из значений $F(\pi, k)$, вычисленных для всех $\pi \in P'$.

Разобьем множество P' на подмножества $P'_1, P'_2, \ldots, P'_m$, отнеся ко множеству P'_l все перестановки

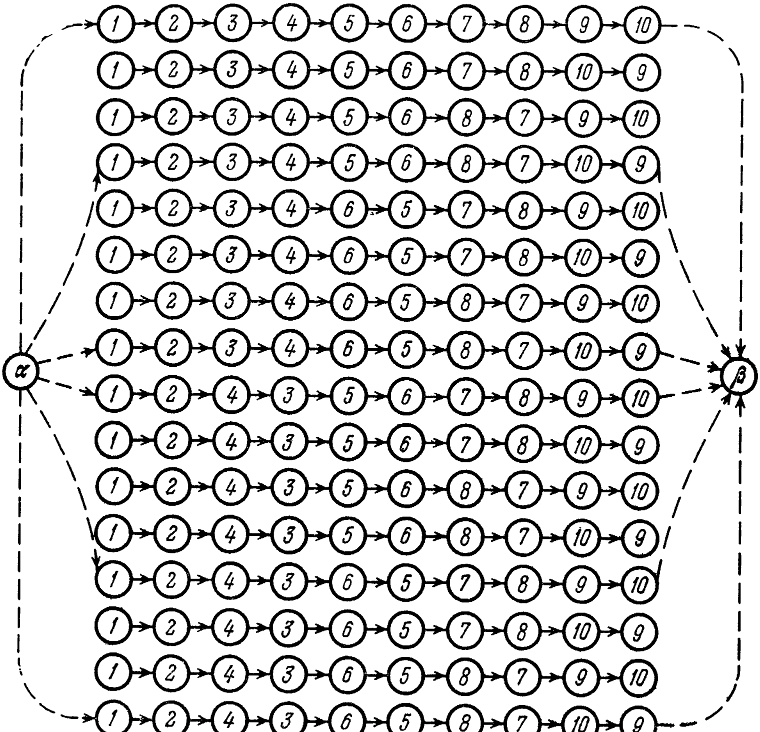

Рис. 2.5.1.

$\pi \in P'$ с одинаковыми окрестностями $\Omega_{s-1,t}(\pi, k-1)$, равными, для определенности, $\Omega^l(k-1)$.

Очевидно,

$$\tilde{F}[\Omega(k)] = \min_{\pi \in P'} \{F(\pi, k)\} = \\ = \min \{\min_{\pi \in P'_1} \{F(\pi, k)\}, \ldots, \min_{\pi \in P'_m} \{F(\pi, k)\}\}. \quad (5.8)$$

Поскольку функция, стоящая в правой части соотношения (5.1), является неубывающей по первому аргументу и Ω_{st} (π, k) одинаковы для всех $\pi \in P'_l$, то

$$\min_{\pi \in P'_l} \{F(\pi, k)\} = \Phi\,[\min_{\pi \in P'_l} \{F(\pi, k-1)\}, \Omega_{s,\,t}(\pi, k)]. \quad (5.9)$$

Покажем, что

$$\min_{\pi \in P'_l} \{F(\pi, k-1)\} = \tilde{F}\,[\Omega^l(k-1)]. \quad (5.10)$$

Действительно, пусть $\tilde{F}\,[\Omega^l\,(k-1)] = F\,(\pi_1, k-1)$ и $\min\limits_{\pi \in P'_l} \{F\,(\pi, k-1)\} = F\,(\pi_2, k-1)$, где $\pi_1 = (i_1^1, i_2^1, \ldots, i_n^1)$ и $\pi_2 = (i_1^2, i_2^2, \ldots, i_n^2)$. По условию $\Omega_{s-1,t}\,(\pi_1, k-1) = \Omega_{s-1,t}\,(\pi_2, k-1)$. Поскольку P — нормализованное множество, то оно содержит перестановку $\pi' = (i_1^1, i_2^1, \ldots, i_{k-1}^1, i_k^2, \ldots, i_n^2)$. Перестановка $\pi' \in P'_l$ и $F\,(\pi', k-1) = F\,(\pi_1, k-1)$, откуда следует справедливость соотношения (5.10).

Таким образом,

$$\tilde{F}\,[\Omega(k)] = \min_{\pi \in P^*} \Phi\,[\tilde{F}\,[\Omega(k-1)], \Omega_{s,\,t}(\pi, k)], \quad (5.11)$$

где P^* — множество всех перестановок $\pi \in P$, для которых $\Omega_{s-1,t}\,(\pi, k) = \Omega\,(k)$ и $F\,(\pi, k-1) = \tilde{F}\,[\Omega\,(k-1)]$ при $\Omega_{s-1,t}\,(\pi, k-1) = \Omega\,(k-1)$.

Соотношение (5.11) позволяет существенно сократить обьем вычислений при $n \gg s + t$.

Оптимальной перестановке π^* соответствует наименьшее значение $\tilde{F}\,[\Omega\,(n)]$. Эта перестановка удовлетворяет условию $F\,(\pi^*, k) = \tilde{F}\,[\Omega_{s-1,t}\,(\pi^*, k)], k=\overline{1,n}$. Тем самым, запоминая значение $\Omega\,(k-1)$, при котором в правой части соотношения (5.11) достигается минимум каждый раз, когда вычисляется $\tilde{F}\,[\Omega\,(k)]$, можно без значительных затруднений восстановить перестановку π^*.

5.4. Описанная ситуация имеет простую графическую интерпретацию. Используя соотношение (5.11) мы, по существу, заменяем задачу поиска пути из α в β, обладающего требуемым свойством в графе типа рис. 2.5.1, задачей поиска пути, обладающего аналогичным свойством в некотором новом графе, получаемом из первого объединением вершин.

Вершины графа рис. 2.5.1 объединяются в одну вершину нового графа, если соответствующие им элементы из N в данной последовательности имеют одинаковые $(s-1, t)$-окрестности.

Рис. 2.5.2 иллюстрирует процесс объединения вершин графа, изображенного на рис. 2.5.1, при а) $s = 2$, $t = 1$; б) $s = 1$, $t = 2$; в) $s = 3$, $t = 1$.

Размер и сложность получаемых графов естественно рассматривать в качестве оценки вычислительной сложности алгоритма построения оптимальной перестановки. С увеличением n, s и t размер этих графов существенно возрастает и тем самым значительно увеличивается необходимый объем вычислений.

5.5. Вычислим основные количественные характеристики алгоритма построения оптимальной последовательности в случае, когда P — множество всех $n!$ перестановок элементов множества N.

Обозначим через π_l — перестановку l элементов из N.

В рассматриваемом случае, используя соотношение (5.1), вычисляем $F(\pi_{s+t}, s)$ для всех возможных π_{s+t}. Полагаем $\tilde{F}[\Omega(s)]$ равными соответствующим $F(\pi_{s+t}, s)$.

По определению $\Omega(k)$ — это упорядоченный набор вида $\langle Q, \sigma_{s-1}, \sigma_t\rangle$. Запишем последовательность (σ_{s-1}, σ_t) в виде $\pi_{s+t-1} = (\pi_{s+t-2}, l)$, где l — последний по порядку элемент в σ_t.

Тогда соотношение (5.11) можно записать как

$$\tilde{F}[\langle Q, (\pi_{s+t-2}, l)\rangle] = \\ = \min_{k\in Q} \Phi[\tilde{F}[\langle Q\setminus k, (k, \pi_{s+t-2})\rangle], \langle Q, (k, \pi_{s+t-2}, l)\rangle]. \quad (5.12)$$

Вычисления проводятся последовательно для всех возможных Q и π_{s+t-2}, не содержащих одинаковых элементов сначала при $|Q| = 2$, затем при $|Q| = 3$ и т. д., наконец, при $|Q| = n - s - t + 1$.

При нахождении $\tilde{F}[\langle Q, (\pi_{s+t-2}, l)\rangle]$ используются только значения $\tilde{F}[\langle Q\setminus k, (k, \pi_{s+t-2})\rangle]$, найденные на предыдущем шаге. В соответствии с соотношением (5.12) вычисляется $|Q|$ значений функции Φ (или, что то же, функции F) и в результате $|Q| - 1$ сравнений этих значений находится значение $\tilde{F}[\langle Q, (\pi_{s+t-2}, l)\rangle]$.

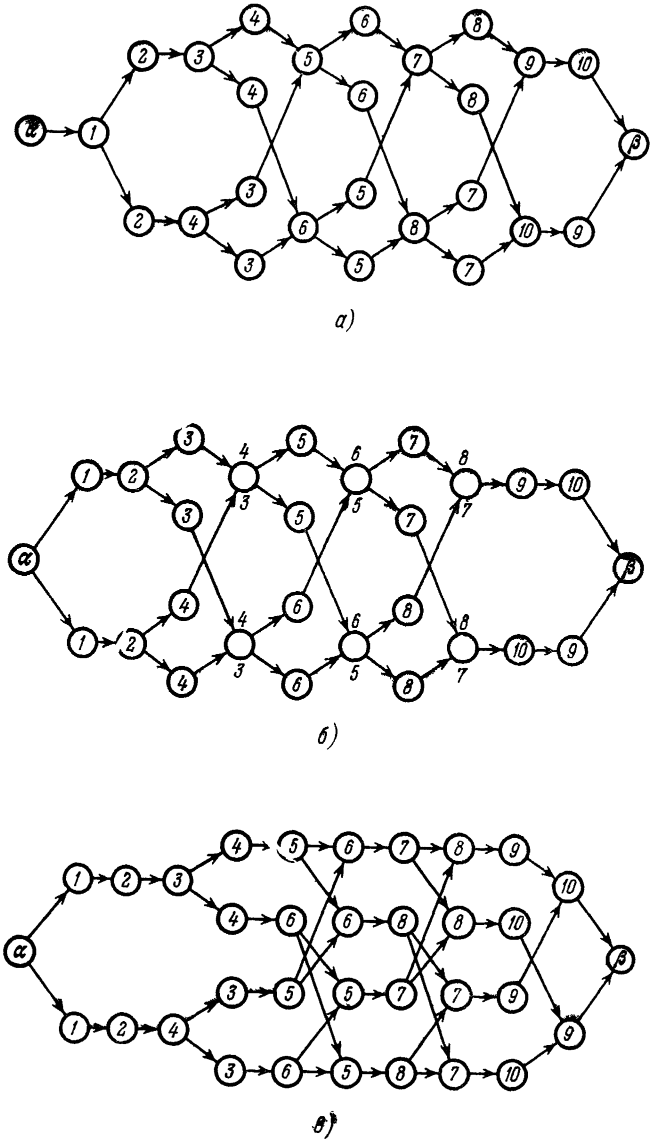

Рис. 2.5.2.

Таким образом, в общем случае требуется вычислить

$$R_F = \sum_{k=1}^{n} (1/(b_k - 1)!)\, A_n^{c_k} \tag{5.13}$$

значений функции $F(\pi, k)$, из них запомнить

$$R_з = \sum_{k=1}^{n-1} (1/b_k!)\, A_n^{c_k} + 1 \tag{5.14}$$

значений, произведя $R_{ср} = R_F - R_з$ их сравнений. При этом одновременно требуется помнить не более

$$R_п = \frac{4d(d-[d])+2}{\left[d+\frac{1}{2}\right]!\left[d+\frac{3}{2}\right]!}\, n! \tag{5.15}$$

значений.

Здесь $b_k = \max(k - s, 0) + 1$, $c_k = \min(t + k, n)$, $d = \frac{1}{2}(n - t - s)$; A_n^m — число размещений из n элементов по m; $[p]$ — целая часть числа p.

При $n \leqslant s + t$ процесс объединения вершин неосуществим, что приводит к необходимости рассмотрения всех перестановок множества N.

При решении большеразмерных практических задач обычно ограничиваются получением приближенного решения. В этом случае целесообразно проводить текущую поэтапную оценку перспективности конструируемых перестановок с соответствующим «достраиванием» наиболее перспективных. Определенный эффект может быть достигнут также в результате разбиения рассматриваемой задачи на ряд задач меньшего размера с использованием получаемой информации о решении последних.

§ 6. Библиографическая справка

Теории графов посвящена обширная литература. В данной книге используется, в основном, терминология К. Бержа [17]. С основными понятиями теории бинарных отношений, в частности отношений порядка, можно более подробно ознакомиться по книге Ю. А. Шрейдера [212].

Теорема 2.1 известна в литературе как теорема Дилворта [263, 153]. Теорема 2.2 доказана В. Г. Спринджуком [163]. Обе теоремы справедливы и для бесконечных множеств. Формулировка задачи нахождения минимального D-разложения в терминах линейного программирования дана Д. Б. Данцигом и А. Дж. Гофманом [59]. Вопросы определения числа допустимых перестановок рассматривались С. С. Кислицыным [77] и В. С. Танаевым [175]. Применению матричных методов к решению этих вопросов посвящена работа [372]. На возможность использования теоремы Дилворта для решения некоторых задач сетевого планирования указано в [245].

Интерпретация задач теории расписаний в терминах смешанных графов проведена в работах Л. П. Матюшкова и В. С. Танаева [113], Е. Балаша [229], Б. Сассмана [415] и (в несколько иной терминологии) В. В. Шкурбы [202], Дж. Геллера и Дж. Логемана [300]. Более подробно эти вопросы рассмотрены в гл. 7. Определению числа элементов множества K_G и нижней его оценки посвящена работа [161]. Формулировка и доказательство теоремы 3.1 содержатся в [92].

Перестановочный прием для минимизации функции (4.1) применялся Г. Г. Харди, Дж. Литтлвудом, Г. Полиа [192]. Интересный цикл работ по минимизации линейных форм на различных подмножествах множества перестановок проведен Д. А. Супруненко, В. С. Айзенштатом, Н. А. Лепешинским, И. М. Кунцевичем, Д. Н. Кравчуком и Н. Н. Метельским. Обзор этих работ приведен в [165].

Функция (4.3), вероятно, впервые исследовалась С. М. Джонсоном [313] и Р. Беллманом [236]. Формулировка и доказательство теоремы 4.2 содержатся в работе [170]. Попытки обобщения перестановочного приема предприняты в работах [382, 311]. В [240] вводится понятие многогранников перестановок.

Систематическое исследование функций, рекуррентно заданных на множестве перестановок, проведено Г. М. Левиным и В. С. Танаевым [93, 178]. Существенно более общая ситуация (по сравнению с описанной) рассмотрена в [94].

ГЛАВА 3
ДЕТЕРМИНИРОВАННЫЕ СИСТЕМЫ ОБСЛУЖИВАНИЯ С ОДНИМ ПРИБОРОМ

Конечный поток требований, поступающих на обслуживание в заданные моменты времени, обслуживается одним прибором. В каждый момент времени прибор обслуживает не более одного требования. Известны времена обслуживания требований. Порядок обслуживания может быть произвольным, либо регламентироваться заданными отношениями частичного порядка. При некоторых постановках задачи могут допускаться прерывания в обслуживании каждого отдельного требования. Необходимо так организовать процесс обслуживания, чтобы он в том или ином смысле был наилучшим.

В этой главе рассматриваются ситуации, в которых требования поступают на обслуживание одновременно и качество расписания характеризуется суммарным или максимальным штрафом за обслуживание. Линейные, экспоненциальные и монотонные (отличающиеся на константу) функции штрафа рассматриваются в § 2, а константно-линейные — в § 3. Вопросы минимизации суммарного времени (или стоимости) пуско-наладочных работ рассматриваются в § 4. Некоторые специальные критерии оптимальности анализируются в § 5. Обобщение ситуаций, рассматриваемых в § 2, на случай частично упорядоченного множества требований проведено в § 6. В § 7 описаны вычислительные схемы построения наилучших расписаний при произвольных (неубывающих) функциях штрафа.

§ 1. Предварительные замечания

Пусть необходимо обслужить множество $N = \{1, 2, \ldots, n\}$ требований. Требование k, $k = \overline{1, n}$, поступает на обслуживание в момент времени $d_k \geqslant 0$ и для обслуживания требует $t_k > 0$ единиц времени.

1.1. Процесс обслуживания может быть описан заданием кусочно-постоянной, непрерывной слева функции $s = s(t)$, принимающей при $0 \leqslant t < \infty$ одно из значений

$0, 1, 2, \ldots, n$. Если $s(t) = k \neq 0$, то в момент времени t прибор обслуживает требование k; если $s(t) = 0$, то в момент времени t ни одно из требований не обслуживается. Эта функция называется *расписанием*.

Расписание должно удовлетворять определенным условиям, вытекающим из самой постановки задачи. В качестве основных могут быть выделены

— условие полноты: для любого $1 \leqslant k \leqslant n$ существует $0 \leqslant t < \infty$ такое, что $s(t) = k$; суммарная длина промежутков, на которых $s(t) = k$, равна t_k;

— условие готовности к обслуживанию: $s(t) \neq k$ для всех $t < d_k$;

— условие упорядоченности: если по условию задачи требование i необходимо обслужить раньше требования j и $s(t') = i$, то $s(t) \neq j$ для всех $t < t'$;

— условие непрерывности: если по условию задачи прерывания в обслуживании требований не допускаются и $s(t') = s(t'') = k \neq 0$, то $s(t) = k$ для всех $t' < t < t''$.

Расписания $s = s(t)$, которые удовлетворяют перечисленным условиям, называются *допустимыми*.

Определенный практический интерес представляют расписания, которые удовлетворяют некоторым дополнительным условиям таким, как условия, связанные с необходимостью переналадок обслуживающего прибора, условия группирования требований при их обслуживании и т. п.

При рассмотрении задач такого рода под допустимым расписанием будем понимать такое расписание $s = s(t)$, которое удовлетворяет всем требованиям, вытекающим из постановки конкретно рассматриваемой задачи.

1.2. Каждому допустимому расписанию $s = s(t)$ соответствует вектор $\bar{t} = (\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n)$ времен завершения обслуживания требований при этом расписании. Значения $\bar{t}_k$, $k = \overline{1, n}$, таковы, что $s(\bar{t}_k) = k$ и $s(t) \neq k$ для всех $t > \bar{t}_k$.

Предполагается, что качество расписания s определяется вектором $\bar{t}$, т. е. каждому расписанию s ставится в соответствие значение некоторой скалярной функции $F(\bar{t})$ вектора $\bar{t}$, определяемого расписанием s.

Функция $F(\bar{t})$ предполагается монотонно возрастающей кусочно-непрерывной относительно всех компонент вектора $\bar{t}$.

Наиболее распространенным является следующий способ задания $F(\bar{t})$. Для каждого требования $k = \overline{1, n}$ задается неубывающая кусочно-непрерывная функция $\varphi_k(x)$, выражающая в количественном отношении «штраф», который необходимо «заплатить», если обслуживание этого требования завершится в момент времени x. В качестве $F(\bar{t})$ выбирается одна из функций $F_\Sigma(\bar{t}) = \sum_{k=1}^{n} \varphi_k(\bar{t}_k)$ или $F_{\max}(\bar{t}) = \max_{1 \leqslant k \leqslant n} \{\varphi_k(\bar{t}_k)\}$. Функции $\varphi_k(x)$ называются *функциями штрафа*.

Расписание, которое удовлетворяет всем условиям рассматриваемой задачи, называется *оптимальным*, если ему соответствует наименьшее значение $F(\bar{t})$.

1.3. Поскольку функция $F(\bar{t})$ является неубывающей, то при поиске оптимального расписания достаточно рассматривать только такие расписания, при которых недопустимы неоправданные задержки в завершении обслуживания требований. Такие расписания получили название *активных*. Характерной их особенностью является невозможность уменьшения времени завершения обслуживания любого из требований без увеличения времен завершения обслуживания каких-либо других требований.

Если прерывания обслуживания каждого отдельного требования запрещены, то активное расписание однозначно определяется заданием последовательности $\pi = (i_1, i_2, \ldots, i_n)$, в которой эти требования обслуживаются. Здесь i_k означает номер требования, которое в последовательности π обслуживается k-м по порядку. В этом случае $\bar{t}_{i_1} = d_{i_1} + t_{i_1}$, $\bar{t}_{i_k} = \max(d_{i_k} + t_{i_k}, \bar{t}_{i_{k-1}} + t_{i_k})$, $k = \overline{2, n}$. Таким образом, если поиск оптимального расписания ограничен классом расписаний, при которых не происходит прерываний процесса обслуживания каждого отдельного требования, то решение задачи может быть получено в результате рассмотрения конечного числа расписаний, определяемых возможными последовательностями обслуживания требований. Такие расписания называются *перестановочными*. В этом случае наряду с обозначением $F(\bar{t})$ будем употреблять обозначение $F(\pi)$.

Если множество N не упорядочено, то число перестановочных расписаний $n!$. Если множество N — частично упорядоченное множество, то число перестановочных расписаний не превышает величины, рассчитанной в соответствии с теоремой 2.3 § 2 гл. 2.

В общем случае, если допустимы прерывания процесса обслуживания требований, можно показать, что оптимальное расписание также может быть получено в результате рассмотрения конечного числа расписаний. Однако для однозначного описания этих расписаний уже недостаточно простого задания последовательности обслуживания требований. Необходимо указать, в какие моменты времени прерывается, а затем возобновляется процесс обслуживания каждого требования.

В этой главе основное внимание уделяется анализу ситуаций, когда все d_k одинаковы и, следовательно, можно положить $d_k = 0$, $k = \overline{1, n}$. Интересной особенностью систем такого рода является возможность исключения из рассмотрения расписаний с прерываниями процесса обслуживания требований (см. гл. 4, § 1, теорема 1.1).

§ 2. Интервалы очередности

В этом параграфе мы приведем весьма простые методы решения задачи оптимального упорядочения требований для некоторых частных случаев функций штрафа. Предполагается, что множество N требований не упорядочено, все требования поступают в очередь на обслуживание одновременно, в момент времени $d = 0$, критерий оптимальности расписания — суммарный штраф.

2.1. Пусть требования i и j обслуживаются непосредственно друг за другом и их обслуживание начинается в момент времени $t \geqslant 0$. Если при этом первым обслуживается требование i, то суммарный штраф за обслуживание этих требований

$$F_\Sigma(t, i, j) = \varphi_i(t + t_i) + \varphi_j(t + t_i + t_j). \tag{2.1}$$

Если первым обслуживается требование j, то суммарный штраф за обслуживание этих двух требований

$$F_\Sigma(t, j, i) = \varphi_j(t + t_j) + \varphi_i(t + t_i + t_j). \tag{2.2}$$

Определим

$$R_{ij}(t) = F_\Sigma(t, i, j) - F_\Sigma(t, j, i). \tag{2.3}$$

Величина $R_{ij}(t)$ показывает, насколько изменяется суммарный штраф при переходе от последовательности, при которой требование i начинало обслуживаться в момент времени t непосредственно перед требованием j, к последовательности, отличающейся от исходной транспозицией элементов i и j.

Если $R_{ij}(t) < 0$, то при решении вопроса, какое именно из двух последовательно обслуживаемых требований i и j необходимо начать обслуживать в момент времени t, предпочтение следует отдать требованию i. Аналогично, если $R_{ij}(t) > 0$, то первым в момент времени t следует обслуживать требование j. Наконец, если $R_{ij}(t) = 0$, то порядок обслуживания этих требований безразличен.

Поскольку функции $\varphi_i(x)$ и $\varphi_j(x)$ предполагаются кусочно-непрерывными в интервале $(0,\ T = \sum_{k=1}^{n} t_k)$, то этот интервал может быть разбит на конечное число интервалов, в каждом из которых величина $R_{ij}(t)$ положительна, отрицательна или равна нулю.

Интервал (θ_1, θ_2) называется *интервалом очередности* типа $i \to j$, если $R_{ij}(t) < 0$ для всех $t \in (\theta_1, \theta_2)$. Если $R_{ij}(t) > 0$ для всех $t \in (\theta_1, \theta_2)$, то интервал (θ_1, θ_2) называется интервалом очередности типа $j \to i$. При $R_{ij}(t) = 0$ для всех $t \in (\theta_1, \theta_2)$ порядок обслуживания требований безразличен.

Величина $R_{ij}(t)$ по определению зависит от значений t, t_i, t_j и функций штрафа $\varphi_i(x)$ и $\varphi_j(x)$. В каждом конкретном случае в результате несложных преобразований могут быть выделены интервалы очередности.

2.2. Пусть, например, $\varphi_i(x) = a_i x^2$ и $\varphi_j(x) = a_j x^2$. Найдем интервалы очередности при различных сочетаниях a_i, a_j, t_i, t_j.

Имеем

$$R_{ij}(t) = a_i(t + t_i)^2 + a_j(t + t_i + t_j)^2 - \\ - a_j(t + t_j)^2 - a_i(t + t_i + t_j)^2 = \\ = 2(a_j t_i - a_i t_j)t + [2t_i t_j(a_j - a_i) + a_j t_i^2 - a_i t_j^2]. \quad (2.4)$$

Пусть $a_i/t_i = a_j/t_j$. В этом случае $R_{ij}(t)$ не зависит от t и определяется величиной

$$Q(a_i, a_j, t_i, t_j) = 2t_i t_j(a_j - a_i) + a_j t_i^2 - a_i t_j^2. \quad (2.5)$$

Если $Q(a_i, a_j, t_i, t_j) < 0$, то при последовательном обслуживании требований i и j требование i всегда обслуживается первым. В этом случае интервал $(-\infty, \infty)$ (а, следовательно, и интересующий нас интервал $(0, T)$) является интервалом очередности типа $i \to j$. Аналогично, если $Q(a_i, a_j, t_i, t_j) > 0$, то $(-\infty, \infty)$ — интервал очередности типа $j \to i$. Наконец, если $Q(a_i, a_j, t_i, t_j) = 0$, то порядок обслуживания требований i и j безразличен, если эти требования обслуживаются непосредственно друг за другом.

Пусть $a_i/t_i \neq a_j/t_j$. Величина $R_{ij}(t) = 0$ при

$$t_0 = \frac{2t_i t_j (a_j - a_i) + a_j t_i^2 - a_i t_j^2}{2(a_i t_j - a_j t_i)}. \tag{2.6}$$

Интервал $(-\infty, \infty)$ разбивается на три интервала очередности $(-\infty, t_0)$, (t_0), (t_0, ∞). Тип интервалов $(-\infty, t_0)$ и (t_0, ∞) определяется знаком $R_{ij}(t)$ в этих интервалах. При $t = t_0$ порядок обслуживания требований i и j безразличен.

По предположению обслуживание всех требований осуществляется во временно́м интервале $(0, T)$ или, во всяком случае, в интервале $(0, \infty)$. Следовательно, в практическом отношении интерес представляет выделение интервалов очередности в интервале планирования $(0, T)$ или $(0, \infty)$. Если $t_0 < 0$ или $t_0 > T - t_i - t_j$, то $(0, T)$, очевидно, является интервалом очередности, тип которого определяется знаком $R_{ij}(t)$. Если $0 < t_0 < T - t_i - t_j$, то интервал $(0, T)$ разбивается на три интервала очередности. Если $t_0 = 0$ или $t_0 = T - t_i - t_j$, то имеем два интервала очередности, один из которых (t_0).

Полагая, для определенности, $t_i = 2$, $t_j = 1$, $a_i = 9$ и $a_j = 5$, имеем $t_0 = 2{,}5$. Следовательно, интервал $(-\infty, \infty)$ разбивается на три интервала очередности $(-\infty; 2{,}5)$, $(2{,}5)$ и $(2{,}5; \infty)$. При $t < 2{,}5$ величина $R_{ij}(t) < 0$, при $t = 2{,}5$ величина $R_{ij}(t) = 0$, наконец, при $t > 2{,}5$ величина $R_{ij}(t) > 0$. Таким образом, если требования i и j обслуживаются непосредственно одно за другим и первое из этих требований необходимо начать обслуживать в момент времени $t \geqslant 0$, то этим требованием будет требование i при $t < 2{,}5$ и требование j при $t > 2{,}5$. Если $t = 2{,}5$, то первым может обслуживаться любое из этих требований.

Это утверждение, вообще говоря, несправедливо, если после завершения процесса обслуживания одного из требований i или j до начала обслуживания второго проходит некоторый промежуток времени, занятый, например, обслуживанием некоторого третьего требования.

2.3. Выделение интервалов очередности может быть проведено для любых пар функций штрафа $\varphi_i(x)$, $\varphi_j(x)$ и любых t_i, t_j. Для этого достаточно решить (относительно t) соответствующее уравнение $R_{ij}(t) = 0$.

Знание интервалов очередности оказывается весьма полезным при поиске оптимальных расписаний методом последовательного конструирования вариантов.

Знак величины $R_{ij}(t)$, а следовательно, и тип интервала очередности определяется в общем случае значениями t, t_i, t_j и конкретным видом функций штрафа $\varphi_i(x)$ и $\varphi_j(x)$. При некоторых сочетаниях t_i, t_j, $\varphi_i(x)$ и $\varphi_j(x)$ знак $R_{ij}(t)$ может не зависеть от t, как это было показано при рассмотрении двух квадратичных функций штрафа в предыдущем пункте.

Найдем такие пары функций $\varphi_i(x)$ и $\varphi_j(x)$, для которых знак $R_{ij}(t)$ не зависит от t при любых сочетаниях t_i и t_j.

Пусть $\varphi_i(x) = a_i x + b_i$ и $\varphi_j(x) = a_j x + b_j$. Имеем

$$R_{ij}(t) = a_i(t + t_i) + a_j(t + t_i + t_j) - a_j(t + t_j) - a_i(t + t_i + t_j) = a_j t_i - a_i t_j.$$

Следовательно, эта пара функций является искомой, поскольку величина и тем самым знак $R_{ij}(t)$ не зависят от t при любых данных t_i и t_j.

Пусть $\varphi_i(x) = a_i \exp(\alpha x) + b_i$ и $\varphi_j(x) = a_j \exp(\alpha x) + b_j$. Имеем

$$R_{ij}(t) = \exp(\alpha t)[a_i \exp(\alpha t_i) - a_j \exp(\alpha t_j) + (a_j - a_i)\exp[\alpha(t_i + t_j)]]. \tag{2.7}$$

В данном случае величина $R_{ij}(t)$ зависит от t, однако знак этой величины определяется только значениями α, a_i, a_j, t_i, t_j, а при фиксированных α, a_i и a_j только значениями t_i и t_j.

Пусть $\varphi_i(x) = \varphi(x) + b_i$ и $\varphi_j(x) = \varphi(x) + b_j$, где $\varphi(x)$ — возрастающая на $(-\infty, \infty)$ функция. Поскольку $R_{ij}(t) = \varphi(t + t_i) - \varphi(t + t_j)$, то знак $R_{ij}(t)$ не зависит от t при любых конкретных значениях t_i и t_j.

Таким образом, указанные три пары функций обладают свойством независимости знака величины $R_{ij}(t)$ от значения $-\infty < t < \infty$. Интервал $(-\infty, \infty)$, а следовательно, и $(0, \infty)$, и $(0, T)$ являются интервалами очередности для этих пар функций. Тип этих интервалов определяется конкретными значениями параметров функций и значениями t_i и t_j.

Теорема 2.1. *Пусть $\varphi_i(x)$ и $\varphi_j(x)$ — строго возрастающие достаточно гладкие (существует третья производная) на $(-\infty, \infty)$ функции. Для того чтобы знак величины $R_{ij}(t)$ не зависел от t, $-\infty < t < \infty$, при любых t_i и t_j, необходимо и достаточно, чтобы*

1) $\varphi_k(x) = a_k x + b_k$, $k = i, j$, *или*

2) $\varphi_k(x) = a_k \exp(\alpha x) + b_k$, $k = i, j$, *или*

3) $\varphi_k(x) = \varphi(x) + b_k$, $k = i, j$.

Доказательство. Поскольку достаточность условия теоремы доказана, переходим к доказательству его необходимости.

Прежде всего сделаем одно замечание. Пусть уравнение вида $F_1(y, z) + \tau F_2(y, z) = 0$ имеет действительное решение (y_0, z_0) при любом $\tau = \tau_0$ и $F_2(y_0, z_0) \neq 0$. Тогда знак функции $F(y, z, v) = F_1(y, z) + \tau(v) F_2(y, z)$ не зависит от v тогда и только тогда, когда $\tau(v) = \text{const}$. Действительно, предполагая, что $\tau(v) \neq \text{const}$, $\tau(v_1) = c_1$, $\tau(v_2) = c_2$, $c_1 < c_2$ и обозначая через (y^*, z^*) решение уравнения $F_1(y, z) + \frac{c_1 + c_2}{2} F_2(y, z) = 0$, имеем

$$F(y^*, z^*, v_1) = F_1(y^*, z^*) + c_1 F_2(y^*, z^*) = \\ = F_1(y^*, z^*) + \frac{c_1 + c_2}{2} F_2(y^*, z^*) + \frac{c_1 - c_2}{2} F_2(y^*, z^*) = \\ = \frac{c_1 - c_2}{2} F_2(y^*, z^*). \quad (2.8)$$

Аналогично, $F(y^*, z^*, v_2) = \frac{c_2 - c_1}{2} F_2(y^*, z^*)$. Следовательно, знак $F(y, z, v)$ зависит от v, если $\tau(v) \neq \text{const}$.

Покажем, что если знак $R_{ij}(t)$ не зависит от t, то $\varphi_i'(x)/\varphi_j'(x) = c = \text{const}$ и при $c \neq 1$ $\varphi_j''(x)/\varphi_j'(x) = \alpha = \text{const}$. Воспользуемся разложениями функции $R_{ij}(t)$ в ряд Тейлора. Имеем

$$R_{ij}(t) = \varphi_j'(t)\,[t_i - (\varphi_i'(t)/\varphi_j'(t))\, t_j] + o(t_i + t_j)^2. \quad (2.9)$$

При достаточно малых t_i, $t_j > 0$ знак $R_{ij}(t)$ определяется знаком первого слагаемого, и поскольку $\varphi_j'(t) > 0$, то знаком выражения, стоящего в квадратных скобках. На основании сделанного замечания знак этого выражения не зависит от t, тогда и только тогда, когда $\varphi_i'(t)/\varphi_j'(t) = c = \text{const}$. Следовательно, $\varphi_i(x) = c\varphi_j(x) + b$.

Пусть $c \neq 1$. Имеем

$$R_{ij}(t) = \varphi_j'(t)\left[t_i - ct_j + \frac{1}{2}\frac{\varphi_j''(t)}{\varphi_j'(t)}(t_i(t_i + 2t_j) - ct_j(t_j + 2t_i))\right] + o(t_i + t_j)^3. \quad (2.10)$$

При достаточно малых t_i, t_j знак $R_{ij}(t)$ определяется знаком выражения, стоящего в квадратных скобках. Поскольку при любом фиксированном значении $\varphi_j''(t)/\varphi_j'(t)$ существуют сколь угодно малые значения t_i, t_j, обращающие это выражение в нуль, то на основании сделанного замечания $\varphi_j''(t)/\varphi_j'(t) = \alpha = \text{const}$.

Таким образом, если $c = 1$, то $\varphi_i(x) = \varphi_j(x) + b$. Если $c \neq 1$, то при $\varphi_j''(t) = 0$ имеем $\varphi_j(x) = a_jx + b_j$ и $\varphi_i(x) = a_ix + b_i$. Наконец, при $\alpha \neq 0$ имеем $\varphi_j''(x) = \alpha\varphi_j'(x)$, $\varphi_j(x) = a_j \exp(\alpha x) + b_j$, $\varphi_i(x) = a_i \exp(\alpha x) + b_i$. Теорема доказана.

2.4. Перечисленные в теореме 2.1 пары функций обладают тем характерным свойством, что интервал $(-\infty, \infty)$ (и тем самым любой заданный временной интервал) является интервалом очередности, тип которого при фиксированных параметрах этих функций определяется значениями t_i и t_j.

Можно показать что при достаточно общих предположениях относительно функций штрафа $\varphi_i(x)$ и $\varphi_j(x)$ соответствующим выбором значений t_i и t_j любой заданный интервал можно превратить в интервал очередности того или иного типа.

Особый интерес представляет выявление условий, при которых заданный временной интервал будет интервалом очередности независимо от конкретных значений t_i и t_j, но с учетом сравнительных соотношений между этими значениями.

Теорема 2.2. *Если функции штрафа $\varphi_i(x)$ и $\varphi_j(x)$ сколь угодно раз дифференцируемы и $\varphi_i^{(l)}(x) \geqslant \varphi_j^{(l)}(x)$*

для всех $l = 1, 2, \ldots$ *на временном интервале* $(\theta_1 - \varepsilon, \theta_2 + \varepsilon)$, *где* $\varepsilon > 0$, *то при* $t_i < t_j$ *интервал* (θ_1, θ_2) *является интервалом очередности типа* $i \to j$.

Доказательство. Разлагая функцию $R_{ij}(t)$ в ряд Тейлора в точке $t \in (\theta_1, \theta_2)$, имеем

$$R_{ij}(t) = \sum_{l=1}^{\infty} \frac{1}{l!} \{\varphi_i^{(l)}(t) t_i^l + \varphi_j^{(l)}(t)(t_i + t_j)^l - \varphi_j^{(l)}(t) t_j^l - \varphi_i^{(l)}(t)(t_i + t_j)^l\} =$$
$$= \sum_{l=1}^{\infty} \frac{1}{l!} \left\{ \sum_{k=1}^{l} C_l^k t_i^k t_j^{l-k} (\varphi_j^{(l)}(t) - \varphi_i^{(l)}(t)) + (\varphi_j^{(l)}(t) t_i^l - \varphi_i^{(l)}(t) t_j^l) \right\}. \quad (2.11)$$

Следовательно, если выполняется условие теоремы, то $R_{ij}(t) < 0$ независимо от значения $t \in (\theta_1, \theta_2)$.

В качестве примера рассмотрим пару функций $\varphi_i(x) = \exp x$ и $\varphi_j(x) = x$. Эти функции сколь угодно раз дифференцируемы и $\varphi_i^{(l)}(x) \geqslant \varphi_j^{(l)}(x)$ на $(0, \infty)$ для всех $l = 1, 2, \ldots$

Если $0 < t_i < t_j$, то знак

$$R_{ij}(t) = t_i - \exp(t + t_i)(\exp t_j - 1) \leqslant$$
$$\leqslant t_i - \exp(t_i)(\exp t_j - 1) \leqslant t_i - (1 + t_i)t_j =$$
$$= (t_i - t_j) - t_i t_j < 0 \quad (2.12)$$

не зависит от $t \geqslant 0$ и конкретных значений t_i и t_j. Интервал $(0, \infty)$ является интервалом очередности типа $i \to j$, что соответствует теореме 2.2.

Если $t_i \geqslant t_j$, то интервал $(0, \infty)$, вообще говоря, разбивается на несколько интервалов очередности в зависимости от конкретных значений t_i и t_j. Эти интервалы могут быть определены способом, описанным в предыдущем пункте.

2.5. Знание интервалов очередности в значительной степени ускоряет процесс поиска оптимальной последовательности обслуживания требований и в ряде случаев приводит к весьма простым способам ее построения.

Теорема 2.3. *Пусть* $d_k = 0$, $k = \overline{1, n}$, *все неубывающие функции штрафа* $\varphi_k(x)$ *принадлежат одному и только одному из следующих трех классов*: а) $\varphi_k(x) =$

$= a_k x + b_k$, б) $\varphi_k(x) = a_k \exp(\alpha x) + b_k$ *и* в) $\varphi_k(x) = \varphi(x) + b_k$, *и значения* $\omega(k)$ *равны соответственно* а) $\omega(k) = a_k/t_k$, б) $\omega(k) = (1 - \exp(\alpha t_k))/a_k \exp(\alpha t_k)$ *и* в) $\omega(k) = -t_k$. *Для того чтобы последовательности* $\pi = (i_1, i_2, \ldots, i_n)$ *соответствовал наименьший суммарный штраф, достаточно, чтобы* $\omega(i_\nu) \geqslant \omega(i_\mu)$ *для всех* $1 \leqslant \nu < \mu \leqslant n$.

Доказательство. Покажем справедливость утверждения теоремы для случая б), предполагая, естественно, $a_k \neq 0$. В остальных случаях доказательство аналогичное.

Выбираем два произвольных требования i и j. В п. 2.3 было показано, что знак величины $R_{ij}(t)$ на интервале $(0, \infty)$ не зависит от t. Из соотношения (2.7) следует, что для того, чтобы $R_{ij}(t) \leqslant 0$ на $(0, \infty)$, достаточно, чтобы

$$\frac{1 - \exp(\alpha t_i)}{a_i \exp(\alpha t_i)} \geqslant \frac{1 - \exp(\alpha t_j)}{a_j \exp(\alpha t_j)}, \tag{2.13}$$

причем равенство имеет место только в случае $R_{ij}(t) = 0$.

Положим $\omega(k) = \dfrac{1 - \exp(\alpha t_k)}{a_k \exp(\alpha t_k)}$. Следовательно, функция $F_\Sigma(\pi)$ такова, что каждому требованию k может быть отнесена некоторая величина $\omega(k)$, обладающая свойством: если $\omega(i) \geqslant \omega(j)$, то $F_\Sigma(\pi) \leqslant F_\Sigma(\pi')$. Здесь π — последовательность, в которой требование i обслуживается непосредственно перед требованием j, а π' отличается от π транспозицией элементов i и j.

Покажем, что если последовательность обслуживания требований $\pi = (i_1, i_2, \ldots, i_n)$ удовлетворяет условию $\omega(i_\nu) \geqslant \omega(i_\mu)$ для всех $1 \leqslant \nu < \mu \leqslant n$, то π — оптимальная последовательность. Действительно, если в оптимальной последовательности $\pi^* = (\overset{*}{i}_1, \overset{*}{i}_2, \ldots, \overset{*}{i}_j, \ldots, \overset{*}{i}_n)$ при некотором $j = k$ значение $\omega(\overset{*}{i}_k) < \omega(\overset{*}{i}_{k+1})$, то последовательности $\tilde{\pi}^*$, отличающейся от π^* транспозицией элементов $\overset{*}{i}_k$ и $\overset{*}{i}_{k+1}$, соответствует меньшее значение F_Σ, что противоречит предположению об оптимальности π^*.

Теорема доказана.

Из приведенных рассуждений следует, что если, например, все функции штрафа $\varphi_k(x) = a_k x + b_k$, то, обслуживая требования в порядке невозрастания величин $\omega(k) = \frac{a_k}{t_k}$, получаем наименьшее значение величины суммарного штрафа.

2.6. Примеры.

1. На одном станке необходимо обработать n различных деталей. Известны длительность обработки каждой детали t_k и срок D, к которому необходимо обработать эти детали. Функции штрафа $\varphi_k(x) = \max(x - D, 0)$. Необходимо определить такой порядок обработки деталей, при котором средний штраф $\frac{1}{n}\sum_{k=1}^{n} \max(\bar{t}_k - D, 0)$ наименьший. Здесь $\bar{t}_k$ — фактическое время завершения обработки k-й детали. Последовательность $\pi = (i_1, i_2, \ldots, i_n)$ такая, что $t_{i_\nu} \leqslant t_{i_{\nu+1}}$, $\nu = \overline{1, n-1}$, является оптимальной, что соответствует классу в) теоремы 2.3.

2. Пакет, содержащий n программ с известными временами их прохождения, выполняется однопрограммной ЭВМ. Необходимо так организовать пакет, т. е. установить такую очередность прохождения программ, чтобы минимизировать величину $\frac{1}{n}\sum_{k=1}^{n} \bar{t}_k^{\alpha}$, $\alpha > 0$, где $\bar{t}_k$ — время завершения выполнения k-й программы. Здесь, как и в примере 1, программы следует упорядочить в порядке неубывания величины t_k, поскольку $\varphi_k(x) = \frac{1}{n}x^{\alpha}$.

3. Последовательно осуществляется n операций. Стоимость k-й операции равна u_k, вероятность брака при ее выполнении равна p_k. При возникновении брака все операции повторяются сначала. Необходимо указать такой порядок $\pi = (i_1, i_2, \ldots, i_n)$ выполнения операций, при котором средняя стоимость выполнения операций

$$U(\pi) = \frac{u_{i_1}}{(1 - p_{i_1})\ldots(1 - p_{i_n})} + \frac{u_{i_2}}{(1 - p_{i_2})\ldots(1 - p_{i_n})} + \ldots + \frac{u_{i_n}}{1 - p_{i_n}} \tag{2.14}$$

будет наименьшей. Производя замену переменной

$1 - p_k = \exp(-t_k)$, можно записать

$$U(\pi') = \sum_{k=1}^{n} u_{i_k} \exp\Big(\sum_{l=1}^{k} t_{i_l}\Big), \tag{2.15}$$

где $\pi' = (i'_1, i'_2, \ldots, i'_n)$ и $i'_k = i_{n-k+1}$. Таким образом, на основании п. б) теоремы 2.3 операции необходимо выполнять в порядке неубывания величины u_k/p_k.

§ 3. Директивные сроки

Директивным сроком называется момент времени $D_k \geqslant 0$, к которому необходимо или, во всяком случае, желательно завершить обслуживание требования $k = \overline{1, n}$. В общем случае не удается построить такого расписания, при котором обслуживание каждого требования k завершается не позднее заданного директивного срока D_k. Появляется объективная необходимость в нарушении отдельных директивных сроков, что сопряжено с определенными потерями. Эти потери, как правило, зависят от того, какие именно требования и на сколько задерживаются с обслуживанием. Иными словами, каждому требованию $k = \overline{1, n}$ отнесена неубывающая кусочно-непрерывная функция штрафа $\varphi_k(x) = 0$ при $x \leqslant D_k$ и $\varphi_k(x) > 0$ при $x > D_k$. Требуется определить расписание обслуживания n требований, при котором суммарный штраф наименьший. Предполагается, что множество требований неупорядочено и значение $d_k = 0$, $k = \overline{1, n}$.

3.1. Найдем необходимые и достаточные условия, при которых все требования могут быть обслужены в заданные директивные сроки и, следовательно, $F_\Sigma(\pi^*) = 0$, где π^* — оптимальная последовательность обслуживания требований.

Пусть π^D — последовательность требований, упорядоченных по неубыванию значений D_k.

Теорема 3.1. *$F_\Sigma(\pi^*) = 0$ тогда и только тогда, когда $F_\Sigma(\pi^D) = 0$.*

Доказательство. Если $F_\Sigma(\pi^D) = 0$, то, очевидно, π^D — оптимальная последовательность и можно положить $\pi^* = \pi^D$.

Если $F_\Sigma(\pi^*) = 0$ и $\pi^* \neq \pi^D$, то в последовательности $\pi^* = (i^*_1, i^*_2, \ldots, i^*_n)$ существуют требования i^*_j и i^*_{j+1}

такие, что $D_{i_j^*} > D_{i_{j+1}^*}$. Рассмотрим последовательность $\tilde{\pi}$, полученную из π^* транспозицией элементов i_j^* и i_{j+1}^*. В расписании, определяемом этой новой последовательностью, все требования, за исключением требований i_j^* и i_{j+1}^*, будут обслужены в те же моменты времени, что и в расписании, определяемом последовательностью π^* Обозначим моменты завершения обслуживания требований i_j^* и i_{j+1}^* относительно π^* через $\bar{t}_{i_j^*}(\pi^*)$ и $\bar{t}_{i_{j+1}^*}(\pi^*)$, а относительно $\tilde{\pi}$ — через $\bar{t}_{i_j^*}(\tilde{\pi})$ и $\bar{t}_{i_{j+1}^*}(\tilde{\pi})$ соответственно. Очевидно, $\bar{t}_{i_j^*}(\tilde{\pi}) = \bar{t}_{i_{j+1}^*}(\pi^*) \leqslant D_{i_{j+1}^*} < D_{i_j^*}$. С другой стороны, $\bar{t}_{i_{j+1}^*}(\tilde{\pi}) < \bar{t}_{i_{j+1}^*}(\pi^*) \leqslant D_{i_{j+1}^*}$. Следовательно, и требования с индексами i_j^* и i_{j+1}^* будут обслужены в срок, если требования обслуживаются в последовательности $\tilde{\pi}$. Проводя аналогичные рассуждения конечное число раз, убеждаемся в справедливости утверждения теоремы. Теорема доказана.

Не нарушая общности, пронумеруем все требования в порядке неубывания D_k. Таким образом, на основании приведенной теоремы, если $\bar{t}_k = \sum_{i=1}^{k} t_i \leqslant D_k$, $k = \overline{1, n}$, то обслуживание всех требований может быть завершено в заданные сроки и оптимальная последовательность обслуживания требований $\pi^* = (1, 2, \ldots, n)$. Если хотя бы при одном значении k значение $\bar{t}_k > D_k$, то не существует последовательности обслуживания требований, при которой эти требования обслуживались бы в заданные сроки. В этом случае заведомо $F_\Sigma(\pi^*) > 0$ и построение оптимальной последовательности сопряжено с определенными вычислительными затруднениями.

3.2. Пусть все $\varphi_k(x)$ — ступенчатые **функции**, т. е. функции вида

$$\varphi_k(x) = \begin{cases} 0, & \text{если } x \leqslant D_k, \\ a_k > 0, & \text{если } x > D_k, \end{cases} \tag{3.1}$$

где $k = \overline{1, n}$

Предположим, что имеется способ, посредством которого из множества $N = \{1, 2, \ldots, n\}$ всех требований может быть выделено подмножество B^*, элементы которого обладают характеристическим свойством: существует оптимальная последовательность π^*, при которой требование k в срок не обслуживается, тогда и только тогда, когда $k \in B^*$. Тогда, очевидно, $F_\Sigma(\pi^*) = \sum\limits_{k \in B^*} a_k$.

Покажем, что в этом случае и построение самой оптимальной последовательности не вызывает затруднений. Действительно, пусть π_{B^*} — произвольная последовательность элементов множества B^*, а $\vec{\pi}_{A^*}$ — упорядоченная по возрастанию последовательность элементов множества $A^* = N \setminus B^*$. Будем обслуживать требования в последовательности $\pi = (\vec{\pi}_{A^*}, \pi_{B^*})$. Учитывая тот факт, что все требования пронумерованы в порядке неубывания значений D_k, на основании теоремы 3.1 требования множества A^* по-прежнему будут обслуживаться в срок. Что касается требований множества B^*, то ни одно из них при этой последовательности не будет обслужено в срок, так как в противном случае исходная последовательность не была бы оптимальной. Следовательно, $F_\Sigma(\vec{\pi}_{A^*}, \pi_{B^*}) = F_\Sigma(\pi^*)$ и $\pi = (\vec{\pi}_{A^*}, \pi_{B^*})$ — оптимальная последовательность.

Если каждому требованию $k = \overline{1, n}$ отнесем переменную x_k, принимающую значение 1, если $k \in B^*$, и значение 0, если $k \in A^*$, то искомое разбиение множества N может быть получено в результате решения задачи целочисленного линейного программирования с n ограничениями и n булевыми переменными:

$$\sum_{k=1}^{n} a_k x_k \to \min, \tag{3.2}$$

$$\sum_{k=1}^{i} t_k (1 - x_k) \leqslant D_i, \quad i = \overline{1, n}, \tag{3.3}$$

$$x_k \in \{0, 1\}, \quad k = \overline{1, n}. \tag{3.4}$$

Положим $z_k = \max(0, \bar{t}_k - D_k)$, где $\bar{t}_k$ — фактическое время завершения обслуживания требования k, а D_k — соответствующий директивный срок. Если требования обслуживаются в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, то $\bar{t}_{i_j} = \sum_{k=1}^{j} t_{i_k}$.

Пусть $B \subset B^*$, $A = N \setminus B$ и $\vec{\pi}_A$ — последовательность требований множества A, расположенных в порядке их возрастания.

На основании предыдущих рассуждений можно заключить, что если $\vec{\pi}_A = (i_1, \ldots, i_p, \ldots, i_q)$, $z_{i_p} > 0$ и $z_{i_j} = 0$ при $j < p$, то в B^* содержится, по крайней мере, одно из требований множества $C = \{i_1, i_2, \ldots, i_p\}$.

Следовательно, задачу (3.2)—(3.4) можно заменить эквивалентной ей задачей целочисленного линейного программирования, в ряде случаев меньшей размерности. Обозначим через S множество элементов последовательности $(1, 2, \ldots, n)$, которым соответствует строго возрастающая последовательность $z_k > 0$. Имеем

$$\sum_{k=1}^{s} a_k x_k \to \min, \tag{3.5}$$

$$\sum_{k=1}^{i} t_k x_k \geqslant z_i, \quad i \in S, \tag{3.6}$$

$$x_k \in \{0, 1\}, \quad k = \overline{1, s}. \tag{3.7}$$

Здесь s — наибольший элемент множества S.

Будем говорить, что множество $\bar{C} \subseteq C$ является *критическим*, если $\sum_{i_j \in \bar{C}} t_{i_j} \geqslant z_{i_p}$ и $\sum_{i_j \in \tilde{C}} t_{i_j} < z_{i_p}$ для всех собственных подмножеств $\tilde{C}$ множества $\bar{C}$. Одно из критических подмножеств множества C содержится в B^*.

Этот факт позволяет организовать целенаправленный поиск оптимального разбиения множества N всех требований на два подмножества A^* и B^*. Простейшая схема организации такого поиска может быть описана следующим образом.

1. Положим $B = B^0 = \phi$, $A = A^0 = N \setminus B^0 = N$ и построим $\vec{\pi}_{A^0} = (1, 2, \ldots, n)$. Если все $z_k = 0$, то $B^* = B^0$, $A^* = A^0$ и $\vec{\pi}_{A^0}$ — искомая последовательность. Пусть $z_p > 0$ и $z_k = 0$ при $k < p$. Образуем все критичес-

кие подмножества $\bar{C}_1^0, \bar{C}_2^0, \ldots, \bar{C}_{r_0}^0$ множества $C = \{1, 2, \ldots, p\}$. Положим $B_l^1 = B^0 \cup \bar{C}_l^0$, $A_l^1 = N \setminus B_l^1$, $l = \overline{1, r_0}$.

2. Переходим к последовательному рассмотрению разбиений (A_l^1, B_l^1). Полагая $B = B_l^1$, $A = A_l^1$, строим $\vec{\pi}_{A_l^1} = (i_1^1, \ldots, i_{p_1}^1, \ldots, i_{q_1}^1)$. Если все $z_{i_j^1} = 0$, то строим $\pi = (\vec{\pi}_{A_l^1}, \pi_{B_l^1})$, вычисляем $F_\Sigma(\pi)$ и приписываем это значение $F_\Sigma(\pi)$ разбиению (A_l^1, B_l^1), Если $z_{i_{p_1}^1} > 0$ и $z_{i_j^1} = 0$ при $j < p_1$, то образуем все критические подмножества $\bar{C}_{l1}^1, \bar{C}_{l2}^1, \ldots, \bar{C}_{lr_l}^1$ множества $C_l^1 = \{i_1^1, \ldots, i_{p_1}^1\}$. Положим $B_{lt}^2 = B_l^1 \cup C_{lt}^1$, $A_{lt}^2 = N \setminus B_{lt}^2$, $t = \overline{1, r_l}$.

3. Переходим затем к последовательному рассмотрению разбиений $(A_{11}^2, B_{11}^2), \ldots, (A_{r_0 r_{r_0}}^2, B_{r_0 r_{r_0}}^2)$ и т. д. Этот процесс, очевидно, конечен. В результате будет получен набор разбиений множества N, каждому из которых будет приписано некоторое значение $F_\Sigma(\pi)$. То из разбиений, которому приписано наименьшее значение $F_\Sigma(\pi)$, является оптимальным разбиением (A^*, B^*), а последовательность $\pi = (\vec{\pi}_{A^*}, \pi_{B^*})$ — искомой оптимальной последовательностью обслуживания требований.

Приведенный алгоритм, по существу, описывает процесс последовательного конструирования оптимальной последовательности и графически может быть представлен в виде графа, изображенного на рис. 3.3.1. Каждой вершине графа приписано выбираемое на данном шаге множество $\bar{C}$. Объединение всех множеств, приписанных всем вершинам пути, соединяющего начальную вершину графа с рассматриваемой, образует множество B, соответствующее рассматриваемой вершине.

Применение алгоритма, очевидно, может привести к образованию одинаковых множеств B. В этом случае из всех вершин, которым соответствуют одинаковые множества B, дальнейшему «развитию» подлежит одна, поскольку рассмотрение остальных не приводит к новым разбиениям множества N. Эти последние вершины будем относить к классу тупиковых.

Некоторого ускорения процесса поиска оптимального разбиения можно добиться следующим простым приемом.

Припишем всем вершинам графа числа $a(B) = \sum_{i \in B} a_i$, где B — множество, соответствующее рассматриваемой вершине. Поскольку все $a_i > 0$, то вершинам, связанным с рассматриваемой вершиной исходящими из нее дугами, будут приписаны числа, каждое из которых больше числа,

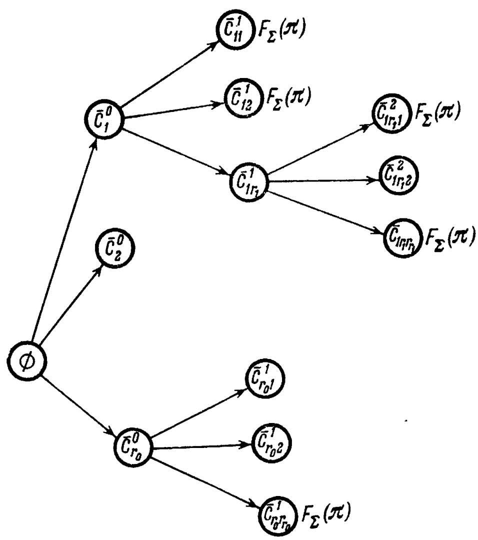

Рис. 3.3.1.

приписанного рассматриваемой вершине. Для конечных вершин, которым приписаны значения $F_\Sigma(\pi)$, окажутся приписанными значения $a(B) = F_\Sigma(\pi)$.

Пусть $a^* = F_\Sigma(\pi^*)$. Все вершины, которым приписаны значения $a(B) > a^*$, очевидно, являются «бесперспективными» и могут не рассматриваться в процессе поиска оптимального разбиения. Поскольку значение a^* заранее неизвестно, то в процессе поиска на каждом шаге целесообразно «развивать» ту из конечных вершин, которой соответствует наименьшее значение $a(B)$. Признаком окончания процесса служит ситуация, когда одной из конечных вершин оказывается приписанным значение $F_\Sigma(\pi)$

и это значение не больше значений $a(B)$, приписанных остальным конечным вершинам (кроме тупиковых).

Отметим, что любая конечная вершина, за исключением тупиковых, будучи «бесперспективной» на данном шаге алгоритма, может подвергнуться «развитию» на последующих его шагах. В этом отношении тупиковые вершины, заведомо не подлежащие дальнейшему «развитию», занимают особое положение среди конечных вершин.

Можно сформулировать ряд правил оценки и выбора направления ветвления, так называемых условий доминирования, применение которых позволяет отнести ту или иную конечную вершину в разряд тупиковых.

Одно из такого рода правил состоит в следующем. Предположим, что на некотором шаге алгоритма одной из конечных вершин соответствует множество B_1 и приписано значение $a(B_1) = a^{(1)}$, а второй — множество B_2 и значение $a(B_2) = a^{(2)}$. Пусть $a^{(1)} \leqslant a^{(2)}$ и множества B_1 и B_2 отличаются одним элементом. Для определенности, пусть множество B_1 содержит элемент i и не содержит элемента j, а множество B_2 содержит элемент j и не содержит элемента i. Предположим далее, что $t_i \geqslant t_j$. Тогда конечная вершина, которой приписано множество B_2, является тупиковой.

Действительно, поскольку $t_i \geqslant t_j$, то при $a^{(1)} \leqslant a^{(2)}$ (и, следовательно, $a_i \leqslant a_j$) множество B^* наряду с требованием j должно содержать требование i либо содержать требование i, но не содержать требования j, либо не содержать ни одного из этих требований. Если $a^{(1)} = a^{(2)}$ (и, следовательно, $a_i = a_j$) и $j \in B^*$, то элемент j в B^* можно, очевидно, заменить элементом i.

3.3. Перейдем к рассмотрению задачи, в которой необходимо в заданные директивные сроки обслужить наибольшее число требований. В этом случае $a_k = 1, k = \overline{1, n}$. По-прежнему будем предполагать, что все требования пронумерованы в порядке неубывания значений D_k.

Поскольку $a_k = 1$, $k = \overline{1, n}$, то, используя описанное выше правило доминирования, при формировании множества B^* нет необходимости в анализе всех возможных альтернатив. Достаточно на каждом шаге из множества C выбирать требование с наибольшим значением t_k.

Алгоритм построения оптимальной последовательности в данном случае может быть описан следующим образом.

1. Если при обслуживании требований в последовательности $\pi^{(n)} = (1, 2, \ldots, n)$ все требования обслуживаются в срок, то эта последовательность оптимальна. Если все $t_k > D_k$, то ни одно из требований не может быть обслужено в срок, и в этом случае любая последовательность обслуживания требований является оптимальной.

2. Пусть на некотором шаге алгоритма последовательность $\pi^{(q)} = (i_1, i_2, \ldots, i_p, \ldots, i_q)$ требований, упорядоченных по неубыванию их индексов, такова, что $z_{i_p} > 0$ и $z_{i_j} = 0$ при $j < p$. Среди требований множества $C = \{i_1, i_2, \ldots, i_p\}$ выберем требование i_l (или одно из требований) с наибольшим значением t_{i_j}. Перейдем к рассмотрению последовательности $\pi^{(q-1)}$, полученной из $\pi^{(q)}$ удалением требования i_l.

3. Повторяя шаг 2 конечное число раз, получаем последовательность $\pi^{(r)} = (i'_1, i'_2, \ldots, i'_r)$, для которой все $z_{i'_j} = 0$, $j = \overline{1, r}$. Последовательность $\pi^* = (\pi^{(r)}, \tilde{\pi})$ является оптимальной. Здесь $\tilde{\pi}$ — произвольная последовательность требований, отличных от требований множества $\{i'_1, i'_2, \ldots, i'_r\}$.

3.4. Пример. Требуется определить последовательность обслуживания требований, при которой наибольшее число требований обслуживается в заданные директивные сроки. Значения времен обслуживания t_k и директивных сроков D_k приведены в следующей таблице:

k	1	2	3	4	5	6	7
t_k	2	1	4	3	2	3	1
D_k	4	5	5	6	8	10	10

1. Обслуживая требования в последовательности $\pi^{(7)} = (1, 2, 3, 4, 5, 6, 7)$, получаем фактические времена $\bar{t}_k$ завершения их обслуживания, равные соответственно 2, 3, 7, 10, 12, 15 и 16. Значения $z_1 = z_2 = 0$, $z_3 = 2$, $z_4 = 4$, $z_5 = 4$, $z_6 = 5$, $z_7 = 6$. Поскольку не все $z_k = 0$, то не все требования могут быть обслужены в заданные директивные сроки.

2. Из множества $C = \{1, 2, 3\}$ требований выберем требование с наибольшим t_k. В данном случае $k = 3$.

Обслуживая требования в последовательности $\pi^{(6)} = (1, 2, 4, 5, 6, 7)$, получаем $\bar{t}_1 = 2, \bar{t}_2 = 3, \bar{t}_4 = 6, \bar{t}_5 = 8, \bar{t}_6 = 11, \bar{t}_7 = 12, z_1 = z_2 = z_4 = z_5 = 0, z_6 = 1, z_7 = 2$.

3. Из множества $C = \{1, 2, 4, 5, 6\}$ выбираем требование с наибольшим t_k. В данном случае $k = 4$. Имеем последовательность $\pi^{(5)} = (1, 2, 5, 6, 7)$, для которой все $z_k = 0$. Следовательно, $\pi^* = (1, 2, 5, 6, 7, 3, 4)$ — оптимальная последовательность обслуживания требований. В заданные директивные сроки обслуживается 5 требований.

3.5. В заключение этого параграфа рассмотрим задачу построения оптимальной (по критерию $F_\Sigma(\pi)$) последовательности обслуживания требований в случае, когда функции штрафа $\varphi_k(x) = \max(x - D_k, 0)$, $k = \overline{1, n}$. Подход к разработке и обоснованию алгоритма ее решения в достаточной мере универсален и может быть использован при решении иных аналогичных задач.

Введем понятие обобщенного интервала очередности.

Наряду с выявлением условий, при которых одно из двух требований следует обслуживать первым в случае, когда эти требования обслуживаются непосредственно друг за другом (см. § 2 гл. 3), определенный интерес представляет выявление условий, при которых одно из двух требований следует обслуживать первым независимо от того, обслуживается ли второе требование непосредственно за первым или между моментом завершения обслуживания первого требования и моментом начала обслуживания второго проходит некоторый промежуток времени, занятый, например, обслуживанием других требований.

Будем говорить, что временной интервал (θ_1, θ_2) является *обобщенным интервалом очередности* типа $i \to j$, если

$$\varphi_i(\underline{t} + t_i) - \varphi_j(\underline{t} + t_j) \leqslant \varphi_i(\bar{t}) - \varphi_j(\bar{t}) \tag{3.8}$$

при $t_i \leqslant t_j$ или

$$\varphi_i(\bar{t} - t_j) - \varphi_j(\underline{t} + t_j) \leqslant \varphi_i(\bar{t}) - \varphi_j(\bar{t}) \tag{3.9}$$

при $t_i > t_j$ для всех $\underline{t}$ и $\bar{t}$, удовлетворяющих условию $\theta_1 \leqslant \underline{t} < \underline{t} + t_i + t_j \leqslant \bar{t} \leqslant \theta_2$.

Содержательно это означает, что если требования обслуживаются, например, в некоторой последовательности

$\pi = (k_1, k_2, \ldots, k_\nu, \ldots, k_\mu, \ldots, k_n)$, $k_\nu = j$, $k_\mu = i$, $\underline{t} = \sum\limits_{l=1}^{\nu-1} t_{k_l}$, $\bar{t} = \sum\limits_{l=1}^{\mu} t_{k_l}$ и $\theta_1 \leqslant \underline{t} < \bar{t} \leqslant \theta_2$, то $F_\Sigma(\pi') \leqslant F_\Sigma(\pi)$, где $\pi' = (k_1, k_2, \ldots, k_{\nu-1}, k_\mu, k_{\nu+1}, \ldots, k_{\mu-1}, k_\nu, k_{\mu+1}, \ldots$ $\ldots, k_n)$ при $t_i \leqslant t_j$ и $\pi' = (k_1, k_2, \ldots, k_{\nu-1}, k_{\nu+1}, \ldots$ $\ldots, k_\mu, k_\nu, k_{\mu+1}, \ldots, k_n)$ при $t_i > t_j$. Действительно, в обоих случаях время завершения обслуживания каждого требования, кроме требований i и j, относительно последовательности π' не больше времени завершения обслуживания этого требования относительно последовательности π. Поскольку все функции штрафа — неубывающие и интервал (θ_1, θ_2) — обобщенный интервал очередности типа $i \to j$, то $F_\Sigma(\pi') \leqslant F_\Sigma(\pi)$.

Используя понятие обобщенных интервалов очередности, в каждом конкретном случае можно сформулировать достаточные условия, при которых некоторое требование из данного множества требований будет обслуживаться первым (или последним), если требования этого множества начинают обслуживание в данный момент времени. Действительно, если рассматриваемое множество требований обозначить через $\tilde{N}$, а через θ_1 и θ_2 обозначить соответственно моменты начала и завершения обслуживания всех требований этого множества, то указанные условия в общем виде означают следующее. Требование $i \in \tilde{N}$ следует обслуживать первым (последним), если интервал (θ_1, θ_2) является обобщенным интервалом очередности типа $i \to j$ (соответственно, типа $j \to i$) для всех $j \in \tilde{N}$, $j \neq i$.

3.6. Теорема 3.2. *Пусть* $\varphi_k(x) = \max(x - D_k, 0)$, $k = i, j$. *Для того чтобы интервал* (θ_1, θ_2) *являлся обобщенным интервалом очередности типа* $i \to j$, *достаточно, чтобы при* $t_i \leqslant t_j$ *значение* $D_i \leqslant \max(\theta_1 + t_j, D_j)$, *а при* $t_i > t_j$ *значение* $D_j \geqslant D_i \geqslant \theta_2 - t_j$.

Доказательство. В рассматриваемом случае условия (3.8)—(3.9) можно записать в виде

$$\max(\underline{t} + t_i, D_i) - \max(\underline{t} + t_j, D_j) \leqslant \max(\bar{t}, D_i) - \max(\bar{t}, D_j) \quad \text{при } t_i \leqslant t_j, \tag{3.10}$$

$$\max(\bar{t} - t_j, D_i) - \max(\underline{t} + t_j, D_j) \leqslant \max(\bar{t}, D_i) - \max(\bar{t}, D_j) \quad \text{при } t_i > t_j. \tag{3.11}$$

Пусть $t_i \leqslant t_j$. По условию $D_i \leqslant \max(\theta_1 + t_j, D_j)$. Поскольку $\bar{t} \geqslant \theta_1 + t_i + t_j$, то $D_i \leqslant \max(\bar{t}, D_j)$. Возможны следующие ситуации: $\bar{t} \leqslant D_j$ или $\bar{t} > D_j$ и, следовательно, $\bar{t} \geqslant D_i$. В обоих случаях непосредственной проверкой убеждаемся в справедливости неравенства (3.10).

Пусть $t_i > t_j$. Левая часть неравенства (3.11) принимает наибольшее значение при $\bar{t} = \theta_2$ и $\underline{t} = \theta_1$. При выполнении неравенств $D_j \geqslant D_i \geqslant \theta_2 - t_j$ неравенство

$$D_i - \max(\theta_1 + t_j, \ D_j) \leqslant \max(\bar{t}, D_i) - \max(\bar{t}, D_j),$$

очевидно, выполняется при любом $\bar{t} \geqslant \theta_1 + t_i + t_j$. Теорема доказана.

Обозначим через $\tilde{N}$ некоторое подмножество множества N требований. Пусть обслуживание требований множества $\tilde{N}$ начинается в момент времени θ_1 и завершается в момент времени $\theta_2 = \theta_1 + \sum_{k \in \tilde{N}} t_k$. Сформулируем достаточные условия, при которых некоторое требование $i \in \in \tilde{N}$ следует обслуживать первым, а некоторое требование $j \in \tilde{N}$ последним среди всех требований множества $\tilde{N}$. Эти условия непосредственно следуют из определения обобщенного интервала очередности и приведенной теоремы.

Следствие 1. *Если $i \in \tilde{N}$ и для всех $j \in \tilde{N}$ выполняются условия $D_i \leqslant \max(\theta_1 + t_j, D_j)$ при $t_i \leqslant \leqslant t_j$ или $\theta_2 - t_j \leqslant D_i \leqslant D_j$ при $t_i > t_j$, то существует оптимальная последовательность обслуживания требований множества $\tilde{N}$, в которой требование i обслуживается первым.*

Следствие 2. *Если $j \in \tilde{N}$ и для всех $i \in \tilde{N}$ выполняются условия $D_i \leqslant \max(\theta_1 + t_j, D_j)$ при $t_i \leqslant t_j$ или $\theta_2 - t_j \leqslant D_i \leqslant D_j$ при $t_i > t_j$, то существует оптимальная последовательность обслуживания требований множества $\tilde{N}$, в которой требование j обслуживается последним.*

На основании этих утверждений можно сформулировать критерии оптимальности некоторых последовательностей.

Следствие 3. *Последовательность $\pi^t = (i_1, i_2, \ldots, i_n)$ всех требований, упорядоченных по неубыванию* *значений* t_k, *является оптимальной, если* $D_{i_k} \leqslant D_{i_j}$ *или* $D_{i_k} \leqslant \sum_{l=1}^{k-1} t_{i_l} + t_{i_j}$, $1 \leqslant k < j \leqslant n$.

Последовательность $\pi^D = (j_1, j_2, \ldots, j_n)$ *всех требований, упорядоченных по неубыванию значений* D_k, *является оптимальной, если* $t_{j_n} + D_{j_i} \geqslant T = \sum_{k=1}^{n} t_k$, $i = \overline{1, n-1}$.

3.7. Следствия из теоремы 3.2 позволяют предложить следующий алгоритм конструирования оптимальной (относительно $F_\Sigma(\pi)$) последовательности обслуживания требований в рассматриваемом случае, т. е. когда $\varphi_k(x) = \max(x - D_k, 0)$, $k = \overline{1, n}$.

1. Упорядочим множество N всех требований по неубыванию значений D_k. Обозначим полученную последовательность $\pi^D = (j_1, j_2, \ldots, j_n)$, $D_{j_i} \leqslant D_{j_{i+1}}$, $i = \overline{1, n-1}$. Если при этом все требования обслуживаются в срок, т. е. $\sum_{k=1}^{i} t_{j_k} \leqslant D_{j_i}$, или $t_{j_n} + D_{j_i} \geqslant T = \sum_{k=1}^{n} t_k$, то π^D — оптимальная последовательность (следствие 3).

2. Упорядочим множество N всех требований по неубыванию значений t_k. Обозначим полученную последовательность $\pi^t = (i_1, i_2, \ldots, i_n)$, $t_{i_j} \leqslant t_{i_{j+1}}$, $j = \overline{1, n-1}$.

Если при этом $D_{i_k} \leqslant D_{i_j}$ или $D_{i_k} \leqslant \sum_{l=1}^{k-1} t_{i_l} + t_{i_j}$ для всех $1 \leqslant k < j \leqslant n$, то π^t — оптимальная последовательность (следствие 3).

3. Из множества N выберем требование l_n с $t_{l_n} = \max_{k \in N} t_k$. Если $D_k \leqslant \max(t_{l_n}, D_{l_n})$ для всех $k \in N$, то в оптимальной последовательности это требование можно обслуживать последним (следствие 2).

Перейдем к рассмотрению множества $N_1 = N \setminus l_n$. Выберем среди требований множества N_1 требование l_{n-1} с $t_{l_{n-1}} = \max_{k \in N_1} t_k$. Если $D_k \leqslant \max(t_{l_{n-1}}, D_{l_{n-1}})$ для всех $k \in N_1$, то требование l_{n-1} в оптимальной последовательности можно обслуживать предпоследним. Перейдем к рассмотрению множества $N_2 = N_1 \setminus l_{n-1}$. Пусть этот процесс удается повторить p раз. В результате получаем множество требований $\{l_{n-p+1}, \ldots, l_n\}$, которые в оптимальной последовательности можно обслужить последними, и порядок их обслуживания $(l_{n-p+1}, \ldots, l_n)$. Множество оставшихся требований обозначим через N_p.

4. Среди требований множества N_p выберем требование l_{n-p} с $D_{l_{n-p}} = \max\limits_{k \in N_p} D_k$. Если $t_{l_{n-p}} + D_k \geqslant \sum\limits_{l \in N_p} t_l$ для всех $k \in N_p$, то в оптимальной последовательности требования множества N_p будут обслуживаться первыми в порядке неубывания значений D_k (следствие 3). В противном случае обозначим N_p через M.

5. На основании теоремы 3.2 разобьем множество M на два подмножества M_1 и M_2 по следующему принципу: если для $j \in M$ существует $i \in M$ такое, что интервал $(\theta_1 = 0,\ \theta_2 = \sum\limits_{k \in M} t_k)$ является интервалом очередности типа $i \to j$, то $j \in M_2$, остальные требования принадлежат множеству M_1. Иными словами, множество M_1 состоит из конкурентноспособных претендентов на первоочередное обслуживание в оптимальной последовательности среди всех требований множества M.

6. Выберем произвольное требование $l_1 \in M_1$ и предположим, что в оптимальной последовательности это требование обслуживается первым среди всех требований множества M. Уменьшим значение D_k для всех $k \in M$, $k \neq l_1$, на величину t_{l_1} и перейдем к выполнению пп. 3—4, полагая $N = M \setminus l_1$. Таким образом, в результате принятия гипотезы, что первым в оптимальной последовательности обслуживается требование l_1, получаем множество $M(l_1) \subset M$ все еще неупорядоченных требований.

Аналогичные рассуждения проводим относительно остальных требований множества M_1.

7. Перейдем к последовательному рассмотрению ситуаций вида: требование l_1 обслуживается первым и $M(l_1)$ — множество все еще неупорядоченных требований. Повторяем пп. 5, 6, полагая $M = M(l_1)$ и принимая в качестве D_k значения исходных директивных сроков, уменьшенные на величину t_{l_1}. Затем перейдем к последовательному рассмотрению ситуаций вида: требования l_1, l_2 обслуживаются первыми в последовательности (l_1, l_2) и $M(l_1, l_2)$

— множество все еще неупорядоченных требований и т. д.

В результате получаем конечное число последовательностей, среди которых выбираем последовательность π^*, которой соответствует наименьшее значение $F_\Sigma(\pi)$. Эта последовательность является искомой.

3.8. Пример. Построим оптимальную (по критерию $F_\Sigma(\pi)$) последовательность π^* обслуживания семи требований одним прибором в предположении, что все функции штрафа $\varphi_k(x) = \max(x - D_k, 0)$. Значения t_k и D_k приведены в следующей таблице:

k	1	2	3	4	5	6	7
t_k	9	10	9	8	5	2	7
D_k	15	20	17	8	10	11	12

1. Последовательности $\pi^D = (4, 5, 6, 7, 1, 3, 2)$ и $\pi^t = (6, 5, 7, 4, (1, 3), 2)$ не удовлетворяют достаточным условиям оптимальности.

2. Выберем требование $l_7 = 2$ с $t_2 = \max_{k\in N} t_k = 10$, $N = \{1, 2,\dots, 7\}$. Поскольку $D_k \leqslant \max(t_2, D_2) = 20$ для всех $k \in N$, то требование 2 в искомой последовательности π^* можно обслуживать последним. Из множества $N_1 = N \setminus \{2\}$ выберем требование $l_6 = 3$ с $t_3 = \max_{k\in N_1} t_k = 9$. Поскольку $D_k \leqslant \max(t_3, D_3) = 17$ для всех $k \in N_1$, то требование 3 в π^* можно обслуживать предпоследним. Аналогично, требование 1 можно обслуживать пятым по порядку. Таким образом, требования 1, 3, 2 могут обслуживаться последними в указанном порядке.

3. Перейдем к рассмотрению задачи упорядочения обслуживания оставшихся требований $M = \{4, 5, 6, 7\}$ во временном интервале $\left(\theta_1 = 0,\ \theta_2 = \sum_{k\in M} t_k = 22\right)$. Интервал $(0, 22)$ является интервалом очередности $\{5, 6\} \to \to 7$. Поэтому $M_1 = \{4, 5, 6\}$ и $M_2 = \{7\}$. Одно из требований множества M_1 может обслуживаться в π^* первым по порядку. Рассмотрим каждую из имеющихся возможностей.

а) Пусть $l_1 = 4$. Уменьшим значения D_k для всех $k \in \in M$, $k \neq 4$, на величину $t_4 = 8$, и перейдем к рассмотрению эквивалентной задачи оптимального обслуживания в интервале (0, 14) требований, значения t_k и D_k которых приведены в следующей таблице:

k	5	6	7
t_k	5	2	7
D_k	2	3	4

Выберем требование $l_4 = 7$ с $t_7 = \max\limits_{k \in \{5, 6, 7\}} t_k = 7$. Поскольку $D_k \leqslant \max(t_7, D_7) = 7$ для $k = 5,6$, то требование 7 может обслуживаться четвертым по порядку. Аналогично получаем $l_3 = 5$.

Таким образом, получаем последовательность (6, 5, 7) с суммарным штрафом 15.

б) Пусть $l_1 = 5$. Уменьшая значения D_k для всех $k \in M$, $k \neq 5$, на величину $t_5 = 5$, перейдем к упорядочению в интервале (0, 17) требований, параметры которых приведены в следующей таблице:

k	4	6	7
t_k	8	2	7
D_k	3	6	7

Выберем требование $l_4 = 4$ с $t_4 = \max\limits_{k \in \{4, 6, 7\}} t_k = 8$. Требование 4 может обслуживаться четвертым по порядку, так как $D_k \leqslant \max(t_4, D_4) = 8$ для $k = 6, 7$. Аналогично находим $l_3 = 7$ и получаем последовательность (6, 7, 4) с суммарным штрафом 16.

в) Пусть $l_1 = 6$. Уменьшая значения D_k для всех $k \in M$, $k \neq 6$, на величину $t_6 = 2$, перейдем к упорядочению в интервале (0, 20) требований, параметры которых приведены в следующей таблице:

k	4	5	7
t_k	8	5	7
D_k	6	8	10

Интервал $(0, 20)$ является интервалом очередности $5 \to \{4, 7\}$. Получаем последовательности $(5, 4, 7)$ и $(5, 7, 4)$ с суммарным штрафом 17 и 16 соответственно.

4. Выберем среди последовательностей, полученных в случаях а), б), в), последовательность с наименьшим штрафом. Это — последовательность $(6, 5, 7)$ с $l_1 = 4$ и суммарным штрафом, равным 15. Таким образом, искомой последовательностью является последовательность $\pi^* = (4, 6, 5, 7, 1, 3, 2)$ и $F_\Sigma(\pi^*) = 84$.

В практическом отношении целесообразно вычислять нижнюю оценку значений суммарного штрафа на множестве всех последовательностей с заданным начальным и конечным фрагментами. Это позволяет в последующем развивать наиболее перспективные направления ветвления. Если для некоторой последовательности π значение $F_\Sigma(\pi)$ оказывается не больше значений суммарного штрафа каждой из остальных полученных последовательностей и не больше нижних границ остальных, подлежащих развитию вариантов, то π, очевидно,— искомая последовательность.

§ 4. Задача коммивояжера

Задача коммивояжера является одной из наиболее известных экстремальных комбинаторных задач. Обширный круг возможных приложений, сосредоточение характерных для задач дискретного программирования трудностей, наряду с простотой и прозрачностью формулировки, привлекает к этой задаче внимание многих исследователей. Многие общеизвестные в настоящее время приемы и методы дискретного программирования впервые были наиболее четко сформулированы при разработке эффективных алгоритмов решения задачи коммивояжера.

4.1. Необходимость рассмотрения задачи коммивояжера и различных ее обобщений в теории расписаний возникает, как правило, при изучении ситуаций, в которых качество расписания существенно зависит от правильной организации разнообразных пуско-наладочных и транспортных работ, связанных с эксплуатацией обслуживающих приборов. Опишем одну из характерных ситуаций такого рода.

Пусть n требований обслуживается одним прибором. Все требования поступают на обслуживание в момент времени $d = 0$. Длительность обслуживания требования k равна t_k единиц времени. Если требование k обслуживается первым, то для подготовки прибора к обслуживанию этого требования необходимо δ_{0k} единиц времени, $k = \overline{1, n}$. Если требование j обслуживается непосредственно после требования i, то для переналадки прибора с обслуживания требования i на обслуживание требования j необходимо δ_{ij} единиц времени, $1 \leqslant i \neq j \leqslant n$. Требуется так организовать процесс обслуживания требований (т. е. указать такую их последовательность), чтобы общее время обслуживания всех требований было наименьшим. Нетрудно видеть, что искомой последовательности при этом должно соответствовать наименьшее общее время пуско-наладочных работ. В ряде случаев желательно также учитывать время δ_{k0}, требуемое на приведение обслуживающего прибора в исходное состояние, если требование k обслуживается последним, $k = \overline{1, n}$. Таким образом, необходимо найти такую последовательность $\pi = (i_1, i_2, \ldots, i_n)$ обслуживания требований, чтобы величина $\delta_{0i_1} + \sum\limits_{k=1}^{n-1} \delta_{i_k i_{k+1}} + \delta_{i_n 0}$ была наименьшей.

Эта задача допускает простую геометрическую интерпретацию. Рассмотрим полный симметрический ориентированный граф $(X, \vec{U})$; $X = \{0, 1, \ldots, n\}$ — множество вершин; $\vec{U}$ — множество дуг. Каждой дуге (i, j) припишем число δ_{ij}, которое будем называть длиной этой дуги. Требуется найти гамильтонов контур, т. е. контур, проходящий через каждую вершину один и только один раз, которому соответствует наименьшая длина. Под длиной контура понимается величина, равная сумме длин его дуг.

Задача коммивояжера обязана своим названием несколько более частной геометрической ситуации. Пусть X — множество городов, а $l_{ij} \geqslant 0$ — расстояние между городами i и j. Вообще говоря, $l_{ij} \neq l_{ji}$, учитывая, например, наличие различных магистралей одностороннего движения, связывающих города i и j. Коммивояжер должен выехать из города 0, объехать все остальные города ровно по одному разу и вернуться в исходный город 0.

Необходимо указать такую последовательность объезда городов, при которой коммивояжеру требуется проехать наименьшее суммарное расстояние.

Задача называется симметрической, если $l_{ij} = l_{ji}$, и геометрической, если при этом значения l_{ij} удовлетворяют всем аксиомам метрики евклидовой плоскости.

4.2. Задача коммивояжера допускает формулировку в терминах смешанного линейного программирования. К сожалению, получаемые при этом модели оказываются весьма громоздкими и, что особенно неблагоприятно, содержат большое число целочисленных переменных. Тем не менее сам процесс постановки задачи коммивояжера как задачи математического программирования представляет определенный интерес.

Введем переменные x_{ij}, $0 \leqslant i \neq j \leqslant n$, принимающие значения $x_{ij} = 1$, если дуга (i, j) принадлежит гамильтонову контуру, и значение $x_{ij} = 0$ в противном случае. Всего гамильтоновых контуров $n!$. Обозначим через $\underline{\Omega}$ совокупность $n!$ наборов $\{x_{ij}\}$ значений x_{ij}, соответствующих этим контурам. Таким образом, задачу коммивояжера можно записать в виде

$$\sum_{i \neq j} l_{ij} x_{ij} \to \min, \tag{4.1}$$

$$\{x_{ij}\} \in \underline{\Omega}. \tag{4.2}$$

Обозначим через Ω совокупность наборов $\{x_{ij}\}$ значений x_{ij}, удовлетворяющих условиям

$$\sum_{j=0}^{n} x_{ij} = 1, \quad i = \overline{0, n}, \quad i \neq j, \tag{4.3}$$

$$\sum_{i=0}^{n} x_{ij} = 1, \quad j = \overline{0, n}, \ j \neq i. \tag{4.4}$$

Нетрудно убедиться, что $\underline{\Omega} \subset \Omega$. Кроме наборов $\{x_{ij}\}$ значений x_{ij}, которым соответствуют гамильтоновы контуры, множество Ω содержит также наборы $\{x_{ij}\}$ значений x_{ij}, которым соответствуют несколько изолированных контуров.

Чтобы исключить подобные ситуации, необходимо наложить некоторые дополнительные условия на наборы

$\{x_{ij}\}$, которые естественно называть условиями связности. Заметим, что задача (4.1), (4.3) и (4.4) является известной задачей о назначениях, для которой существуют весьма эффективные методы решения. Эта задача неадекватна задаче коммивояжера из-за отсутствия условий связности, исключающих возможность получения нескольких изолированных контуров.

Условия связности могут быть записаны (без введения дополнительных переменных) в виде

$$\sum_{i,\,j\in M_k^q} x_{ij}<k,\quad 1\leqslant k\leqslant\left[\frac{n+1}{2}\right],\ 1\leqslant q\leqslant C_{n+1}^k, \qquad (4.5)$$

где M_k^q — q-е подмножество множества $X=\{0,1,2,\ldots,n\}$, содержащее k элементов; $[a]$ — целая часть a.

Всякий набор $\{x_{ij}\}\in\underline{\Omega}$ удовлетворяет, очевидно, ограничениям (4.5). Покажем, что всякий набор $\{x_{ij}\}\in\in\Omega$, удовлетворяющий ограничениям (4.5), принадлежит $\underline{\Omega}$.

Действительно, если набору $\{x_{ij}\}\in\Omega$ соответствует несколько изолированных контуров, то хотя бы один из этих контуров содержит не более $\left[\frac{n+1}{2}\right]$ вершин, пусть это будет контур $(\nu_1, \nu_2,\ldots, \nu_l, \nu_1)$. Следовательно, $x_{\nu_1\nu_2}= x_{\nu_2\nu_3}=\ldots=x_{\nu_l\nu_1}=1$ и $\sum\limits_{i,\,j\in M_l^q} x_{ij}=l$, где $M_l^q=\{\nu_1\nu_2\ldots\nu_l\}$, что противоречит условию (4.5).

Таким образом, задача (4.1), (4.3), (4.4) и (4.5) эквивалентна задаче коммивояжера.

Та же цель может быть достигнута введением дополнительных переменных, играющих роль своеобразных счетчиков дуг или вершин, встречающихся при обходе контура.

В случае организации «счетчика дуг» каждой дуге (i, j) относят переменную $y_{ij}\geqslant 0$, удовлетворяющую условию

$$y_{ij}\leqslant x_{ij},\quad 0\leqslant i\neq j\leqslant n. \qquad (4.6)$$

Тем самым набору $\{x_{ij}\}\in\Omega$ ставится в соответствие некоторый набор $\{y_{ij}\}$ значений y_{ij}. При этом нулевым значениям x_{ij} соответствуют значения y_{ij}, равные нулю. Если $x_{ij} = 1$, то y_{ij}, вообще говоря, не равно нулю.

Поскольку набору $\{x_{ij}\} \in \Omega$ соответствует или гамильтонов контур, или несколько изолированных контуров, то в любом случае можно выбрать контур, содержащий вершину 0. Произведем обход этого контура, начиная с вершины 0. Пусть первой из дуг этого контура приписано значение $y_{ij} = y_0$, второй значение $y_{ij} = y_0 + p$ (или $y_0 \cdot q$), третьей значение $y_{ij} = y_0 + 2p$ (или $y_0 \cdot q^2$) и т. д., где p и q — некоторые действительные числа.

Если рассматриваемый контур является гамильтоновым, то последней дуге должно быть приписано значение $y_{ij} = y_0 + np$ (или $y_0 \cdot q^n$). В противном случае ей окажется приписанным число $y_{ij} = y_0 + n_1 p$ (или $y_0 \cdot q^{n_1}$), где $n_1 < n$.

Таким образом, условия связности могут быть записаны в виде

$$\sum_{j=0}^{n} y_{ij} = \sum_{j=0}^{n} y_{ji} + p, \quad i = \overline{1, n}, \quad i \neq j, \ 0 < p = \text{const} \leqslant \frac{1}{n} \tag{4.7}$$

или

$$\sum_{j=0}^{n} y_{ij} = q \sum_{j=0}^{n} y_{ji}, \quad i = \overline{1, n}, \ i \neq j, \ 0 < q = \text{const} < 1, \tag{4.8}$$

$$\sum_{j=1}^{n} y_{0j} = 1, \ \sum_{j=1}^{n} y_{j0} = q^n \tag{4.9}$$

и задача (4.1), (4.3), (4.4), (4.6) и (4.7) или (4.8), (4.9) эквивалентна задаче коммивояжера.

При организации «счетчика вершин» достаточно ввести n новых переменных $z_i \geqslant 0$, $i = \overline{1, n}$, означающих порядковый «номер» вершины i при обходе контура, содержащего вершину 0. Условия связности в этом случае могут быть записаны в виде

$$z_i - z_j \leqslant n(1 - x_{ij}) - 1, \quad 1 \leqslant i \neq j \leqslant n. \tag{4.10}$$

Действительно, пусть набор $\{x_{ij}\}$ принадлежит Ω и ему соответствует гамильтонов контур $(0, \nu_1, \nu_2, \ldots, \nu_n, 0)$, т. е. $x_{0\nu_1} = x_{\nu_1\nu_2} = \ldots = x_{\nu_{n-1}\nu_n} = x_{\nu_n 0} = 1$, остальные $x_{ij}=0$. Положим $z_{\nu_1}=1$, $z_{\nu_2}=2, \ldots, z_{\nu_n}=n$. Непосредственной проверкой убеждаемся в справедливости условий (4.10). Если $\{x_{ij}\}\in\Omega$ и $\{x_{ij}\}\notin\underline{\Omega}$, то набору $\{x_{ij}\}$ соответствует несколько изолированных контуров. Выберем контур $(\mu_1, \mu_2, \ldots, \mu_l, \mu_1)$, не содержащий вершины 0, $x_{\mu_1\mu_2}=x_{\mu_2\mu_3}=\ldots=x_{\mu_{l-1}\mu_l}=x_{\mu_l\mu_1}=1$. Покажем, что не существует чисел $z_{\mu_1}, z_{\mu_2}, \ldots, z_{\mu_l}\geqslant 0$, удовлетворяющих ограничениям (4.10). По условию $z_{\mu_2}\geqslant z_{\mu_1}+1, \ldots, z_{\mu_l}\geqslant z_{\mu_{l-1}}+1$, т. е. $z_{\mu_l}\geqslant z_{\mu_1}+(l-1)$, но $z_{\mu_1}\geqslant z_{\mu_l}+1$ и $l\geqslant 2$. Получаем противоречие. Следовательно, задача коммивояжера эквивалентна задаче (4.1), (4.3), (4.4) и (4.10).

Введение переменных двух типов и разделение ограничений на две группы, первая из которых обеспечивает «контурность», а вторая «связность» решения, является вполне оправданным, но не необходимым приемом. Вполне очевидна возможность построения линейной модели, переменные которой несут двойную содержательную нагрузку.

Пусть переменная x_{ij}^k принимает значения 1 или 0, в зависимости от того, встречается дуга (i, j) при обходе гамильтонова контура (начиная с вершины 0) k-й по порядку или нет, $0\leqslant i\neq j\leqslant n$, $k=\overline{1, n+1}$. Тогда задача коммивояжера может быть сформулирована в виде

$$\sum_{k=1}^{n+1}\sum_{0\leqslant i\neq j\leqslant n} l_{ij}x_{ij}^k\to\min \tag{4.11}$$

при ограничениях

$$\sum_{0\leqslant i\neq j\leqslant n} x_{ij}^1=\sum_{j=1}^{n} x_{0j}^1=1, \tag{4.12}$$

$$\sum_{i=0}^{n} x_{ij}^k=\sum_{i=0}^{n} x_{ji}^{k+1}, \quad j=\overline{1, n}, \quad k=\overline{1, n}. \tag{4.13}$$

Эти ограничения, по существу, означают, что если дуга (i, j) входит в гамильтонов контур и при его обходе, начиная с вершины 0, встречается k-й по порядку, то одно из значений $x_{jl}^{k+1}=1$.

Перечисленные линейные модели задачи коммивояжера отличаются друг от друга как числом переменных, так и числом и видом ограничений. Априори отдать предпочтение той или иной модели не представляется возможным. Немногочисленные попытки непосредственного применения методов целочисленного линейного программирования к решению задачи коммивояжера не привели к сколь-нибудь обнадеживающим результатам.

Определенный теоретический интерес представляет формулировка задачи коммивояжера в терминах целочисленного квадратичного программирования. Введем переменные x_i^k, $i,\ k = \overline{1, n}$, принимающие значения 0 и 1. Если $x_i^k = 1$, то содержательно это означает, что при обходе гамильтонова контура, начиная с вершины 0, вершина i встречается k-й по порядку. В противном случае $x_i^k = 0$.

Таким образом, значения x_i^k должны удовлетворять ограничениям

$$\sum_{i=1}^{n} x_i^k = 1, \quad k = \overline{1, n}, \tag{4.14}$$

$$\sum_{k=1}^{n} x_i^k = 1, \quad i = \overline{1, n}, \tag{4.15}$$

и доставлять наименьшее значение квадратичной форме

$$\sum_{i=1}^{n} (l_{0i}x_i^1 + l_{i0}x_i^n) + \sum_{k=1}^{n-1} \sum_{1 \leqslant i \neq j \leqslant n} l_{ij}x_i^k x_j^{k+1}. \tag{4.16}$$

Эта задача не является задачей выпуклого программирования.

4.3. Основой многих методов решения задачи коммивояжера служит систематическое использование следующего очевидного свойства решения этой задачи. Пусть последовательность вершин $(0, i_1^*, i_2^*, \ldots, i_k^*, i_{k+1}^*, \ldots, i_{l-1}^*, i_l^*, \ldots \ldots, i_n^*, 0)$ определяет искомый оптимальный контур. Тогда длина пути $(i_k^*, i_{k+1}^*, \ldots, i_{l-1}^*, i_l^*)$ не больше длины пути $(i_k^*, j_{k+1}, \ldots, j_{l-1}, i_l^*)$, где $(j_{k+1}, \ldots, j_{l-1})$ — произвольная последовательность вершин множества $\{i_{k+1}^*, \ldots, i_{l-1}^*\}$.

Задача коммивояжера относится к классу задач, рассмотренных в § 5 гл. 2.

Вычислительная схема в данном случае может быть описана следующим образом.

Обозначим через Q произвольное подмножество множества $N = \{1, 2,\ldots, n\}$, а через $\bar{F}(Q; k)$ — длину кратчайшего пути из вершины 0 в вершину $k \notin Q$, проходящего через вершины множества Q. Величина $\bar{F}(N; 0)$ — длина искомого гамильтонова контура.

По определению $\bar{F}(\phi; k) = l_{0k}$, $k = \overline{1, n}$. Учитывая отмеченное выше свойство решения задачи коммивояжера, можно записать рекуррентное соотношение

$$\bar{F}(Q; k) = \min_{i \in Q} [\bar{F}(Q \setminus i; i) + l_{ik}], \quad Q \subseteq N, \ Q \neq \phi. \tag{4.17}$$

Таким образом, для вычисления значения $\bar{F}(N; 0)$ необходимо располагать n значениями $\bar{F}(N \setminus i; i)$, $i \in N$. Если в соотношении (4.17) минимум достигается при $i = i'$, то $\overset{*}{i}_n = i'$. Положим $Q = N \setminus \overset{*}{i}_n$, $k = \overset{*}{i}_n$. Из соотношения (4.17) может быть найдено значение $i = i''$, при котором достигается минимум. Следовательно, $\overset{*}{i}_{n-1} = i''$. Полагая, далее, $Q = N \setminus \{\overset{*}{i}_n, \overset{*}{i}_{n-1}\}$, аналогичным образом может быть найдено значение $\overset{*}{i}_{n-2}$ и т. д., пока не будет найдено значение $\overset{*}{i}_1$.

Процесс построения кратчайшего гамильтонова контура включает последовательное вычисление значений $\bar{F}(Q; k)$ для $|Q| = 1, 2,\ldots, n-1$, вычисление значения $\bar{F}(N, 0)$ и запоминание тех значений i, при которых в соотношении (4.17) достигается минимум.

Возможны различные схемы организации вычислений. Наиболее распространенной является схема, при которой сначала вычисляются и запоминаются все значения $\bar{F}(Q; k)$ для $|Q| = 1$, затем все значения $\bar{F}(Q, k)$ для $|Q| = 2$ и т. д., пока не будет вычислено значение $\bar{F}(N; 0)$. При реализации этой вычислительной схемы лимитирующим оказывается объем промежуточных результатов, которые необходимо запомнить для проведения последующих вычислений. Так, для вычисления всех значений $\bar{F}(Q; k)$ при $|Q| = 2$ необходимо располагать $(n-1)\,C_n^1$ значениями $\bar{F}(Q; k)$ при $|Q| = 1$, для вычисления всех значений $\bar{F}(Q; k)$ при $|Q| = 3$ необходимо располагать

$(n-2)\,C_n^2$ значениями $\tilde{F}(Q;k)$, вычисленными при $|Q|=2$ и т. д. Тем не менее с использованием средних ЭВМ оказывается возможным решать задачи коммивояжера с $n \leqslant 15$.

Объем одновременно хранимых промежуточных результатов существенно уменьшается, если воспользоваться иной вычислительной схемой. Согласно этой схеме при вычислении значения $\tilde{F}(\overline{Q};\overline{k})$ для каждой конкретной пары $(\overline{Q},\overline{k})$ последовательно вычисляются, начиная с $|Q|=1$, только те значения $\tilde{F}(Q;k)$, которые необходимы для вычисления значения $\tilde{F}(\overline{Q},\overline{k})$. При этом, естественно, происходит значительное дублирование вычислений, однако объем запоминаемой промежуточной информации уменьшается. Тем самым оказывается возможным решать задачи с $n \leqslant 18$ за практически приемлемое время.

При разработке вычислительных процедур для конкретных ЭВМ целесообразно сочетать различные приемы с целью максимального использования возможностей ЭВМ.

4.4 Прежде чем переходить к описанию метода ветвей и границ решения задачи коммивояжера, сделаем одно общее замечание. Пусть, как и прежде, $\|l_{ij}\|$ — матрица расстояний между вершиной i и вершиной j, $0 \leqslant i \neq j \leqslant n$. Положим $l_{ii}=\infty$, $i=\overline{0,n}$. Значения l_{ij} больше или равны 0. Вычтем из какой-либо строки (или столбца) матрицы $\|l_{ij}\|$ положительное число l, не превосходящее элементов этой строки (или столбца). Обозначим полученную в результате матрицу через $\|l'_{ij}\|$, $l'_{ij} \geqslant 0$, $0 \leqslant i \neq j \leqslant n$. Решая задачу коммивояжера с матрицей расстояний $\|l'_{ij}\|$, получаем, очевидно, тот же гамильтонов контур, что и при решении этой задачи с матрицей расстояний $\|l_{ij}\|$, однако длина полученного контура будет на l единиц меньше. Другими словами, длина искомого контура (и тем самым любого гамильтонова контура) не меньше величины l. Это обстоятельство позволяет получать нижнюю оценку длин гамильтоновых контуров в результате последовательного вычитания из строк и столбцов матрицы $\|l_{ij}\|$ неотрицательных величин $l_1, l_2, \ldots, l_r$, пока в каждой строке и каждом столбце результирующей матрицы (с неотрицательными элементами) не будет содержаться хотя бы один нулевой элемент. Искомой оценкой является величина

$$\gamma\|l_{ij}\| = \sum_{k=1}^{r} l_k. \tag{4.18}$$

Процесс построения оптимального гамильтонова контура методом ветвей и границ представляется процессом последовательного разбиения множества всех гамильтоновых контуров на подмножества с последующим определением нижней границы длин контуров каждого подмножества. При этом к одному подмножеству $\{R, S\}$ относят все те гамильтоновы контуры, которые содержат все дуги некоторого множества $R \subset \vec{U}$ и не содержат дуг некоторого другого множества $S \subset \vec{U}$, $R \cap S = \phi$.

Если R и S таковы, что множество $\{R, S\}$ не пусто, то нижняя оценка $\gamma\{R, S\}$ длин гамильтоновых контуров этого множества может быть вычислена следующим образом. Для всех $(i, j) \in R$ в матрице $\|l_{ij}\|$ удаляем i-ю строку и j-й столбец. Для всех $(i, j) \in S$ в матрице $\|l_{ij}\|$ заменяем элемент l_{ij} достаточно большим числом. Обозначим полученную в результате матрицу через $\|\bar{l}_{ij}\|$. Описанным выше способом находим величину $\gamma\|\bar{l}_{ij}\|$. Полагаем

$$\gamma\{R, S\} = \sum_{(i, j)\in R} l_{ij} + \gamma\|\bar{l}_{ij}\|. \tag{4.19}$$

Если множество $\{R, S\} = \phi$, то полагаем значение $\gamma\{R, S\} = \infty$.

На первом этапе множество всех гамильтоновых контуров разбивается на два подмножества вида $\{R_1^1 = (i_1, j_1), S_1^1 = \phi\}$ и $\{R_2^1 = \phi, S_2^1 = (i_1, j_1)\}$. Вычисляются значения $\gamma\{R_1^1, S_1^1\}$ и $\gamma\{R_2^1, S_2^1\}$.

Если $\gamma\{R_1^1, S_1^1\} \leqslant \gamma\{R_2^1, S_2^1\}$, то множество $\{R_1^1, S_1^1\}$ разбивается на два подмножества вида $\{R_{11}^2 = \{R_1^1, (i_2, j_2)\}, S_{11}^2 = \phi\}$ и $\{R_{12}^2 = R_1^1, S_{12}^2 = (i_2, j_2)\}$. Вычисляются значения $\gamma\{R_{11}^2, S_{11}^2\}$ и $\gamma\{R_{12}^2, S_{12}^2\}$. Множество всех гамильтоновых контуров оказывается разбитым на три подмножества вида $\{R_2^1, S_2^1\}$, $\{R_{11}^2, S_{11}^2\}$ и $\{R_{12}^2, S_{12}^2\}$. Для последующего разбиения выбирается то из этих подмножеств, которому соответствует наименьшее значение $\gamma\{R, S\}$.

Если $\gamma\{R_1^1, S_1^1\} > \gamma\{R_2^1, S_2^1\}$, то множество $\{R_2^1, S_2^1\}$ разбивается на два подмножества следующего вида: $\{R_{21}^2 = (i_2, j_2), S_{21}^2 = S_2^1\}$, $\{R_{22}^2 = R_2^1, S_{22}^2 = \{S_2^1, (i_2, j_2)\}\}$. Вычисляются значения $\gamma\{R_{21}^2, S_{21}^2\}$ и $\gamma\{R_{22}^2, S_{22}^2\}$. Множество всех гамильтоновых контуров оказывается разбитым на три подмножества вида $\{R_1^1, S_1^1\}$, $\{R_{21}^2, S_{21}^2\}$ и $\{R_{22}^2, S_{22}^2\}$. Для последующего разбиения выбирается то из этих подмножеств, которому соответствует наименьшее значение $\gamma\{R, S\}$.

Процесс продолжается до тех пор, пока не будет получено разбиение множества всех гамильтоновых контуров на подмножества вида $\{R, S\}$, удовлетворяющее следующему условию. Хотя бы одно из подмножеств $\{R, S\}$ содержит ровно один гамильтонов контур и значение $\gamma(R, S\}$ для этого подмножества (являющееся, по существу, длиной этого контура) не больше значений $\gamma\{R, S\}$ остальных подмножеств разбиения. Указанный контур является искомым.

При разбиении множества $\{R, S\}$ на два подмножества вида $\{\{R, (i', j')\}, S\}$ и $\{R, \{S, (i', j')\}\}$ необходимо располагать некоторым правилом, согласно которому осуществляется выбор дуги (i', j'). Пусть $\|\tilde{l}_{ij}\|$ — матрица, полученная из матрицы $\|l_{ij}\|$ в результате вычитания неотрицательных чисел из элементов ее строк и столбцов. Все элементы матрицы $\|\tilde{l}_{ij}\|$ неотрицательны и в каждой строке и каждом столбце этой матрицы содержится хотя бы один нулевой элемент. В качестве (i', j') естественно выбирать ту из дуг, которой соответствует нулевой элемент в матрице $\|\tilde{l}_{ij}\|$. Среди всех таких дуг выбирается дуга (i', j'), которой соответствует наибольшее значение суммы наименьшего элемента в i'-й строке и наименьшего элемента в j'-м столбце этой матрицы. При этом сам элемент $\tilde{l}_{i'j'}$ не учитывается.

Объем вычислений, которые необходимо провести при решении задачи коммивояжера описанным методом, и объем запоминаемых промежуточных результатов существенно зависят от конкретных условий задачи и могут колебаться в весьма широких пределах.

В гл. 8, § 1, п. 1.3 описан метод решения задачи коммивояжера, использующий тот факт, что формулировка этой задачи в терминах целочисленного линейного программирования отличается от задачи о назначениях только наличием условий связности искомого контура.

4.5. Локальные методы решения задачи коммивояжера составляют обширную группу приближенных методов, основанных на понятии окрестности произвольного допустимого варианта (гамильтонова контура). Окрестность состоит из конечного числа гамильтоновых контуров, в некотором смысле близких к рассматриваемому. Как правило, переход от рассмотрения одного контура к рассмотрению другого осуществляется в случае, когда второй контур принадлежит окрестности первого и имеет меньшую длину. В некоторых вычислительных схемах переход осуществляется только в случае, когда второй контур имеет наименьшую длину среди всех контуров окрестности. Однако в любом случае сохраняется стремление перейти к рассмотрению контура с возможно меньшей длиной. Контур, длина которого не больше длины любого контура его окрестности, является локально оптимальным. В качестве искомого принимается локально оптимальный контур или наилучший среди нескольких найденных локально оптимальных контуров.

При определении окрестности контура наибольшее распространение получил принцип «три в три». Согласно этому принципу окрестностью контура является совокупность контуров, отличающихся от данного ровно тремя дугами. Такой окрестностью является, например, совокупность контуров, отличающихся от данного транспозицией двух рядом встречающихся при его обходе вершин.

Для симметрических задач используется также принцип «два в два», согласно которому окрестность контура определяется таким образом, что любой контур этой окрестности отличается от данного ровно двумя ребрами.

Локальные методы практически применялись для решения симметрических, особенно геометрических, задач коммивояжера. В частности, были получены наилучшие из известных решений для геометрических задач с 48 и 57 вершинами.

§ 5. Взаимосвязанные требования

Во многих практических ситуациях качество расписания определяется тем, насколько полно учтены специфические условия, в которых протекает процесс обслуживания требований. Такие условия могут, например, отражать стремление к определенному группированию требований при их обслуживании, к сокращению временных промежутков между обслуживанием отдельных пар требований, к уменьшению времени «пролеживания» обслуженного требования до завершения обслуживания остальных требований и т. п.

В этом параграфе мы рассмотрим две типичные задачи такого рода, предполагая, что все требования множества $N = \{1, 2, \ldots, n\}$ поступают в очередь на обслуживание в момент времени $d = 0$, обслуживаются последовательно и непрерывно.

5.1. Сопоставим каждой упорядоченной паре требований некоторое действительное число $a_{ij} \geqslant 0$. Числа a_{ij} и a_{ji} называются *коэффициентами взаимосвязанности* требований i и j.

Если требования обслуживаются в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, то обслуживание требования i_k завершается в момент времени $\bar{t}_{i_k} = \sum_{j=1}^{k} t_{i_j}$, где t_{i_j} — время обслуживания требования i_j.

Необходимо указать такую последовательность обслуживания требований, которой соответствует наименьшее значение величины

$$F_{\text{вз}}(\pi) = \sum_{\nu<\mu} a_{i_\nu i_\mu} (\bar{t}_{i_\mu} - \bar{t}_{i_\nu}). \tag{5.1}$$

Примеры.

1. Учебный процесс состоит в последовательном изучении конечного множества $N = \{1, 2, \ldots, n\}$ отдельных тем. Тема l для своего изучения требует t_l единиц времени. Как правило, темы не могут изучаться в произвольной последовательности. На множестве N всех тем задано некоторое отношение строгого порядка, показывающее изучение каких именно тем должно предшествовать во времени изучению данной темы. Если для изучения темы j

необходимо сначала изучить тему i, то естественным является стремление сократить промежуток времени между изучением этих тем. Аналогичная ситуация будет иметь место для любой пары взаимосвязанных тем. «Степень взаимосвязанности» тем i и j в рассматриваемом случае можно охарактеризовать числами $a_{ij} > 0$ и $a_{ji} = R$, где R — достаточно большое число. Тем самым учитывается как крайняя нежелательность изучения темы j до изучения темы i, так и различия в необходимости сокращения промежутков времени между изучением отдельных взаимосвязанных тем. Если темы i и j могут изучаться независимо друг от друга, то полагаем $a_{ij} = a_{ji} = 0$.

2. На магнитной ленте требуется последовательно разместить n массивов информации, каждый из которых имеет длину b. Вероятность обращения к массиву с индексом l равна p_l. Поиск требуемого массива осуществляется движением вдоль ленты вправо или влево, считывание массива производится слева направо с последующей фиксацией считывающей головки в положении, соответствующем правому концу искомого массива — блуждающий поиск, поиск без возврата в исходную позицию.

Если массивы расположены на ленте слева направо в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, то средняя длина пробега при поиске массива равна

$$B(\pi) = \sum_{\nu<\mu} p_{i_\nu} p_{i_\mu} (\mu - \nu - 1) b + \sum_{\nu \leqslant \mu} p_{i_\nu} p_{i_\mu} (\mu - \nu + 1) b =$$

$$= E \sum_{\nu<\mu} p_{i_\nu} p_{i_\mu} (\mu - \nu) + L, \qquad (5.2)$$

где величины E и L не зависят от π.

В данном случае естественно положить $t_i = 1$, $a_{ij} = p_i p_j$, $1 \leqslant i \neq j \leqslant n$.

5.2. Задача построения последовательности π^*, которой соответствует наименьшее значение (5.1), относится к классу задач, рассмотренных в § 5 гл. 2.

Действительно, введем в рассмотрение функцию

$$F(\pi, k) = \sum_{\nu<\mu\leqslant k} a_{i_\nu i_\mu} (\bar{t}_{i_\mu} - \bar{t}_{i_\nu}) + \sum_{\nu<k<\mu} a_{i_\nu i_\mu} (\bar{t}_{i_{k-1}} + t_{i_\mu} - \bar{t}_{i_\nu}), \qquad (5.3)$$

где $k = \overline{1, n}$.

Очевидно, $F_{\text{вз}}(\pi) = F(\pi, n)$. Имеем

$$F(\pi, k) = F(\pi, k-1) + t_{i_{k-1}} \sum_{\nu<k-1<\mu} a_{i_\nu i_\mu} + \sum_{k-1<\mu} a_{i_{k-1} i_\mu} t_{i_\mu}, \tag{5.4}$$

где $F(\pi, 1) = 0$.

Следовательно, значение $F(\pi, k)$ однозначно определяется значением $F(\pi, k-1)$, множеством $\{i_1, i_2, \ldots, i_{k-1}\}$ и значением i_{k-1}. Это соответствует случаю $s = 1$, $t = 0$.

Если все $a_{ij} = 1$, $1 \leqslant i \neq j \leqslant n$, то

$$F_{\text{вз}}(\pi) = 0 \cdot t_{i_1} + (n-1) t_{i_2} + 2(n-2) t_{i_3} + \ldots \\ \ldots + 2(n-2) t_{i_{n-1}} + (n-1) t_{i_n}. \tag{5.5}$$

Коэффициенты при t_{i_j} образуют сначала возрастающую, затем убывающую последовательности.

На основании теоремы 4.1 гл. 2 в этом случае достаточно пронумеровать требования в порядке невозрастания значений t_l. Требование с номером 1 следует обслуживать первым, требование с номером 2 — последним, требование с номером 3 — вторым, требование с номером 4 — предпоследним и т. д., пока не будут исчерпаны все требования.

Аналогичный алгоритм построения оптимальной последовательности имеет место и в случае, когда $a_{ij} = a_i a_j$ и $t_j = t$, $1 \leqslant i \neq j \leqslant n$.

5.3. Рассмотрим задачу минимизации суммарного взвешенного времени «пролеживания» обслуженных требований.

Пусть требования обслуживаются в последовательности $\pi = (i_1, i_2, \ldots, i_n)$. Время завершения обслуживания требования i_k равно $\bar{t}_{i_k} = \sum_{j=1}^{k} t_{i_j}$, а время завершения обслуживания всех требований $T = \sum_{l=1}^{n} t_l$.

Требование i_k, обслуживание которого завершается в момент времени $\bar{t}_{i_k}$, в течение $T - \bar{t}_{i_k}$ единиц времени находится в состоянии ожидания завершения обслуживания остальных требований, что сопряжено с определенными затратами. Пусть $a_{i_k} > 0$ — стоимость пролеживания требования i_k в течение одной единицы времени.

Необходимо указать такую последовательность обслуживания требований, при которой суммарная стоимость пролеживания

$$F_{\text{пр}}(\pi) = \sum_{k=1}^{n} a_{i_k}(T - \bar{t}_{i_k}) \tag{5.6}$$

является наименьшей.

Используя перестановочный прием (см. § 4 гл. 2), приходим к заключению, что требования следует обслуживать в порядке неубывания значений a_l/t_l.

§ 6. Древовидно упорядоченные требования

Рассмотренные во втором параграфе этой главы ситуации позволяют получить оптимальные последовательности обслуживания требований в результате весьма простых рассуждений. Если все функции штрафа линейные, или все экспоненциальные (с одним и тем же коэффициентом при показателе степени), или все отличаются от некоторой монотонной функции на константу, то каждому требованию k можно поставить в соответствие некоторую величину $\omega(k)$ и упорядочить требования в порядке невозрастания этих величин. Значение $\omega(k)$ при этом однозначно определяется параметрами требования k.

Если множество N требований частично упорядочено, то полученные оптимальные последовательности могут противоречить заданному на N порядку. В этом случае желательно располагать эффективными алгоритмами, позволяющими находить оптимальные последовательности среди последовательностей, допустимых относительно заданного на N отношения строгого порядка.

Пусть $(N, \rightarrow)$ — множество N требований с заданным на нем отношением строгого порядка $\rightarrow$. Обслуживание любого требования не может быть начато до того, как закончится обслуживание всех предшествующих ему требований. Предполагается, что редукция отношения порядка $\rightarrow$ представима графом, все компоненты связности которого — прадеревья с корнями. Все требования поступают в очередь на обслуживание в момент времени $d = 0$.

Обозначим через $F_л(\pi)$ и $F_э(\pi)$ — суммарный штраф за обслуживание требований в последовательности π в случаях, когда все функции штрафа $\varphi_k(x) = a_k x + b_k$ $(a_k > 0)$, или $\varphi_k(x) = a_k \exp(\alpha x) + b_k$ $(a_k > 0, \alpha > 0)$, $k = \overline{1, n}$, соответственно.

Приведем алгоритмы построения допустимых (относительно заданного на N древовидного порядка) последовательностей, которым соответствуют наименьшие значения $F_л(\pi)$ и $F_э(\pi)$. В дальнейшем именно эти последовательности будем называть *оптимальными*. Указанные алгоритмы во многом идентичны, что позволяет совместить их описание.

Обозначение $F(\pi)$ будем употреблять при формулировке утверждений, справедливых как для $F_л(\pi)$, так и для $F_э(\pi)$.

В заключение параграфа опишем прием, позволяющий построить оптимальную последовательность обслуживания требований, допустимую относительно порядка, противоположного порядку, заданному на N.

6.1. Прежде чем переходить к непосредственному описанию алгоритмов, сделаем несколько общих замечаний.

Если требования множества $\{\sigma\} = \{i_1, i_2, \dots, i_q\} \subset N$ обслуживаются в последовательности $\sigma = (i_1, i_2, \dots, i_q)$, начиная с момента времени $d = 0$, то суммарный штраф за их обслуживание будем обозначать через $F(\sigma)$.

Нетрудно видеть, что если $\pi = (\sigma^{(1)}, \sigma^{(2)})$, то

$$F_л(\pi) = F_л(\sigma^{(1)}) + F_л(\sigma^{(2)}) + \sum_{i \in \{\sigma^{(2)}\}} a_i \sum_{j \in \{\sigma^{(1)}\}} t_j, \tag{6.1}$$

$$F_э(\pi) = F_э(\sigma^{(1)}) + \exp\left(\alpha \sum_{j \in \{\sigma^{(1)}\}} t_j\right)\left[F_э(\sigma^{(2)}) - \sum_{i \in \{\sigma^{(2)}\}} b_i\right] + \sum_{i \in \{\sigma^{(2)}\}} b_i. \tag{6.2}$$

Следовательно, если известно, что в оптимальной последовательности первыми обслуживаются требования множества $\{\sigma^{(1)}\}$, то исходная задача редуцируется к аналогичной задаче оптимального упорядочения требований множества $\{\sigma^{(2)}\}$.

Отметим, что если $\varphi_k(x) = \varphi(x) + b_k$, $k = \overline{1, n}$, $\varphi(x)$ — неубывающая функция, то оптимальная последовательность обслуживания требований, вообще говоря, зависит от времени начала их обслуживания. Тем самым, если, например, $N = \{1, 2, 3, 4\}$, $1 \to 2 \to 3$ и $1 \to 4$, то оптимальный порядок обслуживания требований 2, 3, 4 может зависеть от длительности обслуживания требования 1. Действительно, положим $\varphi_1(x) = \varphi_2(x) = \varphi_3(x) = \varphi_4(x) = \varphi(x)$, где $\varphi(x) = \max(0, x - 3)$ при $x \leqslant 5$ и $\varphi(x) = 2\min(2, x - 4)$ при $x > 5$, $t_2 = 3$, $t_3 = 1$, $t_4 = 2$. Тогда при $t_1 = 1$ наименьший суммарный штраф соответствует допустимой последовательности (1, 2, 3, 4). При $t_1 = 2$ оптимальной последовательностью является (1, 4, 2, 3).

Пусть $\pi_1 = (\sigma^{(1)}, \sigma^{(2)}, \sigma^{(3)}, \sigma^{(4)})$, $\pi_2 = (\sigma^{(1)}, \sigma^{(3)}, \sigma^{(2)}, \sigma^{(4)})$. Из (6.1) и (6.2) следует, что $F(\pi_1) \leqslant F(\pi_2)$, если $\omega(\sigma^{(2)}) \geqslant \omega(\sigma^{(3)})$, и $F(\pi_1) < F(\pi_2)$, если $\omega(\sigma^{(2)}) > \omega(\sigma^{(3)})$, где

$$\omega(\sigma) = \sum_{i \in \{\sigma\}} a_i \Big/ \sum_{i \in \{\sigma\}} t_i \tag{6.3}$$

при минимизации $F_л(\pi)$ и

$$\omega(\sigma) = \Big[1 - \exp\Big(\alpha \sum_{i \in \{\sigma\}} t_i\Big)\Big] \Big/ \Big[F_э(\sigma) - \sum_{i \in \{\sigma\}} b_i\Big] \tag{6.4}$$

при минимизации $F_э(\pi)$.

Таким образом, если π_1 — искомая оптимальная последовательность, а π_2 — последовательность, допустимая относительно заданного на N порядка, то $\omega(\sigma^{(2)}) \geqslant \omega(\sigma^{(3)})$.

6.2. Алгоритм построения оптимальной последовательности в рассматриваемом случае состоит из двух этапов: 1) вычисление приоритетов обслуживания требований и 2) построение допустимой последовательности в соответствии с вычисленными на предыдущем этапе приоритетами.

Описание алгоритма будем иллюстрировать примером построения оптимальной по критерию $F_э(\pi)$ последовательности обслуживания восьми требований, параметры a_k и t_k которых представлены в табл. 3.6.1, а редукция отношения порядка представлена графом, изображенным на рис. 3.6.1. Значения $b_k = 0$, $k = \overline{1, 8}$, $\alpha = 0{,}1$.

Таблица 3.6.1

k	1	2	3	4	5	6	7	8
t_k	3	4	2	8	7	10	5	12
a_k	2	1	1	0,2	6	6	4	0,1

Разобьем множество N на попарно непересекающиеся непустые подмножества $N_1, N_2, \ldots, N_p$. Здесь N_1 — множество всех максимальных элементов множества N; N_2 — множество всех максимальных элементов множества $N \setminus N_1$, N_3 — множество всех максимальных элементов множества $N \setminus \{N_1 \cup N_2\}$ и т. д.

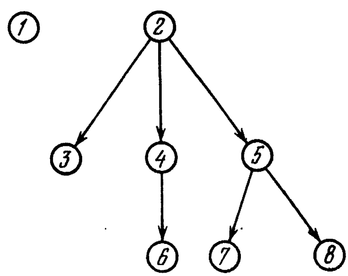

Рис. 3.6.1.

Если $N' \subset N$, то через $M(N')$ обозначим множество всех тех элементов из $N \setminus N'$, каждому из которых непосредственно предшествует хотя бы один элемент множества N'.

Вычисление приоритетов обслуживания требований проводится сначала для всех требований множества N_1, затем для всех требований множества N_2 и т. д. При этом каждому требованию k ставится в соответствие его приоритет $\overline{\omega}(k)$ и некоторая допустимая последовательность $\sigma(k)$ элементов из N. Множество элементов последовательности $\sigma(k)$ будем по-прежнему обозначать через $\{\sigma(k)\}$.

При вычислении приоритетов требований множества N_1 полагаем $\overline{\omega}(k) = \omega(k)$ и $\sigma(k) = (k)$ для всех $k \in N_1$. Значения $\omega(k)$ для $F_л(\pi)$ и $F_э(\pi)$ вычисляются согласно теореме 2.3 пп. а) и б) соответственно (см. § 2 гл. 3).

В условиях рассматриваемого примера $N_1 = \{1, 3, 6, 7, 8\}$, $N_2 = \{4, 5\}$, $N_3 = \{2\}$. Используя указанную теорему, вычисляем $\overline{\omega}(k) = \omega(k)$ для всех $k \in N_1$. Получаем (с точностью до второго знака после запятой) $\overline{\omega}(1) = -0{,}13$, $\overline{\omega}(3) = -0{,}18$, $\overline{\omega}(6) = -0{,}10$, $\overline{\omega}(7) = -0{,}098$ и $\overline{\omega}(8) = -6{,}99$. Полагаем $\sigma(k) = (k)$ для всех $k \in N_1$.

Пусть известны $\overline{\omega}(k)$ и $\sigma(k)$ для всех $k \in N_1 \cup N_2 \cup \ldots \cup N_{l-1}$. Приоритет обслуживания требования $k \in N_l$, $l \leqslant p$, вычисляется следующим образом. На первом шаге полагаем $\sigma_1 = (k)$. Обозначим через i_1 то из требований множества $M(\{\sigma_1\})$, которому соответствует наибольшее значение приоритета $\overline{\omega}(i_1)$. Если $\omega(k) \geqslant \overline{\omega}(i_1)$, то полагаем $\overline{\omega}(k) = \omega(k)$ и $\sigma(k) = \sigma_1 = (k)$.

Если $\omega(k) < \overline{\omega}(i_1)$, то переходим ко второму шагу. Полагаем $\sigma_2 = (\sigma_1, \sigma(i_1))$ и вычисляем значение $\widetilde{\omega}_2 = \omega(\sigma_2)$ по формулам (6.3), (6.4). Обозначим через i_2 то из требований множества $M(\{\sigma_2\})$, которому соответствует наибольшее значение приоритета $\overline{\omega}(i_2)$. Если $\widetilde{\omega}_2 \geqslant \overline{\omega}(i_2)$, то полагаем $\overline{\omega}(k) = \widetilde{\omega}_2$ и $\sigma(k) = \sigma_2$. Если $\widetilde{\omega}_2 < \overline{\omega}(i_2)$, то переходим к следующему шагу.

В общем случае на шаге $r = 3, 4, \ldots$ полагаем $\sigma_r = (\sigma_{r-1}, \sigma(i_{r-1}))$ и вычисляем значение $\widetilde{\omega}_r = \omega(\sigma_r)$. Если $M(\{\sigma_r\}) \neq \phi$, то обозначим через i_r то из требований множества $M(\{\sigma_r\})$, которому соответствует наибольшее значение приоритета $\overline{\omega}(i_r)$. Если $M(\{\sigma_r\}) = \phi$ или $\widetilde{\omega}_r \geqslant \overline{\omega}(i_r)$, то вычисления закончены. В этом случае $\overline{\omega}(k) = \widetilde{\omega}_r$ и $\sigma(k) = \sigma_r$. Если $\widetilde{\omega}_r < \overline{\omega}(i_r)$, то переходим к следующему шагу и т. д., пока не будет вычислено значение $\overline{\omega}(k)$. Требованию k окажется приписана последовательность $\sigma(k)$.

Процесс вычисления приоритетов заканчивается рассмотрением всех требований множества N.

В условиях рассматриваемого примера для требования 4 на первом шаге полагаем $\sigma_1 = (4)$ и вычисляем $\omega(4) = -2{,}75$. Множество $M(\{\sigma_1\}) = \{6\}$. Поскольку $\overline{\omega}(6) = -0{,}10 > \omega(4)$, то полагаем $\sigma_2 = (\sigma_1, \sigma(6)) = (4, 6)$ и по формуле (6.4) вычисляем $\widetilde{\omega}_2 = \omega(\sigma_2) = -0{,}14$. Поскольку $M(\{\sigma_2\}) = \phi$, то полагаем $\overline{\omega}(4) = \widetilde{\omega}_2 = -0{,}14$ и $\sigma(4) = \sigma_2 = (4{,}6)$. Для требования 5 на первом шаге полагаем $\sigma_1 = (5)$ и вычисляем $\omega(5) = -0{,}08$. Множество $M(\{\sigma_1\}) = \{7{,}8\}$. Поскольку $\omega(5) > \overline{\omega}(7) > \overline{\omega}(8)$, то полагаем $\overline{\omega}(5) = \omega(5) = -0{,}08$ и $\sigma(5) = (5)$.

Для требования 2 на первом шаге полагаем $\sigma_1 = (2)$ и вычисляем $\omega(2) = -0{,}33$. Множество $M(\{\sigma_1\}) = \{3, 4, 5\}$. Так как $\overline{\omega}(5) > \overline{\omega}(4) > \overline{\omega}(3)$, то обозначим через i_1 требование 5. Поскольку $\overline{\omega}(i_1) > \omega(2)$, то переходим ко второму шагу. Полагаем $\sigma_2 = (\sigma_1, \sigma(i_1)) =$

$= (2,5)$ и вычисляем $\tilde{\omega}_2 = \omega(\sigma_2) = -0{,}10$. Множество $M(\{\sigma_2\}) = \{3, 4, 7, 8\}$. Поскольку $\bar{\omega}(7) > \bar{\omega}(4) > \bar{\omega}(3) > \bar{\omega}(8)$, то обозначим через i_2 требование 7. Так как $\bar{\omega}(i_2) = -0{,}098 > \tilde{\omega}_2$, то переходим к третьему шагу. Полагаем $\sigma_3 = (\sigma_2, \sigma(i_2)) = (2, 5, 7)$ и вычисляем $\tilde{\omega}_3 = \omega(\sigma_3) = -0{,}10$. Поскольку $M(\{\sigma_3\}) = \{3, 4, 8\}$ и $\tilde{\omega}_3 > \bar{\omega}(4) > \bar{\omega}(3) > \bar{\omega}(8)$, то вычисления закончены. Полагаем $\bar{\omega}(2) = \tilde{\omega}_3 = -0{,}10$ и $\sigma(2) = \sigma_3 = (2, 5, 7)$.

Построение допустимой последовательности обслуживания требований в соответствии с вычисленными на предыдущем этапе приоритетами проводится следующим образом. Полагаем $N^{(1)} = N$, $N^{(l)} = N^{(l-1)} \setminus \{\sigma(k_{l-1})\}$, где k_{l-1} — требование (или одно из требований) с наибольшим приоритетом среди всех минимальных элементов множества $N^{(l-1)}$, $l = 2, 3, \ldots$ Пусть $N^{(\nu)} \neq \phi$ и $N^{(\nu+1)} = \phi$. Построим последовательность $\pi^* = (\sigma(k_1), \sigma(k_2), \ldots, \sigma(k_\nu))$.

В рассматриваемом примере среди минимальных элементов 1 и 2 множества $N^{(1)} = N = \{1, 2, 3, 4, 5, 6, 7, 8\}$ элемент 2 имеет наибольший приоритет $\bar{\omega}(2) = -0{,}10$. Полагаем $N^{(2)} = N^{(1)} \setminus \{\sigma(2)\} = \{1, 3, 4, 6, 8\}$. Среди минимальных элементов 1, 3, 4, 8 множества $N^{(2)}$ элемент 1 имеет наибольший приоритет. Полагаем $N^{(3)} = N^{(2)} \setminus \{\sigma(1)\}$. Продолжая эти рассуждения, приходим к последовательности $\pi^* = (\sigma(2), \sigma(1), \sigma(4), \sigma(3), \sigma(8)) = (2, 5, 7, 1, 4, 6, 3, 8)$.

6.3. Покажем, что построенная последовательность $\pi^* = (\sigma(k_1), \sigma(k_2), \ldots, \sigma(k_\nu))$ является *допустимой* относительно заданного на множестве N порядка $\rightarrow$.

Запишем эту последовательность в виде $\pi^* = (i_1^*, i_2^*, \ldots, i_p^*, \ldots, i_q^*, \ldots, i_n^*)$. Предположим, что $i_q^* \rightarrow i_p^*$, $p < q$, $i_p^* \in \{\sigma(k_s)\}$ и $i_q^* \in \{\sigma(k_t)\}$. По построению частичные последовательности $\sigma(k_1), \sigma(k_2), \ldots, \sigma(k_\nu)$ являются допустимыми и, следовательно, $s < t$, $k_s \rightarrow i_p^*$ или $k_s = i_p^*$, $k_t \rightarrow i_q^*$ или $k_t = i_q^*$.

Если $k_s = i_p^*$, то $k_t \rightarrow k_s$, что противоречит принятому способу выбора элементов k_l. Если $k_s \neq i_p^*$, то $k_s \rightarrow i_p^*$ и (по предположению) $i_q^* \rightarrow i_p^*$. Элементы k_s и i_q^* не могут быть независимыми, так как это противоречило бы условию представимости отношения строгого порядка прадеревьями с корнями. Если $k_s \to i_q^*$, то $k_s \to i_q^* \to i_p^*$ и, следовательно, $i_q^* \in \{\sigma(k_s)\}$, т. е. $s = t$. Получаем противоречие. Таким образом, всякий раз, когда $i_\gamma^* \to i_\delta^*$, имеем $\gamma < \delta$, и, следовательно, π^* — допустимая последовательность.

Доказательство *оптимальности* последовательности π^* проведем по индукции относительно числа элементов множества N.

При $|N| \leqslant 2$ указанная последовательность, очевидно, оптимальна.

Пусть при $|N| = n - 1$ последовательность, построенная в соответствии с приведенным алгоритмом, является оптимальной, а π' — оптимальная последовательность обслуживания $|N| = n$ требований.

Если в последовательности π' первым обслуживается требование r, то на основании соотношений (6.1), (6.2) и предположения индукции последовательность π'', в которой первым элементом является требование r, а остальные $n - 1$ требований упорядочены согласно приведенному алгоритму, также является оптимальной.

Обозначим через p тот из минимальных элементов множества N ($|N| = n$), которому соответствует наибольший приоритет $\overline{\omega}(p)$. Если $\overline{\omega}(p) = \overline{\omega}(r)$, то, выбирая в качестве p требование r, приходим к заключению, что $F(\pi^*) = F(\pi'')$.

Пусть $\overline{\omega}(p) > \overline{\omega}(r)$. В общем случае последовательность π'' можно записать в виде $\pi'' = (r, \sigma(l_1), \ldots, \sigma(l_\mu), \sigma(p), \sigma(s), \ldots)$, где $l_1, \ldots, l_\mu$ — некоторый набор элементов из N, для которых r является предшествующим элементом в N. В частности, может быть $\{\sigma(l_i)\} = \phi$, $i = \overline{1, \mu}$.

Построим последовательность $\tilde{\pi} = (\sigma(p), r, \sigma(l_1), \ldots, \sigma(l_\mu), \sigma(s), \ldots)$. Эта последовательность является допустимой. Согласно описанной процедуре вычисления приоритетов и построения последовательности π'' имеем $\overline{\omega}(r) \geqslant \omega(r, \sigma(l_1), \ldots, \sigma(l_\mu))$ и, следовательно, $\omega(\sigma(p)) = \overline{\omega}(p) > \overline{\omega}(r) \geqslant \omega(r, \sigma(l_1), \ldots, \sigma(l_\mu))$. Приходим к заключению, что $F(\tilde{\pi}) < F(\pi'')$ и π'' не является оптимальной последовательностью.

Таким образом, существует оптимальная последовательность, в которой первыми обслуживаются требования множества $\{\sigma(p)\}$ в последовательности $\sigma(p)$. На основании соотношений (6.1) и (6.2) и предположения индукции последовательность π^*, построенная с использованием приведенного алгоритма, является оптимальной.

6.4. Рассмотрим метод определения оптимальной последовательности обслуживания требований, удовлетворяющих порядку, противоположному по отношению к порядку $\rightarrow$, заданному на N.

Пусть на N заданы отношения строгого порядка α и β (не обязательно древовидные), причем из условия $i\alpha j$ следует $j\beta i$, $i, j \in N$, и наоборот. Пусть $\vec{\pi} = (i_1, i_2, \dots \dots, i_{n-1}, i_n)$ — допустимая последовательность обслуживания требований частично упорядоченного множества (N, α) и $\overleftarrow{\pi} = (i_n, i_{n-1}, \dots, i_2, i_1)$. Тогда $\overleftarrow{\pi}$ является допустимой последовательностью обслуживания требований частично упорядоченного множества (N, β).

Обозначим через $F'_{\text{л}}(\pi)$ и $F'_{\text{э}}(\pi)$ суммарный штраф за обслуживание требований в последовательности π при условии, что все функции штрафа $\varphi_k(x) = a'_k x + b'_k$ или $\varphi_k(x) = a'_k \exp(\alpha' x) + b'_k$, $k = \overline{1, n}$, соответственно, а время обслуживания требования k равно t'_k.

Полагая $a'_k = t_k, t'_k = a_k$ и $b'_k = b_k$, имеем $F_{\text{л}}(\vec{\pi}) = F'_{\text{л}}(\overleftarrow{\pi})$.

Аналогично, полагая $a'_k = a_k \exp \alpha \left(\sum\limits_{i \in N} t_i + t_k \right)$, $\alpha' = -\alpha$, $t'_k = t_k$ и $b'_k = b_k$, имеем $F_{\text{э}}(\vec{\pi}) = F'_{\text{э}}(\overleftarrow{\pi})$.

Тем самым описанный алгоритм построения оптимальной последовательности может быть использован и в случае, когда редукция отношения строгого порядка, заданного на N, представима в виде графа, все компоненты связности которого получаются из прадеревьев с корнями заменой ориентации всех дуг на противоположные.

§ 7. Общий случай

В этом параграфе мы рассмотрим методы построения оптимальных последовательностей обслуживания требований в предположении, что функции штрафа $\varphi_k(x)$ — произвольные неубывающие функции. Множество N требований, вообще говоря, частично упорядочено. Поиск оптимальных расписаний ограничен классом перестановочных расписаний. Критерии оптимальности — суммарный или максимальный штраф за обслуживание.

7.1. Пусть все требования поступают в очередь на обслуживание в момент времени $d = 0$.

Рассмотрим задачу построения допустимой относительно заданного на N отношения строгого порядка $\rightarrow$ последовательности π^*, которой соответствует наименьшее значение *максимального штрафа* за обслуживание

$$F_{\max}(\pi) = \max_{1 \leqslant k \leqslant n} \left\{ \varphi_{i_k} \left(\sum_{j=1}^{k} t_{i_j} \right) \right\}. \tag{7.1}$$

Здесь $\pi = (i_1, i_2, \ldots, i_n)$ — допустимая последовательность, i_k — требование, обслуживаемое k-м по порядку.

Пусть $\bar{N}$ — множество всех максимальных элементов N. Пусть $j_n \in \bar{N}$ и

$$\varphi_{j_n} \left(\sum_{k \in N} t_k \right) = \min_{l \in \bar{N}} \left\{ \varphi_l \left(\sum_{k \in N} t_k \right) \right\}. \tag{7.2}$$

Аналогично, пусть $N_1 = N \setminus \{j_n\}$, $\bar{N}_1$ — множество всех максимальных элементов N_1, $j_{n-1} \in \bar{N}_1$ и

$$\varphi_{j_{n-1}} \left(\sum_{k \in N_1} t_k \right) = \min_{l \in \bar{N}_1} \left\{ \varphi_l \left(\sum_{k \in N_1} t_k \right) \right\}. \tag{7.3}$$

Продолжая этот процесс, получаем в результате последовательность $\hat{\pi} = (j_1, j_2, \ldots, j_n)$. Покажем, что эта последовательность является искомой оптимальной последовательностью.

По построению последовательность $\hat{\pi}$ допустима относительно заданного на N строгого порядка $\rightarrow$.

Пусть $\pi^* = (j_1^*, j_2^*, \ldots, j_n^*)$, $j_n^* = j_n$, $j_{n-1}^* = j_{n-1}$, $\ldots$ $\ldots$, $j_{\nu+1}^* = j_{\nu+1}$ и $j_\nu^* \neq j_\nu$. Тогда среди элементов $j_1^*, j_2^*, \ldots, j_{\nu-1}^*$ последовательности π^* содержится элемент $j_\mu^* = j_\nu$. Рассмотрим последовательность $\tilde{\pi} = (j_1^*, \ldots, j_{\mu-1}^*, j_{\mu+1}^*, \ldots, j_\nu^*, j_\mu^*, j_{\nu+1}^*, \ldots, j_n^*)$. Эта последовательность является допустимой, поскольку допустима последовательность $\hat{\pi}$.

Покажем, что $F_{\max}(\tilde{\pi}) = F_{\max}(\pi^*)$. Будем через $\bar{t}_k^*$ и $\tilde{t}_k$ обозначать времена завершения обслуживания требования с индексом k, если все требования обслуживаются в последовательности π^* и $\tilde{\pi}$ соответственно. Имеем $\tilde{t}_{j_k^*} = \bar{t}_{j_k^*}^*$ для $k = \overline{1, \mu - 1}$ и $k = \overline{\nu + 1, n}$, $\tilde{t}_{j_k^*} < \bar{t}_{j_k^*}^*$

для $k = \overline{\mu + 1, \nu}$ и $\tilde{\bar{t}}_{j_\mu^*} > \bar{t}^*_{j_\mu^*}$. Поскольку функции штрафа $\varphi_k(x)$, $k = \overline{1, n}$, неубывающие и $\varphi_{j_\mu^*}(t) \leqslant \varphi_{j_\nu^*}(t)$, где $t = \bar{t}^*_{j_\nu^*} = \tilde{\bar{t}}_{j_\mu^*}$, то $F_{\max}(\tilde{\pi}) \leqslant F_{\max}(\pi^*)$. С другой стороны, по определению $F_{\max}(\tilde{\pi}) \geqslant F_{\max}(\pi^*)$. Следовательно, $F_{\max}(\tilde{\pi}) = F_{\max}(\pi^*)$.

Повторяя эти рассуждения конечное число раз, приходим к заключению, что $\hat{\pi}$ — оптимальная последовательность обслуживания требований.

Таким образом, независимо от конкретного вида функции штрафа, решение задачи минимизации наибольшего штрафа требует вычисления не более $n + (n - 1) + \ldots + 1 = n(n + 1)/2$ значений функций штрафа и проведения не более $n(n - 1)/2$ их сравнений.

7.2. Примеры. 1. Рассмотрим задачу размещения n информационных массивов на магнитной ленте в случае, когда поиск этих массивов осуществляется от нулевого адреса. Массив $k = \overline{1, n}$ имеет длину l_k. Массивы размещаются на ленте последовательно, начиная с левого ее конца. Исходное положение считывающей головки — левый конец ленты. При поиске массива осуществляется перемещение ленты относительно считывающей головки слева направо со скоростью $v_п$, считывание массива осуществляется слева направо со скоростью $v_с$, возврат в исходное положение — со скоростью $v_в$. Вероятность обращения к массиву k равна p_k.

Если массивы расположены в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, то максимальное среднее время обращения равно

$$T_{\max}(\pi) = \max_{1 \leqslant k \leqslant n} p_{i_k} \left[\sum_{j < k} l_{i_j} / v_п + l_{i_k} / v_с + \sum_{j \leqslant k} l_{i_j} / v_в \right]. \quad (7.4)$$

Для построения последовательности π^*, которой соответствует наименьшее значение $T_{\max}(\pi)$, может быть непосредственно использован описанный алгоритм, если положить

$$t_k = l_k, \quad \varphi_k(x) = a_k x + b_k,$$

где

$$a_k = (1/v_п + 1/v_в)\, p_k, \quad b_k = (1/v_с - 1/v_п)\, p_k l_k.$$

Таблица 3.7.1

k	1	2	3	4	5
t_k	2	3	1	4	4
$\varphi_k(x)$	$2x+1$	$2^{x-10}-3$	$\max(x-3,2)$	x^2-5	$x-3$

2. Найдем оптимальную по критерию $F_{\max}(\pi)$ последовательность обслуживания пяти требований, времена обслуживания t_k и функции штрафа $\varphi_k(x)$ которых приведены в табл. 3.7.1. Значения $d_k = d = 0$.

Таблица 3.7.2

$x \backslash k$	1	2	3	4	5
14		13		191	11
10		−2		95	
7	15			44	
5				20	
1			2		

Отношение строгого частичного порядка, заданное на множестве требований $N = \{1, 2, 3, 4, 5\}$, изображено на рис. 3.7.1.

Результаты последовательного вычисления значений $\varphi_k(x)$ приведены в табл. 3.7.2.

Последовательность $\hat{\pi} = (3, 4, 1, 2, 5)$ является искомой, $F_{\max}(\hat{\pi}) = 20$.

7.3. При минимизации *суммарного штрафа* $F_\Sigma(\pi)$ за обслуживание требований в последовательности π вычислительная схема существенно усложняется.

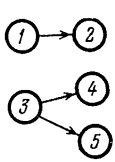

Рис. 3.7.1.

Пусть, как и в предыдущем пункте, все требования поступают на обслуживание в момент времени $d = 0$, множество требований, вообще говоря, частично упорядочено отношением $\rightarrow$.

Задача минимизации $F_\Sigma(\pi)$ относится к классу задач, рассмотренных в § 5 гл. 2. Действительно, если $\pi = (i_1, i_2, \ldots, i_n)$, то, обозначая через $F(\pi, k)$ наименьший суммарный штраф за обслуживание первых k требований в последовательности π, имеем

$$F(\pi, k) = F(\pi, k-1) + \varphi_{i_k}\left(\sum_{j=1}^{k} t_{i_j}\right), \qquad (7.5)$$

где $F(\pi, 0) = 0$.

Поскольку в данном случае $s = 1$, $t = 0$, то соотношение (5.11) гл. 2 можно записать в виде

$$\tilde{F}(Q) = \min_{k \in \bar{Q}} \left\{ \tilde{F}(Q \setminus k) + \varphi_k \left(\sum_{l \in Q} t_l \right) \right\}. \tag{7.6}$$

Здесь множество $Q \subseteq N$ и удовлетворяет условию: не существует $k_1 \in Q$ и $k_2 \in N \setminus Q$ таких, что $k_2 \to k_1$; $\bar{Q}$ — множество максимальных относительно $\to$ элементов Q; $\tilde{F}(\phi) = 0$.

Для построения оптимальной последовательности достаточно вычислить значения $\tilde{F}(Q)$ для всех допустимых в указанном выше смысле множеств $Q \subseteq N$. При этом сначала вычисляются значения $\tilde{F}(Q)$ для всех Q с числом элементов $|Q| = 1$, затем для всех Q с числом элементов $|Q| = 2$ и т. д., наконец, для $Q = N$. Вычисляя значение $\tilde{F}(Q)$, необходимо запоминать значения $k \in \bar{Q}$, при которых в (7.6) достигается минимум. Если при $Q = N$ значение $k = i_n$, при $Q = N \setminus \{i_n\}$ значение $k = i_{n-1}$ и т. д., то $(i_1, i_2, \ldots, i_n)$ — искомая последовательность.

Объем вычислений определяется структурой отношения порядка, заданного на N. Если N — неупорядоченное множество, то при вычислении $\tilde{F}(Q)$ необходимо провести $|Q| - 1$ сравнений, используя значения этой функции, вычисленные на предыдущем шаге. Значения $\tilde{F}(Q)$ вычисляются для всех $2^n - 1$ непустых подмножеств множества N. Для частично упорядоченного множества эти величины являются верхними оценками соответственно числа определяемых значений функции $\tilde{F}(Q)$ и числа проводимых при этом сравнений.

Следует отметить, что при практической реализации этой вычислительной схемы существенные затруднения возникают из-за необходимости запоминания значений функции $\tilde{F}(Q)$, вычисленных на предыдущем шаге, в качестве исходной информации для вычислений ее значений на следующем шаге. Поэтому в ряде случаев предпочтительно проводить дополнительный объем дублирующих вычислений, сокращая при этом объем запоминаемой промежуточной информации.

7.4. Пример. Применим этот метод построения оптимальной по критерию $F_\Sigma(\pi)$ последовательности обслуживания требований множества $N = \{1, 2, 3, 4, 5, 6\}$, поступающих в очередь на обслуживание в момент времени $d = 0$. Граф отношения строгого порядка, заданного на N, представлен на рис. 3.7.2. Значения t_k и функции штрафа $\varphi_k(x)$ приведены в табл. 3.7.3.

Элементы множества Q, значения $\bar{F}(Q)$ и запоминаемые значения k приведены в табл. 3.7.4.

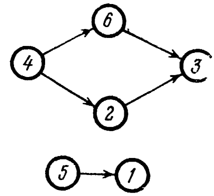

Рис. 3.7.2.

Значение i_6 равно 1 или 3, значение i_5 равно 3 или 1 соответственно. Значения $i_4 = 2$, $i_3 = 6$, $i_2 = 4$, $i_1 = 5$. Все эти значения выделены в таблице.

Оптимальные последовательности $\pi_1^0 = (5, 4, 6, 2, 3, 1)$ и $\pi_2^0 = (5, 4, 6, 2, 1, 3)$. Суммарный штраф за обслуживание равен 1.

Описанные в предыдущих пунктах методы построения оптимальных последовательностей могут быть обобщены и на случай различных значений $d_k \geqslant 0$, $k = \overline{1, n}$.

7.5. В практическом отношении весьма эффективными оказались схемы последовательного конструирования решений с использованием оценочных функций и правил доминирования. Применимость этих схем, как правило, не зависит от значений d_k.

Таблица 3.7.3

k	1	2	3
t_k	2	3	4
$\varphi_k(x)$	$x-5$	$\max[(x-6)^3, 0]$	$2\max(x-10, 0)$
k	4	5	6
t_k	2	1	1
$\varphi_k(x)$	2^x-3x	$x+1$	x^2-7x+1

Множество всех допустимых последовательностей обслуживания требований по некоторому признаку разбивается на подмножества. Вычисляются нижние границы значений оптимизируемой функции на каждом из этих подмножеств. Если в процессе последовательного разбиения исходного множества значение оптимизируемой функции, соответствующее некоторой последовательности обслуживания требований, оказывается не больше нижних

Таблица 3.7.4

$\lvert Q\rvert$	Q	$\tilde{F}(Q)$	k	$\lvert Q\rvert$	Q	$\tilde{F}(Q)$	k
0	∅	0	—	4	{2, 3, 4, 6}	−13	3
1	{4}	−2	4		{2, 4, 5, 6}	−9	2*
	{5}	2	5*		{1, 2, 4, 5}	4	1
2	{2, 4}	−2	2		{1, 4, 5, 6}	−9	1
	{4, 6}	−13	6	5	{2, 3, 4, 5, 6}	−7	3
	{4, 5}	1	4*		{1, 2, 4, 5, 6}	−5	1*
	{1, 5}	0	1	6	{1, 2, 3, 4, 5, 6}	1	1,3*
3	{2, 4, 6}	−13	2				
	{2, 4, 5}	1	2				
	{4, 5, 6}	−10	6*				
	{1, 4, 5}	1	1				

границ значений этой функции на каждом из рассматриваемых на данном шаге подмножеств, то указанная последовательность, очевидно, является искомой.

В качестве признака, согласно которому множество последовательностей разбивается на подмножества, в рассматриваемом случае естественно выбрать частичную последовательность обслуживания первых $1 \leqslant l \leqslant n$ требований. При этом две последовательности принадлежат одному и тому же подмножеству, если первые l элементов у них одинаковы.

§ 8. Библиографическая справка

Начало систематическому использованию интервалов очередности в задачах теории расписаний положили работы В. В. Шкурбы [204] и Г. Эммонса [272]. Отмеченные в п. 2.3 пары функций рассматривались В. Е. Смитом [409], Дж. Р. Джексоном [309], М. Роткопфом [387], В. С. Танаевым [176, 177]. Теорема 2.1 доказана Г. К. Кладовым и Э. М. Лифшицем [79], теорема 2.2 — А. Д. Глубочанским [52]. Доказательство теоремы 2.3 содержится в [176]. Примеры п. 2.6 рассмотрены соответственно в [386, 409, 103].

Теорема 3.1 доказана Р. Мак-Нотоном [344]. Различные ее обобщения рассматриваются в [388, 171, 56]. Ступенчатые функции штрафа исследовались Е. Л. Лоулером и Дж. М. Муром [327], В. С. Гордоном и В. С. Танаевым [55]. Вопросы минимизации числа запаздывающих требований рассматривались в работах Дж. М. Мура [352], Л. Б. Штурма [413] и В. Л. Максвелл [339]. Обобщение этих результатов на случай, когда задано подмножество требований, каждое из которых должно быть обслужено в срок, рассмотрено Дж. Сиднеем [404].

Теорема 3.2 (в несколько иной интерпретации) доказана Г. Эммонсом [272]. Соответствующий алгоритм описан в [272, 411]. Сравнительный экспериментальный анализ известных методов решения этой задачи (при различных D_k) провели К. Бакер и Дж. Мартин [226]. Простые методы минимизации суммарного или максимального запаздывания в обслуживании требований (разрешены прерывания) предложены В. Хорном [304].

Первые попытки использовать методы линейного программирования для решения задачи коммивояжера были предприняты Дж. Данцигом, Д. Фалкерсоном и С. Джонсоном [258, 259]. Условия связности (4.5) приведены в [174]. Условия (4.6), (4.7) сформулированы А. Е. Залесским [71], (4.6), (4.8), (4.9) — В. И. Мудровым [125], а (4.10) — С. Миллером, А. Таккером и Р. Землиным [349]. Формулировка задачи коммивояжера в виде (4.11)—(4.13) приведена в работе Дж. Мортона и А. Лэнда [353], в виде (4.14)—(4.16) — в работе Б. Кортэ [319]. Использованию методов линейного программирования для решения задачи коммивояжера посвящены также работы [347, 357, 358, 427].

Решение этой задачи методом динамического программирования осуществлено Р. Беллманом [235]. Некоторое обобщение описано в [301]. Своеобразный подход предложен В. К. Коробковым и Р. Е. Кричевским [84].

Метод ветвей и границ для решения задачи коммивояжера был впервые предложен Дж. Литтлом, К. Мурти, Д. Суини и К. Кэрел [332]. Дальнейшее развитие этот метод получил в работах [216, 297, 317, 318, 372, 424], в основном, в части уточнения оценок.

Описанию методов направленного перебора посвящены также работы [32, 197, 377, 397]. В [396] приведен метод определения всех оптимальных решений задачи коммивояжера.

Подробное описание нескольких локальных алгоритмов содержится в работах Дж. Круса [256]. С. Рэйтера и К. Шермана [384], С. М. Робертса и Б. Флоренса [385], С. Мака [334—336], Н. Христофайдса и С. Эйлона [251], Л. С. Гельтмана и С. Я. Гусмана [44].

Для решения большеразмерных практических задач предложены разнообразные приближенные методы, основанные на различных эвристических соображениях, В. А. Леонтьевым [98], Е. И. Радевым [145], М. М. Телемтаевым [179], Р. Л. Каргом и Дж. Л. Томпсоном [315] и другими авторами [9, 105, 151, 277, 289, 330, 371, 414]. В работе М. Вебба [426] приведены результаты экспериментов по решению задач коммивояжера с числом городов до 2,5 тысячи. Декомпозиционный подход предложен Т. Ху [305].

Задача коммивояжера допускает сравнительно простое решение в случае, когда $l_{ij} = \max(0, s_j - r_i)$, где r_i и s_j $(i, j = \overline{1, n})$ — неотрицательные числа [285]. Практически важное обобщение этого результата содержится в [283, 284]. В [42] рассмотрен случай, когда $l_{ij} = \min(r_i, s_j)$.

Решению задачи коммивояжера на узкие места посвящены работы [43, 283, 402].

Обобщение на случай нескольких коммивояжеров рассмотрено в работах [66, 84, 249].

Задача коммивояжера в стохастическом варианте изучалась в [261], с подвижными клиентами ($l_{ij} = l_{ij}(t)$) — в [159]. Иные обобщения, связанные, в частности, с определенным группированием «городов», неоднократным их «посещением», рассмотрены в работах [87, 88, 91, 172, 301, 390].

Взаимосвязь задачи коммивояжера с задачей о назначениях изучается в [238], с задачей о деревьях-остовах — в [299]. Асимптотический подход к решению этих задач предложен в [134].

Среди обзорных работ следует отметить статьи Дж. Немхаузера и М. Беллмора [370], Е. Габовича [41].

Задача минимизации функции (5.1) рассматривалась С. Е. Элмаграби [269], А. В. Никитиным [127], функции (5.6) — В. Гаппом, П. Манкекаром и Л. Миттэном [276]. Вопросы оптимального размещения информационных массивов на магнитных лентах рассматривались В. Н. Бурковым и В. Б. Соколовым [30].

В основу параграфа 6 положены работы [303, 58]. Вопросы упорядочения «цепочек» требований рассмотрены в [254, 211].

Применению метода динамического программирования для решения задачи минимизации суммарного штрафа посвящены работы [36, 108, 143, 247, 298], методов последовательного анализа вариантов — работы [78, 176, 246, 262, 270, 360, 403]. Алгоритм минимизации максимального штрафа разработан Э. М. Лифшицем [104] и Е. Л. Лоулером [325].

Приближенные методы для решения различных задач одностадийного обслуживания описаны в работах [267, 278, 378, 394, 395]. Применению методов линейного программирования посвящены работы [288, 326, 368].

Задача построения последовательности обслуживания, минимизирующей суммарные переналадки оборудования, рассматривалась Ю. А. Заком [67, 68, 70] и С. Р. Глассэем [286].

Попытки учесть вероятностный характер величин d_k и t_k предприняты в работах Л. И. Фейгина [188], Б. П. Банержи [230], Д. В. Файфа [273], С. Усна [425], Р. Блау [239].

Связь рассматриваемых задач с задачами распределения ресурсов, минимизации производственных затрат, затрат на хранение и т. п. устанавливается в [276, 331].

Несколько более специальные вопросы, связанные с планированием работы одного обслуживающего прибора, рассмотрены в [20, 147, 296, 301, 391].

ГЛАВА 4

ПАРАЛЛЕЛЬНЫЕ ПРИБОРЫ

Конечный поток требований, поступающих на обслуживание в заданные моменты времени, обслуживается несколькими идентичными приборами. В каждый момент времени отдельный прибор обслуживает не более одного требования. Каждое требование может обслуживаться любым прибором. Известны времена обслуживания требований. Необходимо так организовать процесс обслуживания, чтобы он был в некотором смысле наилучшим.

В § 1 устанавливаются достаточные условия существования оптимальных расписаний, допускающих прерывания процесса обслуживания только в моменты времени поступления требований. В § 2 приводятся необходимые и достаточные условия существования расписаний, при которых все требования могут быть обслужены в заданные сроки. В § 3 рассматриваются две частные задачи оптимального обслуживания требований параллельными идентичными приборами. В первой задаче необходимо минимизировать суммарное время запаздывания в обслуживании требований. Во второй — минимизировать общее время обслуживания частично упорядоченного множества требований двумя приборами.

§ 1. Прерывания

Рассмотрим систему обслуживания, состоящую из M параллельных идентичных приборов. На обслуживание поступает конечный поток n требований, каждое из которых $k = \overline{1, n}$ поступает в момент времени $d_k \geqslant 0$ и для своего обслуживания требует $t_k > 0$ единиц времени.

1.1. *Расписанием* $s = s(t) = \{s_1(t), \ldots, s_M(t)\}$ называется совокупность M кусочно-постоянных, непрерывных слева функций $s_L = s_L(t)$, $L = \overline{1, M}$, каждая из которых задана на промежутке $0 \leqslant t < \infty$ и принимает значения $0, 1, \ldots, n$ причем если $s_L(t') = k \neq 0$ при некотором

$t = t'$, то $s_H(t') \neq k$ для любых $1 \leqslant L \neq H \leqslant M$. Выражение $s_L(t') = k \neq 0$ означает, что в момент времени $t = t'$ прибор L обслуживает требование k. Если $s_L(t') = 0$, то в момент времени $t = t'$ прибор L свободен.

Расписание должно удовлетворять

— условию полноты: суммарная длина промежутков, на которых функции $s_L(t)$, $L = \overline{1, M}$, принимают значение $k \neq 0$, равна t_k;

— условию готовности к обслуживанию: $s_L(t) \neq k$, $L = \overline{1, M}$, при $t \leqslant d_k$, $t = \overline{1, n}$.

Будем говорить, что расписание $s = s(t)$ допускает *прерывания* в процессе обслуживания требований, если существуют такие $1 \leqslant k \leqslant n$, $1 \leqslant L \neq H \leqslant M$ и $0 \leqslant \leqslant t' < t < t'' < \infty$, что выполняется хотя бы одно из условий: 1) $s_L(t') = s_L(t'') = k$, но $s_L(t) \neq k$ и 2) $s_L(t') = = s_H(t'') = k$.

Содержательно процесс обслуживания требований без прерываний удовлетворяет следующему условию. Каждое требование обслуживается только одним прибором. Если обслуживание некоторого требования k начинается в момент времени $\underline{t}_k$, то оно протекает непрерывно и завершается в момент времени $\underline{t}_k + t_k$.

Предполагается, что количество разрывов каждой из функций $s_L(t)$ конечно.

Каждому расписанию s соответствует вектор $\bar{t}(s) = = (\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_n)$ моментов завершения обслуживания требований. Здесь $\bar{t}_k$ — наибольшее значение t, при котором $\prod_{L=\overline{1,M}} (s_L(t) - k) = 0$.

Качество расписания s характеризуется значением действительной кусочно-непрерывной монотонно возрастающей функции $F(\bar{x}) = F(x_1, x_2, \ldots, x_n)$ при $\bar{x} = \bar{t}(s)$. Из двух расписаний лучшим считается то, которому соответствует меньшее значение $F(\bar{x})$. Расписание, которому соответствует наименьшее значение $F(\bar{x})$, называется *оптимальным расписанием*.

Наиболее распространенным является способ задания $F(\bar{x})$, при котором каждому требованию $k = \overline{1, n}$ относят некоторую кусочно-непрерывную неубывающую функцию — функцию штрафа $\varphi_k(x_k)$, выражающую в количественном отношении «штраф», который необходимо «заплатить», если обслуживание этого требования завершается в момент времени $x_k = \bar{t}_k$. Качество расписания s характеризуется суммарным или максимальным штрафом, который необходимо заплатить при обслуживании требований по расписанию s, т. е. $F(\bar{t}(s)) = \sum_{k=1}^{n} \varphi_k(\bar{t}_k)$ или $F(\bar{t}(s)) = \max_{1 \leqslant k \leqslant n} \{\varphi_k(\bar{t}_k)\}$.

1.2. В общем случае оптимальное расписание содержит прерывания процесса обслуживания требований. Приведем достаточные условия существования оптимальных расписаний, не содержащих прерываний в моменты времени, отличные от d_k, $k = \overline{1, n}$.

Теорема 1.1. *Если $M = 1$ и $F(\bar{x})$ — неубывающая функция, то для любого расписания s существует расписание s^* такое, что $F(\bar{t}(s^*)) \leqslant F(\bar{t}(s))$ и прерывания процесса обслуживания при расписании s^* имеют место не чаще, чем в моменты времени d_k, $k = \overline{1, n}$.*

Доказательство. Пусть $d_1 = \ldots = d_{i_1} < d_{i_1+1} = \ldots = d_{i_2} < \ldots < d_{i_r} = \ldots = d_n$. Обозначим через $\beta_1, \ldots, \beta_{r-1}, \beta_r$ временные промежутки $(d_{i_1}, d_{i_2}], \ldots$ $\ldots, (d_{i_{r-1}}, d_{i_r}], (d_{i_r}, \infty)$ соответственно.

1. Не нарушая общности рассуждений, можно рассматривать только такие расписания s, при которых обслуживание каждого требования k в любом промежутке β_j проходит непрерывно.

Действительно, если $s(t) = k$ на промежутках $(t_1', t_2']$ и $(t_1'', t_2'']$ $(t_2' < t_1'')$ промежутка β_j, но $s(t) \neq k$ при $t_2' < t \leqslant t_1''$, то расписание s можно преобразовать в расписание s', полагая $s'(t) = s(t)$ при $t \leqslant t_1'$ и $t > t_1''$, $s'(t) = s(t + (t_2' - t_1'))$ при $t_1' < t \leqslant t_1'' - (t_2' - t_1')$ и $s'(t) = k$ при $t_1'' - (t_2' - t_1') < t \leqslant t_1''$. Полученное расписание не уступает по качеству исходному и содержит меньшее число прерываний.

2. Если в некотором промежутке β_j, $j < r$, происходит прерывание обслуживания $m \geqslant 2$ требований (с последующим возобновлением их обслуживания в некоторых иных промежутках), то исходное расписание s можно преобразовать в неуступающее ему по качеству расписание s', при

котором в рассматриваемом промежутке происходит прерывание обслуживания не более $m-1$ требований.

Введем в рассмотрение следующую операцию преобразования расписаний.

Пусть $s(t)=a$ на $(t', t'']$ и $\hat{t} \geqslant t''$. Выберем на $(\hat{t}, \infty)$ один из промежутков $(t_1, t_2]$, на котором $s(t)=b$.

Если $t_2-t_1 \geqslant t''-t'$, то преобразуем s в s', полагая $s'(t)=b$ при $t'<t \leqslant t''$, $s'(t)=a$ при $t_1<t \leqslant t_1+(t''-t')$ и $s'(t)=s(t)$ при остальных значениях t.

Если $t_2-t_1<t''-t'$, то преобразуем s в $\tilde{s}$, полагая $\tilde{s}(t)=b$ при $t'<t \leqslant t'+(t_2-t_1)$, $\tilde{s}(t)=a$ при $t_1<t \leqslant t_2$ и $\tilde{s}(t)=s(t)$ при остальных значениях t. Рассматривая $\tilde{s}$ в качестве s и выбирая в качестве $(t', t'']$ промежуток $(t'+t_2-t_1, t'']$, повторяем указанные преобразования и т. д. до тех пор, пока либо $\tilde{s}(t)=b$ на $(t', t'']$, либо в промежутке $(\hat{t}, \infty)$ уже не содержатся промежутки $(t_1, t_2]$, на которых $\tilde{s}(t)=b$. Полученную в результате функцию $\tilde{s}(t)$ обозначим через $s'(t)$.

Если s' — расписание, то будем говорить, что оно получено из расписания s в результате применения операции $O(a, b, t', t'', \hat{t})$.

Пусть в промежутке β_j происходит прерывание обслуживания требований k и l, $s(t)=k$ при $t \in (t_1', t_2'] \in \beta_j$, $s(t)=l$ при $t \in (t_1'', t_2''] \in \beta_j$ и, для определенности, $\bar{t}_k<\bar{t}_l$.

Применим к расписанию s операцию $O(l, k, t_1'', t_2'', d_{i_{j+1}})$. В результате получаем расписание s', при котором либо требование k завершает обслуживание в промежутке β_j, либо в этом промежутке не обслуживается требование l.

В любом случае получаем расписание, не уступающее по качеству исходному и допускающее в промежутке β_j прерывания обслуживания не более $m-1$ требований.

3. Если в некотором промежутке β_j, $j<r$, происходит прерывание только одного требования k (обслуживание которого возобновляется в некотором последующем промежутке), то исходное расписание s можно преобразовать в новое расписание, при котором если в промежутке β_j и происходит прерывание, то оно происходит только в момент времени $d_{i_{j+1}}$.

Действительно, пусть $s(t) = k$ на промежутке $(t_1, t_2] \in \beta_j$, $s(t) = l$ на промежутке $(t', t'']$, $t_2 < d_{i_{j+1}}$, $t' < d_{i_{j+1}} \leqslant t''$.

а) Если $l = 0$ или (при $l \neq 0$) требование l завершает обслуживание в момент времени t'', то преобразуем расписание s в расписание s', полагая $s'(t) = s(t)$ при $t \leqslant t_1$ и $t > t''$, $s'(t) = s(t + (t_2 - t_1))$ при $t_1 < t \leqslant t'' - (t_2 - t_1)$ и $s'(t) = k$ при $t'' - (t_2 - t_1) < t \leqslant t''$.

б) Если обслуживание требования l прерывается в момент времени t'', то преобразуем расписание s в расписание s', используя операцию $O(l, k, t', d_{i_{j+1}}, t'')$, если $\bar{t}_k < \bar{t}_l$, или операцию $O(k, l, t_1, t_2, t'')$, если $\bar{t}_l < \bar{t}_k$.

В любом случае полученное расписание может быть преобразовано в искомое приемами пп. 1 и 2.

4. Поскольку число прерываний конечно, в результате конечного числа указанных преобразований исходного расписания s может быть получено расписание s^*, при котором прерывания обслуживания требований происходят не чаще, чем в моменты времени d_k. При этом промежутки β_j следует рассматривать последовательно слева направо. Каждое из вновь получаемых расписаний по качеству не уступает исходному. Теорема доказана.

Эта теорема указывает точную верхнюю границу минимального количества прерываний, имеющих место при оптимальном расписании для систем обслуживания с одним прибором.

С л е д с т в и е. *Если $d_k = 0$, $k = \overline{1, n}$, то при $M = 1$ и неубывающей функции $F(\bar{x})$ существует оптимальное расписание без прерываний процесса обслуживания.*

1.3. Перейдем к рассмотрению случая нескольких приборов.

Введем понятие квазивогнутой функции $F(\bar{x})$.

Обозначим через E множество всех n-мерных векторов, компонентами которых являются числа 0, 1 или —1. Будем предполагать, что функция $F(\bar{x})$ является монотонно возрастающей по каждому аргументу и, следовательно, по каждому направлению, определяемому вектором $\bar{e} \in E$ с неотрицательными компонентами. Потребуем, чтобы поведение функции $F(\bar{x})$ по направлению, определяемому вектором $\bar{e} \in E$, который содержит как положительные, так и отрицательные компоненты, удовлетворяло следующему условию. Если существует вектор $\bar{x} = \bar{x}_0$ и число $\alpha > 0$ такие, что $F(\bar{x}_0) < F(\bar{x}_0 + \alpha\bar{e})$, то $F(\bar{x}_0) \geqslant F(\bar{x}_0 - \beta\bar{e})$ для всех $\beta \geqslant 0$. Значения $\bar{x}_0$, α, β, вообще говоря, не могут быть произвольными, поскольку функция $F(\bar{x})$ определена для $\bar{x} > 0$, не содержащих более M одинаковых компонент. Функцию $F(\bar{x})$, обладающую указанными свойствами, будем называть *квазивогнутой функцией*. Класс квазивогнутых функций содержит, в частности, все вогнутые, не убывающие в положительном октанте функции.

Теорема 1.2. *Если $M \geqslant 1$, $d_k = 0$, $k = \overline{1, n}$ и $F(\bar{x})$ — квазивогнутая функция, то для любого расписания s существует расписание s^* без прерываний процесса обслуживания требований такое, что $F(\bar{t}(s^*)) \leqslant F(\bar{t}(s))$.*

Доказательство. 1. Введем некоторые операции преобразования расписаний.

Операция $O_1(Q, R, t')$. Будем говорить, что расписание $s' = \{s_1'(t), \ldots, s_M'(t)\}$ получено из расписания s в результате операции $O_1(Q, R, t')$, если $s_L'(t) = s_L(t)$, $0 \leqslant t < \infty$, для всех $L \neq Q, R$; $s_Q'(t) = s_Q(t)$ и $s_R'(t) = s_R(t)$, $0 \leqslant t \leqslant t'$; $s_Q'(t) = s_R(t)$ и $s_R'(t) = s_Q(t)$, $t' < t < \infty$. Эта операция позволяет в момент времени t' приборы Q и R поменять ролями. Поскольку приборы идентичны, то расписания s и s' одинаковы по качеству, т. е. $F(\bar{t}(s)) = F(\bar{t}(s'))$.

Операция $O_2(Q, t', \pm a)$. Эта операция позволяет увеличить или уменьшить величину простоя некоторого заранее выбранного прибора на заданную величину $a > 0$. В результате ее применения к расписанию s получаем совокупность функций $s' = \{s_1'(t), \ldots, s_M'(t)\}$, где $s_L'(t) = s_L(t)$, $0 \leqslant t < \infty$, для всех $L \neq Q$; $s_Q'(t) = s_Q(t)$, $0 \leqslant t \leqslant t'$, $s_Q'(t) = 0$, $t' < t \leqslant t' + a$, и $s_Q'(t) = s_Q(t - a)$, $t' + a < t < \infty$, если a имеет знак плюс; $s_Q'(t) = s_Q(t)$, $0 \leqslant t \leqslant t' - a$, и $s_Q'(t) = s_Q(t + a)$, $t' - a < t < \infty$, если a имеет знак минус. Если s' — расписание, то в силу неубывания функции $F(\bar{x})$ оно по качеству не уступает исходному расписанию s в случае, когда a имеет знак минус.

Операция $O_3(Q, R, t', t'', a)$. Пусть t', t'' и a — неотрицательные числа. Рассмотрим совокупность функций $s' = \{s'_1(t), \ldots, s'_M(t)\}$, где $s'_L(t) = s_L(t)$, $0 \leqslant t < \infty$, для всех $L \neq Q, R$; $s'_Q(t) = s_Q(t)$ при $0 \leqslant t \leqslant t' - a$ и $s'_Q(t) = s_Q(t + a)$ при $t' - a < t < \infty$; $s'_R(t) = s_R(t)$ при $0 \leqslant t \leqslant t'$; $s'_R(t) = s_Q(t + t' - t'' - a)$ при $t'' < t \leqslant \leqslant t'' + a$ и $s'_R(t) = s_R(t - a)$ при $t'' + a < t < \infty$. В случае, когда $Q = R$ и $t' < t''$, положим $s'_Q(t) = s_Q(t)$ при $0 \leqslant t \leqslant t' - a$ и $t > t'' + a$, $s'_Q(t) = s_Q(t + a)$ при $t' - a < t \leqslant t''$ и $s'_Q(t) = s_Q(t + t' - t'' - a)$ при $t'' \leqslant \leqslant t \leqslant t'' + a$. Если s' — расписание, то будем говорить, что оно получено из расписания s в результате применения операции $O_3(Q, R, t', t'', a)$.

2. Не нарушая общности рассуждений, можно рассматривать только такие расписания s, что $s_L(t) \neq 0$ на некотором промежутке $(0, T_L)$ и $s_L(t) = 0$ при $t > T_L$ или $s_L(t) = 0$ для всех $t \geqslant 0$, $L = \overline{1, M}$. Иными словами, каково бы ни было исходное расписание s, существует расписание $\hat{s}$, при котором каждый прибор при обслуживании требований функционирует непрерывно, и это расписание по качеству не уступает исходному.

Действительно, пусть, например, $s_R(t') = 0$ и $s_R(t) \neq \neq 0$ при некотором $t > t' \geqslant 0$. Поскольку расписание s допускает конечное число прерываний, то R и t' можно выбрать таким образом, чтобы t' было наибольшим. Пусть при этом $s_L(t') = \nu_L$, а $s_L(t' + \delta) = \mu_L$, $L = \overline{1, M}$. Значения ν_L и μ_L не обязательно различны. Выберем $\delta > 0$ таким образом, чтобы $s_L(t' + \delta_1) = \mu_L$ для всех $0 < \delta_1 < < \delta$, $L = \overline{1, M}$.

Если существует такое $1 \leqslant Q \leqslant M$, что $\nu_Q \neq 0$ и $\mu_Q \neq 0$, то применим к расписанию s операцию $O_1(R, Q, t')$, получая в результате новое расписание s'.

Если все $\nu_L = 0$, $L = \overline{1, M}$, то применим к расписанию s операцию $O_2(R, t', -(t' - t''))$, получая в результате расписание s''. Значение $t'' \geqslant 0$ выбирается наибольшим среди всех значений, удовлетворяющих условию $s_L(t) = = 0$, $t'' < t \leqslant t'$, для всех $L = \overline{1, M}$, и, по крайней мере, для одного $1 \leqslant L \leqslant M$ значение $s_L(t'') \neq 0$.

Наконец, если не имеет место ни одна из перечисленных ситуаций, то существует $1 \leqslant H \leqslant M$ такое, что $\mu_H \neq 0$ и $\mu_H \neq \nu_L$ для всех $L = \overline{1, M}$. В частности, может оказаться $H = R$. Применим к расписанию s последовательно операцию $O_1(R, H, t')$, затем $O_2(R, t', -(t' - \tilde{t}))$, получая в результате расписание $\tilde{s}$. Значение $\tilde{t} > 0$ выбирается наибольшим среди всех значений, удовлетворяющих условию $s_L(t) \neq \mu_H$, $L = \overline{1, M}$, $\tilde{t} \leqslant t < t'$, и $s_H(t) = \mu_H$, $t' < t \leqslant t' + (t' - \tilde{t})$.

Таким образом, в любом случае удается сократить величину простоя прибора R, не увеличивая простоя остальных приборов. Качество получаемых расписаний s', s'' или $\tilde{s}$ оказывается не хуже качества исходного расписания. Повторяя аналогичные рассуждения конечное число раз, приходим к искомому заключению.

3. Пусть расписание s допускает прерывания только в момент времени $t = t^{(1)}$ и в этот момент времени происходит прерывание процесса обслуживания $v \leqslant M$ требований $k_1, k_2, \ldots, k_v$. Требование k_j сначала обслуживается прибором Q_j в течение t'_{k_j} единиц времени, затем прибором R_j в течение t''_{k_j} единиц времени. Если $Q_j = R_j$, то при достаточно малом $\varepsilon > 0$ значение $s_{Q_j}(t^{(1)} + \varepsilon) \neq k_j$. Длину временного интервала между моментом времени возобновления обслуживания требования k_j и моментом времени $t^{(1)}$ обозначим через Δ_j. Пусть $\Delta_{j*} = \min\limits_{1 \leqslant j \leqslant v} \Delta_j$.

Если $Q_{j*} = R_{j*}$, то применим к расписанию s операцию $O_3(Q_{j*}, Q_{j*}, t^{(1)}, t^{(1)} + \Delta_{j*} - t'_{k_{j*}}, t'_{k_{j*}})$, получая в результате расписание $s^{(1)}$. Вектор времени завершения обслуживания требований при этом расписании удовлетворяет условию $\bar{t}(s^{(1)}) \leqslant \bar{t}(s)$. Следовательно, $F(\bar{t}(s^{(1)})) \leqslant$ $\leqslant F(\bar{t}(s))$.

Пусть $Q_{j*} \neq R_{j*}$. Если $\Delta_{j*} = 0$, то, применяя к расписанию s операцию $O_1(R_{j*}, Q_{j*}, t^{(1)})$, получаем расписание $s^{(2)}$, по качеству эквивалентное исходному. Если $\Delta_{j*} > 0$, то применим к расписанию s операцию $O_3(Q_{j*}, R_{j*}, t^{(1)}, t^{(1)} + \Delta_{j*} + t''_{k_{j*}}, t'_{k_{j*}})$. Полученная совокупность функций $s^{(3)}$ также является расписанием. Если $F(\bar{t}(s^{(3)})) > F(\bar{t}(s))$, то применим к s операцию

$O_3(R_{j*}, Q_{j*}, t^{(1)} + \Delta_{j*} + t''_{k_{j*}}, t^{(1)}, t''_{k_{j*}})$. Обозначим полученную совокупность функций через $s^{(4)}$. Эта совокупность функций является расписанием. Векторы $\bar{t}(s^{(3)})$ и $\bar{t}(s^{(4)})$ времен завершения обслуживания требований при расписаниях $s^{(3)}$ и $s^{(4)}$ соответственно связаны с вектором $\bar{t}(s)$ соотношениями $\bar{t}(s^{(3)}) = \bar{t}(s) + t'_{k_{j*}} \cdot \bar{e}$ и $\bar{t}(s^{(4)}) = \bar{t}(s) - t''_{k_{j*}} \cdot \bar{e}$, где $\bar{e}$ — некоторый n-мерный вектор с компонентами 0, 1 и —1. Поскольку $F(\bar{x})$ — квазивогнутая функция и по предположению $F(\bar{t}(s^{(3)})) > F(\bar{t}(s))$, то $F(\bar{t}(s^{(4)})) \leqslant F(t(s))$.

Таким образом, всегда можно построить расписание, по качеству не уступающее исходному расписанию s. Это расписание, так же как и расписание s, допускает прерывания только в момент времени $t = t^{(1)}$. Однако прерыванию подвергается не более $v - 1$ требований. Следовательно, существует и расписание без прерываний процесса обслуживания требований, по качеству не уступающее расписанию s.

4. Для завершения доказательства теоремы достаточно показать, что если расписание s допускает прерывания только в моменты времени $t^{(1)}, t^{(2)}, \ldots, t^{(w)}$, то существует расписание s', допускающее прерывания только в моменты времени $t^{(1)}, t^{(2)}, \ldots, t^{(w-1)}$ и при этом $F(\bar{t}(s')) \leqslant F(\bar{t}(s))$.

Расписание s удовлетворяет условиям предыдущего пункта при $t > t^{(w-1)}$. Следовательно, можно положить $t^{(w-1)} = 0$ и воспользоваться приведенными выше рассуждениями. В результате получаем расписание, не допускающее прерываний при $t > t^{(w-1)}$, совпадающее с исходным расписанием при $t \leqslant t^{(w-1)}$ и не уступающее ему по качеству. Теорема доказана.

Из определения квазивогнутой функции и доказанной теоремы следует, в частности, что при минимизации суммарного штрафа за обслуживание для достаточно широкого класса функций штрафа можно рассматривать расписания без прерываний процесса обслуживания. При этом, однако, предполагается, что все требования поступают в очередь на обслуживание одновременно.

Если требования поступают неодновременно и число приборов $M > 1$, то ситуация значительно усложняется. В этом случае даже при одинаковых для всех требований линейных функциях штрафа оптимальное расписание может содержать прерывания процесса обслуживания в моменты времени, отличные от моментов времени поступления требований.

Конструктивные приемы доказательства приведенных в этом параграфе теорем позволяют в соответствующих ситуациях преобразовать любое оптимальное расписание в оптимальное расписание с минимально возможным числом прерываний.

§ 2. Обслуживание в заданные сроки

Рассмотрим детерминированную систему обслуживания, состоящую из M параллельных идентичных приборов.

На обслуживание поступает n требований, каждое из которых может обслуживаться на любом из приборов. При этом в каждый момент времени любой прибор может обслуживать не более одного требования, а каждое требование может обслуживаться одним прибором. Требование k поступает в очередь на обслуживание в момент времени $d_k \geqslant 0$ и требует для обслуживания $t_k > 0$ единиц времени. Известен момент времени (директивный срок) $D_k \geqslant \geqslant 0$, к которому необходимо завершить обслуживание требования k. Будем считать, что d_k, t_k и D_k — целые числа, $k = \overline{1, n}$. Предполагается, что возможны прерывания процесса обслуживания требований. Под прерыванием понимается перенос обслуживания требования с одного прибора на другой либо прекращение обслуживания требования до его полного завершения (с последующим возобновлением обслуживания на этом же или на другом приборе). Определенный интерес представляет выявление условий, при которых все требования могут быть обслужены в заданные директивные сроки.

2.1. Присвоим временным промежуткам единичной длины, начиная с момента $d = 0$, номера $1, 2, \ldots, \max\limits_{1 \leqslant k \leqslant n} D_k$. Будем предполагать, что единичный промежуток с номером θ имеет вид $(\theta - 1, \theta]$. Поскольку требование k поступает в очередь на обслуживание в момент времени d_k и должно быть обслужено не позднее момента времени D_k, то это требование может обслуживаться в течение временных промежутков с номерами $d_k + 1,\ d_k + 2, \ldots, D_k$.

Поставим множеству номеров временных промежутков $\{1, 2, \ldots, \theta, \ldots, \max\limits_{1 \leqslant k \leqslant n} D_k\}$, множеству требований $\{1, 2, \ldots, n\}$ и набору чисел d_k, t_k, D_k, $k = \overline{1, n}$, в соответствие транспортную сеть G, образованную вершинами $x_1, x_2, \ldots, x_{\max\limits_k D_k}, y_1, y_2, \ldots, y_n$, входом x_0 и выходом z; вершина x_θ соответствует временному промежутку с номером θ, вершина y_k соответствует требованию

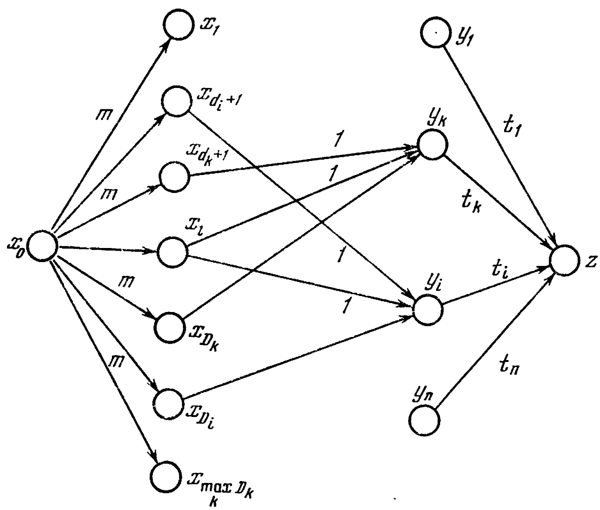

Рис. 4.2.1.

k; вершины x_θ, $\theta = \overline{d_k + 1, D_k}$, соединим дугами пропускной способности $c(x_\theta, y_k) = 1$ с вершиной y_k; вершины x_0 и x_θ — дугой пропускной способности $c(x_0, x_\theta) = M$ $(\theta = \overline{1, \max\limits_k D_k})$, вершины y_k и z — дугой пропускной способности $c(y_k, z) = t_k$, $k = \overline{1, n}$ (рис. 4.2.1).

Каждый поток, насыщающий выходные дуги сети G, определяет расписание, при котором все требования обслуживаются без нарушения директивных сроков. С другой стороны, каждое расписание, при котором все требования обслуживаются в заданные сроки, определяет поток, насыщающий выходные дуги сети G.

Пусть $\tilde{N} \subseteq N = \{1, 2, \ldots, n\}$. Обозначим через $\varepsilon(\tilde{N}, \theta)$ число требований $k \in \tilde{N}$ таких, что $d_k < \theta \leqslant D_k$. Тогда в силу известной теоремы о насыщении, необходимым и достаточным условием существования потока, насыщающего выходные дуги, является выполнение неравенства

$$\sum_{k \in \tilde{N}} t_k \leqslant \sum_{\theta = \min\limits_{k \in \tilde{N}} d_k + 1}^{\max\limits_{k \in \tilde{N}} D_k} \min[\varepsilon(\tilde{N}, \theta), M] \tag{2.1}$$

для всех $\tilde{N} \subseteq N$.

Таким образом, для того чтобы существовало расписание, при котором все требования обслуживаются без нарушения директивных сроков, необходимо и достаточно, чтобы неравенство (2.1) выполнялось для всех $\tilde{N} \subseteq N$.

2.2. Сформулируем условия обслуживания требований в заданные сроки для некоторых частных случаев постановки задачи.

а) $d_k \geqslant 0$, $D_k = D$, $k = \overline{1, n}$. Пронумеруем требования множества N в порядке поступления их в очередь на обслуживание, т. е. будем считать $d_1 \leqslant d_2 \leqslant \ldots \leqslant d_n$. Пусть $\tilde{N} \subseteq N$ и $\tilde{N} = \{i_1, i_2, \ldots, i_l\}$, где $i_j < i_k$, если $j < k$. Тогда неравенство (2.1) преобразуется следующим образом:

$$\begin{aligned} \sum_{j=1}^{l} t_{i_j} \leqslant \sum_{\theta = d_{i_1}+1}^{D} \min[\varepsilon(\tilde{N}, \theta), M] &= \\ = (d_{i_2} - d_{i_1}) \cdot 1 + (d_{i_3} - d_{i_2}) \min(2, M) &+ \ldots \\ \ldots + (D - d_{i_l}) \min(l, M). & \end{aligned} \tag{2.2}$$

Неравенство (2.2) должно выполняться для всех $\tilde{N} = \{i_1, i_2, \ldots, i_l\} \subseteq N$, $1 \leqslant l \leqslant n$. При $l \leqslant M$ и $l > M$ из (2.2) получим соответственно

$$\sum_{j=1}^{l} t_{i_j} \leqslant D \cdot l - \sum_{j=1}^{l} d_{i_j} \tag{2.3}$$

и

$$\sum_{j=1}^{l} t_{i_j} \leqslant D \cdot M - \sum_{j=1}^{} d_{i_j}. \tag{2.4}$$

Для выполнения этих неравенств при любых $\tilde{N} \subseteq N$ необходимо и достаточно выполнение условий

$$d_k + t_k \leqslant D, \quad k = \overline{1, n}, \tag{2.5}$$

$$\sum_{j=1}^{M} (d_{i_j} + t_{i_j}) + \sum_{k=i_M+1}^{n} t_k \leqslant D \cdot M \tag{2.6}$$

для всех $\tilde{N} = \{i_1, i_2, \ldots, i_M\} \subseteq N$.

Таким образом, для обслуживания требований без нарушения директивного срока в случае а) необходимо и достаточно выполнение условий (2.5), (2.6).

б) $d_k = 0$, $D_k = D$, $k = \overline{1, n}$. Из (2.5), (2.6) непосредственно следует, что условием обслуживания требований в заданный срок для случая б) является выполнение неравенств

$$t_k \leqslant D, \quad k = \overline{1, n}, \tag{2.7}$$

$$\sum_{k=1}^{n} t_k \leqslant D \cdot M. \tag{2.8}$$

в) $d_k = 0$, $D_k \geqslant 0$, $k = \overline{1, n}$. Пронумеруем требования по неубыванию директивных сроков D_k. Пусть $\tilde{N} \subseteq N$ и $N = \{i_1, i_2, \ldots, i_l\}$, где $i_j < i_k$, если $j < k$. Тогда неравенство (2.1) запишется в виде

$$\sum_{j=1}^{l} t_{i_j} \leqslant \sum_{\theta=1}^{D_{i_l}} \min [\varepsilon(\tilde{N}, \theta), M] =$$
$$= D_{i_1} \min (l, M) + (D_{i_2} - D_{i_1}) \min (l-1, M) + \ldots$$
$$\ldots + (D_{i_{l-1}} - D_{i_{l-2}}) \min (2, M) + (D_{i_l} - D_{i_{l-1}}). \tag{2.9}$$

Неравенство (2.9) должно выполняться для всех $\tilde{N} = \{i_1, i_2, \ldots, i_l\} \subseteq N$, $1 \leqslant l \leqslant n$.

При $l \leqslant M$ и $l > M$ получим из (2.9) соответственно

$$\sum_{j=1}^{l} t_{i_j} \leqslant \sum_{j=1}^{l} D_{i_j} \tag{2.10}$$

и

$$\sum_{j=1}^{l} t_{i_j} \leqslant \sum_{j=l-M+1}^{l} D_{i_j}. \tag{2.11}$$

Для выполнения последних неравенств при любых $\tilde{N} \subseteq N$ необходимо и достаточно, чтобы выполнялись следующие условия:

$$t_k \leqslant D_k, \quad k = \overline{1, n}, \tag{2.12}$$

$$\sum_{k=1}^{i_{l-M+1}} t_k + \sum_{j=l-M+2}^{l} t_{i_j} \leqslant \sum_{j=l-M+1}^{l} D_{i_j} \tag{2.13}$$

для всех $\tilde{N} = \{i_1, i_2, \ldots, i_l\} \subseteq N$, $l > M$.

Итак, для того чтобы существовало расписание, при котором требования обслуживаются без нарушения директивных сроков, необходимо и достаточно выполнение условий

$$t_k \leqslant D_k, \quad k = \overline{1, n}, \tag{2.14}$$

$$\sum_{k=1}^{i_1} t_k + \sum_{j=2}^{M} t_{i_j} \leqslant \sum_{j=1}^{M} D_{i_j} \tag{2.15}$$

для всех $\tilde{N} = \{i_1, i_2, \ldots, i_M\} \subseteq N$.

г) $d_k \geqslant 0$, $D_k \geqslant 0$, $k = \overline{1, n}$, $M = 1$. Пронумеруем требования так же, как в случае в). Пусть $\tilde{N} \subseteq N$ и $\tilde{N} = \{i_1, i_2, \ldots, i_l\}$, где $i_j < i_k$, если $j < k$. Тогда неравенство (2.1) преобразуется следующим образом:

$$\sum_{j=1}^{l} t_{i_j} \leqslant \sum_{\theta = \min\limits_{1 \leqslant j \leqslant l} d_{i_j} + 1}^{D_{i_l}} \min[\varepsilon(\tilde{N}, \theta), 1] = D_{i_l} - \min_{1 \leqslant j \leqslant l} d_{i_j}. \tag{2.16}$$

Неравенство (2.16) должно выполняться для всех $\tilde{N} = \{i_1, i_2, \ldots, i_l\} \subseteq N$, $1 \leqslant l \leqslant n$. Пусть $d_k \geqslant d_{k+1}$, $k = \overline{1, n-1}$. Тогда из (2.16) получаем

$$\sum_{i=1}^{k} t_i + d_k \leqslant D_k, \quad k = \overline{1, n}. \tag{2.17}$$

Пусть $d_k \leqslant d_{k+1}$, $k = \overline{1, n-1}$. Тогда из (2.16) получаем

$$\sum_{i=k}^{l} t_i \leqslant D_l - d_k \tag{2.18}$$

для любых $1 \leqslant k \leqslant l \leqslant n$.

Таким образом, условия (2.17), (2.18) являются необходимыми и достаточными условиями для того, чтобы в случае г) соответственно при $d_k \geqslant d_{k+1}$ и $d_k \leqslant d_{k+1}$, $k = \overline{1, n-1}$, существовало расписание обслуживания требований, при котором не нарушаются директивные сроки.

д) $d_k = 0$, $D_k \geqslant 0$, $k = \overline{1, n}$, $M = 1$. Из п. г) следует, что условием существования расписания, при котором все требования обслуживаются без нарушения директивных сроков, является в этом случае соблюдение следующих неравенств:

$$\sum_{i=1}^{k} t_i \leqslant D_k, \quad k = \overline{1, n}. \tag{2.19}$$

Приведенные условия существования расписаний, при которых все требования обслуживаются в заданные сроки, справедливы только при разрешении прерываний процесса обслуживания требований (за исключением условия (2.19)).

§ 3. Некоторые задачи упорядочения

Рассмотрим сравнительно частные задачи построения оптимальных расписаний обслуживания n требований M идентичными параллельными приборами в предположении, что процесс обслуживания каждого требования протекает непрерывно.

3.1. Одной из задач такого рода является задача построения расписания, которому соответствует наименьшее значение *суммы времен запаздываний* в обслуживании требований относительно заданных директивных сроков. Директивные сроки будем по-прежнему обозначать через D_k, а времена завершения обслуживания требований при расписании s через $\bar{t}_k$. Предполагается, что процесс обслуживания каждого требования может быть начат в любой момент времени, т. е. $d_k = 0$, $k = \overline{1, n}$, и протекает без прерываний до завершения обслуживания этого требования.

Необходимо так распределить требования множества $N = \{1, 2, \ldots, n\}$ по обслуживающим приборам и указать такие последовательности их обслуживания каждым прибором, чтобы величина $F_\Sigma(s) = \sum_{k=1}^{n} \max(0, \bar{t}_k - D_k)$ была наименьшей.

Ограничимся рассмотрением ситуации, когда все директивные сроки одинаковы и равны $D \geqslant 0$. Все рассматриваемые ниже расписания предполагаются активными.

Пусть требования множества $N = \{1, 2, \ldots, n\}$ пронумерованы в порядке неубывания значений t_k и $N_L \subseteq N$ — множество требований, обслуживаемых L-м прибором, $L = \overline{1, M}$. На основании теоремы 2.3 гл. 3 поиск оптимального расписания можно ограничить рассмотрением расписаний, при которых каждый прибор обслуживает требования в порядке возрастания их номеров. Таким образом, искомое расписание однозначно определяется разбиением множества N на (не обязательно непустые) подмножества $N_1, N_2, \ldots, N_M$, попарно не имеющие общих элементов. Если упорядоченной по возрастанию последовательностью элементов множества N_L является последовательность $(i_1, i_2, \ldots, i_{|N_L|})$, то время начала обслуживания требования i_j равно $\underline{t}_{i_j} = \sum_{k=1}^{j-1} t_{i_k}$, а время завершения его обслуживания $\bar{t}_{i_j} = \underline{t}_{i_j} + t_{i_j}$.

Разобьем множество N_L на два подмножества Q_L и R_L по следующему признаку: $i_j \in Q_L$, если $\underline{t}_{i_j} < D$, в противном случае $i_j \in R_L$. Положим $Q = \bigcup_{L=\overline{1,M}} Q_L$ и $R = \bigcup_{L=\overline{1,M}} R_L$.

Теорема 3.1. *Существует оптимальное расписание, при котором*

1) $Q = \{1, 2, \ldots, |Q|\}$,

2) *если* $|Q| < n$, *то* $\sum_{k \in Q_L} t_k \geqslant D$ *и* R_L *содержит те и только те элементы* $R = \{|Q| + 1, \ldots, n\}$, *которые отличаются от* $|Q| + L$ *на величину, кратную* M, $L = \overline{1, M}$.

Доказательство. а) Пусть s^* — оптимальное расписание и при этом расписании L-й прибор обслуживает (в порядке возрастания номеров) требования множества N_L^*, $Q_L = Q_L^*$, $R_L = R_L^*$, $L = \overline{1, M}$, $Q = Q^*$ и $R = R^*$. Если $|Q^*| = n$, то справедливость утверждения теоремы очевидна. Пусть $|Q^*| < n$. Если предположить, что при некотором $1 \leqslant L \leqslant M$ значение $\sum\limits_{k \in Q_L^*} t_k < D$, то приходим к заключению, что расписание s^* не является оптимальным, поскольку одно из требований множества R^* можно было бы обслужить L-м прибором, начиная с момента времени $\sum\limits_{k \in Q_L^*} t_k < D$. Следовательно, $\sum\limits_{k \in Q_L^*} t_k \geqslant D$ для всех $L = \overline{1, M}$.

б) Выберем наибольший элемент $j \in Q^*$ и наименьший элемент $i \in R^*$. Предположим, что $i < j$. Если $t_i = t_j$, то эти требования эквивалентны и их можно поменять местами. Пусть $t_i < t_j$. Поскольку каждый прибор обслуживает требования в порядке возрастания их номеров, то требования i и j обслуживаются разными приборами. Пусть, для определенности, требование i обслуживается прибором L, а требование j — прибором H. Очевидно, можно полагать, что $\underline{t}_j < D$, $\bar{t}_j \geqslant D$, $\underline{t}_i \geqslant D$ и не существует $k \in R_L$ с $\underline{t}_k < \underline{t}_i$.

Рассмотрим новое расписание s_1, отличающееся от расписания s^* тем, что требование i обслуживается прибором H непосредственно перед требованием j. Суммарный штраф при этом изменяется на величину $t_i(|R_H^*| - |R_L^*| + 2) + \max(\underline{t}_j + t_i - D, 0) - (\bar{t}_i - D)$.

Если $|R_L^*| \geqslant |R_H^*| + 2$, то эта величина неположительна. Если $|R_L^*| = |R_H^*| + 1$, то эта величина также неположительна при $\underline{t}_j + t_i \leqslant D$.

Если не имеет места ни одна из указанных ситуаций, то рассмотрим расписание s_2, отличающееся от исходного оптимального расписания s^* тем, что на приборе L вместо требования i обслуживается требование j, а на приборе H вместо требования j обслуживается требование i. Суммарный штраф при этом изменится на величину $(t_j - t_i) \times (|R_L^*| - |R_H^*|) + \max(\underline{t}_j + t_i - D, 0) - (\bar{t}_j - D)$.

Если $|R_L^*| \leqslant |R_H^*|$, то эта величина неположительна. Она неположительна и в случае, когда $|R_L^*| = |R_H^*| + 1$ при условии, что $\underline{t}_j + t_i > D$.

Таким образом, в любом случае существует расписание, которому соответствует суммарный штраф, не превосходящий по величине суммарного штрафа, соответствующего оптимальному расписанию s^*. Множество Q, определяемое этим расписанием, отличается от Q^* только тем, что оно содержит элемент i и, возможно, не содержит элемента j.

Повторяя аналогичные рассуждения конечное число раз, приходим к заключению, что существует оптимальное расписание, удовлетворяющее условию 1) теоремы. Если $|Q| < n$, то в этом расписании $\sum_{k \in Q_L} t_k \geqslant D$ для всех $L = \overline{1, M}$. Не нарушая общности, можно полагать, что $D \leqslant \sum_{k \in Q_1^*} t_k \leqslant \ldots \leqslant \sum_{k \in Q_M^*} t_k$. В дальнейшем символ s^* будем употреблять для обозначения оптимального расписания, обладающего указанными свойствами.

в) Для завершения доказательства теоремы достаточно показать, что существует оптимальное расписание, при котором требование $|Q^*| + 1$ обслуживается первым прибором, требование $|Q^*| + 2$ — вторым прибором и т. д., требование $|Q^*| + M$ — M-м прибором, требование $|Q^*| + M + 1$ — первым прибором и т. д., пока не будет исчерпано множество $R^* = \{|Q^*| + 1, \ldots, n\}$. Предполагается, естественно, что $|Q^*| < n$.

Пусть при расписании s^* указанным образом обслуживаются требования $|Q^*| + 1$, $|Q^*| + 2$, ..., $|Q^*| + lM + (L - 1)$. Множество этих требований обозначим через $\bar{R}$, а множество остальных требований из R^* — через $\bar{\bar{R}}$. Предположим, что $\bar{\bar{R}} \neq \phi$ и наименьший номер требования из $\bar{\bar{R}}$, обслуживаемого L-м прибором, равен $j \neq |Q^*| + lM + L$. Тогда требование $i = |Q^*| + lM + L$ обслуживается некоторым прибором $H \neq L$. Таким образом, по предположению оптимальное расписание s^* таково, что прибор L обслуживает последовательно требования $|Q^*| + L$, $|Q^*| + M + L$, ..., $|Q^*| + (l - 1) M + L$, j, ..., а прибор H обслуживает последовательно требования $|Q^*| + H$, $|Q^*| + M + H$, ..., $|Q^*| + (l - 1) M + H$, $i = |Q^*| + lM + L$, ..., если $H > L$, или требования $|Q^*| + H$, $|Q^*| + M + H$, ..., $|Q^*| + lM + H$, $i = |Q^*| + lM + L$, ..., если $H < L$. В любом случае $t_j \leqslant t_i$.

Аналогично п. б) рассмотрим расписания s_1 и s_2. Первое из них отличается от s^* только тем, что требование i обслуживается прибором L непосредственно перед требованием j. Второе отличается от s^* тем, что на приборе L вместо требования j обслуживается требование i, а на приборе H вместо требования i обслуживается требование j.

В зависимости от числа требований, обслуживаемых прибором L, начиная с момента времени $\underline{t}_j$, и числа требований, обслуживаемых прибором H, начиная с момента времени $\underline{t}_i$, одно из расписаний s_1 или s_2 окажется оптимальным. Поскольку в обоих случаях прибором L обслуживается требование $i = |Q^*| + lM + L$, то, повторяя аналогичные рассуждения конечное число раз, убеждаемся в справедливости теоремы.

Следствие 1. *Расписанию s, при котором каждый прибор $L = \overline{1, M}$ обслуживает требования в последовательности L, $M + L$, $2M + L$, ..., соответствует наименьшее значение суммы времен завершения обслуживания всех требований.*

Следствие 2. *Пусть обслуживание требований L-м прибором не может быть начато раньше момента времени $d_L \geqslant 0$, $L = \overline{1, M}$. Расписанию s, при котором каждое очередное требование $k = 1, 2, \ldots, n$ назначается на обслуживание на тот прибор, который раньше других оказывается свободным, соответствует наименьшее значение суммы времен завершения обслуживания всех требований.*

Обозначим через $S(\mu)$ класс расписаний, удовлетворяющих условию теоремы 3.1 и дополнительному условию $|Q| = \mu$, μ — натуральное число. Оптимальное расписание s^*, очевидно, принадлежит хотя бы одному из классов $S(\mu)$ при некотором $\mu = \mu^*$.

Число различных непустых классов $S(\mu)$ не превышает M. Действительно, пусть $\underline{\mu}$ — наименьшее значение μ, при котором $S(\mu) \neq \phi$. Если $\underline{\mu} = n$, то $S(\underline{\mu})$ — единственный непустой класс. Если $\underline{\mu} < n$, то по определению класса $S(\mu)$ значения $z_L = \sum_{k \in Q_L} t_k - D \geqslant 0$, $L = \overline{1, M}$, для любого расписания из этого класса и, следовательно, $\sum_{L=1}^{M} z_L + \sum_{i=1}^{h} t_{|Q|+i} < \sum_{i=1}^{M} t_{|Q|+i}$ только при $h \leqslant M - 1$.

Обозначим через $\bar{\mu}$ наибольшее значение μ, при котором $S(\mu) \neq \phi$. Таким образом, $\bar{\mu} - \underline{\mu} \leqslant M - 1$. Необходимо отметить, что при некоторых $\underline{\mu} < \mu < \bar{\mu}$ может оказаться $S(\mu) = \phi$.

Найдем достаточные условия, при которых расписанию $s \in S(\mu)$ соответствует наименьший суммарный штраф среди всех расписаний, принадлежащих классу $S(\mu)$, $\underline{\mu} \leqslant \mu \leqslant \bar{\mu}$. Обозначим через $c(\mu)$ число, равное остатку от деления $(n - \mu)$ на M при $n - \mu > 0$ и равное M в противном случае.

Теорема 3.2. *Расписание $s \in S(\mu)$ является оптимальным в $S(\mu)$, если при этом расписании величина*

$$Z(\mu) = \sum_{L=1}^{c(\mu)} \max(0, z_L) \tag{3.1}$$

достигает наименьшего значения.

Доказательство. Если $\mu = n$, то $Z(\mu) = F_\Sigma(s)$. Пусть $\mu < n$ и, следовательно, $z_L > 0$ для всех $L = \overline{1, M}$. Не нарушая общности, будем предполагать, что $z_1 \leqslant z_2 \leqslant \ldots \leqslant z_M$.

Суммарный штраф $F_\Sigma(s)$, соответствующий любому расписанию $s \in S(\mu)$, можно представить в виде

$$F_\Sigma(s) = \sum_{L=1}^{M} |R_L| z_L + \widetilde{F}_\Sigma, \tag{3.2}$$

где величина $\widetilde{F}_\Sigma$ не зависит от $s \in S(\mu)$.

По условию $|R_1| = |R_2| = \ldots = |R_{c(\mu)}| = |R_{c(\mu)+1}| + 1 = \ldots = |R_M| + 1$ и $\sum_{L=1}^{M} z_L = \text{const}$ для всех $s \in S(\mu)$. Следовательно, $F_\Sigma(s)$ достигает наименьшего значения, если величина $\sum_{L=1}^{c(\mu)} z_L$ минимальна, что и требовалось доказать.

Таким образом, при построении оптимального расписания s^* можно воспользоваться следующей вычислительной схемой. Сначала необходимо определить значение $\underline{\mu}$. Затем, используя теоремы 3.1, 3.2, построить оптимальное в $S(\mu)$ расписание s_μ для $\mu = \underline{\mu}, \underline{\mu}+1, \ldots, \overline{\mu}$. Среди полученных расписаний, число которых не превосходит M, выбрать то, которому соответствует наименьший суммарный штраф. Это расписание является искомым.

В общем случае поиск оптимального расписания требует перебора большого числа возможных вариантов. Следующая теорема позволяет оценить и выбрать, если необходимо, в качестве приближенного решения некоторое из «промежуточных» расписаний.

Теорема 3.3. *Если расписание* $s \in S(\mu)$, *то*

$$F_\Sigma(s) - F_\Sigma(s^*) \leqslant Z(\mu). \tag{3.3}$$

Доказательство. Не нарушая общности, можно предполагать, что $s^* \in S(\mu^*)$. Неравенство (3.3), очевидно, выполняется при $\mu^* = \mu$.

Пусть $\mu^* \neq \mu$. Расписанию s соответствуют множества $Q, R_1, R_2, \ldots, R_M$, а расписанию s^* — множества $Q^*, R_1^*, R_2^*, \ldots, R_M^*$. По определению $|Q| = \mu$ и $|Q^*| = \mu^*$.

Рассмотрим случай, когда $\mu^* = \mu + h$, $0 < h \leqslant M - 1$. Множества

$$R_1^* = R_{h+1}, \ldots, R_{M-h}^* = R_M,$$

$$R_{M-h+1}^* = R_1 \setminus \{|Q|+1\}, \ldots, R_M^* = R_h \setminus \{|Q|+h\}.$$

Значения z_L, определяемые расписанием s, неотрицательны и

$$\sum_{L=1}^{M} \max(0, z_L^*) = \sum_{L=1}^{M} z_L + \sum_{i=1}^{h} t_{|Q|+i},$$

где значения z_L^* определяются расписанием s^*. Возможны следующие ситуации. Если $h < c(\mu)$, то

$$F_\Sigma(s) - F_\Sigma(s^*) = \sum_{L=1}^{c(\mu)} z_L - \sum_{L=1}^{c(\mu)-h} \max(0, z_L^*).$$

Если $h = c(\mu)$, то

$$F_\Sigma(s) - F_\Sigma(s^*) = \sum_{L=1}^{c(\mu)} z_L,$$

Если $h > c(\mu)$, то

$$F_\Sigma(s) - F_\Sigma(s^*) = \sum_{L=1}^{M} z_L + \sum_{L=1}^{c(\mu)} z_L + \sum_{i=1}^{c(\mu)} t_{|Q|+i} - \sum_{L=1}^{M-h+c(\mu)} \max(0, z_L^*) = \sum_{L=1}^{c(\mu)} z_L - \Bigl(\sum_{i=c(\mu)+1}^{h} t_{|Q|+i} - \sum_{L=M-h+c(\mu)+1}^{M} \max(0, z_L^*) \Bigr).$$

Таким образом, неравенство 3.3 выполняется при $\mu^* \geqslant \mu$.

Пусть $\mu = \mu^* + h$, $0 < h \leqslant M - 1$. В этом случае $R_1^* = R_{M-h+1} \cup \{|Q| - h + 1\}, \ldots, R_h^* = R_M \cup \{|Q|\}$, $R_{h+1}^* = R_1, \ldots, R_M^* = R_{M-h}$. Значения z_L^* неотрицательны и $\sum_{L=1}^{M} \max(0, z_L) = \sum_{L=1}^{M} z_L^* + \sum_{i=1}^{h} t_{|Q|+1-i}$. Если $h \leqslant M - c(\mu)$, то

$$F_\Sigma(s) - F_\Sigma(s^*) = \sum_{L=1}^{c(\mu)} \max(0, z_L) - \sum_{L=1}^{c(\mu)+h} z_L^*.$$

Если $h > M - c(\mu)$, то

$$F_\Sigma(s) - F_\Sigma(s^*) = \sum_{L=1}^{c(\mu)} \max(0, z_L) - 2 \sum_{L=1}^{h+c(\mu)-M} z_L^* - \sum_{L=h+c(\mu)-M+1}^{M} z_L^* - \sum_{i=1}^{h+c(\mu)-M} t_{|Q|-h+i} \leqslant \sum_{L=1}^{c(\mu)} \max(0, z_L).$$

Следовательно, неравенство (3.3) выполняется и при $\mu < \mu^*$. Теорема доказана.

3.2. Рассмотрим задачу оптимального обслуживания n идентичных частично упорядоченных требований двумя идентичными приборами. Не нарушая общности, будем предполагать, что каждое требование может быть обслужено любым из двух приборов в течение одной единицы времени.

На множестве $N = \{1, 2, \ldots, n\}$ всех требований задано отношение $\rightarrow$ строгого частичного порядка, регламентирующее возможную последовательность их обслуживания. Если $i \to j$, то требование j должно быть обслужено после того, как будет обслужено требование i. В качестве критерия оптимальности выберем *общее время обслуживания* требований.

Процесс обслуживания в рассматриваемом случае может быть описан заданием последовательности векторов $s = \langle (p_1, q_1), \ldots, (p_t, q_t), \ldots, (p_T, q_T) \rangle$, где p_t и q_t, $t = \overline{1, T}$, принимают одно из значений $0, 1, 2, \ldots, n$. Если $p_t = k \neq 0$ и $q_t = l \neq 0$, то в t-й единичный интервал времени один из приборов обслуживает требование k, второй — требование l. Условие $p_t = 0$ или $q_t = 0$ означает, что один из приборов в t-й единичный интервал времени простаивает. Эту последовательность естественно называть расписанием.

Расписания s должны удовлетворять следующим двум условиям: 1) $p_t \neq q_t$, $t = \overline{1, T}$, и для каждого $1 \leqslant k \leqslant n$ существует одно и только одно $1 \leqslant t \leqslant T$ такое, что $(p_t - k)(q_t - k) = 0$; 2) если $i \to j$, $(p_{t_1} - i)(q_{t_1} - i) = 0$ и $(p_{t_2} - j)(q_{t_2} - j) = 0$, то $t_1 < t_2$.

Оптимальному расписанию s^* соответствует наименьшее значение общего времени $T = T^*$ обслуживания требований. Тем самым это расписание должно содержать максимально возможное число векторов, компоненты которых отличны от нуля.

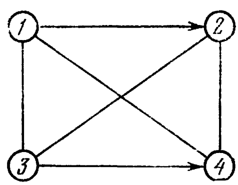

Рис. 4.3.1.

Рассмотрим неориентированный граф (N, U), где N — множество вершин (требований); U — множество ребер. Множество U не содержит ребро $[i, j]$ тогда и только тогда, когда $i \to j$ или $j \to i$.

В расписание s^* в качестве векторов (p_t, q_t) могут войти только те ребра $[i, j]$ графа (N, U), которые попарно не имеют общих вершин (условие 1)). Обозначим наибольшее число таких ребер через m^*. Таким образом, приходим к заключению, что $T^* \geqslant n - m^*$. Определение числа m^* и самих ребер может быть проведено методом чередующихся цепей.

Однако не всякая совокупность ребер графа (N, U), попарно не имеющих общих вершин, может входить в одно и то же расписание s. Действительно, ребра $[1, 4]$ и $[2, 3]$ (рис. 4.3.1) не могут входить в одну и ту же последовательность s, поскольку при любом их упорядочении в этом расписании будет нарушено условие 2).

Теорема 3.4. *Существует расписание s с $T = n - m^*$.*

Доказательство. Пусть $\tilde{N} \subseteq N$ и $\tilde{U}$ — произвольное непустое множество ребер $[i, j]$ графа (N, U), попарно не имеющих общих вершин. Будем говорить, что элемент $k \in \tilde{N}$ является максимальным в $\tilde{N}$, если не существует $l \in \tilde{N}$ такого, что $k \to l$.

Возможны следующие ситуации:

а) существует максимальный элемент k в $\tilde{N}$ и $k \neq i, j$ для всех ребер $[i, j]$ из $\tilde{U}$;

б) существует ребро $[i, j]$ в $\tilde{U}$ такое, что элементы i и j являются максимальными в $\tilde{N}$;

в) существуют ребра $[i_1, j_1]$ и $[i_2, j_2]$ в $\tilde{U}$ такие, что элементы i_1 и i_2 являются максимальными в $\tilde{N}$, а $[j_1, j_2] \in U$.

Действительно, если не имеет место ситуация а), то все максимальные в $\tilde{N}$ элементы инцидентны ребрам $\tilde{U}$.

Пусть $i^{(1)}$ — один из таких элементов, $[i^{(1)}, j^{(1)}] \in \tilde{U}$. Если $j^{(1)}$ — максимальный в $\tilde{N}$ элемент, то имеет место ситуация б). В противном случае в $\tilde{N}$ может быть выбран элемент $i^{(2)}$, который является максимальным в $\tilde{N}$, и $j^{(1)} \to i^{(2)}$. Очевидно, $i^{(2)} \neq i^{(1)}$ и существует ребро $[i^{(2)}, j^{(2)}] \in \tilde{U}$. Если $j^{(2)} = i^{(1)}$, то имеет место ситуация б). Если $[j^{(1)}, j^{(2)}] \in U$, то имеет место ситуация в). Наконец, если $j^{(2)} \neq i^{(1)}$ и $[j^{(1)}, j^{(2)}] \notin U$, то $j^{(1)} \to j^{(2)}$ и в $\tilde{N}$ может быть выбран максимальный элемент $i^{(3)}$, для которого выполняется условие $j^{(2)} \to i^{(3)}$. Очевидно, $i^{(3)} \neq i^{(1)}$ и $i^{(3)} \neq i^{(2)}$. Повторяя описанные рассуждения конечное число раз (число элементов, а следовательно, и максимальных в $\tilde{N}$ элементов конечно), приходим к заключению, что приведенные ситуации являются исчерпывающими.

Перейдем теперь к непосредственному построению искомого расписания s с $T = n - m^*$.

Пусть $\tilde{U}$ — максимальное множество ребер графа (N, U), попарно не имеющих общих вершин, $|\tilde{U}| = m^*$, а $\tilde{N}$ — множество тех вершин графа (N, U), которые не инцидентны ни одному из указанных ребер.

Если $\tilde{U} = \phi$, то N является линейно упорядоченным множеством и последовательность, имеющая вид $s = \langle (i_1, 0), (i_2, 0), \ldots, (i_n, 0) \rangle$, где $t_1 < t_2$, при

$i_{t_1} \to i_{t_2}$, является последовательностью обслуживания требований с общим временем обслуживания $T = n$.

Если $\bar{N} = \phi$, то относительно $\bar{U}$ и N могут иметь место ситуации б) и (или) в). При ситуации б) в момент времени $t = m^*$ на одном из приборов может быть обслужено требование i, на втором — требование j. При ситуации в) в момент времени $t = m^*$ могут быть обслужены требования i_1 и i_2. В первом случае ребро $[i, j]$ удаляется из множества $\bar{U}$, во втором — ребра $[i_1, j_1]$ и $[i_2, j_2]$ удаляются из $\bar{U}$ и заменяются ребром $[j_1, j_2]$. В обоих случаях получаем новое множество $\bar{U}$ с $|\bar{U}| = m^* - 1$, что позволяет продолжить приведенные рассуждения для $t = m^* - 1$, $m^* - - 2, \ldots,$ и т. д. до тех пор, пока не будет построено расписание $s = \langle (p_1, q_1), (p_2, q_2), \ldots, (p_{m^*}, q_{m^*}) \rangle$. В этом расписании ни одно p_t и q_t не равно нулю.

Наконец, если $\bar{U} \neq \phi$ и $\bar{N} \neq \phi$, то наряду с ситуациями б) и в) может иметь место ситуация а). В этом случае в рассматриваемый момент времени $t = n - m^*$, $n - - m^* - 1, \ldots, 1$ необходимо в первую очередь обслужить требование k и удалить это требование из множества $\bar{N}$. В остальном процесс построения расписания s аналогичен ранее описанному.

Таким образом, существует расписание s, при котором общее время обслуживания всех требований $T = n - m^*$. Это расписание является оптимальным, поскольку величина $n - m^*$ является нижней границей значений T на множестве всех расписаний s. Процесс его построения включает определение множества $\bar{U}$, содержащего наибольшее число ребер графа (N, U), попарно не имеющих общих вершин, и последующее выявление векторов (p_{n-m^*}, q_{n-m^*}), $(p_{n-m^*-1}, q_{n-m^*-1})$, $\ldots$, (p_1, q_1) описанным выше методом.

§ 4. Библиографическая справка

Вопросы существования оптимальных расписаний без прерываний процесса обслуживания рассматривались Р. Мак-Нотоном (линейные функции штрафа) [344] и М. Роткопфом (экспоненциальные функции штрафа) [387]. Теорема 1.1 приведена в [171] и [54], теорема 1.2 — в [171]. Обобщение теоремы 1.2 на случай неодновременного поступления требований проведено в [56].

Основу § 2 составили работы В. С. Танаева и В. С. Гордона [54, 57].

Первая из задач § 3 рассматривалась Дж. Рутом [386]. Эта задача существенно упрощается, если $\varphi_k(x) = a\tau_k$ и, в случае неидентичных приборов, время обслуживания $t_{kL} = t_k R_L$, $k = \overline{1,n}$, $L = \overline{1, M}$ [254]. Вторая задача § 3 рассматривалась в [274]. Случай древовидных порядков рассматривался ранее Т. С. Ху [306]. Дальнейшее развитие эти результаты получили в работах Р. Мюнтца и Е. Коффмана [355, 356] (случай $M \geqslant 2$, различные t_k, разрешены прерывания). Задача минимизации суммарного и максимального запаздывания в обслуживании требований рассматривалась В. А. Хорном [304]. Определению наименьшего числа прерываний посвящены работы Б. Н. Панайоти, Л. Я. Пьянзиной и В. А. Чебакова [131], В. А. Таланова [168]. Вопросы существования расписания заданной длины рассмотрены К. Г. Николаевым и Л. Н. Плужниковым [128].

Для построения оптимальных (в основном, по быстродействию) расписаний предложены разнообразные схемы метода «ветвей и границ» [8, 69, 107, 198, 242, 250, 269, 366, 326], динамического программирования [298, 327, 387], приближенные методы [6, 11, 14, 34, 162, 210, 287, 326, 399]. При этом, как правило, рассматриваются несколько более общие ситуации (приборы неидентичны, множество требований частично упорядочено и т. п.).

Некоторые вопросы рационального использования ограниченного числа операторов, обслуживающих параллельные приборы, рассмотрены В. К. Тютюкиным [181], Е. В. Левнером [97], Э. И. Волынским и Л. А. Красиным [38].

Использованию методов линейного программирования для упорядочения на параллельных приборах посвящены работы [112, 365, 428].

ГЛАВА 5

ДЕТЕРМИНИРОВАННЫЕ СИСТЕМЫ ОБСЛУЖИВАНИЯ С ДВУМЯ ПОСЛЕДОВАТЕЛЬНЫМИ ПРИБОРАМИ

Некоторое множество требований необходимо обслужить двумя приборами. Каждый прибор одновременно обслуживает не более одного требования. Для каждого требования указан порядок и характер его обслуживания приборами. Прерывания в обслуживании не допускаются. Необходимо построить в некотором смысле оптимальные (в основном по быстродействию) расписания процесса обслуживания требований.

§ 1. Последовательное обслуживание

Рассмотрим систему, состоящую из двух приборов A и B. На обслуживание в момент времени $d = 0$ поступают требования множества $N = \{1, 2, \ldots, n\}$. Для обслуживания требования $k = \overline{1, n}$ прибором A необходимо t_{kA} единиц времени, а прибором B необходимо t_{kB} единиц времени. Если обслуживание требования k прибором A начинается в момент времени $\underline{t}_{kA}$, то оно протекает непрерывно до момента времени $\bar{t}_{kA} = \underline{t}_{kA} + t_{kA}$. Аналогично, если обслуживание требования k прибором B начинается в момент времени $\underline{t}_{kB}$, то оно протекает непрерывно и завершается в момент времени $\bar{t}_{kB} = \underline{t}_{kB} + t_{kB}$. Предполагается, что каждый прибор одновременно не может обслуживать более одного требования и каждое требование одновременно обслуживается не более чем одним прибором.

В общем случае не обязательно все требования множества N должны быть обслужены сначала прибором A, затем прибором B. Обозначим через N_{AB} множество указанных требований. Аналогично, обозначим через N_{BA} множество требований, которые должны быть обслужены сначала прибором B, затем прибором A. Наконец, пусть N_A и N_B — множества требований, которые должны быть обслужены только прибором A или прибором B соответственно. Для любого требования $k \in N_{AB}$ величина $\underline{t}_{kB} \geqslant \bar{t}_{kA}$, для любого требования $k \in N_{BA}$ величина $\underline{t}_{kA} \geqslant \bar{t}_{kB}$.

Предполагается, что множества N_{AB}, N_{BA}, N_A и N_B попарно не пересекаются и $N = N_{AB} \cup N_{BA} \cup N_A \cup N_B$.

Нас будет интересовать определение таких последовательностей обслуживания требований каждым из приборов A и B, при которых общее время обслуживания всех требований будет наименьшим.

1.1. Начнем с рассмотрения случая, когда $N_{AB} = N$, т. е. все требования множества N сначала обслуживаются прибором A, затем прибором B.

Пусть прибором A требования обслуживаются в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, а прибором B — в последовательности $\pi' = (j_1, j_2, \ldots, j_n)$. По условию задачи прибор A может начать обслуживание требования i_1 в момент времени $d = 0$ и требования i_k — после завершения обслуживания требования i_{k-1}, $k = \overline{2, n}$.

Прибор B может начать обслуживание требования j_k, $k = \overline{2, n}$, если, во-первых, завершено обслуживание этого требования прибором A и, во-вторых, завершено обслуживание требования j_{k-1} прибором B. Что касается требования j_1, то оно может обслуживаться прибором B после завершения его обслуживания прибором A. Таким образом, $\underline{t}_{j_1B} \geqslant \bar{t}_{j_1A}$ и $\underline{t}_{j_kB} \geqslant \max(\bar{t}_{j_kA}, \bar{t}_{j_{k-1}B})$ при $k = \overline{2, n}$. Общее время обслуживания всех требований $T(\pi, \pi') = \bar{t}_{j_nB}$.

Величина $T(\pi, \pi')$ достигает наименьшего значения (при фиксированных π и π'), если во всех приведенных соотношениях знак нестрогого неравенства заменить знаком равенства. В дальнейшем именно это значение общего времени обслуживания будем обозначать через $T(\pi, \pi')$.

Покажем, что общее время обслуживания не увеличится, если потребовать, чтобы каждый прибор обслуживал требования непрерывно одно за другим. Это условие выполняется для прибора A, т. е. можно полагать $\underline{t}_{i_2A} = \bar{t}_{i_1A}, \underline{t}_{i_3A} = \bar{t}_{i_2A}, \ldots, \underline{t}_{i_nA} = \bar{t}_{i_{n-1}A}$.

Пусть $\underline{t}_{j_2B} = \bar{t}_{j_1B}, \ldots, \underline{t}_{j_lB} = \bar{t}_{j_{l-1}B}$ и $\underline{t}_{j_{l+1}B} = \bar{t}_{j_lB} + h$, где $h > 0$. Положим $\underline{t}'_{j_kB} = \underline{t}_{j_kB} + h$ и $\bar{t}'_{j_kB} = \bar{t}_{j_kB} + h$ для всех $k = \overline{1, l}$. Значение $\underline{t}_{j_{l+1}B}$ и, следовательно, величина $T(\pi, \pi')$ при этом не изменятся. С другой стороны, требования $j_1, j_2, \ldots, j_l$ могут быть начаты обслуживаться прибором B в моменты времени $\underline{t}'_{j_1B}, \underline{t}'_{j_2B}, \ldots, \underline{t}'_{j_lB}$ соответственно, поскольку $\underline{t}'_{j_1B} = \underline{t}_{j_1B} + h \geqslant \bar{t}_{j_1A} + h > \bar{t}_{j_1A}$ и $\underline{t}'_{j_kB} = \underline{t}_{j_kB} + h \geqslant \max(\bar{t}_{j_kA}, \bar{t}_{j_{k-1}B}) + h \geqslant \max(\bar{t}_{j_kA}, \bar{t}'_{j_{k-1}B})$ для всех $k = \overline{2, l}$. Применяя эту процедуру конечное число раз, получаем искомые значения $\underline{t}_{j_kB}$.

Таким образом, для определения величины $T(\pi, \pi')$ достаточно рассматривать расписания, при которых как прибор A, так и прибор B обслуживают требования множества N непрерывно одно за другим в последовательности π и π' соответственно. Естественно полагать, что прибор A начинает обслуживать требования в момент времени $d_A = d = 0$. Тогда $\underline{t}_{i_1A} = 0$, $\underline{t}_{i_2A} = \bar{t}_{i_1A}, \ldots, \underline{t}_{i_nA} = \bar{t}_{i_{n-1}A}$. Значения $\bar{t}_{i_kA} = \underline{t}_{i_kA} + t_{i_kA}$, $k = \overline{1, n}$. Если прибор B начинает обслуживать требования в момент времени d_B, то $\underline{t}_{j_1B} = d_B, \underline{t}_{j_2B} = \bar{t}_{j_1B}, \ldots, \underline{t}_{j_nB} = \bar{t}_{j_{n-1}B}$. Значения $\bar{t}_{j_kB} = \underline{t}_{j_kB} + t_{j_kB}$, $k = \overline{1, n}$. Следовательно, $T(\pi, \pi') = d_B + \sum_{k=1}^{n} t_{kB}$, где второе слагаемое не зависит от π и π'. Значение $d_B = d_B(\pi, \pi') = \max_{1 \leqslant u \leqslant n} (\Delta_u(\pi, \pi'))$. Здесь $\Delta_u(\pi, \pi') = \sum_{k=1}^{u} t_{i_kA} - \sum_{k=1}^{v-1} t_{j_kB}$ и значение v таково, что $j_v = i_u$.

Для вычисления d_B достаточно рассматривать только такие u и v, при которых наряду с выполнением условия $j_v = i_u$ выполняется следующее условие. Для любого k', $1 \leqslant k' \leqslant v$, существует k, $1 \leqslant k \leqslant u$, такое, что $j_{k'} = i_k$. Действительно, если u и v не удовлетворяют этому условию, т. е. среди элементов $j_1, j_2, \ldots, j_v$ существует элемент j_{k_1} такой, что $j_{k_1} = i_r$, $r > u$, то

$$\sum_{k=1}^{r} t_{i_kA} - \sum_{k=1}^{k_1-1} t_{j_kB} \geqslant \sum_{k=1}^{u} t_{i_kA} - \sum_{k=1}^{-1} t_{j_kB}. \qquad (1.1)$$

Покажем, что при любых π и π' имеет место соотношение $d_B(\pi, \pi) \leqslant d_B(\pi, \pi')$, а следовательно, и соотношение $T(\pi, \pi) \leqslant T(\pi, \pi')$. Пусть $\pi = (i_1, i_2, \ldots, i_n)$, $\pi' = (j_1 = i_1, \ldots, j_l = i_l, \quad j_{l+1} \neq i_{l+1}, \ldots, j_{l+p} =$

$= i_{l+1}, \ldots, j_n)$ и $\pi'' = (j'_1 = i_1, \ldots, j'_l = i_l, j'_{l+1} = i_{l+1}, j'_{l+2} = j_{l+1}, \ldots, j'_{l+p} = j_{l+p-1}, j'_{l+p+1} = j_{l+p+1}, \ldots, j'_n = j_n)$. Пусть $u_1, u_2, \ldots, u_\alpha$ — все те значения $1 \leqslant u \leqslant n$ и $v_1, v_2, \ldots, v_\alpha$ — соответствующие им значения $1 \leqslant v \leqslant n$, при которых $i_{u_k} = j_{v_k}$ и множество $\{j_1, j_2, \ldots, j_{v_k}\}$ содержится во множестве $\{i_1, i_2, \ldots, i_{u_k}\}$, $k = \overline{1, \alpha}$. Аналогично, пусть $u'_1, u'_2, \ldots, u'_{\alpha+1}$ — все те значения $1 \leqslant u' \leqslant n$ и $v'_1, v'_2, \ldots, v'_{\alpha+1}$ — соответствующие им значения $1 \leqslant v' \leqslant n$, при которых $i_{u'_k} = j'_{v'_k}$ и множество $\{j'_1, j'_2, \ldots, j'_{v'_k}\}$ содержится во множестве $\{i_1, i_2, \ldots, i_{u'_k}\}$, $k = \overline{1, \alpha + 1}$. Не нарушая общности, можно полагать $u_k = u'_k$ для всех $k = \overline{1, \alpha}$ и $u'_{\alpha+1} = l + 1$.

Поскольку $u_k = v_k = u'_k = v'_k = k$ при $1 \leqslant k \leqslant l$, то $\Delta_{u_k}(\pi, \pi') = \Delta_{u'_k}(\pi, \pi'')$ для всех $k = \overline{1, l}$. Если $u_k > l$, то $\Delta_{u_k}(\pi, \pi') - \Delta_{u'_k}(\pi, \pi'') = t_{i_{l+1}}$ и, следовательно, $\Delta_{u'_k}(\pi, \pi'') \leqslant \Delta_{u_k}(\pi, \pi')$ для всех $k = \overline{l + 1, \alpha}$. Наконец, $\Delta_{u'_{\alpha+1}}(\pi, \pi'') \leqslant \Delta_{u_{l+1}}(\pi, \pi')$. Следовательно, $d_B(\pi, \pi'') \leqslant d_B(\pi, \pi')$.

Если $\pi'' \neq \pi$, то аналогичные рассуждения можно провести относительно последовательности π''. В результате получим последовательность π''', у которой, по крайней мере, первые $l + 2$ элемента будут совпадать с пе выми $l + 2$ элементами исходной последовательности π и $d_B(\pi, \pi''') \leqslant d_B(\pi, \pi'')$. Повторяя эту процедуру конечное число раз, приходим к заключению, что $d_B(\pi, \pi) \leqslant d_B(\pi, \pi')$.

Таким образом, поиск оптимального по быстродействию расписания последовательного обслуживания n требований двумя приборами можно ограничить рассмотрением таких ситуаций, при которых каждый прибор обслуживает требования в одной и той же последовательности π непрерывно одно за другим. Если последовательность $\pi = (i_1, i_2, \ldots, i_n)$, то общее время обслуживания всех требований $T(\pi) = \max\limits_{1 \leqslant u \leqslant n} (\Delta_u(\pi)) + \sum\limits_{k=1}^{n} t_{kB}$, где

$$\Delta_u(\pi) = \sum_{k=1}^{u} t_{i_k A} - \sum_{k=1}^{u-1} t_{i_k B}.$$

Тем самым задача построения оптимального расписания сводится к нахождению перестановки π, которой соответствует наименьшее значение

$$d_B(\pi) = \max_{1 \leqslant u \leqslant n} \left[\sum_{k=1}^{u} t_{i_k A} - \sum_{k=1}^{u-1} t_{i_k B} \right]. \tag{1.2}$$

1.2. Воспользуемся перестановочным приемом. Пусть перестановка π_1 отличается от перестановки $\pi = (i_1, \ldots, i_{l-1}, i_l, i_{l+1}, i_{l+2}, \ldots, i_n)$ элементов множества N только транспозицией элементов i_l и i_{l+1}, т. е. $\pi_1 = (i_1, \ldots, i_{l-1}, i_{l+1}, i_l, i_{l+2}, \ldots, i_n)$. Найдем достаточные условия, при которых $d_B(\pi) \leqslant d_B(\pi_1)$.

Имеем $\Delta_u(\pi) = \Delta_u(\pi_1)$ для всех $u = \overline{1, l-1}$ и $u = \overline{l+2, n}$. Следовательно, если

$$\max \{\Delta_l(\pi), \Delta_{l+1}(\pi)\} \leqslant \max \{\Delta_l(\pi_1), \Delta_{l+1}(\pi_1)\}, \tag{1.3}$$

или что то же

$$\min (t_{i_l A}, t_{i_{l+1} B}) \leqslant \min (t_{i_{l+1} A}, t_{i_l B}), \tag{1.4}$$

то $d_B(\pi) \leqslant d_B(\pi_1)$.

Таким образом, если требования ν и μ обслуживаются непосредственно друг за другом и

$$\min (t_{\nu A}, t_{\mu B}) \leqslant \min (t_{\mu A}, t_{\nu B}), \tag{1.5}$$

то требование ν следует обслуживать первым.

Иными словами, в этом случае интервал $(0, \infty)$ является интервалом очередности типа $\nu \to \mu$.

Пусть требования ν, μ и γ таковы, что

$$\min (t_{\nu A}, t_{\mu B}) \leqslant \min (t_{\mu A}, t_{\nu B}) \tag{1.6}$$

и

$$\min (t_{\mu A}, t_{\gamma B}) \leqslant \min (t_{\gamma A}, t_{\mu B}). \tag{1.7}$$

Покажем, что тогда и

$$\min (t_{\nu A}, t_{\gamma B}) \leqslant \min (t_{\gamma A}, t_{\nu B}). \tag{1.8}$$

Действительно, если, например, $t_{\nu A} \leqslant t_{\mu B}$ и $t_{\mu A} \leqslant t_{\nu B}$, то из (1.6) следует, что $t_{\nu A} \leqslant t_{\mu A}$. Если $t_{\mu A} \leqslant t_{\gamma B}$ и $t_{\gamma A} \leqslant t_{\mu B}$, то из (1.7) $t_{\mu A} \leqslant t_{\gamma A}$. Следовательно, $t_{\nu A} \leqslant t_{\gamma A}$ и $t_{\nu A} \leqslant t_{\gamma B}$ и неравенство (1.8) справедливо как при $t_{\nu A} \leqslant t_{\gamma B}$, так и при $t_{\nu A} > t_{\gamma B}$.

Аналогичным образом можно убедиться в справедливости этого неравенства во всех остальных случаях.

Учитывая результаты § 2 гл. 3, получаем следующую теорему.

Теорема 1.1. *Общее время последовательного обслуживания требований двумя приборами достигает наименьшего значения, если каждый прибор обслуживает требования непрерывно в одной и той же последовательности* $\pi = (i_1, i_2, \ldots, i_n)$, *удовлетворяющей условию* (1.4) *при всех* $l = \overline{1, n-1}$.

Эта теорема позволяет предложить весьма простой алгоритм построения оптимального по быстродействию расписания обслуживания требований. Сопоставим каждому требованию $k \in N$ величину $\theta_k = \min(t_{kA}, t_{kB})$.

Среди всех требований множества N выберем требование (или одно из требований) $k = k_1$, которому соответствует наименьшее значение θ_k. Если $\theta_{k_1} = t_{k_1 A}$, то требование k_1 будем обслуживать первым как на приборе A, так и на приборе B. Если $\theta_{k_1} = t_{k_1 B}$, то требование k_1 будем обслуживать последним как на приборе A, так и на приборе B. Удалим это требование из рассмотрения.

Пусть упорядочено обслуживание r требований, при этом r_1 требований обслуживается первыми в некоторой последовательности $(i_1, i_2, \ldots, i_{r_1})$, а $r_2 = r - r_1$ требований — последними в порядке $(i_{n-r_2+1}, \ldots, i_n)$. Среди оставшихся требований выберем требование $k = k_{r+1}$ с наименьшим значением θ_k. Если $\theta_{k_{r+1}} = t_{k_{r+1}A}$, то требование k_{r+1} будем обслуживать непосредственно после требования i_{r_1}. Если $\theta_{k_{r+1}} = t_{k_{r+1}B}$, то требование k_{r+1} будем обслуживать непосредственно перед требованием i_{n-r_2+1}. В результате получим последовательность $\pi = (i_1, i_2, \ldots, i_n)$, удовлетворяющую условию теоремы. В качестве моментов времени начала и завершения обслуживания каждого требования на каждом приборе могут быть выбраны величины $\underline{t}_{i_1 A} = 0$, $\underline{t}_{i_k A} = \underline{t}_{i_{k-1} A} + t_{i_{k-1} A}$, $k = \overline{2, n}$, $\underline{t}_{i_1 B} = \max\limits_{1 \leqslant u \leqslant n} (\Delta_u(\pi))$, $\underline{t}_{i_k B} = \underline{t}_{i_{k-1} B} + t_{i_{k-1} B}$, $k = \overline{2, n}$, $\bar{t}_{i_k A} = \underline{t}_{i_k A} + t_{i_k A}$, $\bar{t}_{i_k B} = \underline{t}_{i_k B} + t_{i_k B}$. При этом как прибор A, так и прибор B будут обслуживать требования непрерывно одно за другим, прибор A начнет обслуживание требований в момент времени $d_A = 0$, а прибор B — в момент времени $d_B = t_{i_1B}$.

Для построения оптимальной последовательности обслуживания требований можно воспользоваться также следующим алгоритмом (см. п. 4.4 § 4 гл. 2).

Каждому требованию k поставим в соответствие величину

$$\omega(k) = \operatorname{sign}(t_{kA} - t_{kB})\,[M - \min(t_{kA}, t_{kB})], \qquad (1.9)$$

где $M > \max\limits_{1 \leqslant i \leqslant n} \min(t_{iA}, t_{iB})$, и упорядочим требования в порядке неубывания этих величин.

В общем случае, первый из приведенных алгоритмов порождает большее число оптимальных последовательностей, среди которых, естественно, содержатся все последовательности, порождаемые вторым алгоритмом.

Таблица 5.1.1

k	1	2	3	4	5	6
t_{kA}	2	6	8	4	4	2
t_{kB}	5	1	3	4	7	4

1.3. Пример. Необходимо построить оптимальную (по быстродействию) последовательность обслуживания шести требований двумя приборами A и B при условии, что каждое требование обслуживается сначала прибором A, затем прибором B. Совмещение процессов обслуживания одного и того же требования обоими приборами не допускается. Времена обслуживания требований каждым прибором приведены в табл. 5.1.1.

Используя первый из приведенных алгоритмов, получаем $\theta_1 = 2$, $\theta_2 = 1$, $\theta_3 = 3$, $\theta_4 = 4$, $\theta_5 = 4$, $\theta_6 = 2$.

Поскольку $\min\limits_{1 \leqslant k \leqslant 6} \theta_k = \theta_2 = t_{2B}$, то требование 2 можно обслуживать последним.

Среди оставшихся требований наименьшее значение θ_k соответствует требованиям 1 и 6, причем $\theta_1 = t_{1A}$ и $\theta_6 = t_{6A}$.

Первыми в оптимальной последовательности можно обслуживать требования 1 и 6 в произвольном порядке.

Среди еще не рассмотренных требований наименьшее значение θ_k соответствует требованию 3, причем $\theta_3 = t_{3B}$. Требование 3 можно обслуживать предпоследним.

Наконец, $\theta_4 = \theta_5 = 4$, причем $\theta_4 = t_{4A} = t_{4B}$, а $\theta_5 = t_{5A}$.

Получаем в результате четыре оптимальные последовательности $\pi_1 = (1, 6, 4, 5, 3, 2)$, $\pi_2 = (1, 6, 5, 4, 3, 2)$, $\pi_3 = (6, 1, 4, 5, 3, 2)$ и $\pi_4 = (6, 1, 5, 4, 3, 2)$. Общее время обслуживания равно 27.

Используя второй алгоритм, полагая, для определенности, $M = 10$, получаем $\omega(1) = -8$, $\omega(2) = 9$, $\omega(3) = 7$, $\omega(4) = 0$, $\omega(5) = -6$, $\omega(6) = -8$. Следовательно, оптимальными последовательностями являются π_2 и π_4.

Отметим, что в условиях этого примера существуют и другие оптимальные последовательности, например $\pi_5 = (1, 4, 6, 5, 3, 2)$.

1.4. Прежде чем перейти к рассмотрению общего случая, сделаем несколько замечаний.

Если $N_{BA} = N$, т. е. каждое требование сначала обслуживается прибором B, затем прибором A, то все утверждения предыдущего пункта остаются справедливыми, если приборы A и B поменять ролями. В частности, при обслуживании требований обоими приборами в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, удовлетворяющей для всех $l = \overline{1, n-1}$ условию

$$\min(t_{i_l B}, t_{i_{l+1} A}) \leqslant \min(t_{i_{l+1} B}, t_{i_l A}), \tag{1.10}$$

получаем наименьшее значение общего времени обслуживания всех требований.

Если хотя бы одно из множеств N_A и N_B непусто, то все эти требования можно отнести или ко множеству N_{AB}, или ко множеству N_{BA}, полагая $t_{kB} = 0$, если $k \in N_A$, и $t_{kA} = 0$, если $k \in N_B$.

Следовательно, построение оптимальной по быстродействию последовательности обслуживания требований не вызывает особых затруднений, если $N_{AB} = \phi$ или $N_{BA} = \phi$.

Пусть $N_{AB} \neq \phi$, $N_{BA} \neq \phi$, прибором A необходимо обслужить n_1 требований, прибором B необходимо обслужить n_2 требований. Обозначим через $T(\pi, \pi')$ наименьшее время обслуживания всех требований при условии, что прибор A обслуживает требования в последовательности $\pi = (i_1, i_2, \ldots, i_{n_1})$, а прибор B — в последовательности $\pi' = (j_1, j_2, \ldots, j_{n_2})$.

Последовательности π и π' не могут быть произвольными. Будем говорить, что последовательности π и π' непротиворечивы, если не существует такой пары требований $\nu \in N_{AB}$ и $\mu \in N_{BA}$, порядок обслуживания которых прибором A противоположен порядку их обслуживания прибором B и требование μ прибором A обслуживается первым.

Если π и π' непротиворечивы, то им соответствует бесконечно много расписаний процесса обслуживания требований, определяемых в рассматриваемом случае заданием моментов $\underline{t}_{kA}$ и $\underline{t}_{lB}$ времени начала обслуживания каждого требования каждым прибором.

Обозначим $N'_{AB} = N_{AB} \cup N_A$ и $N'_{BA} = N_{BA} \cup N_B$. Если $i_1 \in N'_{AB}$, то $\underline{t}_{i_1A} = d_A \geqslant 0$, в противном случае $\underline{t}_{i_1A} \geqslant \bar{t}_{i_1B}$. Если $j_1 \in N'_{BA}$, то $\underline{t}_{j_1B} = d_B \geqslant 0$, в противном случае $\underline{t}_{j_1B} \geqslant \bar{t}_{j_1A}$. Для всех $k = \overline{2, n_1}$ значение $\underline{t}_{i_kA} \geqslant \bar{t}_{i_{k-1}A}$, если $i_k \in N'_{AB}$, и $\underline{t}_{i_kA} \geqslant \max(\bar{t}_{i_{k-1}A}, \bar{t}_{i_kB})$, если $i_k \in N_{BA}$.

Аналогично, для всех $k = \overline{2, n_2}$ значение $\underline{t}_{j_kB} \geqslant \bar{t}_{j_{k-1}B}$, если $j_k \in N'_{BA}$, и $\underline{t}_{j_kB} \geqslant \max(\bar{t}_{j_{k-1}B}, \bar{t}_{j_kA})$, если $j_k \in N_{AB}$. Значения $\bar{t}_{i_kA} = \underline{t}_{i_kA} + t_{i_kA}$ для всех $k = \overline{1, n_1}$ и $\bar{t}_{j_kB} = \underline{t}_{j_kB} + t_{j_kB}$ для всех $k = \overline{1, n_2}$.

Набор значений $\underline{t}_{i_kA}$ и $\underline{t}_{j_kB}$, удовлетворяющих перечисленным неравенствам, определяет некоторое допустимое расписание. Общее время обслуживания требований равно $\max(\bar{t}_{i_{n_1}A}, \bar{t}_{j_{n_2}B})$ и достигает наименьшего значения (при фиксированных π и π'), если набор значений $\underline{t}_{i_kA}$ и $\underline{t}_{j_kB}$ удовлетворяет всем этим неравенствам как равенствам.

Пусть в последовательности $\pi = (i_1, i_2, \ldots, i_{n_1})$ требование $i_l \in N_{BA}$, требование $i_{l+1} \in N'_{AB}$ и $\hat{\pi} = (i_1, i_2, \ldots, i_{l-1}, i_{l+1}, i_l, i_{l+2}, \ldots, i_{n_1})$. Покажем, что $T(\hat{\pi}, \pi') \leqslant T(\pi, \pi')$.

Если $\underline{t}_{i_kA}$ и $\underline{t}_{j_kB}$ — моменты времени начала обслуживания требований при обслуживании их в последовательности π на приборе A и последовательности π' на приборе B, то длина временного интервала $(\underline{t}_{i_lA}, \bar{t}_{i_{l+1}A})$ не меньше $t_{i_lA} + t_{i_{l+1}A}$.

Рассмотрим расписание, при котором требование i_{l+1} начинает обслуживаться прибором A в момент времени $\underline{t}'_{i_{l+1}A} = \underline{t}_{i_lA}$, а требование i_l начинает обслуживаться в момент времени $\underline{t}'_{i_l A} = \underline{t}'_{i_l A} + t_{i_{l+1} A}$. Значения остальных $\underline{t}_{i_k A}$ и $\underline{t}_{j_k B}$ сохраняются неизменными.

Это расписание допустимо, поскольку $\bar{t}'_{i_{l+1} A} = \underline{t}'_{i_{l+1} A} + t_{i_{l+1} A} < \bar{t}_{i_{l+1} A} \leqslant \underline{t}_{i_{l+1} B}$, если $i_{l+1} \in N_{AB}$, и $\underline{t}'_{i_l A} > \underline{t}_{i_l A} \geqslant \bar{t}_{i_l B}$, если $i_l \in N_{BA}$. Общее время обслуживания требований при этом не больше $T(\pi, \pi')$.

Аналогично можно показать, что если $j_l \in N_{AB}$ и $j_{l+1} \in N'_{BA}$, то, обслуживая требования прибором B в последовательности $\hat{\pi}' = (j_1, j_2, \ldots, j_{l-1}, j_{l+1}, j_l, j_{l+2}, \ldots, j_{n_2})$, получаем $T(\pi, \hat{\pi}') \leqslant T(\pi, \pi')$.

Таким образом, в дальнейшем можно рассматривать только такие последовательности π и π' обслуживания требований, при которых прибор A сначала обслуживает требования множества N'_{AB}, затем — требования множества N_{BA}, а прибор B сначала обслуживает требования множества N'_{BA}, затем — требования множества N_{AB}.

Поскольку $N'_{AB} \cap N'_{BA} = \phi$, то прибор A может обслуживать требования множества N'_{AB} непрерывно одно за другим, начиная с момента времени $d_A = 0$, а прибор B может обслуживать требования множества N'_{BA} непрерывно одно за другим, также начиная с момента времени $d_B = 0$.

Аналогично п. 1.1 можно показать, что общее время обслуживания всех требований не увеличится, если потребовать, чтобы прибор A непрерывно обслуживал требования множества N_{BA}, начиная с некоторого момента времени $d'_A(\pi, \pi')$, а прибор B непрерывно обслуживал требования множества N_{AB}, начиная с некоторого момента времени $d'_B(\pi, \pi')$. Очевидно, $d'_A \geqslant \sum\limits_{k \in N'_{AB}} t_{kA}$ и $d'_B \geqslant \sum\limits_{k \in N'_{BA}} t_{kB}$. Общее время обслуживания всех требований $T(\pi, \pi') = $

$$= \max\left(d'_A(\pi, \pi') + \sum_{k \in N_{BA}} t_{kA},\ d'_B(\pi, \pi') + \sum_{k \in N_{AB}} t_{kB}\right).$$

Таким образом, решение исходной задачи сводится к рассмотрению двух однотипных задач оптимального (по быстродействию) упорядочения требований. В первой необходимо упорядочить обслуживание требований множества N'_{AB} (полагая $t_{kB} = 0$, если $k \in N_A$), во второй — обслуживание требований множества N'_{BA} (полагая $t_{kA} = 0$, если $k \in N_B$). Обе задачи относятся к классу задач, рассматриваемых в п. 1.2.

Отметим, что если $d'_A(\pi, \pi') = \sum\limits_{k \in N'_{AB}} t_{kA} + h$, $h > 0$, то $d'_B(\pi, \pi') = \sum\limits_{k \in N'_{BA}} t_{kB}$ и без увеличения $T(\pi, \pi')$ можно положить $d_A = h$. Аналогично, если $d'_B(\pi, \pi') = \sum\limits_{k \in N'_{BA}} t_{kB} + h$, $h > 0$, то $d'_A(\pi, \pi') = \sum\limits_{k \in N'_{AB}} t_{kA}$ и можно положить $d_B = h$. В обоих случаях каждый прибор будет обслуживать требования непрерывно одно за другим: прибор A, начиная с момента времени d_A, а прибор B, начиная с момента времени d_B. Тем самым доказана

Теорема 1.2. *Общее время обслуживания требований множества $N = N_{AB} \cup N_{BA} \cup N_A \cup N_B$ приборами A и B достигает наименьшего значения, если прибор A обслуживает требования в последовательности N_{AB}, N_A, N_{BA}, а прибор B — в последовательности N_{BA}, N_B, N_{AB}. Требования множества N_{AB} упорядочены согласно условию (1.4), требования множества N_{BA} — согласно условию (1.10), требования множеств N_A и N_B упорядочены произвольным образом. При этом каждый прибор может обслуживать требования непрерывно одно за другим.*

1.5. Пример. Рассмотрим задачу построения оптимального по быстродействию расписания обслуживания десяти требований, условия обслуживания которых приведены в табл. 5.1.2.

Упорядочим требования множества N_{AB} так, чтобы соответствующая последовательность $\pi(N_{AB})$ удовлетворяла условию (1.4). Используя алгоритм п. 1.2, получаем $\pi(N_{AB}) = (1, 4, 2, 3)$. Упорядочим требования множества N_{BA} так, чтобы соответствующая последовательность удовлетворяла условию (1.10). Получаем $\pi(N_{BA}) = (7, 6, 5)$.

Согласно теореме 1.2 прибор A может обслуживать требования в последовательности $\pi = (1, 4, 2, 3, 8, 7, 6, 5)$,

Таблица 5.1.2

	N_{AB}				N_{BA}			N_A	N_B	
k	1	2	3	4	5	6	7	8	9	10
t_{kA}	1	4	2	2	1	3	2	1	—	—
t_{kB}	3	2	1	3	2	4	1	—	2	3

а прибор B — в последовательности $\pi' = (7, 6, 5, (9, 10), 1, 4, 2, 3)$.

Если прибор A начинает обслуживание требований в момент времени d_A и обслуживает их непрерывно одно за другим, то обслуживание всех требований этот прибор завершает в момент времени $d_A + 16$. Аналогично, прибор B завершает обслуживание всех требований в момент времени $d_B + 21$, где d_B — момент времени начала непрерывного обслуживания требований этим прибором. Поскольку $\sum\limits_{k \in N_{AB}} t_{kA} = 9 < \sum\limits_{k \in N'_{BA}} t_{kB} = 12$ и $\sum\limits_{k \in N_{BA}} t_{kB} = 7 < \sum\limits_{k \in N'_{AB}} t_{kA} = 10$, то можно положить $d_A = d_B = 0$ и, следовательно, $T(\pi, \pi') = 21$.

§ 2. Параллельно-последовательное обслуживание

Задача построения оптимальных по быстродействию расписаний обслуживания требований двумя последовательными приборами в значительной степени усложняется, если допустить частичное совмещение во времени процессов обслуживания требований. На практике такого рода ситуации имеют место каждый раз, когда под требованием понимается, например, партия деталей, требующих обработки сначала на одном станке, затем на другом. В этом случае отдельная деталь из рассматриваемой партии деталей может, вообще говоря, обрабатываться на втором станке спустя некоторое время (связанное с ее транспортировкой и установкой) или непосредственно после завершения ее обработки на первом станке.

2.1. Обозначим через t_{kA} и t_{kB} время обслуживания требования $k=\overline{1, n}$ приборами A и B соответственно. Пусть каждое требование обслуживается, начиная с прибора A. Если требование k начинает обслуживаться прибором A в момент времени $\underline{t}_{kA}$, то момент времени начала его обслуживания прибором B $\underline{t}_{kB} \geqslant \underline{t}_{kA} + \xi_k$, где $\xi_k \geqslant 0$ — данное число. Будем предполагать также заданными наименьшие допустимые промежутки времени $\delta_{kl}^A \geqslant 0$ и $\delta_{kl}^B \geqslant 0$ между моментами времени начала обслуживания требований k и l приборами A и B соответственно при условии, что требование k начинает обслуживаться первым. Пусть $\tau_{kA} \geqslant 0$ и $\tau_{kB} \geqslant 0$ — заданные моменты времени, раньше которых не может быть начато обслуживание требования k приборами A и B соответственно.

Если положить $\xi_k = \delta_{kl}^A = t_{kA}$, $\delta_{kl}^B = t_{kB}$ и $\tau_{kA} = \tau_{kB} = 0$, $k, l = \overline{1, n}$, то получим ситуацию, описанную в предыдущем параграфе, п. 1.1.

Найдем наименьшее общее время $T(\pi, \pi')$ обслуживания всех требований при условии, что прибор A обслуживает требования в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, а прибор B — в последовательности $\pi' = (j_1, j_2, \ldots, j_n)$. По условию требование i_1 может быть начато обслуживаться прибором A в момент времени $\underline{t}_{i_1A} = \tau_{i_1A}$, требование i_2 — в момент времени $\underline{t}_{i_2A} = \max(\tau_{i_2A}, \underline{t}_{i_1A} + \delta_{i_1i_2}^A)$, требование i_3 — в момент времени $\underline{t}_{i_3A} = \max(\tau_{i_3A}, \underline{t}_{i_1A} + \delta_{i_1i_3}^A, \underline{t}_{i_2A} + \delta_{i_2i_3}^A)$ и т. д., наконец, требование i_n — в момент времени $\underline{t}_{i_nA} = \max(\tau_{i_nA}, \underline{t}_{i_1A} + \delta_{i_1i_n}^A, \ldots, \underline{t}_{i_{n-1}} + \delta_{i_{n-1}i_n}^A)$.

Самый ранний момент времени обслуживания требования j_1 прибором B равен $\underline{t}_{j_1B} = \max(\tau_{j_1B}, \underline{t}_{j_1A} + \xi_{j_1})$, требования j_2 равен $\underline{t}_{j_2B} = \max(\tau_{j_2B}, \underline{t}_{j_2A} + \xi_{j_2}, \underline{t}_{j_1B} + \delta_{j_1j_2}^B)$ и т. д., требования j_n равен $\underline{t}_{j_nB} = \max(\tau_{j_nB}, \underline{t}_{j_nA} + \xi_{j_n}, \underline{t}_{j_1B} + \delta_{j_1j_n}^B, \ldots, \underline{t}_{j_{n-1}B} + \delta_{j_{n-1}j_n}^B)$.

Величина $T(\pi, \pi') = \max\limits_{1 \leqslant k \leqslant n} (\bar{t}_{i_kA}, \bar{t}_{j_kB})$, где $\bar{t}_{i_kA} = \underline{t}_{i_kA} + t_{i_kA}$ и $\bar{t}_{j_kB} = \underline{t}_{j_kB} + t_{j_kB}$.

В общем случае последовательности π и π', которым соответствует наименьшее значение $T(\pi, \pi')$, различны. Тем самым при построении оптимального расписания необходимо рассмотреть $(n!)^2$ возможных вариантов с последующим выбором наилучшего.

2.2. Сформулируем достаточные условия, при которых неравенство $T(\pi, \pi) \leqslant T(\pi, \pi')$ выполняется для любых π и π'. Не нарушая общности рассуждений, будем полагать $\pi = (1, 2, \ldots, n)$ и $\pi' = (j_1, j_2, \ldots, j_n)$.

Если при любых $1 \leqslant k \neq l \neq r \leqslant n$ имеют место неравенства

$$\delta_{kl}^A \leqslant \delta_{rl}^A + \delta_{kr}^A \quad \text{и} \quad \tau_{lA} - \tau_{kA} \leqslant \delta_{kl}^A, \tag{2.1}$$

то значение

$$\underline{t}_{kA} = \tau_{1A} + \sum_{l=1}^{k-1} \delta_{l,l+1}^A. \tag{2.2}$$

Аналогично, если при любых $1 \leqslant k \neq l \neq r \leqslant n$ имеют место неравенства

$$\delta_{kl}^B \leqslant \delta_{kr}^B + \delta_{rl}^B \quad \text{и} \quad \tau_{kB} - \tau_{kA} \leqslant \xi_k, \tag{2.3}$$

то значение

$$\underline{t}_{j_k B} = \max_{1 \leqslant r \leqslant k} \left(\underline{t}_{j_r A} + \xi_{j_r} + \sum_{l=r}^{k-1} \delta_{j_l j_{l+1}}^B \right). \tag{2.4}$$

Следовательно, при выполнении условий (2.1) и (2.3) имеем

$$T(\pi, \pi') = \max_{1 \leqslant r \leqslant k \leqslant n} \left(\tau_{1A} + \sum_{l=1}^{k-1} \delta_{l,l+1}^A + t_{kA},\ \tau_{1A} + \right. \\ \left. + \sum_{l=1}^{j_r-1} \delta_{l,l+1}^A + \xi_{j_r} + \sum_{l=r}^{k-1} \delta_{j_l, j_{l+1}}^B + t_{jkB} \right). \tag{2.5}$$

Потребуем, чтобы при любых $1 \leqslant k \neq l \leqslant n$ выполнялись неравенства

$$t_{kA} - t_{kB} \leqslant \xi_k \quad \text{и} \quad t_{kB} - t_{lB} \leqslant \delta_{kl}^B. \tag{2.6}$$

Тогда

$$T(\pi, \pi') = \max_{1 \leqslant r \leqslant n} \left(\tau_{1A} + \sum_{l=1}^{j_r-1} \delta_{l,l+1}^A + \xi_{j_r} + \sum_{l=r}^{n-1} \delta_{j_l, j_{l+1}}^B + t_{j_n B} \right). \tag{2.7}$$

Теорема 2.1. *Если при параллельно-последовательном обслуживании требований двумя приборами выполняются условия* (2.1), (2.3), (2.6), $\delta_{kl}^B = t_{kB}$ *и* $\xi_k - \xi_l \leqslant \delta_{kl}^A$ *для всех* $1 \leqslant k \neq l \leqslant n$, *то* $T(\pi, \pi) \leqslant T(\pi, \pi')$, *где* π *и* π' — *произвольные последовательности обслуживания требований приборами* A *и* B *соответственно.*

Доказательство. Не нарушая общности рассуждений, по-прежнему будем полагать $\pi = (1, 2, \ldots, n)$. Пусть $\pi' = (j_1, \ldots, j_{p-1}, j_p, \ldots, j_{p+u}, \ldots, j_n)$, где $j_l = l$, $l = \overline{1, p-1}$, $j_p \neq p$, $j_{p+u} = p$. Обозначим $\tilde{\pi}' = (1, 2, \ldots, p-1, p, j_p, \ldots, j_{p+u-1}, j_{p+u+1}, \ldots, j_n)$. Для доказательства теоремы достаточно показать, что $T(\pi, \tilde{\pi}') \leqslant T(\pi, \pi')$.

В рассматриваемом случае

$$T(\pi, \pi') = \max_{1 \leqslant r \leqslant n} \left(\tau_{1A} + \sum_{l=1}^{j_r - 1} \delta_{l,\, l+1}^A + \xi_{j_r} + \sum_{l=r}^{n} t_{j_l B} \right). \quad (2.8)$$

Аналогично можно записать выражение для $T(\pi, \tilde{\pi}')$. По построению единственной компонентой в выражении для $T(\pi, \pi')$, значение которой увеличивается при переходе к последовательности $\tilde{\pi}'$, является компонента с $r = p$.

По предположению $\xi_k - \xi_l \leqslant \delta_{kl}^A$ для всех $1 \leqslant k \neq l \leqslant n$ и, следовательно, $\xi_p - \xi_{j_p} \leqslant \sum_{l=p}^{j_p - 1} \delta_{l,\, l+1}$. Таким образом, среди компонент в выражении для $T(\pi, \pi')$ существует компонента с $r = j_p$, значение которой не меньше значения компоненты с $r = p$ в выражении для $T(\pi, \tilde{\pi}')$. Теорема доказана.

Эта теорема позволяет в некоторых случаях ограничить поиск оптимального по быстродействию расписания процесса обслуживания требований двумя приборами рассмотрением только таких расписаний, при которых требования каждым прибором обслуживаются в одной и той же последовательности.

2.3. В качестве примера рассмотрим задачу оптимального упорядочения обслуживания n требований, каждое k-е из которых сначала обслуживается прибором A в течение t_{kA} единиц времени, затем прибором B в течение t_{kB} единиц времени. В каждый момент времени как прибор A, так и прибор B обслуживает не более одного требования. Предполагается, что между моментами времени начала обслуживания этого требования приборами A и B должно пройти не менее $\xi_k \geqslant 0$ единиц времени. Если $\xi_k < t_{kA}$, то процессы обслуживания требования k двумя приборами могут совмещаться во времени. Если $\xi_k \geqslant t_{kA}$, то эти процессы не могут совмещаться во времени, а при $\xi_k > t_{kA}$ между моментом завершения обслуживания требования k прибором A и моментом времени начала его обслуживания прибором B должно пройти не менее $\xi_k - t_{kA}$ единиц времени. В последнем случае величина $\xi_k - t_{kA}$ может означать, например, время, необходимое для транспортировки или передачи требования k с прибора A на прибор B.

Пусть $n = 2$, $t_{1A} = t_{2A} = t_{2B} = 3$, $t_{1B} = 2$, $\xi_1 = 8$, $\xi_2 = 2$. Если обслуживать требования в одной и той же последовательности обоими приборами, то общее время их обслуживания не менее 13. Если прибором A обслуживать сначала требование 1, затем требование 2, а прибором B — сначала требование 2, а затем требование 1, то наименьшее общее время обслуживания равно 10. Таким образом, в условиях рассматриваемой задачи при построении оптимального по быстродействию расписания следует, вообще говоря, перебрать $(n!)^2$ возможных последовательностей обслуживания требований.

Используя приведенную теорему, можно сформулировать достаточно простые и часто реализуемые на практике условия, при выполнении которых нет необходимости рассматривать последовательности обслуживания требований прибором B, отличающиеся от последовательности их обслуживания прибором A. Действительно, поскольку $\tau_{kA} = \tau_{kB} = 0$, $\delta_{kl}^A = t_{kA}$, $\delta_{kl}^B = t_{kB}$, $1 \leqslant k \neq l \leqslant n$, то $T(\pi, \pi) \leqslant T(\pi, \pi')$ для любых π и π', если $-\xi_l \leqslant t_{kA} - \xi_k \leqslant t_{kB}$ для всех $1 \leqslant k \neq l \leqslant n$.

Эти условия содержательно означают, что если процессы обслуживания одного и того же требования k двумя приборами могут совмещаться во времени (случай $\xi_k < t_{kA}$), то процесс обслуживания этого требования прибором A должен завершиться во всяком случае не позднее, чем процесс его обслуживания прибором B. Если прибор B не может начинать обслуживание требования k непосредственно после завершения процесса обслуживания этого требования прибором A (случай $\xi_k > t_{kA}$), то заданное минимальное время $\xi_k - t_{kA}$ транспортировки (пролеживания) требования k не должно превышать значений ξ_l остальных требований. В частности, если для всех требований $\xi_k \geqslant t_{kA}$, то время транспортировки каждого отдельного требования должно быть не больше времени обслуживания прибором A любого из остальных требований.

2.4. Опишем отдельные классы задач, решение которых требует просмотра и оценки сравнительно небольшого числа конкурентноспособных вариантов.

На двух машинах A и B в порядке AB требуется обработать n комплектов деталей (в комплект входят одинаковые детали). Каждый комплект деталей обрабатывается как машиной A, так и машиной B непрерывно. Время обработки k-го комплекта деталей на машинах A и B равно t_{kA} и t_{kB}, а время обработки одной детали этого комплекта θ_{kA} и θ_{kB} соответственно. Детали обрабатываются каждой машиной в одной и той же последовательности, причем одновременно каждая машина обрабатывает не более одной детали. Требуется найти такой порядок запуска комплектов деталей в обработку, при котором общее время обработки всех деталей наименьшее.

В рассматриваемом случае $\delta_{kl}^A = t_{kA}$, $\delta_{kl}^B = t_{kB}$, $\tau_{kA} = \tau_{kB} = 0$, $\xi_k = \max(\theta_{kA}, \ t_{kA} + \theta_{kB} - t_{kB})$, $1 \leqslant k \neq l \leqslant n$. Если комплекты деталей на каждой машине обрабатываются в одной и той же последовательности $\pi = (i_1, i_2, \ldots, i_n)$ (i_k — номер комплекта, обрабатываемого k-м по порядку), то наименьшее общее время обработки всех деталей

$$T(\pi) = \max_{1 \leqslant u < n} \left[\sum_{k=1}^{u} t_{i_k A} + \max(\theta_{i_u B}, \ t_{i_u B} + \theta_{i_u A} - t_{i_u A}) + \sum_{k=u+1}^{n} t_{i_k B} \right]. \qquad (2.9)$$

Таким образом, задача сводится к нахождению последовательности π, которой соответствует наименьшее значение функции

$$\widetilde{T}(\pi) = \max_{1 \leqslant u < n} \left[\sum_{k=1}^{u} (t_{i_k A} - t_{i_k B}) + \max(\theta_{i_u B}, \ t_{i_u B} + \theta_{i_u A} - t_{i_u A}) \right]. \qquad (2.10)$$

Полагая $\alpha_k = t_{kA} - t_{kB}$, $\beta_k = \max(\theta_{kB}, t_{kB} + \theta_{kA} - t_{kA})$ для всех $k = \overline{1, n}$, на основании теоремы 4.2 гл. 2 комплекты следует запускать в обработку в порядке неубывания их весов:

$$\omega(k) = \operatorname{sign} \alpha_k [M - \min(\alpha_k + \beta_k, \beta_k)].$$

Отметим, что общее время обработки всех деталей, вообще говоря, можно уменьшить, если эти детали обрабатывать на машине B не обязательно в той же последовательности, что и на машине A. Однако и в этом случае указанное правило приводит к получению оптимального по быстродействию расписания, если для всех $1 \leqslant k \neq \neq l \leqslant n$ выполняются неравенства

$$\max(\theta_{kA} - t_{kA}, \theta_{kB} - t_{kB}) \leqslant t_{lA} + \max(\theta_{lA} - t_{lA}, \theta_{lB} - t_{lB}), \tag{2.11}$$

что непосредственно следует из теоремы 2.1.

2.5. Аналогично решается задача групповой обработки деталей на двух взаимосвязанных рабочих местах A и B. Эта задача состоит в определении такого порядка запуска в обработку групп и подгрупп каждой группы деталей, при котором общее время обработки всех деталей наименьшее. Число групп n; каждая группа k состоит из n_k подгрупп деталей $k_1, k_2, \ldots, k_{n_k}$. Времена обработки групп и подгрупп деталей на рабочих местах A и B известны и равны соответственно $\tilde{t}_{kA}$, $\tilde{t}_{kB}$, t_{k_iA} и t_{k_iB}.

При определении очередности запуска в обработку подгрупп деталей k-й группы задача сводится к определению последовательности $\pi_k = (j_1, j_2, \ldots, j_{n_k})$, которой соответствует наименьшее значение функции

$$L_k(\pi_k) = \max_{1 \leqslant u \leqslant n_k} \sum_{l=1}^{u} (t_{k_{j_l}A} - t_{k_{j_l}B}). \tag{2.12}$$

Пусть $\Delta_k = \max[0, \min\limits_{\pi_k} L_k(\pi_k)]$. Для определения последовательности запуска в обработку групп деталей достаточно найти последовательность $\pi = (p_1, p_2, \ldots, p_n)$, которой соответствует наименьшее значение функции

$$L(\pi) = \max_{1 \leqslant u \leqslant n} \left[\sum_{k=1}^{u-1} (\tilde{t}_{p_kA} - \tilde{t}_{p_kB}) + \Delta_{p_u}\right]. \tag{2.13}$$

Используя теорему 4.2 гл. 2, получаем следующее правило оптимального упорядочения. Каждой группе k деталей приписать вес $\omega(k) = \operatorname{sign}(\tilde{t}_{kA} - \tilde{t}_{kB})\,[M - \min(\Delta_k,\ \tilde{t}_{kB} - \tilde{t}_{kA} + \Delta_k)]$, каждой подгруппе k_i — вес $\omega(k_i) = \operatorname{sign}(t_{k_iA} - t_{k_iB})\,(M - \min(0,\ t_{k_iA} - t_{k_iB})]$.

Каждую группу деталей и каждую подгруппу в группе запускать в обработку в порядке неубывания весов.

2.6. Рассмотрим несколько более общую, по сравнению с описанными в предыдущем пункте, задачу оптимальной организации параллельно-последовательного обслуживания требований.

Пусть выполняются условия теоремы 2.1 и, кроме того, $\delta_{kl}^A = \delta_k^A$, $1 \leqslant k \neq l \leqslant n$. Общее время обслуживания всех требований достигает наименьшего значения, если требования обслуживаются обоими приборами в одной и той же последовательности. Наименьшее время обслуживания всех требований при условии, что эти требования обслуживаются в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, равно

$$T(\pi) = \max_{1 \leqslant u \leqslant n}\left(\tau_{i_1A} + \sum_{k=1}^{u-1}\delta_{i_k}^A + \xi_{i_u} + \sum_{k=u}^{n} t_{i_kB}\right). \quad (2.14)$$

Если известно, какое именно требование в последовательности π обслуживается первым (пусть, например, $i_1 = v$), то задача нахождения оптимальной по быстродействию последовательности обслуживания требований сводится к задаче минимизации функции

$$\hat{T}(\pi_{i_1=v}) = \max_{2 \leqslant u \leqslant n}\left[\sum_{k=2}^{u}(\delta_{i_k}^A - t_{i_kB}) + \xi_{i_u} - \delta_{i_u}^A + t_{i_uB}\right], \quad (2.15)$$

где $\pi_{i_1=v}$ — перестановка элементов множества $N = \{1, 2, \ldots, n\}$ вида $(v, i_2, i_3, \ldots, i_n)$.

Полагая $\alpha_k = \delta_k^A - t_{kB}$, $\beta_k = \xi_k - \delta_k^A + t_{kB}$ и используя теорему 4.2 гл. 2, получаем следующее правило оптимального упорядочения. Каждому требованию k приписать вес

$$\omega(k) = \operatorname{sign}(\delta_k^A - t_{kB})\,[M - \min(\xi_k,\ \xi_k - \delta_k^A + t_{kB})].$$

Вычислить наименьшее общее время обслуживания требований при условии, что первым обслуживается требование $v = 1, 2, \ldots, n$, а остальные требования обслуживаются в порядке неубывания их весов. В качестве искомой следует выбрать ту из рассмотренных n перестановок $\pi_{i_1=v}$, которой соответствует наименьшее значение $\hat{T}(\pi_{i_1=v})$.

2.7. Рассмотрим конкретный пример, в котором требуется упорядочить обслуживание пяти требований двумя

Таблица 5.2.1

k	τ_{kA}	τ_{kB}	δ_k^A	$\delta_k^B = t_{kB}$	ξ_k	t_{kA}	$\omega(k)$
1	0	5	6	1	8	5	7
2	2	0	2	4	3	3	—7
3	0	3	5	2	5	6	8
4	1	2	8	3	2	5	13
5	0	4	3	4	6	4	—4

приборами A и B. Все параметры и значения весов этих требований приведены в табл. 5.2.1.

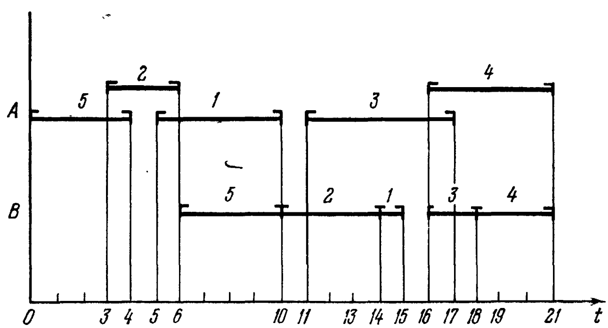

Рис. 5.2.1.

Непосредственной проверкой можно убедиться, что условия теоремы 2.1 выполняются и $\delta_{kl}^A = \delta_k^A$ для всех $1 \leqslant k \neq l \leqslant n$.

Следовательно, достаточно рассмотреть пять перестановок $\pi_{i_1=1} = (1, 2, 5, 3, 4)$, $\pi_{i_1=2} = (2, 5, 1, 3, 4)$,

$\pi_{i_1=3} = (3,\ 2,\ 5,\ 1,\ 4)$, $\pi_{i_1=4} = (4,\ 2,\ 5,\ 1,\ 3)$ и $\pi_{i_1=5} = (5,\ 2,\ 1,\ 3,\ 4)$. Соответствующие значения $\hat{T}\ (\pi_{i_1=v})$ равны 23, 23, 22, 30 и 21.

Таким образом, требования необходимо обслуживать в последовательности (5, 2, 1, 3, 4). Общее время обслуживания при этом равно 21. График соответствующего расписания приведен на рис. 5.2.1.

§ 3. Библиографическая справка

Теорема 1.1 известна в литературе как теорема Беллмана — Джонсона [236, 237, 313]. Теорема 1.2 доказана Дж. Джексоном [310, 109, 224], теорема 2.1 — С. А. Баркан [12].

Изучению параллельно-последовательных процессов обслуживания посвящены работы Л. Дж. Миттэна [350], В. Шварца [416], И. Набешимы [359], С. М. Джонсона [314], Ф. И. Парамонова [132]. Обобщение этих результатов содержится в работах В. С. Танаева [170] и Е. В. Левнера [95].

Алгоритм определения всех оптимальных расписаний описан в [73], классификация задач Беллмана — Джонсона с двумя последовательными приборами приведена в [99].

В. Я. Бурдюк и Т. А. Мазня [25], Б. А. Власюк [35] рассмотрели естественное обобщение задачи двухстадийного обслуживания в предположении, что на каждой стадии имеется несколько параллельно работающих приборов.

В [223] приведен алгоритм построения последовательности обслуживания, минимизирующей величину $\sum_{i=1}^{n} w_i/p_i$, где w_i — общее время пролеживания i-го требования в ожидании обслуживания, p_i — общее время его обслуживания на обоих приборах. Следует отметить также работы [225, 399].

В работах [24, 321, 337] рассматриваются стохастические варианты задачи двухстадийного последовательного обслуживания.

ГЛАВА 6

M ПОСЛЕДОВАТЕЛЬНЫХ ПРИБОРОВ. ОДИНАКОВЫЕ МАРШРУТЫ

Рассматривается задача построения оптимального по быстродействию расписания последовательного обслуживания n требований M приборами. Каждое требование обслуживается сначала прибором 1, затем прибором 2 и т. д., наконец, прибором M. Время обслуживания требования k прибором L равно t_{kL}. Предполагается, что все требования обслуживаются каждым прибором в одной и той же последовательности. Каждый прибор обслуживает одновременно не более одного требования и приступает к обслуживанию каждого очередного требования после завершения предыдущего без неоправданного простоя.

В § 1 приводятся формулы расчета общего времени обслуживания всех требований при различных допущениях относительно характера их обслуживания. В § 2 описывается один из вариантов метода ветвей и границ. В § 3 рассматриваются специфические условия, позволяющие исключить из рассмотрения заведомо бесперспективные варианты обслуживания. В § 4 исследуются некоторые частные случаи рассматриваемой задачи.

§ 1. Общие замечания

1.1. Задача построения оптимального (по быстродействию) расписания последовательного обслуживания n требований M приборами существенно усложняется при переходе от $M = 2$ приборов к числу приборов $M > 2$.

В этом случае последовательность обслуживания требований одним прибором при оптимальном расписании, вообще говоря, отличается от последовательности их обслуживания другими приборами.

Однако можно показать, что существует оптимальное расписание, при котором прибором 1 требования обслуживаются в той же последовательности, что и прибором 2, а прибором M в той же последовательности, что и прибором $M - 1$. Поиск оптимального расписания, таким образом, можно ограничить рассмотрением $(n!)^{M-2}$ вариантов обслуживания требований. В частности, если $M = 3$, то оптимальное расписание можно искать в классе расписаний, при которых каждый прибор обслуживает требования в одной и той же последовательности.

Таблица 6.1.1

$k \backslash L$	1 (A)	2 (B)	3 (C)	4 (D)
1	2	4	4	1
2	4	2	1	4
3	5	2	1	1

Если $M \geqslant 4$, то, как показывает простой пример (значения t_{kL} приведены в табл. 6.1.1), оптимальным расписанием может быть расписание, при котором приборы обслуживают требования в *разных* последовательностях. В рассматриваемом случае приборы 1 и 2 должны обслуживать требования в последовательности (1, 2, 3), а приборы 3 и 4 — в последовательности (2, 1, 3). График этого расписания приведен на рис. 6.1.1.

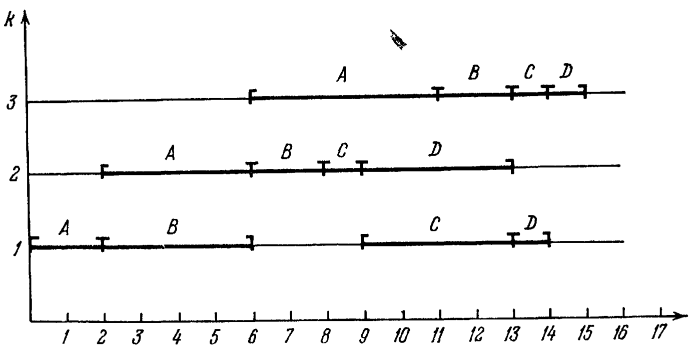

Рис. 6.1.1.

В отличие от рассмотренного в гл. 5, § 1, п. 1.1 случая $M = 2$ в общем случае условие непрерывности обслуживания требования каждым прибором (за исключением приборов 1 и M) может привести к увеличению общего времени их обслуживания.

Мы ограничимся рассмотрением ситуаций, в которых все требования обслуживаются каждым прибором в одной и той же последовательности. Если при этом потребовать, чтобы каждый прибор приступал к обслуживанию очередного поступающего к нему требования сразу после завершения обслуживания предыдущего требования, то расписание однозначно определяется заданием последовательности $\pi = (i_1, i_2, \ldots, i_n)$ обслуживания требований.

1.2. Если требования обслуживаются каждым прибором в последовательности $\pi = (i_1, i_2, \ldots, i_n)$, то момент времени начала обслуживания требования i_1 прибором 1 равен $\underline{t}_{i_1 1} = 0$, а требования i_k равен $\underline{t}_{i_k 1} = \bar{t}_{i_{k-1} 1}$, $k = \overline{2, n}$. Аналогично, $\underline{t}_{i_1 L} = \bar{t}_{i_1\, L-1}$ и $\underline{t}_{i_k L} = \max(\bar{t}_{i_k\, L-1}, \bar{t}_{i_{k-1} L})$, $k = \overline{2, n}$, $L = \overline{2, M}$. Здесь $\bar{t}_{i_k L} = \underline{t}_{i_k L} + t_{i_k L}$ — момент времени завершения обслуживания требования i_k прибором L.

Обозначим через $T(\pi)$ наименьшее общее время обслуживания всех требований при обслуживании их каждым прибором в последовательности π. Очевидно, $T(\pi) = \bar{t}_{i_n M}$.

Зависимость $T(\pi)$ от t_{kL} и π можно записать в виде

$$T(\pi) = \max_{1 \leqslant u_1 \leqslant u_2 \leqslant \ldots \leqslant u_{M-1} \leqslant n} \left[\sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} + \ldots + \sum_{k=u_{M-1}}^{n} t_{i_k M} \right]. \quad (1.1)$$

Нас будут интересовать вопросы, связанные с построением последовательности π^*, которой соответствует наименьшее значение $T(\pi)$.

1.3. Определенный практический интерес представляют ситуации, в которых необходимо соблюдать одно из двух дополнительных условий — каждый прибор должен обслуживать требования непрерывно одно за другим или каждое требование должно обслуживаться непрерывно одним прибором за другим.

В первом случае общее время обслуживания равно

$$T'(\pi) = \sum_{L=1}^{M-1} \max_{1 \leqslant u \leqslant n} \left[\sum_{k=1}^{u} t_{i_k L} - \sum_{k=1}^{u-1} t_{i_k L+1} \right] + \sum_{k=1}^{n} t_{i_k M}. \quad (1.2)$$

Во втором случае общее время обслуживания равно

$$T''(\pi) = \sum_{k=1}^{n-1} \max_{1\leqslant U\leqslant M} \left[\sum_{L=1}^{U} t_{i_k L} - \sum_{L=1}^{U-1} t_{i_{k+1} L}\right] + \sum_{L=1}^{M} t_{i_n L}. \quad (1.3)$$

Величину $T''(\pi)$ можно рассматривать в качестве длины гамильтонова контура $(0, \pi, 0)$ в полном графе с вершинами $\{0, 1, \ldots, n\}$, если длины l_{ij} его дуг (i, j) положить равными $l_{0j} = 0$, $l_{i0} = \sum_{L=1}^{M} t_{iL}$, $l_{ij} = \max_{1\leqslant U\leqslant M} \left[\sum_{L=1}^{U} t_{iL} - \sum_{L=1}^{U-1} t_{jL}\right]$. Следовательно, для построения последовательности с наименьшим значением $T''(\pi)$ достаточно решить соответствующую задачу коммивояжера.

§ 2. Конструктивный подход

В основе методов последовательного конструирования, анализа и отсеивания вариантов лежит идея пошагового построения решения. Если при таком последовательном конструировании на основании некоторых свойств решения можно ввести понятия «доминирования» одних вариантов над другими, «перспективности» одних вариантов перед другими, оперируя отдельными, построенными частями этих вариантов, то тогда удается разработать простые вычислительные схемы отыскания оптимального варианта.

В этом параграфе мы рассмотрим одну из схем последовательного конструирования оптимальной последовательности π^* обслуживания n требований множества $N = \{1, 2, \ldots, n\}$ M приборами. Эта схема получила название *схемы ветвей и границ*.

2.1. Процесс построения искомой последовательности можно представить как процесс пошагового «развития» частичной (возможно пустой) последовательности σ обслуживания некоторых требований из N до полной последовательности π обслуживания всех требований N. Подобное «развитие» может проводиться по различным «направлениям» путем приписывания справа различных элементов из N.

Естественно, что дальнейшему развитию должна подвергаться та из рассматриваемых на данном шаге частичных последовательностей, развитие которой представляется наиболее перспективным. Способы оценки перспективности частичных последовательностей в рассматриваемом случае весьма разнообразны. Они отличаются друг от друга как по сложности, так и по точности вычисления оценок. В схемах последовательного анализа вариантов выбор того или иного метода обусловлен двумя противоречивыми обстоятельствами. Во-первых, уточнение оценок позволяет обычно сократить процесс ветвления. Во-вторых, всякое уточнение сопряжено с увеличением объема вычислений, что может привести к компенсации выигрыша, получаемого от сокращения процесса ветвления.

Мы приведем описание одного из методов вычисления оценок для случая $M = 3$. Обобщение на случай $M > 3$ не вызывает особых затруднений и может быть проведено читателем самостоятельно.

Обозначим приборы через A, B и C. Каждое требование сначала обслуживается прибором A в течение a_k единиц времени, затем прибором B в течение b_k единиц времени, наконец, прибором C в течение c_k единиц времени.

Если требования множества $\tilde{N} \subset N$ обслуживаются в последовательности $\sigma = (i_1, i_2, \ldots, i_r)$, $r = |\tilde{N}|$, а остальные требования обслуживаются в последовательности $\sigma' = (i_{r+1}, i_{r+2}, \ldots, i_n)$, то общее время обслуживания всех требований равно

$$T(\sigma, \sigma') = \max_{1 \leqslant u_1 \leqslant u_2 \leqslant n} \left[\sum_{k=1}^{u_1} a_{i_k} + \sum_{k=u_1}^{u_2} b_{i_k} + \sum_{k=u_2}^{n} c_{i_k} \right]. \qquad (2.1)$$

Прибор A завершает обслуживание требований множества $\tilde{N}$ в момент времени

$$T_A(\sigma) = \sum_{k=1}^{r} a_{i_k}. \qquad (2.2)$$

Прибор B завершает обслуживание требований множества $\tilde{N}$ в момент времени

$$T_B(\sigma) = \max_{1 \leqslant u \leqslant r} \left[\sum_{k=1}^{u} a_{i_k} + \sum_{k=u}^{r} b_{i_k} \right]. \qquad (2.3)$$

Прибор C завершает обслуживание требований множества $\tilde{N}$ в момент времени

$$T_C(\sigma) = \max_{1 \leqslant u_1 \leqslant u_2 \leqslant r} \left[\sum_{k=1}^{u_1} a_{i_k} + \sum_{k=u_1}^{u_2} b_{i_k} + \sum_{k=u_2}^{r} c_{i_k} \right]. \qquad (2.4)$$

Определим оценку $\gamma_{\text{пр}}(\sigma)$ последовательности σ (по приборам), полагая

$$\gamma_{\text{пр}}(\sigma) = \max \begin{cases} T_A(\sigma) + \sum\limits_{k \in N \setminus \tilde{N}} a_k + \min\limits_{k \in N \setminus \tilde{N}} (b_k + c_k), \\ T_B(\sigma) + \sum\limits_{k \in N \setminus \tilde{N}} b_k + \min\limits_{k \in N \setminus \tilde{N}} c_k, \\ T_C(\sigma) + \sum\limits_{k \in N \setminus \tilde{N}} c_k. \end{cases} \qquad (2.5)$$

Очевидно, $\gamma_{\text{пр}}(\sigma) \leqslant T(\sigma, \sigma')$.

Аналогично определим оценку $\gamma_{\text{тр}}(\sigma)$ последовательности σ (по требованиям), полагая

$$\gamma_{\text{тр}}(\sigma) = \max \begin{cases} T_A(\sigma) + \max\limits_{i \in N \setminus \tilde{N}} \left[a_i + b_i + c_i + \sum\limits_{\substack{k \in N \setminus \tilde{N} \\ k \neq i}} \min(a_k, c_k) \right], \\ T_B(\sigma) + \max\limits_{i \in N \setminus \tilde{N}} \left[b_i + c_i + \sum\limits_{\substack{k \in N \setminus \tilde{N} \\ k \neq i}} \min(b_k, c_k) \right], \\ T_C(\sigma) + \sum\limits_{i \in N \setminus \tilde{N}} c_i. \end{cases} \qquad (2.6)$$

Очевидно, $\gamma_{\text{тр}}(\sigma) \leqslant T(\sigma, \sigma')$.

Таким образом, в качестве нижней оценки $\gamma(\sigma)$ значений $T(\sigma, \sigma')$ при различных σ' может быть выбрана любая из величин $\gamma_{\text{пр}}(\sigma)$ или $\gamma_{\text{тр}}(\sigma)$. При вычислении обеих этих величин в качестве $\gamma(\sigma)$ естественно выбрать наибольшую из них.

В результате некоторого усложнения формул вычисления $\gamma_{\text{пр}}(\sigma)$ и $\gamma_{\text{тр}}(\sigma)$ могут быть получены более точные нижние оценки $\gamma_*(\sigma)$ значений $T(\sigma, \sigma')$.

Конструирование оптимальной последовательности включает построение частичных последовательностей, оценку этих последовательностей, выбор для последующего «развития» наиболее перспективной частичной последовательности и ее «развитие» по возможным направлениям.

Этот процесс является, по существу, процессом последовательного разбиения множества всех $n!$ возможных перестановок на подмножества и вычисления нижних оценок значений оптимизируемой функции на каждом из этих подмножеств. При этом к одному подмножеству относятся все перестановки, начинающиеся с одной и той же частичной перестановки.

Таблица 6.2.1

k	1	2	3	4	5
a_k	5	4	7	2	3
b_k	8	4	1	8	5
c_k	1	9	4	4	1

Конструирование оптимальной последовательности завершается наступлением ситуации, в которой окажется построенной некоторая перестановка π^* и значение $T(\pi^*)$ будет не больше оценок $\gamma(\sigma)$ для всех рассматриваемых на данном шаге множеств перестановок вида (σ, σ').

2.2. Пример. Рассмотрим простейшую задачу построения оптимального по быстродействию расписания обслуживания пяти требований тремя приборами A, B и C. Времена обслуживания каждого требования каждым прибором приведены в табл. 6.2.1.

Рассмотрим пять множеств перестановок вида (σ, σ'), $\sigma = (k)$, $k = \overline{1, 5}$, σ' — произвольная перестановка элементов множества $N \setminus \{k\}$, $N = \{1, 2, 3, 4, 5\}$. Каждое из этих множеств содержит $4! = 24$ перестановки.

В качестве нижней оценки $\gamma(\sigma)$ значений оптимизируемой функции $T(\pi)$ на множестве перестановок вида (σ, σ') выберем, для определенности, величину $\gamma_{\text{пр}}(\sigma)$.

Имеем $\gamma(1) = 32$, $\gamma(2) = 31$, $\gamma(3) = 34$, $\gamma(4) = 29$, $\gamma(5) = 30$.

Выбираем перестановку $\sigma = (4)$ с наименьшим значением $\gamma(\sigma)$.

Рассмотрим четыре множества перестановок вида (σ, σ'), $\sigma = (4, k)$, $k = 1, 2, 3, 5$, σ' — произвольная перестановка элементов множества $N \setminus \{4, k\}$. Каждое из этих множеств содержит $3! = 6$ перестановок.

Имеем $\gamma(4, 1) = 33$, $\gamma(4, 2) = 29$, $\gamma(4, 3) = 29$, $\gamma(4, 5) = 30$.

Ветвящийся процесс построения оптимальной перестановки иллюстрируется рис. 6.2.1.

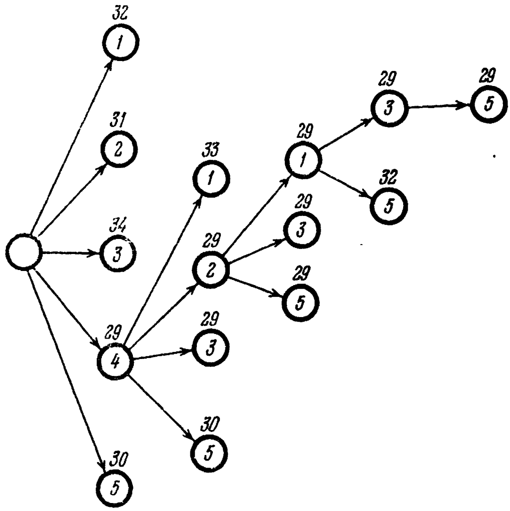

Рис. 6.2.1.

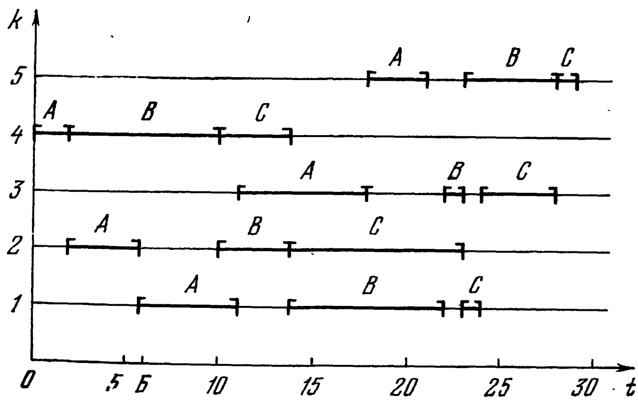

Рис. 6.2.2.

Дальнейшему «развитию» естественно подвергнуть перестановку (4, 2) или (4, 3).

Выберем, для определенности, $\sigma = (4,\ 2)$. Имеем $\gamma(4,\ 2,\ 1) = 29$, $\gamma(4,\ 2,\ 3) = 29$, $\gamma(4,\ 2,\ 5) = 29$. Аналогично, $\gamma(4, 2, 1, 3) = 29$ и $\gamma(4, 2, 1, 5) = 32$. Наконец, $T(4, 2, 1, 3, 5) = 29$.

Множество всех перестановок оказалось разбитым на 11 подмножеств, соответствующих конечным вершинам прадерева, изображенного на рис. 6.2.1. При этом найдена перестановка $\pi^* = (4,\ 2,\ 1,\ 3,\ 5)$, которой соответствует значение $T(\pi)$, не большее нижних оценок значений $T(\pi)$ на остальных множествах перестановок. Следовательно, обслуживая требования в последовательности π^*, получаем оптимальное по быстродействию расписание. График этого расписания приведен на рис. 6.2.2.

§ 3. Элиминация

Сформулируем достаточные условия, при которых возможно исключить (элиминировать) из рассмотрения все последовательности обслуживания n требований M приборами, начинающиеся с данной частичной последовательности σ. Оставшееся множество последовательностей должно, естественно, содержать хотя бы одну оптимальную последовательность.

3.1. Каждое требование множества $N = \{1, 2, \ldots, n\}$ обслуживается последовательно приборами $1, 2, \ldots$ $\ldots, L, \ldots, M$. Время обслуживания требования k прибором L равно t_{kL}. Предполагается, что все приборы обслуживают требования в одной и той же последовательности без неоправданных простоев.

Пусть $\tilde{N} \subset N$ и $\sigma = (i_1, i_2, \ldots, i_r)$, $\sigma^{(1)}$, $\sigma^{(2)}$ — произвольные перестановки элементов множества $\tilde{N}$, $r = |\tilde{N}|$.

Если требования множества $\tilde{N}$ обслуживаются первыми в последовательности σ, начиная с момента времени $d = 0$, то прибор L завершает их обслуживание в момент времени

$$T_L(\sigma) = \max_{1 \leqslant u_1 \leqslant u_2 \leqslant \ldots \leqslant u_{L-1} \leqslant r} \left[\sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} + \ldots + \sum_{k=u_{L-1}}^{r} t_{i_k L} \right]. \tag{3.1}$$

Предположим, что $T_L(\sigma^{(1)}) \leqslant T_L(\sigma^{(2)})$ для всех $L = \overline{1, M}$. Тогда, очевидно, $T(\sigma^{(1)}, \sigma^*) \leqslant T(\sigma^{(2)}, \sigma^*)$ для всех перестановок σ^* элементов множества $N \setminus \tilde{N}$. Напомним, что $T(\pi)$ — общее время обслуживания всех требований при обслуживании их в последовательности π.

Таким образом, если $T_L(\sigma^{(1)}) \leqslant T_L(\sigma^{(2)})$ для всех $L = \overline{1, M}$, то при поиске оптимальной (по быстродействию) последовательности обслуживания требований можно отсеить $|N \setminus \tilde{N}|!$ последовательностей, начинающихся с частичной последовательности $\sigma^{(2)}$.

Указанное условие элиминации является простейшим, достаточно жестким и в реальных задачах упорядочения выполняется сравнительно редко. Значительно более широкое распространение получило следующее условие.

Будем через $\{\sigma\}$ обозначать множество элементов в перестановке σ, т. е. в рассматриваемом случае $\{\sigma\} = \tilde{N}$. Пусть $a, b \in N \setminus \{\sigma\}$.

Обозначим через σ' и σ'' некоторые перестановки элементов из множества $N \setminus \{\sigma, a, b\}$. Предполагается, что $\{\sigma'\} \cap \{\sigma''\} = \phi$ и $\{\sigma'\} \cup \{\sigma''\} = N \setminus \{\sigma, a, b\}$. В частности, эти перестановки могут быть пустыми.

Сравним значения $T(\sigma, a, b, \sigma', \sigma'')$ и $T(\sigma, b, \sigma', a, \sigma'')$. Если нам удастся сформулировать условия, при которых

$$T(\sigma, a, b, \sigma', \sigma'') \leqslant T(\sigma, b, \sigma', a, \sigma'') \qquad (3.2)$$

для всех возможных σ' и σ'', то множество последовательностей, начинающихся с частичной последовательности (σ, b), можно удалить из рассмотрения.

Действительно, в этом случае для любой последовательности вида $\pi^{(1)} = (\sigma, b, \sigma', a, \sigma'')$ можно указать последовательность, начинающуюся с частичной последовательности (σ, a), а именно последовательность $\pi^{(2)} = (\sigma, a, b, \sigma', \sigma'')$ такую, что $T(\pi^{(2)}) \leqslant T(\pi^{(1)})$.

3.2. Обозначим

$$\Delta_L = T_L(\sigma, a, b) - T_L(\sigma, b). \qquad (3.3)$$

Теорема 3.1. *Если*

$$\Delta_{L-1} \leqslant \Delta_L \leqslant t_{aL}, \quad L = \overline{2, M}, \qquad (3.4)$$

то соотношение (3.2) *справедливо для всех возможных* σ' *и* σ''.

Доказательство. 1. Пусть выполняется соотношение (3.4) и $\sigma' = (p_1, p_2, \ldots, p_q)$. Покажем, что

$$T_L(\sigma, a, b, \sigma') - T_L(\sigma, b, \sigma') \leqslant \Delta_L. \tag{3.5}$$

Начнем с рассмотрения случая $q = 1$. Имеем

$$T_1(\sigma, a, b, p_1) - T_1(\sigma, b, p_1) = t_{a1} = \Delta_1. \tag{3.6}$$

Пусть соотношение (3.5) справедливо для приборов $1, 2, \ldots, L-1$. Покажем, что оно справедливо и для прибора L (значение q по-прежнему равно 1).

Имеем

$$\begin{gathered} T_L(\sigma, a, b, p_1) - T_L(\sigma, b, p_1) = \\ = \max\{T_{L-1}(\sigma, a, b, p_1), T_L(\sigma, a, b)\} + t_{p_1L} - \\ - \max\{T_{L-1}(\sigma, b, p_1), T_L(\sigma, b)\} - t_{p_1L} \leqslant \\ \leqslant \max\{T_{L-1}(\sigma, a, b, p_1) - T_{L-1}(\sigma, b, p_1), T_L(\sigma, a, b) - \\ - T_L(\sigma, b)\} \leqslant \max\{\Delta_{L-1}, \Delta_L\} = \Delta_L. \end{gathered} \tag{3.7}$$

Таким образом, если $q = 1$, то соотношение (3.5) выполняется для всех $L = \overline{1, M}$.

Пусть $q = 2$ и соотношение (3.5) выполняется для приборов $1, 2, \ldots, L-1$. Покажем, что оно справедливо и для прибора L. Для $L = 1$ справедливость (3.5) очевидна.

Имеем

$$\begin{gathered} T_L(\sigma, a, b, p_1, p_2) - T_L(\sigma, b, p_1, p_2) \leqslant \\ \leqslant \max\{T_{L-1}(\sigma, a, b, p_1, p_2) - T_{L-1}(\sigma, b, p_1, p_2), \\ T_L(\sigma, a, b, p_1) - T_L(\sigma, b, p_1)\} \leqslant \max\{\Delta_{L-1}, \Delta_L\} = \Delta_L. \end{gathered} \tag{3.8}$$

Аналогично можно показать справедливость соотношения (3. 5) и для $q > 2$.

2. По условию $\Delta_L \leqslant t_{aL}$ и, следовательно,

$$T_L(\sigma, a, b, \sigma') \leqslant T_L(\sigma, b, \sigma') + t_{aL}, \quad L = \overline{1, M}. \tag{3.9}$$

По определению правая часть соотношения (3.9) не превосходит величины $T_L(\sigma, b, \sigma', a)$.

Таким образом, для всех $L = \overline{1, M}$

$$T_L(\sigma, a, b, \sigma') \leqslant T_L(\sigma, b, \sigma', a). \tag{3.10}$$

3. Для завершения доказательства теоремы достаточно показать, что если ε' и ε'' — различные последователь-

ности обслуживания одних и тех же требований из N, а ε — произвольная последовательность обслуживания остальных требований множества N, то из условия

$$T_L(\varepsilon') \leqslant T_L(\varepsilon'') \tag{3.11}$$

для всех $L = \overline{1, M}$ следует

$$T_L(\varepsilon', \varepsilon) \leqslant T_L(\varepsilon'', \varepsilon) \tag{3.12}$$

для всех $L = \overline{1, M}$.

Пусть $\varepsilon = (p_1, p_2, \ldots, p_q)$, $q = 1$, и утверждение справедливо для приборов $1, 2, \ldots, L-1$. При $L = 1$ справедливость утверждения очевидна.

Имеем

$$T_L(\varepsilon', p_1) = \max\{T_L(\varepsilon'), T_{L-1}(\varepsilon', p_1)\} + t_{p_1L} \leqslant$$
$$\leqslant \max\{T_L(\varepsilon''), T_{L-1}(\varepsilon'', p_1)\} + t_{p_1L} \leqslant T_L(\varepsilon'', p_1). \tag{3.13}$$

Если $q = 2$, то, полагая ε' равным (ε', p_1), а ε'' равным (ε'', p_1), можно повторить предыдущие рассуждения. Аналогичным образом поступаем и в случае $q > 2$. Теорема доказана.

Теорема 3.2. *Для того чтобы выполнялись условия* (3.4), *достаточно, чтобы*

$$\Delta_{L-1} \leqslant t_{aL} \quad \text{и} \quad T_{L-1}(\sigma, a) \leqslant T_{L-1}(\sigma, b) \tag{3.14}$$

для всех $L = \overline{2, M}$.

Доказательство. 1. Обозначим $t_{\sigma L} = \sum_{k \in \{\sigma\}} t_{kL}$, $Q_L(\sigma) = T_L(\sigma) - t_{\sigma L}$, $K_L(\sigma, a) = T_{L-1}(\sigma, a) - t_{\sigma L}$. Величина $t_{\sigma L}$ — суммарное время обслуживания всех требований множества $\{\sigma\}$ прибором L. Величина $Q_L(\sigma)$ — время простоя прибора L при обслуживании приборами $1, 2, \ldots, L$ требований множества $\{\sigma\}$ в последовательности σ. Наконец, величина $K_L(\sigma, a)$ — разность между общим временем обслуживания приборами $1, 2, \ldots, L-1$ требований множества $\{\sigma, a\}$ в последовательности (σ, a) и суммарным временем обслуживания требований множества $\{\sigma\}$ прибором L.

Введенные в рассмотрение величины связаны соотношением

$$Q_L(\sigma, a) = \max\{Q_L(\sigma), K_L(\sigma, a)\}, \tag{3.15}$$

которое непосредственно следует из соотношения

$$T_L(\sigma,\ a) = \max\{T_L(\sigma),\ T_{L-1}(\sigma,\ a)\} + t_{aL}. \qquad (3.16)$$

2. Пусть выполняются соотношения (3.14) для всех $L = \overline{2, M}$ и, следовательно, для всех $L = \overline{2, M}$:

$$\max\{K_L(\sigma,\ a),\ K_L(\sigma,\ a,\ b)\} \leqslant K_L(\sigma,\ b). \qquad (3.17)$$

Покажем, что в этом случае $\Delta_L \leqslant t_{aL}$, $L = \overline{2, M}$. Имеем

$$\max\{Q_L(\sigma),\ K_L(\sigma,\ a),\ K_L(\sigma,\ a,\ b)\} \leqslant \max\{Q_L(\sigma),\ K_L(\sigma,\ b)\}, \qquad (3.18)$$

откуда, учитывая (3.15), имеем

$$\max\{Q_L(\sigma,\ a),\ K_L(\sigma,\ a,\ b)\} \leqslant Q_L(\sigma,\ b), \qquad (3.19)$$

или

$$Q_L(\sigma,\ a,\ b) \leqslant Q_L(\sigma,\ b). \qquad (3.20)$$

Полученное соотношение можно переписать в виде

$$T_L(\sigma,\ a,\ b) - t_{(\sigma,\ a,\ b)\,L} \leqslant T_L(\sigma,\ b) - t_{(\sigma,\ b)\,L}, \qquad (3.21)$$

откуда следует, что $\Delta_L \leqslant t_{aL}$ для всех $L = \overline{2, M}$.

3. Для завершения доказательства теоремы необходимо показать, что $\Delta_{L-1} \leqslant \Delta_L$ для всех $L = \overline{2, M}$.

Из (3.16), следует, что

$$T_L(\sigma, b) - T_{L-1}(\sigma,\ b) \geqslant t_{bL}. \qquad (3.22)$$

Пусть в этом соотношении имеет место знак равенства. Прибавляя к обеим частям величину Δ_L, получаем

$$T_L(\sigma,\ a,\ b) - T_{L-1}(\sigma,\ b) = \Delta_L + t_{bL}. \qquad (3.23)$$

Из (3.16) следует, что

$$T_L(\sigma,\ a,\ b) \geqslant T_{L-1}(\sigma,\ a,\ b) + t_{bL}. \qquad (3.24)$$

Вычитая из обеих частей этого неравенства величину $T_{L-1}(\sigma,\ b)$, получаем

$$T_L(\sigma,\ a,\ b) - T_{L-1}(\sigma,\ b) \geqslant \Delta_{L-1} + t_{bL}. \qquad (3.25)$$

Из (3.23) и (3.25) следует, что $\Delta_L \geqslant \Delta_{L-1}$.

Пусть в соотношении (3.22) имеет место знак строгого неравенства. Тогда, учитывая (3.16), имеем соотношения $T_L(\sigma, b) = T_L(\sigma) + t_{bL}$ и

$$T_{L-1}(\sigma, a) \leqslant T_{L-1}(\sigma, b) < T_L(\sigma, b) - t_{bL} = T_L(\sigma)$$

или, учитывая (3.16),

$$T_L(\sigma, a) = T_L(\sigma) + t_{aL}. \tag{3.26}$$

Таким образом,

$$\begin{aligned} T_L(\sigma, a, b) &= \max\{T_L(\sigma, a), T_{L-1}(\sigma, a, b)\} + t_{bL} = \\ &= \max\{T_L(\sigma) + t_{aL}, T_{L-1}(\sigma, b) + \Delta_{L-1}\} + t_{bL} = \\ &= T_L(\sigma) + t_{aL} + t_{bL}. \end{aligned} \tag{3.27}$$

Окончательно

$$\begin{aligned} \Delta_L = T_L(\sigma, a, b) - T_L(\sigma, b) = T_L(\sigma) + t_{aL} + t_{bL} - \\ - T_L(\sigma) - t_{bL} = t_{aL} \geqslant \Delta_{L-1}. \end{aligned} \tag{3.28}$$

Теорема доказана.

3.3. Пример. Рассмотрим задачу построения оптимального (по быстродействию) расписания обслуживания требований множества $N = \{1, 2, \ldots, n\}$ тремя приборами 1, 2 и 3. Пусть требования $a, b, c \in N$ и времена их обслуживания приведены в табл. 6.3.1.

Таблица 6.3.1

$k \backslash L$	1	2	3
a	2	10	40
b	3	6	20
c	4	4	15

Выберем $\sigma = (c)$, тогда

$$\Delta_L = T_L(c, a, b) - T_L(c, b).$$

Вычислим Δ_L для $L = 1, 2, 3$. Имеем $\Delta_1 = 9-7 = 2$, $\Delta_2 = 24-14 = 10$, $\Delta_3 = 83-43 = 40$.

Непосредственной проверкой убеждаемся, что условие теоремы 3.1 выполняется. Следовательно, при поиске оптимальной последовательности можно исключить из рассмотрения все последовательности вида $(c, b, i_3, i_4, \ldots, i_n)$.

Отметим, что в данном случае соотношения (3.14) не выполняются, так как $T_2(c, a) = 18 > 14 = T_2(c, b)$.

Систематическое использование условий элиминации в схемах последовательного конструирования вариантов позволяет существенно сузить область поиска оптимального варианта.

§ 4. Вырожденные случаи

В этом параграфе рассматриваются некоторые частные случаи задачи построения оптимального (по быстродействию) расписания обслуживания n требований M последовательными приборами. Показано, что если матрица $\| t_{kL} \|$ времен обслуживания требований удовлетворяет определенным условиям, то оптимальная последовательность обслуживания требований может быть построена в результате весьма простых рассуждений.

4.1. Зададим на множестве столбцов матрицы $\| t_{kL} \|$ бинарное отношение, полагая $L_1 \models L_2$, если $t_{iL_1} \leqslant t_{jL_2}$ для всех $1 \leqslant i \neq j \leqslant n$. Величина t_{kL} — время обслуживания требования k прибором L.

Будем говорить, что *матрица* $\| t_{kL} \|$ *обладает свойством* $C_1(\lambda)$ (при некотором $1 \leqslant \lambda \leqslant M-1$), если

$$\lambda \models (\lambda - 1) \models \ldots \models 1, \quad (\lambda + 1) \models (\lambda + 2) \models \ldots \models M.$$

Покажем, что если при некотором $\lambda = \lambda^*$ матрица $\| t_{kL} \|$ обладает свойством $C_1(\lambda)$, то общее время обслуживания всех требований при обслуживании их в последовательности $\pi = (i_1, i_2, \ldots, i_n)$ равно

$$T(\pi) = \max_{1 \leqslant u \leqslant n} \left[\sum_{k=1}^{u} t_{i_k 1} + \sum_{L=2}^{M-1} t_{i_u L} + \sum_{k=u}^{n} t_{i_k M} \right]. \qquad (4.1)$$

Иными словами, в этом случае в соотношении (1.1) максимум достигается при $u_1 = u_2 = \ldots = \overset{*}{u_\lambda} = \ldots \ldots = u_{M-1} = u$.

Действительно, поскольку $2 \models 1$, то

$$\sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} \leqslant \sum_{k=1}^{u_2} t_{i_k 1} + t_{i_{u_2} 2}. \qquad (4.2)$$

Поскольку $3 \models 2$, то

$$\sum_{k=1}^{u_2} t_{i_k 1} + t_{i_{u_2} 2} + \sum_{k=u_2}^{u_3} t_{i_k 3} \leqslant \sum_{k=1}^{u_3} t_{i_k 1} + t_{i_{u_3} 2} + t_{i_{u_3} 3}. \qquad (4.3)$$

Повторяя аналогичные рассуждения (поскольку $4 \models 3, \ldots \ldots, \lambda^* \models \lambda^* - 1$), получаем

$$\sum_{k=1}^{u_1} t_{i_k 1} + \ldots + \sum_{k=u_{\lambda^*-1}}^{u_{\lambda^*}} t_{i_k \lambda^*} \leqslant \sum_{k=1}^{u_{\lambda^*}} t_{i_k 1} + \sum_{L=2}^{\lambda^*} t_{i_{u_{\lambda^*}} L} \qquad (4.4)$$

и (поскольку $(M-1) \models M$, $(M-2) \models (M-1)$, ..., $(\lambda^*+1) \models (\lambda^*+2)$)

$$\sum_{k=u_{\lambda^*}}^{u_{\lambda^*+1}} t_{i_k \lambda^*+1} + \ldots + \sum_{k=u_{M-1}}^{n} t_{i_k M} \leqslant \sum_{L=\lambda^*+1}^{M-1} t_{i_{u_{\lambda^*}} L} + \sum_{k=u_{\lambda^*}}^{n} t_{i_k M}, \tag{4.5}$$

откуда непосредственно следует справедливость соотношения (4.1).

Прибавим и вычтем из правой части этого соотношения величину $\sum_{L=2}^{M-1} \sum_{k=1}^{n} t_{i_k L}$, не зависящую от π. Обозначим $\beta_j = \sum_{L=2}^{M} t_{jL}$ и $\alpha_j = \sum_{L=1}^{M-1} t_{jL} - \beta_j$.

Имеем

$$T(\pi) = \max_{1 \leqslant u \leqslant n} \left(\sum_{k=1}^{u} \alpha_{i_k} + \beta_{i_u} \right) + A, \tag{4.6}$$

где A — не зависящая от π величина.

Таким образом, если матрица $\| t_{kL} \|$ обладает свойством $C_1(\lambda)$ при некотором $\lambda = \lambda^*$, то построение оптимальной последовательности обслуживания требований сводится к нахождению перестановки π^*, которой соответствует наименьшее значение функции

$$f_2(\pi) = \max_{1 \leqslant u \leqslant n} \left(\sum_{k=1}^{u} \alpha_{i_k} + \beta_{i_u} \right). \tag{4.7}$$

Последняя задача подробно рассмотрена в гл. 2, § 4, п. 4.4.

4.2. Пример. Рассмотрим задачу построения оптимального (по быстродействию) расписания обслуживания восьми требований пятью приборами; времена t_{kL} обслуживания требования $k = \overline{1,8}$ прибором $L = \overline{1,5}$ приведены в табл. 6.4.1.

В рассматриваемом случае матрица $\| t_{kL} \|$ обладает свойством $C_1(2)$.

Значения α_j и β_j и вычисленные в соответствии с гл. 2, § 4 значения $\omega(j)$ (полагаем $M = 100$) приведены в табл. 6.4.2.

Таблица 6.4.1

k \ L	1	2	3	4	5
1	22	15	20	24	34
2	18	10	14	32	40
3	34	13	10	30	38
4	40	14	12	25	36
5	16	18	12	28	40
6	21	16	22	20	37
7	26	12	20	31	42
8	32	12	14	30	45

Таблица 6.4.2

j	1	2	3	4	5	6	7	8
α_j	—12	—22	—4	4	—24	—16	—16	—13
β_j	93	96	91	87	98	95	105	101
$\omega(j)$	—19	—26	—13	13	—26	—21	—11	—12

Оптимальные последовательности обслуживания $\pi^* = ((2, 5), 6, 1, 3, 8, 7, 4)$. Значение $T(\pi^*) = 386$.

4.3. Будем говорить, что *матрица* $\| t_{kL} \|$ *обладает свойством* $C_2(\lambda)$ (при некотором $1 \leqslant \lambda \leqslant M$), если

$$1 \models 2 \models \ldots \models \lambda \quad \text{и} \quad M \models (M-1) \models \ldots \models \lambda.$$

Покажем, что если матрица $\| t_{kL} \|$ обладает свойством $C_2(\lambda)$ при некотором $\lambda = \lambda^*$, то

$$T(\pi) = \sum_{L=1}^{\lambda^*-1} t_{i_1 L} + \sum_{k=1}^{n} t_{i_k \lambda^*} + \sum_{L=\lambda^*+1}^{M} t_{i_n L}. \qquad (4.8)$$

Действительно, в этом случае

$$\sum_{k=1}^{u_1} t_{i_k 1} + \sum_{k=u_1}^{u_2} t_{i_k 2} + \ldots + \sum_{k=u_{\lambda^*-1}}^{u_{\lambda^*}} t_{i_k \lambda^*} \leqslant t_{i_k 1} + \sum_{k=1}^{u_2} t_{i_k 2} + \ldots$$

$$\ldots + \sum_{k=u_{\lambda^*-1}}^{u_{\lambda^*}} t_{i_k \lambda^*} \leqslant \ldots \leqslant \sum_{L=1}^{\lambda^*-1} t_{i_1 L} + \sum_{k=1}^{u_{\lambda^*}} t_{i_k \lambda^*}. \qquad (4.9)$$

Аналогично,

$$\sum_{k=u_{\lambda^*}}^{u_{\lambda^*+1}} t_{i_k\lambda^*+1} + \ldots + \sum_{k=u_{M-1}}^{n} t_{i_kM} \leqslant \sum_{k=u_{\lambda^*}}^{u_{\lambda^*+1}} t_{i_k\lambda^*+1} + \ldots$$
$$\ldots + \sum_{k=u_{M-2}}^{n} t_{i_kM-1} + t_{i_nM} \leqslant \ldots \leqslant \sum_{k=u_{\lambda^*}+1}^{n} t_{i_k\lambda^*} + \sum_{L=\lambda^*+1}^{M} t_{i_nL}, \tag{4.10}$$

откуда следует справедливость соотношения (4.8).

Рассмотрим отдельно случаи $\lambda^* = 1$, $\lambda^* = M$ и $1 < \lambda^* < M$. Если $\lambda^* = 1$, то

$$T(\pi) = \sum_{k=1}^{n} t_{i_k1} + \sum_{L=2}^{M} t_{i_nL}. \tag{4.11}$$

Поскольку первое слагаемое от π не зависит, то $T(\pi)$ принимает наименьшее значение, если последним обслуживается требование j с наименьшим значением $\sum_{L=2}^{M} t_{jL}$. Остальные требования можно обслуживать в произвольном порядке.

Если $\lambda^* = M$, то

$$T(\pi) = \sum_{L=1}^{M-1} t_{i_1L} + \sum_{k=1}^{n} t_{i_kM}. \tag{4.12}$$

Поскольку второе слагаемое от π не зависит, то $T(\pi)$ принимает наименьшее значение, если первым обслуживать требование j с наименьшим значением $\sum_{L=1}^{M-1} t_{jL}$. Остальные требования можно обслуживать в произвольном порядке.

Пусть, наконец, $1 < \lambda^* < M$. Образуем множество J_1 требований j, которым соответствует наименьшее значение $v_j = \sum_{L=1}^{\lambda^*-1} t_{jL}$, и множество J_2 требований j, которым соответствует наименьшее значение $w_j = \sum_{L=\lambda^*+1}^{M} t_{jL}$.

Если можно выбрать $j_1 \in J_1$ и $j_2 \in J_2$ такие, что $j_1 \neq j_2$, то в оптимальной последовательности требование j_1 следует обслуживать первым, а требование j_2 — последним. Остальные требования можно обслуживать в произвольном порядке.

Пусть $J_1 = J_2 = \{j'\}$. Среди требований множества $N \setminus \{j'\}$ выберем требование j'' с наименьшим значением v_j и требование j''' с наименьшим значением w_j. Требования j'' и j''' не обязательно различны.

Если $v_{j'} + w_{j'''} \leqslant v_{j''} + w_{j'}$, то в оптимальной последовательности требование j' следует обслуживать первым, а требование j''' последним. Если $v_{j'} + w_{j'''} > v_{j''} + w_{j'}$, то первым следует обслуживать требование j'', а последним требование j'. В обоих случаях порядок обслуживания остальных требований безразличен.

§ 5. Библиографическая справка

Рассматриваемая в данной главе задача известна в литературе как задача $M \times n$ Беллмана — Джонсона.

Конструктивный подход к решению экстремальных задач, в том числе задач теории расписаний, развит в работах Р. Беллмана и С. Дрейфуса [15, 16], Н. Н. Моисеева [124], В. С. Михалевича, Ю. М. Ермольева, В. В. Шкурбы и Н. З. Шора [119—122], [211]. Вероятно, первый вариант метода ветвей и границ для решения задачи $M \times n$ Беллмана — Джонсона был предложен З. Ломницким сначала для $M = 3$ [333], затем (совместно с А. Броуном) для произвольного M [244]. Мы в изложении следовали работе [343]. В настоящее время предложены разнообразные варианты метода ветвей и границ Б. Банком и Е. Сэйфартом [231, 232], Х. Гринбергом [290], Г. В. Черновой [195, 196] и другими авторами [183, 216, 308]. Достаточно общее описание этого метода дано Е. Балашем [228], обзор литературы — Е. Лоулером и Д. Вудом [328].

Использованию метода динамического программирования для решения рассматриваемой задачи посвящена работа Р. Беллмана и О. Круса [237]. Систематические исследования в этой области проведены И. Набешимой [361—364], Е. Х. Драхлиным и Р. М. Якимовым [62, 214], Б. А. Власюком [37].

Разнообразные приближенные методы предложены С. Л. Журженко [64], В. М. Португалом [140], В. М. Озерным и Л. П. Рябовым [129], В. Н. Резниченко [146], Д. С. Палмером [376] и другими авторами [227, 234, 282, 392, 423]. Использованию методов сортировки посвящены работы Е. Пэйджа [373—375], организации направленного поиска,— работа Д. И. Голенко и Ю. Я. Тарнопольского [53].

В ряде работ (см. следующую главу) предлагается использовать методы (целочисленного) линейного программирования.

Сравнительный анализ некоторых точных и приближенных методов решения задачи $M \times n$ Беллмана — Джонсона проведен Е. Сэйфартом [400] (243 тестовые задачи, $4 \leqslant n \leqslant 100$, $4 \leqslant M \leqslant 60$) и С. Ашуром [222] (550 тестовых задач, $6 \leqslant n \leqslant 12$, $3 \leqslant M \leqslant 5$).

Выявлению условий элиминации посвящены работы Р. Смита, Р. Дудека и О. Тэйтона [266, 407, 408], Дж. Мак-Махона [341, 342] и В. Шварца [419]. Результаты машинных экспериментов с использованием соответствующих алгоритмов приведены в [293].

Параграф 4 написан на основе работы В. Я. Бурдюка [23]. Аналогичные вырожденные ситуации для случая $M = 3$ рассматривались С. М. Джонсоном [313] и В. Шварцем [417, 418]. Несколько иные ситуации рассмотрены в работах [219, 182, 95].

Задача минимизации функционала (1.2) рассматривалась В. Я. Бурдюком [21], функционала (1.3) — Н. А. Лепешинским [100], Е. Х. Драхлиным и Р. М. Якимовым [61] и другими авторами [380, 383, 401]. Обобщение последней задачи на случай нескольких возможных графиков обслуживания требований приведено в работах Р. Солиха [160, 410]. Еще одно естественное обобщение этой задачи рассматривается в § 1 гл. 8.

Асимптотическое поведение решений большеразмерных задач исследовано в [139]. Задача $M \times n$ Беллмана — Джонсона с некоторыми дополнительными ограничениями рассмотрена в работах [74, 316, 348, 369]. П. Талвар [422] и Л. И. Фейгин [189] рассмотрели вариант этой задачи, предполагая, что времена обслуживания достоверно не известны.

ГЛАВА 7

M ПОСЛЕДОВАТЕЛЬНЫХ ПРИБОРОВ РАЗЛИЧНЫЕ МАРШРУТЫ

Имеется n требований и M обслуживающих приборов. Каждое требование обслуживается приборами в заданной, специфической для него последовательности. Процесс обслуживания требования k может включать повторные обращения к одним и тем же приборам. Каждый прибор одновременно обслуживает не более одного требования и каждое требование одновременно обслуживается не более чем одним прибором. Известны времена обслуживания каждого требования каждым прибором, участвующим в его обслуживании (при каждом обращении к этому прибору). Прерывания в обслуживании каждого отдельного требования отдельным прибором не допускаются. Необходимо построить оптимальное по быстродействию расписание процесса обслуживания всех требований. Предполагается, что времена транспортировок требований и переналадок приборов равны нулю.

В § 1 приводится сетевая интерпретация этой задачи. В § 2 анализируются разнообразные используемые на практике правила предпочтения. В § 3 обсуждаются вопросы организации поиска оптимальных расписаний методами статистических испытаний. Наконец, в § 4 задача построения оптимального по критерию быстродействия расписания формулируется в терминах целочисленного линейного программирования.

§ 1. Сетевое представление

Рассмотрим задачу построения оптимального (по быстродействию) расписания обслуживания требований множества $N = \{1, 2, \ldots, n\}$ приборами множества $\{1, 2, \ldots \ldots, L, \ldots, M\}$. Требование $k \in N$ обслуживается приборами в заданной последовательности $(L_1^k, L_2^k, \ldots, L_{r_k})$. Приборы L_j^k, $j = \overline{1, r_k}$, не обязательно различны. Каждый прибор обслуживает требования последовательно. Известны времена обслуживания требований приборами при каждом к ним обращении.

1.1. В рассматриваемом случае определенные затруднения возникают уже на стадии построения допустимых расписаний.

Введем понятие *операции* как процесса обслуживания отдельного требования отдельным прибором при некотором конкретном обращении к этому прибору. В этой терминологии процесс обслуживания требования k состоит в последовательном выполнении r_k операций. Если требование k обслуживается прибором L в q-й по очереди раз, то эту операцию будем обозначать через (k, L, q), а длительность ее выполнения через $t(k, L, q)$.

Представим все операции в виде точек (кружков) на плоскости. Каждые две операции (k_1, L_1, q_1) и (k_2, L_2, q_2) могут быть зависимыми или независимыми в том смысле, что календарное время выполнения одной из них оказывает или не оказывает влияние на календарное время выполнения другой. В условиях рассматриваемой задачи целесообразно выделить три вида бинарных межоперационных отношений.

Если $k_1 \neq k_2$ и $L_1 \neq L_2$, то операции (k_1, L_1, q_1) и (k_2, L_2, q_2) являются *независимыми*. Графически они не соединяются никакими видами связей — ребрами, дугами и т. п.

Если $k_1 = k_2 = k$, то по условию задачи одна из операций, для определенности, (k, L_2, q_2), *следует* во времени за второй. В данном случае операция (k, L_2, q_2) не может быть начата раньше, чем закончится операция (k, L_1, q_1). Графически операции (k, L_1, q_1) и (k, L_2, q_2) естественно соединять дугой, направленной от первой операции ко второй.

Наконец, если $L_1 = L_2 = L$ и $k_1 \neq k_2$, то операции (k_1, L, q_1) и (k_2, L, q_2) не могут выполняться одновременно, однако очередность их выполнения *заранее не оговорена*. В этом случае соединим операции ребром.

В результате получаем смешанный граф $(X, \vec{U}, U)$, где X — множество операций (вершин), $\vec{U}$ — множество дуг, U — множество ребер.

1.2. Пример. Рассмотрим три требования 1, 2, 3, обслуживаемых четырьмя приборами A, B, C, D.

Последовательности (маршруты) обслуживания приведены в табл. 7.1.1.

Процесс обслуживания требования 1 состоит в последовательном выполнении операций (1, A, 1), (1, B, 1) и (1, D, 1). При обслуживании требования 2 последовательно выполняются операции (2, A, 1), (2, B, 1) и (2, A, 2). Наконец, при обслуживании требования 3 последовательно выполняются операции (3, A, 1), (3, C, 1), (3, B, 1) и (3, D, 1).

Таблица 7.1.1

1	A	B	D	—
2	A	B	A	—
3	A	C	B	D

Операции (1, B, 1) и (2, A, 2) независимы, поскольку эти операции осуществляются над разными требованиями с использованием различных приборов. Независимыми являются также пары операций (1, B, 1) и (3, A, 1), (2, B, 1) и (3, D, 1) и т. д.

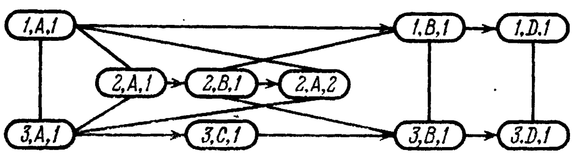

Рис. 7.1.1.

По условию операция (1, D, 1) следует во времени за операцией (1, B, 1), которая в свою очередь следует за операцией (1, A, 1).

Операции (1, A, 1) и (2, A, 2) осуществляются с использованием одного и того же прибора A. Одновременно эти операции выполняться не могут, однако очередность их выполнения не фиксирована. Аналогичная ситуация имеет место и в случае операций (2, A, 1) и (3, A, 1), (1, D, 1) и (3, D, 1) и т. д. Соответствующий смешанный граф $(X, \vec{U}, U)$ представлен на рис. 7.1.1.

1.3. Каждое (допустимое) расписание определяет календарные сроки проведения каждой операции и тем самым однозначно фиксирует определенные последовательности выполнения операций каждым прибором. Иными словами, каждому расписанию соответствует некоторый ориентированный бесконтурный граф, порождаемый данным смешанным графом в результате замены всех его ребер дугами.

В свою очередь каждый бесконтурный ориентированный граф определяет бесконечное число (допустимых) расписаний. Действительно, припишем каждой дуге этого графа, соединяющей вершину (k_1, L_1, q_1) с вершиной (k_2, L_2, q_2), число $t(k_1, L_1, q_1)$ — длительность операции (k_1, L_1, q_1). В результате получаем сетевой график. Используя обычную технику сетевого планирования, можно определить время начала и окончания каждой операции, т. е. построить расписание обслуживания требований приборами. Таких расписаний, очевидно, можно построить сколь угодно много.

Поскольку нас интересуют оптимальные (по быстродействию) расписания, то среди указанного множества расписаний целесообразно выделить расписание, при котором каждая операция начинает выполняться в момент времени завершения всех предшествующих ей операций. Тем самым каждому бесконтурному графу, порождаемому рассматриваемым смешанным графом, будет соответствовать одно вполне определенное расписание.

Таким образом, задача построения оптимального (по быстродействию) расписания обслуживания n требований M приборами может быть решена перебором конечного числа возможных вариантов расписаний. Этот перебор определяется числом бесконтурных графов, порождаемых данным смешанным графом. Методы оценки и прямого определения указанного числа описаны в § 3 гл. 2.

1.4. Вернемся к рассмотрению примера, дополнив его условие заданием времен выполнения операций. Пусть время обслуживания требования $k = 1, 2, 3$ каждым из обслуживающих его приборов равно k^2.

На рис. 7.1.2,*а*) и 7.1.2, *б*) приведены два бесконтурных графа (сетевые графики), порождаемых графом, изображенным на рис. 7.1.1, в результате замены всех его ребер дугами. Каждой дуге приписана длительность выполнения, каждой вершине — календарное время начала выполнения соответствующей операции. Сетевые графики, изображенные на рис. 7.1.2, *а*) и 7.1.2, *б*), дополнены вершиной R — окончание всех операций.

В соответствии с графиком, изображенным на рис. 7.1.2, *а*), обслуживание всех требований можно завершить в течение 43 единиц времени. В соответствии с графиком, изображенным на рис. 7.1.2, *б*), обслуживания всех требований можно завершить в течение 50 единиц времени.

Каждый из графиков фиксирует последовательности обслуживания требований каждым прибором, и обратно, фиксированные (допустимые) последовательности обслуживания определяют некоторый сетевой график.

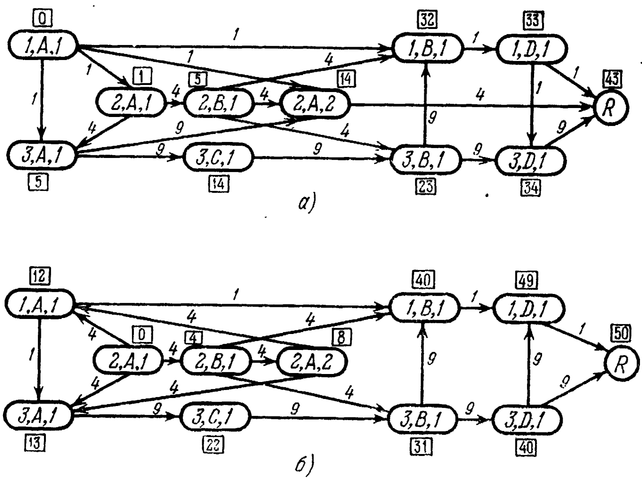

Рис. 7.1.2.

1.5. Процесс построения *допустимых* расписаний, таким образом, может быть разделен на два этапа: 1) построение бесконтурного графа (сетевого графика), порождаемого исходным смешанным графом в результате замены всех его ребер дугами, и 2) построение расписания по данному сетевому графику. Такой подход позволяет создавать *программные генераторы допустимых расписаний* в результате незначительной модификации существующих программ расчета временных параметров сетевых графиков. Для этого достаточно дополнить их программой генерирования самих сетевых графиков.

В соответствии с гл. 2, § 3, п. 3.2 для построения сетевого графика можно воспользоваться следующей формальной процедурой.

а) В списке операций выбираем одну из операций, которая не следует ни за одной операцией списка.

б) Заменяем все ребра в рассматриваемом смешанном графе, соединяющие выбранную операцию и другие операции списка, на исходящие из соответствующей ей вершины дуги. Удаляем операцию из списка.

в) Если список операций исчерпан, процедура окончена. В противном случае переходим к п. а).

Выбирая в п. а) различные операции, получаем в результате многократного применения данной процедуры все допустимые (относительно исходного смешанного графа) сетевые графики. Последующее построение расписания по сетевому графику проводится общеизвестными методами сетевого планирования.

В условиях рассматриваемого примера на первом шаге может быть выбрана любая из трех операций (вершин): (1, A, 1), (2, A, 1) и (3, A, 1). Если выбрать вершину (1, A, 1), то ребра, соединяющие эту вершину с вершинами (2, A, 1), (3, A, 1) и (2, A, 2), необходимо заменить на исходящие из нее дуги (см. рис. 7.1. 2, *а*). Если выбрать вершину (2, A, 1), то ребра, соединяющие эту вершину с вершинами (1, A, 1) и (3, A, 1), меняются на исходящие из вершины (2, A, 1) дуги (см. рис. 7.1.2, *б*).

Предположим, что выбрана вершина (1, A, 1). Удалим эту вершину из рассмотрения (вместе с инцидентными ей дугами).

На следующем шаге может быть выбрана любая из трех операций, (2, A, 1), (3, A, 1) и (1, B, 1), поскольку ни одна из этих операций не следует ни за одной из операций, отличных от (1, A, 1).

Процесс продолжается до тех пор, пока не окажутся рассмотренными все операции (вершины). В результате получаем бесконтурный ориентированный граф. Этот граф вместе с информацией о временных длительностях операций представляет собой один из допустимых сетевых графиков выполнения операций.

Наша задача заключается в построении такого графика, при котором общее время выполнения всех операций окажется наименьшим. Иными словами, необходимо выбрать операции в такой последовательности, чтобы получаемый в результате сетевой график позволял осуществить все операции в кратчайший срок.

В общем случае приходится принимать во внимание не только время выполнения отдельных операций, но и временные затраты, связанные с переналадкой приборов, транспортировкой требований и т. п. В этом случае каждой дуге, соединяющей вершины (k_1, L_1, q_1) и (k_2, L_2, q_2), следует приписать длительность промежутка времени, спустя который операция (k_2, L_2, q_2) может быть начата после начала выполнения операции (k_1, L_1, q_1). В остальном процесс построения допустимых расписаний не претерпевает изменений.

§ 2. Генераторы допустимых расписаний

2.1. Описанный в предыдущем параграфе метод построения допустимых расписаний включает построение бесконтурного ориентированного графа и последующее вычисление необходимых временных характеристик по полученному сетевому графику.

Конструирование графа осуществляется пошагово и на каждом шаге осуществляется выбор очередной операции из множества возможных претендентов. От того, насколько правильно на каждом шаге осуществляется этот выбор, зависит качество получаемого в дальнейшем допустимого расписания.

Поскольку анализ всех возможных последствий выбора той или иной операции практически не реализуем, появляется необходимость принимать решения в условиях относительной неопределенности, на основе анализа ограниченного объема информации, имеющейся к моменту принятия решения. Поэтому представляется целесообразным *совместить* процессы конструирования сетевого графика и расчеты соответствующих временных характеристик.

Процедура построения допустимых расписаний в этом случае может быть описана следующим образом.

а) Положим $(X^{(0)}, \vec{U}^{(0)}, U^{(0)}) = (X, \vec{U}, U)$. Используя граф $(X^{(0)}, \vec{U}^{(0)})$, стандартными методами сетевого планирования вычисляем самые ранние сроки ${}^{(0)}(k, L, q)$ начала всех операций. Очевидно, искомые ремена начала операций $\underline{t}(k, L, q) \geqslant \underline{t}^{(0)}(k, L, q)$.

б) В графе $(X^{(p)}, \vec{U}^{(p)})$ выбираем одну из вершин (k', L', q'), в которую не заходит ни одной дуги.

в) Заменим в графе $(X^{(p)}, \vec{U}^{(p)}, U^{(p)})$ все ребра, инцидентные вершине (k', L', q'), на исходящие из нее дуги. Полученный граф обозначим через $(X^{(p)\prime}, \vec{U}^{(p)\prime}, U^{(p)\prime})$.

г) Используя граф $(X^{(p)\prime}, \vec{U}^{(p)\prime})$ и вычисленные на предыдущем шаге значения $\underline{t}^{(p)}(k, L, q)$, вычисляем самые ранние сроки $\underline{t}^{(p+1)}(k, L, q)$ начала операций множества $X^{(p)\prime}$.

д) Полагаем $\underline{t}(k', L', q') = \underline{t}^{(p)}(k', L', q')$ и удаляем вершину (k', L', q') из рассмотрения вместе с инцидентными ей дугами. Полученный в результате граф обозначим через $(X^{(p+1)}, \vec{U}^{(p+1)}, U^{(p+1)})$.

е) Если $U^{(p+1)} = \phi$, то $\underline{t}(k, L, q) = \underline{t}^{(p+1)}(k, L, q)$ для всех операций множества $X^{(p+1)}$. В противном случае переходим к п. б.), заменяя граф $(X^{(p)}, \vec{U}^{(p)}, U^{(p)})$ графом $(X^{(p+1)}, \vec{U}^{(p+1)}, U^{(p+1)})$.

Применяя эту формальную процедуру к исходному графу $(X, \vec{U}, U)$, через конечное число шагов получаем искомые времена $\underline{t}(k, L, q)$ начала всех операций. Поскольку процесс выполнения каждой отдельной операции непрерывен, то найденные значения $\underline{t}(k, L, q)$ однозначно определяют расписание.

Нетрудно заметить, что в п. б) необходимо уточнить принцип, согласно которому производится выбор той или иной вершины (операции) из множества возможных претендентов. К этому моменту известны нижние границы $\underline{t}^{(p)}(k, L, q)$ времен начала всех операций множества $X^{(p)}$ и все «технологические» условия, которым должен удовлетворять процесс их выполнения, т. е. $\vec{U}^{(p)}, U^{(p)}$, и длительности $t(k, L, q)$ выполнения операций.

Выбор операции должен проводиться с более или менее полным учетом имеющейся информации с тем, чтобы обеспечить достаточно высокое качество получаемого расписания. С другой стороны, степень полноты учета этой информации определяет скорость построения отдельного расписания, следовательно, число допустимых расписаний, генерируемых за приемлемый промежуток времени.

2.2. В настоящее время известны различные по быстродействию и качеству получаемых расписаний генераторы.

Г (R) — генератор с равновероятной выборкой — выбирает любую операцию из множества l возможных на данном шаге претендентов с вероятностью $1/l$.

Г (А) — генератор активных расписаний — если среди множества l возможных на p-м шаге претендентов существует операция (k', L', q'), выбор которой не изменяет ранних сроков начала других операций, т. е. $\underline{t}^{(p+1)}(k, L, q) = \underline{t}^{(p)}(k, L, q)$, то осуществляется выбор этой операции. В противном случае выбирается любая операция с вероятностью $1/l$.

Первый из этих генераторов за достаточно большой промежуток времени позволяет построить все допустимые расписания. Во втором предусмотрена возможность исключения из рассмотрения заведомо неконкурентноспособных расписаний.

Наиболее распространены на практике генераторы, использующие разнообразные правила предпочтения (приоритеты). Эти генераторы не гарантируют, вообще говоря, получения оптимального расписания.

Г (FIFO) — генератор, использующий правило «первый пришел — первый обслуживается», (First In — First On). Из множества возможных на p-м шаге претендентов выбирается операция с наименьшим значением $\underline{t}^{(p)}(k', L', q')$. При наличии нескольких таких операций с равной вероятностью выбирается любая из них. В основу правила FIFO положено соображение, что чем меньше сохраняется в очереди операция, тем скорее разгружается вся очередь.

Г (LIFO) — генератор, использующий правило «последним пришел — первым обслуживается» (Last In — First On). Из множества претендентов на p-м шаге выбирается операция с наибольшим значением $\underline{t}^{(p)}(k', L', q')$. При наличии нескольких таких операций с равной вероятностью выбирается любая из них. Правило LIFO неявно утверждает, что если какая-то операция задержалась, то желательно, чтобы она не мешала разгрузке очереди других операций.

Г (SIO) — генератор, использующий правило «кратчайшей операции» (Shortest Imminent Operation). Выбирается операция с наименьшей длительностью $t(k', L', q')$. Если таких операций несколько, то с равной вероятностью выбирается любая из них. Правило SIO применяется в предположении, что чем меньшая по длительности операция выполняется, тем скорее разгружается очередь.

Г (FOFO) — генератор, использующий правило «первым уйдет — первым обслуживается» (First Off — First On). Выбирается операция с наименьшим значением $t^{(p)}(k', L', q') + t(k', L', q')$. Если таких операций несколько, то с равной вероятностью выбирается любая из них. В правиле FOFO выбирается операция, которая раньше всех может быть выполнена и, по-видимому, обеспечивает и быструю разгрузку очереди.

Г (LRT) — генератор, использующий правило «наибольшего оставшегося времени обслуживания» (Longest Remaining Time). В правиле LRT заложено стремление поскорее обслужить требование, общая длительность обслуживания которого является наибольшей.

Применение каждого из этих правил имеет под собой более или менее правдоподобное основание, хотя нетрудно видеть, например, что правила FIFO и LIFO явно противоречивы. Эффективность того или иного генератора может быть проверена в каждом конкретном случае, для каждой конкретной ситуации.

2.3. Определенное распространение получили так называемые *рандомизированные генераторы* Γ_R, получаемые из перечисленных заменой условия «... выбрать операцию, обладающую некоторым свойством» условием «... с большей вероятностью выбрать операцию, обладающую этим свойством». Тем самым за достаточно большой промежуток времени рандомизированные генераторы Γ_R позволяют получить все допустимые расписания, с большей вероятностью генерируя те же расписания, что и соответствующие генераторы Г.

Второе, не менее естественное направление в создании программных генераторов заключается в максимальном сужении класса генерируемых расписаний. Это достигается заменой в обычном генераторе условия «... при наличии операций, обладающих одним и тем же свойством, с равной вероятностью выбрать любую из них» условием «... выбрать ту из них, которая обладает некоторым другим свойством». Повторяя этот прием, можно создать «генератор», порождающий только одно допустимое расписание. Так, генератор Г (FIFO → LRT) осуществлияет выбор операции по правилу FIFO; при наличии нескольких равноправных относительно этого правила операций — по правилу LRT; если и относительно этого

Таблица 7.2.1

Требования	Порядок прохождения приборов и время обслуживания (в скобках)					
1	3(1)	1(3)	2(6)	4(7)	6(3)	5(6)
2	2(8)	3(5)	5(10)	6(10)	1(10)	4(4)
3	3(5)	4(4)	6(8)	1(9)	2(1)	5(7)
4	2(5)	1(5)	3(5)	4(3)	5(8)	6(9)
5	3(9)	2(3)	5(5)	6(4)	1(3)	4(1)
6	2(3)	4(3)	6(9)	1(10)	5(4)	3(1)

Таблица 7.2.2

Требования	Порядок прохождения приборов и время обслуживания (в скобках)				
1	1(29)	2(9)	3(49)	4(62)	5(44)
2	1(43)	2(75)	4(69)	3(46)	5(72)
3	2(91)	1(39)	3(90)	5(12)	4(45)
4	2(81)	1(71)	5(9)	3(85)	4(22)
5	3(14)	2(22)	1(26)	4(21)	5(72)
6	3(84)	2(52)	5(48)	1(47)	4(6)
7	2(46)	1(61)	3(32)	4(32)	5(30)
8	3(31)	2(46)	1(32)	4(19)	5(36)
9	1(76)	4(76)	3(85)	2(40)	5(26)
10	2(85)	3(61)	1(64)	4(47)	5(90)
11	2(78)	4(36)	1(11)	5(56)	3(21)
12	3(90)	1(11)	2(28)	4(46)	5(30)
13	1(85)	3(74)	2(10)	4(89)	5(33)
14	3(95)	1(99)	2(52)	4(98)	5(43)
15	1(6)	2(61)	5(69)	3(49)	4(53)
16	2(2)	1(95)	4(72)	5(65)	3(25)
17	1(37)	3(13)	2(21)	4(89)	5(55)
18	1(86)	2(74)	5(88)	3(48)	4(79)
19	2(69)	3(51)	1(11)	4(89)	5(74)
20	1(13)	2(7)	3(76)	4(52)	5(45)

правила имеется несколько равноправных операций, то осуществляется равновероятная выборка любой из них.

Создание подобных генераторов не лишено практического интереса в условиях расчетов «вручную» и машинного решения задач большого размера.

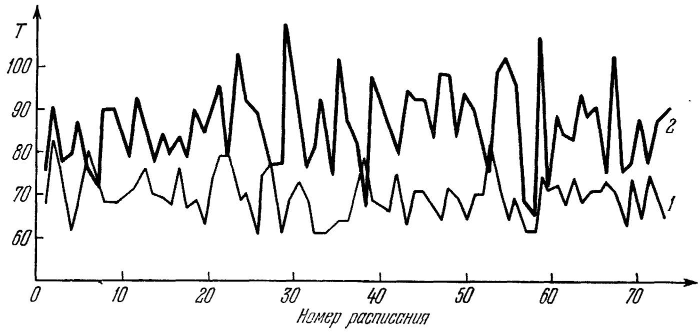

Рис. 7.2.1.

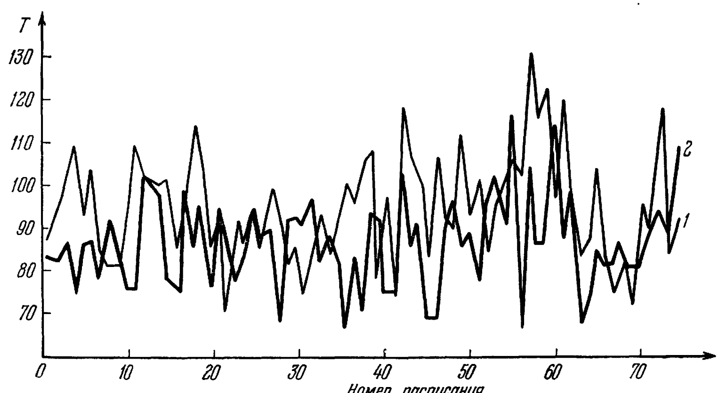

Рис. 7.2.2.

2.4. В качестве иллюстрации рассмотрим две простейшие задачи последовательного обслуживания n требований M приборами. Каждое требование k последовательно обслуживается приборами $1, 2, \ldots, M$. В первой задаче (задаче A) $n = 6$, $M = 6$ и значения t_{kL} приведены в табл. 7.2.1. Во второй задаче (задаче B) $n = 20$, $M = 5$ и значения t_{kL} приведены в табл. 7.2.2.

На рис. 7.2.1 приведены результаты экспериментов с использованием генераторов Г (FIFO) (линия 1) и $Г_R$ (FIFO) (линия 2), на рис. 7.2.2 — с использованием генераторов Г (R) (линия 1) и Г (A) (линия 2) для решения задачи A. Общее число построенных расписаний равно 75. Для каждого расписания приведено общее время T выполнения всех операций.

Решая эту задачу с использованием Г (FIFO $\to$ FOFO), получаем расписание с общим временем обслуживания всех требований $T = 88$.

§ 3. Случайный поиск с обучением

3.1. Программные генераторы допустимых расписаний могут быть использованы как при массовом решении серии однотипных, незначительно отличающихся друг от друга задач теории расписаний, так и для эпизодического решения существенно различных задач. Если в первом случае экономически оправдано создание специального генератора, то эта возможность, как правило, совершенно исключается во втором случае. Подобное положение вызывает серьезные затруднения.

Действительно, заранее оценить эффективность использования того или иного генератора при решении каждой конкретной задачи практически невозможно. Последовательное опробование всех имеющихся в распоряжении генераторов — достаточно трудоемкий процесс, не дающий к тому же сколько-нибудь определенных гарантий, так как совершенно не исключены ситуации, в которых все эти генераторы оказываются неэффективными. В частности, подобная ситуация может потребовать использования различных принципов выбора операций на разных шагах, что не предусмотрено в описанных выше генераторах. Это обстоятельство делает, по существу, необозримым набор генераторов, требуемых для решения различных задач теории расписаний.

Поэтому весьма перспективными являются попытки автоматического конструирования в процессе решения каждой конкретной задачи относительно сложного генератора, учитывающего в той или иной мере ее специфику.

3.2. Одна из наиболее удачных и легко реализуемых попыток этого рода заключается в следующем. Выбираются несколько различных принципов выборки операций (приоритетов) $\Pi_1, \Pi_2, \ldots, \Pi_r$ и неотрицательные числа $\alpha_1^p, \alpha_2^p, \ldots, \alpha_r^p$, $1 \leqslant p \leqslant |X|$, такие, что

$$\sum_{j=1}^{r} \alpha_j^p = 1.$$

В частности, полагаем

$$\alpha_j^p = \alpha_j$$

для всех p и j. Обозначим через $\Gamma(\alpha_1^p\Pi_1, \alpha_2^p\Pi_2, \ldots, \alpha_r^p\Pi_r)$ генератор, выбирающий на p-м шаге операцию по принципу Π_1 с вероятностью α_1^p, по принципу Π_2 с вероятностью α_2^p и т. д., по принципу Π_r с вероятностью α_r^p.

В процессе решения задачи проводится корректировка величин α_j^p в зависимости от качества генерируемых расписаний. Иными словами, организуется поиск в пространстве матриц $\|\alpha_j^p\|$ такого набора значений α_j^p, которому соответствует наилучшее расписание. Подобный поиск, естественно, может быть организован самыми разнообразными способами.

Простейший из них заключается в случайной выборке нескольких наборов α_j^p и последующем генерировании при каждом из этих наборов по одному или по серии расписаний. Более сложные включают элементы поиска в собственном смысле слова, когда изменения α_j^p, приводящие к улучшению качества расписания, закрепляются и развиваются в последующем.

При организации направленного поиска желательно обеспечить быструю сходимость процесса поиска и оптимальность получаемого в результате расписания. Эти условия являются достаточно жесткими для задач теории расписаний, что приводит к необходимости включения в процесс направленного поиска значительного объема элементов зондирования области поиска. Одна из возможных процедур такого рода состоит в следующем.

Пусть для простоты даны два генератора $\Gamma(\Pi_1)$ и $\Gamma(\Pi_2)$. Реализуется генерирование $s+1$ серий расписаний генераторами $\Gamma\left(\frac{k}{s}\Pi_1, \frac{s-k}{s}\Pi_2\right)$, $k = 0, 1, \ldots, s$. Из последних выбирается генератор ($k = k_0$), с использованием которого получено лучшее расписание. Затем множество операций разбивается на несколько групп (не обязательно содержащих одинаковое число элементов), из которых выбирается одна. Реализуется генерирование $s_1 + 1$ серий расписаний с использованием генераторов $\Gamma\left(\frac{k}{s_1}\Pi_1, \frac{s_1-k}{s_1}\Pi_2\right)$, $k = 0, 1, \ldots, s_1$, если отмечаемая операция принадлежит выбранной группе, в противном случае — генератором $\Gamma\left(\frac{k_0}{s}\Pi_1, \frac{s-k_0}{s}\Pi_2\right)$. Закрепляется

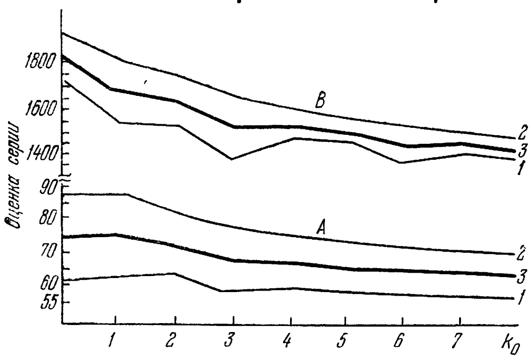

Рис. 7.3.1. Зависимость оценок серии расписаний от значения k_0. Задачи A и B.

значение $k = k_1$, при котором получилось лучшее расписание. Если все расписания оказываются хуже полученного на предыдущем шаге, то полагаем $k = k_0$.

В дальнейшем подобным образом поступаем со второй группой операций, сохраняя для первой группы $k = k_1$, для всех остальных $k = k_0$ и т. д. В случае необходимости процесс может быть повторен, начиная с первой группы, при сохранении для остальных полученных на предыдущих этапах значений k. Возможно последующее разбиение групп операций на более мелкие подгруппы.

Выбор значений k_0, $k_1, \ldots$ проводится по признаку «лучшее расписание в серии» на основании непосредственного генерирования серий расписаний для всех значений k из некоторого диапазона. Этот признак не является единственно возможным. На рис. 7.3.1 показана зависимость различных оценок серии расписаний от выбранных значений k_0 для генераторов Г (FIFO) и Γ_R (A $\to$ $\to$ FIFO) в условиях задач A и B. Оценки проводились по признакам «лучшее расписание в серии» (1), «лучшая в среднем серия» (2) и признаку, являющемуся усреднением первых двух (3). Значение $s = 8$. Последние два признака дают возможность (во всяком случае в условиях рассматриваемых задач) организовать направленный поиск k_0 из-за отсутствия явно выраженных локальных экстремумов.

Это соображение и желание получить генератор, дающий «хорошие в среднем» расписания, обусловили выбор третьего из указанных признаков.

Далее, поскольку разбиение множества операций на группы в значительной степени определяется интуицией экспериментатора, процедура была несколько видоизменена в результате введения «переменных» групп. В этом случае из n последовательно отмечаемых при построении расписания операций первые n_1 операций автоматически относились к первой группе, следующие n_2 операций — ко второй и т. д.

Таким образом, состав групп изменялся в процессе генерирования расписаний.

3.3. На рис. 7.3.2 приведены результаты работы генератора Г (FIFO) (линия 1) и генератора Г $\langle 6;\ 8\rangle$ (линия 2), полученного из Г (FIFO) и Γ_R (A $\to$ FIFO) описанным методом при $n_1 = n_2 = \ldots = n_6 = 6$ и $s = s_1 = \ldots \ldots = s_6 = 8$. Параметры последнего $k_1 = k_3 = k_4 = k_6 = 8$, $k_2 = 4$, $k_5 = 7$. Рассматривается задача A.

Рис. 7.3.3 иллюстрирует распределение числа расписаний заданной длительности, полученных с использованием генераторов Г $\langle 6;\ 8\rangle$ (линия 2) и Г $\langle 4;\ 8\rangle$ (линия 1), являющегося результатом применения описанного метода для случая $n_1 = n_2 = n_3 = n_4 = 9$ и $s = s_1 = \ldots = s_4 = 8$. Параметры последнего $k_1 = 4$, $k_2 = 3$, $k_3 = k_4 = 8$.

Рис. 7.3.4 аналогичен рис. 7.3.2 для случая задачи B и $n_1 = n_2 = \ldots = n_{20} = 5$, $s = s_1 = \ldots = s_{20} = 8$. Лучший результат — расписание с общим временем 1281. В обоих случаях для определения k_0 генерировалось 50 расписаний, для определения $k_1, k_2, \ldots$ — два расписания. Уточнение значений $k_1, k_2, \ldots$ в результате повторения процесса не проводилось.

3.4. Качество расписаний, получаемых в результате использования вероятностных «смесей» различных принципов выбора операций, определяется тем, насколько

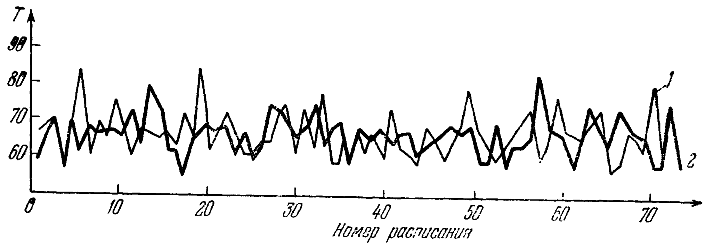

Рис. 7.3.2.

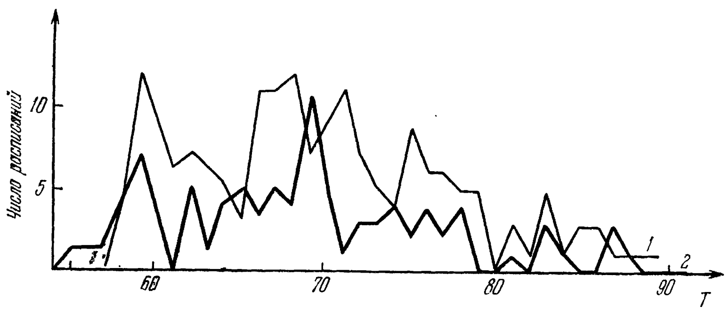

Рис. 7.3.3.

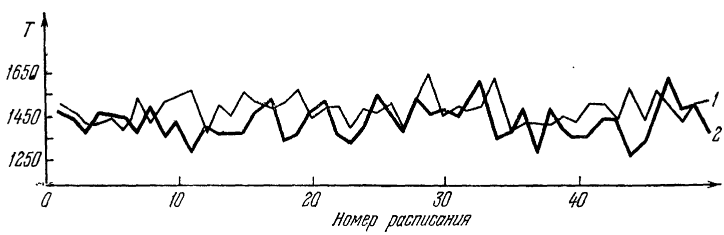

Рис. 7.3.4.

обосновано выбраны принципы для решения каждой конкретной задачи. Этот факт становится очевидным, если учесть, что организация эффективного поиска оптимального расписания с использованием большого числа правил приоритета встречает значительные затруднения.

Простейшим выходом из создавшегося положения является создание генераторов $\Gamma(\alpha_1^p\Pi_1, \alpha_2^p\Pi_2, \ldots, \alpha_r^p\Pi_r)$, в которых принципы $\Pi_1, \Pi_2, \ldots, \Pi_r$ взаимно дополняют друг друга в том смысле, что при различных значениях α_j^p могут быть получены все допустимые, точнее, все конкурентноспособные расписания. Такие генераторы могут быть использованы для решения разнообразных задач. Однако оценить заранее их эффективность не представляется возможным.

Указанное решение не является единственным. Рассмотрим одну из процедур поиска оптимального расписания, в основе которой лежит стремление желательным образом развивать ранговую структуру расписания.

Пусть построен один из сетевых графиков, порождаемых данным смешанным графом. Этот график, очевидно, однозначно определяет распределение операций по рангам. К первому рангу относят операции, каждая из которых не следует ни за одной операцией списка всех операций. Ко второму — операции, каждая из которых не следует ни за одной операцией списка, полученного в результате удаления операций первого ранга, и т. д.

Предположим, что известен способ построения нового графика, отличающегося от исходного небольшим изменением распределения всех или части операций по рангам. Этому графику соответствует, вообще говоря, некоторое новое значение общего времени выполнения всех операций T. Если это значение меньше ранее полученного, то последняя схема выбирается в качестве исходной для следующего шага. В противном случае исходную схему следует подвергнуть другим изменениям.

В реально испытанной процедуре этого рода все построения велись в классе активных расписаний.

На первом этапе с использованием $\Gamma(A)$ генерировалась серия расписаний, из которых выбиралось наилучшее и определялось распределение операций по рангам при этом расписании. Задавалось число $0 \leqslant \alpha \leqslant 1$ и генерировалось новое расписание по следующему принципу. Каждый раз, когда требовалось выбрать ту или иную операцию, из множества возможных претендентов выделялась группа операций с наименьшим относительно исходного расписания рангом. Генератор с вероятностью α проводил выбор среди операций этой группы и с вероятностью $1-\alpha$ — среди всех возможных претендентов.

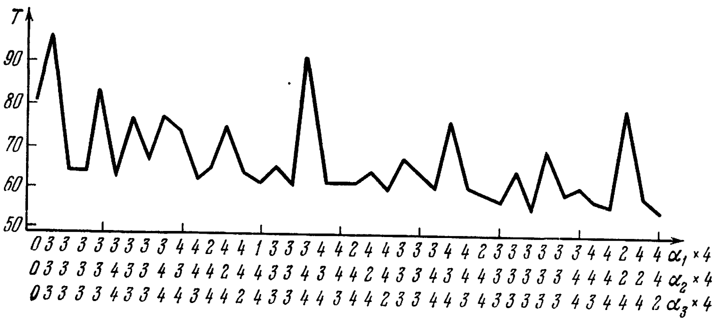

Рис. 7.3.5. Последовательное изменение ранговой структуры. Задача A.

Если полученное расписание оказывалось лучше, то оно выбиралось в качестве исходного для следующего шага. В противном случае при заданном значении α процесс повторялся p раз, после чего изменялось значение α.

Полученное в результате расписание на втором этапе аналогичным образом обрабатывалось «по частям». Иными словами, при построении нового расписания описанный принцип выбора операций использовался только на первых n_1 шагах, если не наступало улучшения — только на вторых n_2 шагах и т. д. На остальных шагах выбор производился из группы операций с наименьшим рангом.

В случае необходимости предусматривалась возможность повторения описанного процесса, начиная с первого этапа и используя полученное расписание в качестве исходного.

3.5. Решение задачи A этим методом показано на рис. 7.3.5. Начальному расписанию соответствовало $T = 82$. Поскольку на втором этапе проводилась последовательная обработка трех частей расписания ($n_1 = n_2 = n_3 = 12$), на этом рисунке приведены три значения α, равные соответственно α_1, α_2 и α_3.

На первом этапе $\alpha_1 = \alpha_2 = \alpha_3$ полагались равными 0,75 без последующих изменений. На втором — значения

Таблица 7.3.1

Требования	Время начала обслуживания требования приборами					
	Приборы					
	1	2	3	4	5	6
1	1	16	0	30	49	42
2	38	0	8	48	13	28
3	18	27	1	6	42	10
4	13	8	22	27	30	45
5	48	22	13	52	25	38
6	28	13	42	16	38	19

α_1, α_2 и α_3 изменялись независимо с шагом 0,25. Значение p на первом этапе равнялось 2, на втором 1.

Процесс поиска оптимального расписания с $T = 55$ (табл. 7.3.1) потребовал построения 40 расписаний.

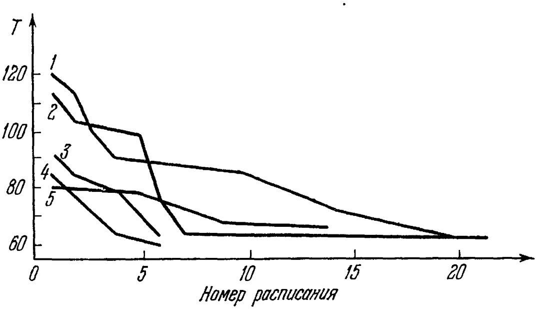

Рис. 7.3.6. Сходимость процесса поиска оптимального расписания. Задача A.

На рис. 7.3.6 приведены значения T последовательно улучшаемых расписаний, начиная с различных генерируе-

Таблица 7.3.2

Требования	Время начала обслуживания требования прибором				
	Приборы				
	1	2	3	4	5
1	158	199	218	301	373
2	187	254	501	382	606
3	935	842	983	1092	1073
4	774	683	898	1143	857
5	37	14	0	63	84
6	845	147	58	1137	225
7	553	488	656	767	866
8	316	208	14	363	417
9	82	943	739	173	1026
10	614	403	547	720	767
11	924	764	1073	888	970
12	443	534	267	576	678
13	678	933	824	1003	1092
14	454	562	357	622	720
15	76	86	452	523	156
16	348	0	1094	451	541
17	0	58	45	84	273
18	230	329	608	924	453
19	763	614	688	799	896
20	63	79	142	249	328

мых Г (А) расписаний. Приведено также общее число построенных в каждом случае расписаний. Эти данные получены в результате использования только первой части описанной процедуры при $\alpha = 0{,}81$ и $p = 3$.

Решение задачи B привело к расписанию с $T = 1178$. Значения $n_1 = n_2 = \ldots = n_{10} = 10$, значения $\alpha_1, \alpha_2, \ldots, \alpha_{10}$ изменялись так же, как и в случае задачи A. Всего потребовалось построить 69 расписаний. Незначительная модификация полученного расписания, проведенная вручную, позволила построить расписание с $T = 1165$, являющееся оптимальным в условиях рассматриваемой задачи (табл. 7.3.2).

§ 4. Линейные модели

В этом параграфе мы приведем несколько формулировок задачи построения оптимального (по быстродействию) расписания в терминах (частично) целочисленного линейного программирования.

4.1. Будем предполагать, что все операции (см. § 1) пронумерованы числами $1, 2, \ldots, R-1$, t_i — (известная) длительность операции с номером i, а $\underline{t}_i$ — (искомое) время начала этой операции. Введем в рассмотрение также «операцию» с номером R и нулевой длительностью. Предполагается, что эта операция следует во времени за каждой из остальных операций и время $\underline{t}_R$ начала ее «выполнения» совпадает с общим временем выполнения всех операций.

Для каждой пары операций i и j известно, являются ли эти операции зависимыми или независимыми. Если эти операции зависимы, то они не могут выполняться одновременно. Последовательность их выполнения либо задана, либо должна быть определена.

Выберем в качестве искомых величин значения $x_i = \underline{t}_i$ — времена начала выполнения операций.

Если известно, что операция j следует во времени за операцией i, то, очевидно, должно выполняться неравенство

$$x_j - x_i \geqslant t_i. \tag{4.1}$$

Аналогично, если операции i и j одновременно не могут выполняться, но последовательность их выполнения заранее не оговорена, то должно быть справедливо одно из неравенств

$$x_j - x_i \geqslant t_i \quad \text{или} \quad x_i - x_j \geqslant t_j. \tag{4.2}$$

Отнесем каждой паре таких операций i и j величину y_{ij}, принимающую значение $y_{ij} = 1$, если операцию i решено выполнять раньше операции j, и $y_{ij} = 0$ в противном случае. Тогда условие (4.2) можно записать в виде

$$By_{ij} - t_j \geqslant x_j - x_i \geqslant t_i - B(1 - y_{ij}), \tag{4.3}$$

где B — достаточно большое число.

Задача построения оптимального по быстродействию расписания, таким образом, сводится к нахождению значений x_i и y_{ij}, которые удовлетворяют неравенствам (4.1) для всех $(i, j) \in \vec{U}$ и неравенствам (4.3) для всех $[i, j] \in U$ и при которых величина $x_R = \underline{t}_R$ достигает наименьшего значения.

4.2. В практическом отношении достаточно рассматривать ситуации, в которых времена t_i выполнения операций — целые числа. Пронумеруем временные промежутки единичной длины вида $(\theta - 1, \theta]$ числами $\theta = 1, 2, \ldots, T$, где T — интервал планирования.

Обозначим через $x_{i\theta}$ величину, принимающую значение 1, если i-я операция выполняется в θ-м временном промежутке, и значение 0 в противном случае.

Для того чтобы за T единиц времени могли быть выполнены все операции, необходимо, чтобы для всех $i = \overline{1, R}$ выполнялось условие

$$\sum_{\theta=1}^{T} x_{i\theta} = t_i. \tag{4.4}$$

Поскольку каждая операция по условию выполняется непрерывно, то необходимо, чтобы неравенство

$$t_i x_{i\theta_0} - t_i x_{i\theta_0+1} + \sum_{\theta=\theta_0+2}^{T} x_{i\theta} \leqslant t_i \tag{4.5}$$

было справедливо для каждой операции i и всех $1 \leqslant \theta_0 \leqslant T - 2$.

Если операции i и j не могут выполняться одновременно (т. е. $[i, j] \in U$), то

$$x_{i\theta} + x_{j\theta} \leqslant 1 \tag{4.6}$$

для всех $\theta = \overline{1, T}$.

Если, кроме того, операция j по условию следует за операцией i (т. е. $(i, j) \in \vec{U}$), то

$$t_i x_{j\theta_0} - \sum_{\theta=1}^{\theta_0-1} x_{i\theta} \leqslant 0 \tag{4.7}$$

для всех $2 \leqslant \theta_0 \leqslant T$.

Задача заключается в нахождении таких значений $x_{i\theta}$, которые удовлетворяют перечисленным неравенствам и при этом величина T минимальна.

4.3. Своеобразным объединением обоих описанных выше подходов служит модель, в которой переменные $z_{i\theta}$ принимают значения 1 или 0 в зависимости от того, начинается или нет выполнение операции i в θ-м промежутке времени. Значения t_i по-прежнему предполагаются целочисленными. В этом случае, очевидно, необходимо, чтобы

$$\sum_{\theta=1}^{T} z_{i\theta} = 1 \tag{4.8}$$

для всех операций i.

Если операция i предшествует операции j (т. е. $(i, j) \in \vec{U}$), то должно выполняться неравенство

$$\sum_{\theta=1}^{\theta_0} z_{j\theta} \leqslant \sum_{\theta=1}^{\theta_0-t_i} z_{i\theta} \tag{4.9}$$

для всех $\theta_0 > t_i$ и

$$\sum_{\theta=1}^{\theta_0} z_{j\theta} = 0, \tag{4.10}$$

если $\theta_0 \leqslant t_i$.

Наконец, если операции i и j не могут выполняться одновременно, но порядок их выполнения заранее не оговорен (т. е. $[i, j] \in U$), то необходимо, чтобы выполнялось неравенство

$$\sum_{\theta=\theta_i}^{\theta_0} z_{i\theta} + \sum_{\theta=\theta_j}^{\theta_0} z_{j\theta} \leqslant 1, \tag{4.11}$$

где $\theta_k = \max(1, \theta_0 - t_k + 1)$ и $1 \leqslant \theta_0 \leqslant T$. По-прежнему задача состоит в минимизации значения T.

Линейные модели практически всех задач теории расписаний имеют большую размерность и содержат значительное число целочисленных переменных.

§ 5. Библиографическая справка

Сетевые формы представления рассматриваемых в этой главе ситуаций предложены в работах Л. П. Матюшкова и В. С. Танаева [113], Е. Балаша [229], В. В. Шкурбы [202], Дж. Геллера и Дж. Логемана [300].

В основу изложения §§ 2, 3 положена работа [113]. Генератор Г (А) описан (в несколько иной терминологии) Б. Гиффлером и Дж. Томпсоном [281]. Р. Конвей и У. Максвелл [82] исследовали правило «кратчайшей операции». Генератор Г (FIFO → LRT) использовался для решения задач календарного планирования В. М. Португалом [138]. Исследование влияния правил приоритета на отдельные стороны производственного процесса проведено А. В. Бондаренко [19], В. Гере [279]. Различные по сложности правила предпочтения предложены Э. Г. Иоффе [72], П. К. Выгнаном и Б. И. Кузиным [40], Д. Р. Джексоном [60], Н. Б. Мироносецким [118] и другими авторами. Идея построения рандомизированных генераторов наиболее четко описана в работе В. В. Шкурбы [205].

Сравнительный экспериментальный анализ разнообразных правил предпочтения провели Р. Конвей, Б. Джонсон, У. Максвелл [81, 253], Б. Гиффлер, Дж. Томпсон, В. Ван-Нессе [46], В. Гере [279]. Условия иллюстративных задач A и B заимствованы из работы [191].

Вопросам выбора правил предпочтения, оптимальной их композиции и настройке посвящены работы Г. Фишера и Г. Л. Томпсона [191], У. Максвелла и М. Мехры [340], Т. П. Подчасовой [137], М. Х. Андгуладзе [7], Л. В. Кочнева [86], А. Б. Ароновича [10]. Моделирующая система для исследования эффективности приоритетов предложена в [243].

Линейная модель (4.1), (4.3) описана в работе А. Манна [338], (4.4)—(4.7) — в работе Е. Боумана [241], (4.8)—(4.11) — в работе А. Л. Лурье [110]. Дж. Данциг [257] построил линейную модель с использованием специфического сетевого представления задачи. В [322] предложена модель, в которой все коэффициенты в ограничениях и все переменные булевы.

Систематическое исследование применимости методов линейного программирования для решения задач календарного планирования проведено Ю. В. Глебским и В. Н. Шевченко [50, 51], [199—201]. Этим же вопросам посвящены работы [75, 324, 381, 392]. Распределение работ по параллельным машинам рассмотрено в [320, 323]. Несколько частных ситуаций оптимального планирования рассмотрено в [164, 233, 252, 312, 329, 351, 367, 393].

Вопросы существования расписаний, реализующих процесс обслуживания в соответствии с заданными технологическими маршрутами, рассматривали Л. И. Фейгин [187] и Г. П. Акимов [3].

Попытки использовать декомпозиционный подход к решению большеразмерных задач предприняты в [221, 275, 294].

Метод ветвей и границ (применительно к сетевой постановке рассматриваемой задачи) описан в [229, 398].

С. Е. Эйкерс и Дж. Фридман [217, 218] предложили простой графический метод решения задачи оптимального (по быстродействию) обслуживания двух требований M приборами. Этот метод получил дальнейшее усовершенствование и обобщение (на случай произвольного числа требований) в работах [47, 295, 346, 420, 421]. Специфический подход предложен в [5]. Верхние оценки длины оптимального расписания установлены Н. И. Глебовым и В. А. Перепелицей [48, 49], В. Э. Хенкиным [193].

ГЛАВА 8
МНОГООПЕРАТОРНЫЕ ПРОЦЕССЫ ОБСЛУЖИВАНИЯ

Рассматривается ряд вопросов, связанных с оптимизацией процесса функционирования систем с M последовательными приборами, ограниченным числом операторов переноса, осуществляющих передачу требований от одного прибора к другому, и заданными временными графиками обслуживания требований.

§ 1. Процессы с неограниченным числом операторов переноса

1.1. Рассмотрим обслуживающую систему, содержащую M последовательных приборов, на вход которой в момент времени $d = 0$ поступает n требований r типов ($n = \sum_{k=1}^{r} n_k$, где n_k — число требований k-го типа). Время обслуживания требования k-го типа L-м прибором равно t_{kL}. Для осуществления передачи (транспортировки) требований от одного прибора к другому используется s однотипных операторов переноса. Обозначим время, затрачиваемое оператором на перемещение обслуживаемого требования k-го типа от L-го к $(L + 1)$-му прибору, через σ_{kL}, а время перемещения незагруженного оператора от $(L_1 + 1)$-го прибора к L_2-му прибору через $\delta_{L_1L_2}$. Пусть θ_L^{ij} означает время, необходимое для переналадки L-го прибора с обслуживания требования i-го типа на обслуживание требования типа j. Предполагается, что процесс обслуживания каждого требования непрерывен. Если обслуживание некоторого p-го по порядку требования, принадлежащего k-му типу, начинается в момент времени ν_p^k первым прибором, то процесс протекает непрерывно до момента времени $\nu_p^k + t_{k1}$. Затем в течение σ_{k1} единиц времени один из операторов переноса доставляет это требование ко второму прибору. В момент времени $\nu_p^k + t_{k1} + \sigma_{k1}$ начинается процесс обслуживания этого требования вторым прибором и т. д., пока в момент времени $\nu_p^k + z_M^k$, где $z_L^k = \sum_{Q=1}^{L} t_{kQ} + \sum_{Q=1}^{L-1} \sigma_{kQ}$, не будет завершен процесс обслуживания этого требования. Каждый прибор одновременно обслуживает, а каждый оператор переноса осуществляет транспортировку не более одного требования. Все приборы обслуживают требования в одной и той же последовательности.

Для описания процесса функционирования систем такого рода достаточно задать моменты $\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}$ времени начала обслуживания каждого из n требований и временные графики перемещения каждого из s операторов переноса.

В этом параграфе мы рассмотрим вопросы построения оптимальных по быстродействию расписаний обслуживания всех требований в предположении, что число операторов переноса достаточно для реализации любого или, по крайней мере, найденного оптимального расписания. Затем рассмотрим задачу определения наименьшего числа операторов переноса, необходимых для реализации построенного расписания.

1.2. Пусть при заданном порядке обслуживания требований в системе p-е требование принадлежит i-му типу, а $(p+1)$-е — j-му типу. Для того чтобы эти два требования не обслуживались одновременно L-м прибором, очевидно, необходимо и достаточно, чтобы (с учетом переналадки)

$$\nu_{p+1}^j - \nu_p^i \geqslant z_L^i - z_L^j + t_{jL} + \theta_L^{ij}, \tag{1.1}$$

для всех $L = \overline{1, M}$.

Если каждая пара чисел $\nu_1^{i_1} < \nu_2^{i_2} < \ldots < \nu_n^{i_n}$, где $i_k \in \{1, 2, \ldots, r\}$, удовлетворяет условию (1.1) при всех $L = \overline{1, M}$, то набор этих чисел будем называть расписанием и обозначать $R\,\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$. Общее время обслуживания всех требований при этом равно $\nu_n^{i_n} + z_M^{i_n}$. Расписание $R\,\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$ будем называть *оптимальным* (по быстродействию), если $\nu_n^{i_n} + z_M^{i_n} \leqslant \nu_n^{j_n} + z_M^{j_n}$, где $R\{\nu_1^{j_1}, \nu_2^{j_2}, \ldots, \nu_n^{j_n}\}$ — произвольное расписание. Оптимальное расписание будем обозначать

$$R_{\text{опт}}\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}.$$

Положим

$$l_{ij} = \max_{1\leqslant L\leqslant M}\{z_L^i - z_L^j + t_{jL} + \theta_L^{ij}\}, \qquad 1\leqslant i,\ j\leqslant r. \qquad (1.2)$$

Поскольку обслуживание каждого требования на всех приборах идет непрерывно и в одной и той же последовательности, то каждая пара чисел $\{\nu_p^{i_p}, \nu_{p+1}^{i_{p+1}}\}$ из набора $R_{\text{опт}}\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$ должна удовлетворять условию

$$\nu_{p+1}^{i_{p+1}} - \nu_p^{i_p} = l_{i_p i_{p+1}}, \qquad 1\leqslant p\leqslant n-1. \qquad (1.3)$$

Таким образом, если число операторов достаточно велико, то оптимальное расписание полностью определяется порядком, в котором требования поступают на обслуживание. Этот порядок естественно называть оптимальным. Формула (1.3) дает возможность определить моменты времени начала обслуживания требований и, в силу непрерывности процесса обслуживания каждого требования, календарные сроки обслуживания каждого требования каждым прибором.

Рассмотрим полный ориентированный симметрический граф $(X, \vec{U})$, где $X = \{0, 1, \ldots, r\}$, а $\vec{U}$ — множество дуг (i, j), $0\leqslant i, j\leqslant r$. Каждой дуге (i, j) отнесем число l_{ij}, которое будем называть *длиной* этой дуги, полагая $l_{0j}=0$, $l_{j0} = z_M^j$, $j=\overline{1, r}$, и $l_{00}=\infty$. Для нахождения оптимального порядка обслуживания требований, очевидно, достаточно найти контур в графе $(X, \vec{U})$, проходящий через каждую вершину k ровно n_k раз ($n_0 = 1$) и имеющий наименьшую длину. Под длиной контура понимается суммарная длина составляющих его дуг.

Тем самым, если $n_k = 1$, $k=\overline{1, r}$, то задача определения оптимального порядка обслуживания требований сводится к задаче коммивояжера. Если $n_k\geqslant 1$, $k=\overline{1, r}$, то получаем естественное обобщение задачи коммивояжера. В этом случае вместо поиска контура наименьшей длины в графе $(X, \vec{U})$, проходящего через каждую вершину k

ровно n_k раз, можно осуществить поиск гамильтонова контура наименьшей длины в графе $(X_1, \vec{U}_1)$. Здесь множество вершин $X_1 = \{0, 1^{(1)}, \ldots, 1^{(n_1)}, \ldots, r^{(1)}, \ldots, r^{(n_r)}\}$, $\vec{U}_1$ — множество всех дуг, соединяющих эти вершины. Каждой дуге $(i^{(k)}, j^{(l)})$ отнесено число l_{ij}, длина этой дуги, $1 \leqslant i, j \leqslant r$, $1 \leqslant k \leqslant n_i$, $1 \leqslant l \leqslant n_j$, $l_{0j^{(l)}} = 0$ и $l_{i^{(k)}0} = z_M^i$. Однако сведение исходной задачи к задаче коммивояжера при $n_k \geqslant 1$, особенно при $n \gg r$, нецелесообразно из-за отсутствия эффективных методов решения последней.

1.3. Опишем один из методов решения указанной обобщенной задачи коммивояжера, так называемый *метод последовательного восстановления связности*. В этом методе существенно используется тот факт, что формулировка (обобщенной) задачи коммивояжера в терминах целочисленного линейного программирования отличается от соответствующей формулировки (обобщенной) задачи о назначениях только наличием ограничений, обеспечивающих связность получаемых в результате контуров. Поскольку для решения последней известны весьма эффективные методы, то появляется реальная возможность организации поиска решения исходной задачи среди допустимых решений соответствующей задачи о назначениях.

Обозначая через x_{ij} число дуг вида (i, j), содержащихся в искомом контуре, аналогично гл. 3, § 4, п. 4.2, рассматриваемую (обобщенную) задачу коммивояжера можно записать в виде

$$\sum_{i, j=0}^{r} l_{ij} x_{ij} \to \min, \tag{1.4}$$

$$\sum_{j=0}^{r} x_{ij} = \sum_{j=0}^{r} x_{ji} = n_i, \quad i = \overline{0, r}, \tag{1.5}$$

$$x_{ij} \geqslant 0, \quad i, j = \overline{0, r}. \tag{1.6}$$

Кроме того, для каждого подмножества $\tilde{N}$ множества $\{0, 1, 2, \ldots, r\}$, содержащего $1 \leqslant \mu \leqslant [(r+1)/2]$ элементов, должно выполняться условие

$$\sum_{i, j \in \tilde{N}} x_{ij} \leqslant \sum_{i \in \tilde{N}} n_i - 1. \tag{1.7}$$

Совокупность условий (1.7) обеспечивает связность получаемого в результате контура. Переменные x_{ij} могут принимать только целочисленные значения.

Метод последовательного восстановления связности обеспечивает целенаправленный поиск решения задачи (1.4) — (1.7) среди допустимых решений (планов) задачи (1.4) — (1.6). При этом на каждом шаге осуществляется решение задачи (1.4) — (1.6) с некоторой совокупностью дополнительных ограничений вида $x_{ij} \leqslant b_{ij}$, где b_{ij} — известные целые неотрицательные числа.

Таблица 8.1.1

$i \backslash j$	0	1	2	3	4
0	∞	0	0	0	0
1	26	5	6	6	6
2	38	13	7	8	10
3	34	11	9	8	9
4	32	7	6	4	6

Описание метода проиллюстрируем примером, в котором требуется найти контур наименьшей длины в полном графе (с петлями), содержащем 5 вершин — 0, 1, 2, 3, 4, длины дуг которого даны в табл. 8.1.1, при условии, что этот контур проходит через вершину 0 один раз, вершину 1 — четыре раза, вершину 2 — восемь раз, вершину 3 — пять раз, вершину 4 — семь раз.

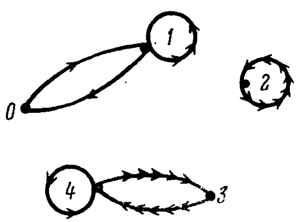

Рис. 8.1.1.

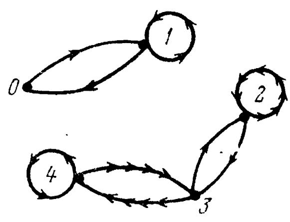

Рис. 8.1.2.

Пусть $\{x_{ij}^0\}$ — (целочисленное) решение задачи (1.4)—(1.6). Если $\{x_{ij}^0\}$ удовлетворяет условиям (1.7), то $\{x_{ij}^0\}$ определяет искомый связный контур.

В рассматриваемом случае получаем три изолированных контура (рис. 8.1.1). Суммарная длина дуг равна 174.

Выделим одну из компонент связности, содержащую, для определенности, наименьшее число q различных дуг.

Пусть эта компонента содержит $a_{\alpha\beta}$ дуг (α, β), $a_{\gamma\delta}$ дуг (γ, δ) и т. д. $a_{\nu\mu}$ дуг (ν, μ) и не содержит других дуг.

В рассматриваемом случае наименьшее число различных дуг, а именно дугу (2, 2), содержит компонента связности с вершиной 2.

Искомое решение должно удовлетворять хотя бы одному из ограничений $x_{\alpha\beta} \leqslant a_{\alpha\beta} - 1$, $x_{\gamma\delta} \leqslant a_{\gamma\delta} - 1, \ldots$ $\ldots, x_{\nu\mu} \leqslant a_{\nu\mu} - 1$. В результате приходим к q задачам (1.4) — (1.6) с одним дополнительным ограничением вида $x_{ij} \leqslant b_{ij}$. Решая каждую из них, получаем (целочисленные) значения $\{x_{ij}\}$, равные $\{x_{ij}^1\}$, $\{x_{ij}^2\}, \ldots, \{x_{ij}^q\}$, и значения линейной формы (1.4), равные $c^1, c^2, \ldots, c^q$ соответственно.

Если $c^l = \min\limits_{1 \leqslant k \leqslant q} (c^k)$ и $\{x_{ij}^l\}$ определяет связный контур, то $\{x_{ij}^l\}$ — искомое решение задачи. В противном случае $\{x_{ij}^l\}$ определяет несколько изолированных контуров и, повторяя приведенные рассуждения, приходим к q_l задачам минимизации линейной формы (1.4) при ограничениях (1.5) — (1.6), ранее наложенном дополнительном ограничении вида $x_{ij} \leqslant b_{ij}$ и налагаемом на данном шаге еще одном ограничении вида $x_{ij} \leqslant b_{ij}$.

В условиях рассматриваемого примера, решая задачу (1.4) — (1.6) с дополнительным ограничением $x_{22} \leqslant 7$, получаем два изолированных контура с суммарной длиной дуг, равной 177 (рис. 8.1.2). Компонента связности с вершинами 0 и 1 содержит наименьшее число различных дуг — дуги (0, 1), (1, 0) и (1, 1). Решаем три задачи (1.4)—(1.6) при $x_{22} \leqslant 7$ и дополнительном ограничении $x_{01} = 0$, $x_{10} = 0$ и $x_{11} \leqslant 2$ соответственно. Полученные в результате решения этих задач контуры представлены на рис. 8.1.3, *а*, *б*, *в*. Суммарная длина дуг равна а) 178, б) 179 и в) 181.

Ветвящийся процесс наложения дополнительных ограничений вида $x_{ij} \leqslant b_{ij}$ на задачу (1.4) — (1.6) представим древовидной структурой. При выборе наименьшего значения линейной формы сравниваются между собой значения этой формы, соответствующие концевым вершинам этого дерева. На каждом шаге ветвлению подвергается одна из концевых вершин рассматриваемого на данном шаге дерева.

Этот процесс конечен и завершается наступлением ситуации, при которой значения $\{x_{ij}\}$, соответствующие наименьшему из рассматриваемых значений линейной формы, определяют связный контур.

В рассматриваемом случае контур, изображенный на рис. 8.1.3, *а*), является искомым.

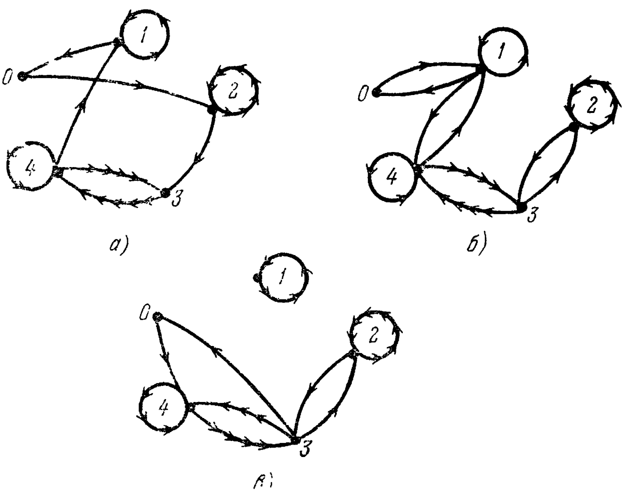

Рис. 8.1.3.

Метод последовательного восстановления связности позволяет получить решение (обобщенной) задачи коммивояжера сравнительно большого размера. На рис. 8.1.4 представлено решение задачи (1.4) — (1.6) в условиях табл. 8.1.2. Суммарная длина дуг равна 4102. Искомый контур представлен на рис. 8.1.5. Длина его равна 4451. Он получен в результате наложения на задачу (1.4) — (1.6) дополнительных ограничений $x_{19,19} = 0$, $x_{14,14} \leqslant 2$, $x_{13,5} \leqslant 2$, $x_{11,17} \leqslant 2$.

1.4. Пусть $R\,\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$ — произвольное (в частности, оптимальное) расписание обслуживания требований. Рассмотрим задачу определения наименьшего числа s операторов переноса, необходимых для реализации этого расписания.

Расписание $R\{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$ однозначно определяет календарные сроки переноса (транспортировки)

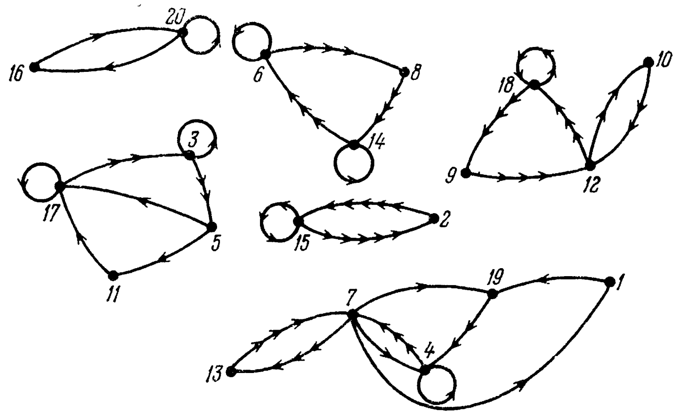

Рис. 8.1.4.

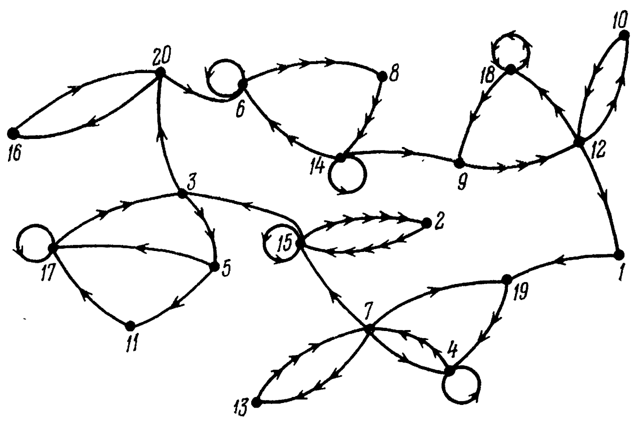

Рис. 8.1.5.

требований от прибора к прибору, т. е. множество W временных интервалов

$$w_{i_pL} = (\nu_p^{i_p} + z_L^{i_p},\ \nu_p^{i_p} + z_L^{i_p} + \sigma_{i_pL}). \qquad (1.8)$$

Зададим на множестве W отношение строгого порядка. Будем говорить, что интервал $w_{i_{p_1}L_1}$ предшествует

Таблица 8.1.2

i \ j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	276	142	314	150	178	284	70	163	312	283	467	160	515	237	174	416	141	545	312	410
1	174	625	700	374	890	346	218	471	516	520	180	292	372	613	21	662	145	564	130	210
2	807	140	80	437	200	372	464	413	662	307	633	373	870	682	156	713	36	326	210	317
3	412	998	516	34	823	173	62	712	172	340	259	918	686	145	484	160	707	876	46	562
4	267	819	57	152	912	997	993	722	759	264	814	297	252	319	399	662	392	601	648	154
5	815	390	139	520	828	73	150	264	860	517	630	496	416	88	689	900	161	943	156	154
6	135	625	174	93	890	341	698	284	463	485	961	763	94	137	546	516	880	343	127	215
7	364	170	362	291	455	46	339	154	172	752	467	333	578	302	607	174	703	440	178	743
8	163	910	773	477	264	563	401	172	473	214	770	548	957	131	538	438	94	32	778	763
9	700	524	708	312	805	778	283	336	221	927	538	39	571	597	368	171	311	318	664	543
10	680	150	971	620	79	691	792	609	814	637	982	870	171	256	942	190	810	522	144	139
11	264	488	503	668	852	976	700	213	80	71	519	147	853	281	967	698	281	850	413	320
12	141	645	823	417	256	148	97	417	256	270	163	192	656	162	154	441	877	149	393	247
13	413	145	702	172	636	982	562	32	188	162	496	297	154	117	930	198	913	802	607	316
14	961	16	181	274	414	903	139	481	730	381	466	905	662	244	68	225	151	545	650	701
15	312	774	188	186	453	326	176	764	801	313	172	434	224	967	342	416	935	447	150	37
16	178	665	163	227	107	271	154	282	197	399	114	721	307	818	593	613	50	319	371	413
17	398	145	451	482	278	648	215	465	196	520	806	81	518	333	238	165	320	16	750	197
18	116	162	237	181	270	500	120	453	268	665	719	947	755	668	524	174	156	918	437	215
19	413	340	140	686	571	513	828	152	662	538	239	264	215	181	516	35	229	171	215	128
n_i	1	5	3	4	2	4	6	3	3	2	1	5	3	4	8	1	3	7	2	2

интервалу $w_{t_{p_2}L_2}$, если

$$\nu_{p_2}^{i_{p_2}} - \nu_{p_1}^{i_{p_1}} \geqslant z_{L_1}^{i_{p_1}} - z_{L_2}^{i_{p_2}} + \sigma_{i_{p_1}L_1} + \delta_{L_1L_2}. \qquad (1.9)$$

Поскольку одним оператором переноса можно осуществить те и только те перемещения требований, соответствующие временные интервалы которых образуют линейно упорядоченное подмножество множества W, то для определения наименьшего числа операторов переноса, требуемых для реализации расписания $R\ \{\nu_1^{i_1}, \nu_2^{i_2}, \ldots, \nu_n^{i_n}\}$, достаточно найти наименьшее число попарно непересекающихся линейно упорядоченных подмножеств (цепей) множества W, покрывающих это множество. По теореме 2.1 гл. 2 наименьшее число таких цепей равно максимальному числу попарно несравнимых элементов множества W. Эти элементы и сами цепи могут быть определены методами линейного программирования (см. гл. 2, § 2, п. 2.3).

Таблица 8.1.3

k	t_{k1}	t_{k2}	t_{k3}	t_{k4}	t_{k5}	t_{k6}	σ_{k1}	σ_{k2}	σ_{k3}	σ_{k4}	σ_{k5}	n_k	θ_L^{k1}	θ_L^{k2}	θ_L^{k3}	θ_L^{k4}
1	5	3	5	4	2	0	1	1	1	2	2	4	0	1	1	1
2	4	6	2	7	6	5	2	2	1	2	1	8	1	0	1	1
3	8	3	6	4	6	2	1	1	1	1	1	5	1	1	0	1
4	3	5	4	6	3	3	2	1	1	2	2	7	1	1	1	0

1.5. В качестве примера рассмотрим задачу построения оптимального расписания процесса обслуживания требований четырех типов, параметры которых приведены в табл. 8.1.3.

Значения l_{ij}, вычисленные по формуле (1.2), приведены в табл. 8.1.1.

Используя полученное в предыдущем пункте решение соответствующей (обобщенной) задачи коммивояжера, получаем оптимальный порядок обслуживания требований

$$(2, 2, 2, 2, 2, 2, 2, 2, 3, 4, 3, 4, 3, 4, 3, 4, 3, 4, 4, 4, 1, 1, 1, 1)$$

(см. рис. 8.1.3, *а*). Сначала последовательно обслуживаются все требования типа 2, затем попеременно типа 3 и 4, затем требования типа 4 и, наконец, требования типа 1.

Таблица 8.1.4

Номер требования	Тип требования	Прибор 1		Прибор 2		Прибор 3		Прибор 4		Прибор 5		Прибор 6	
		нач.	оконч.	нач.	оконч.	нач.	оконч.	нач.	оконч.	нач.	оконч.	нач.	оконч.
1	2	0	4	6	12	14	16	17	24	26	32	33	38
2	2	7	11	13	19	21	23	24	31	33	39	40	45
3	2	14	18	20	26	28	30	31	38	40	46	47	52
4	2	21	25	27	33	35	37	38	45	47	53	54	59
5	2	28	32	34	40	42	44	45	52	54	60	61	66
6	2	35	39	41	47	49	51	52	59	61	67	68	73
7	2	42	46	48	54	56	58	59	66	68	74	75	80
8	2	49	53	55	61	63	65	66	73	75	81	82	87
9	3	57	65	66	69	70	76	77	81	82	88	89	91
10	4	66	69	71	76	77	81	82	88	90	93	95	98
11	3	70	78	79	82	83	89	90	94	95	101	102	104
12	4	79	82	84	89	90	94	95	101	103	106	108	111
13	3	83	91	92	95	96	102	103	107	108	114	115	117
14	4	92	95	97	102	103	107	108	114	116	119	121	124
15	3	96	104	105	108	109	115	116	120	121	127	128	130
16	4	105	108	110	115	116	120	121	127	129	132	134	137
17	3	109	117	118	121	122	128	129	133	134	140	141	143
18	4	118	121	123	128	129	133	134	140	142	145	147	150
19	4	124	127	129	134	135	139	140	146	148	151	153	156
20	4	130	133	135	140	141	145	146	152	154	157	159	162
21	1	137	142	143	146	147	152	153	157	159	161	—	—
22	1	142	147	148	151	152	157	158	162	164	166	—	—
23	1	147	152	153	156	157	162	163	167	169	171	—	—
24	1	152	157	158	161	162	167	168	172	174	176	—	—

Календарные сроки обслуживания каждого требования каждым прибором приведены в табл. 8.1.4. Предполагается, что процесс обслуживания требований начинается в момент времени $d = 0$. Общее время обслуживания всех требований равно 178 (с учетом транспортировки требований типа 1 от 5-го к 6-му прибору). Пронумеруем интервалы w_{i_pL}, определяемые табл. 8.1.4, приписывая каждому интервалу w_{i_pL} номер $5(p-1)+L$. Так, интервал (4, 6) получает номер 1, интервал (12, 14) — номер 2 и т. д., интервал (32, 33) — номер 5 и т. д., интервал (162, 163) — номер 113. Число интервалов равно 120, и следовательно, множество W содержит 120 элементов.

Используя соотношение (1.9), на множестве W зададим отношение частичного порядка. Для определенности, положим $\delta_{L_1L_2} = 0$ для всех $1 \leqslant L_1, L_2 \leqslant M$. Таким образом, интервал с номером 1 предшествует интервалу с номером 2, интервал с номером 11 предшествует интервалу с номером 9 и т. д. Интервалы с номерами 21 и 5 оказываются несравнимыми.

Максимальное число попарно несравнимых элементов множества W равно четырем. Такими элементами являются, в частности, интервалы с номерами 79, 83, 87 и 91. Следовательно, если все $\delta_{L_1L_2} = 0$, то оптимальное расписание можно реализовать с использованием четырех операторов переноса и нельзя реализовать с использованием меньшего числа операторов.

Отметим, что если все $\delta_{L_1L_2} > 0$, то оптимальное расписание нельзя реализовать с числом операторов, меньшим пяти. Действительно, в этом случае интервалы с номерами 40, 44, 48, 52, 56 оказываются попарно несравнимыми.

Для нахождения допустимых графиков передвижения операторов от прибора к прибору и их загрузки во времени достаточно найти минимальное D-разложение множества W. Вопросы рационального использования операторов переноса рассмотрены в конце следующего параграфа.

§ 2. Процессы с ограниченным числом операторов переноса

Построение оптимального (по быстродействию) расписания процесса обслуживания требований в системе, описанной в предыдущем параграфе, в случае, когда число операторов s оказывается недостаточным для реализации в системе допустимого расписания, представляет значительные затруднения. Мы ограничимся рассмотрением практически важного частного случая этой задачи, предполагая, что все обслуживаемые требования однотипны. Такого рода ситуации имеют место, в частности, в гальванических цехах промышленных предприятий. В качестве критерия оптимальности расписания, как и раньше, выберем общее время обслуживания всех требований, если число этих требований конечно, и так называемую плотность расписания, если процесс обслуживания требований бесконечен.

2.1. Пусть, как и раньше, ν_p означает момент начала обслуживания p-го по порядку требования, $p = \overline{1, n}$. Найдем условия, которым должны удовлетворять числа ν_1, ν_2, . . ., ν_n, чтобы процесс обслуживания требований можно было реализовать в системе с заданными параметрами (поскольку требования одного типа, у всех параметров опущены индексы типа).

Так как каждый прибор системы не может одновременно обслуживать более одного требования, то необходимо, чтобы (с учетом переналадки)

$$\begin{gathered}\nu_{p+1} - \nu_p \geqslant t,\\ p = 1,\end{gathered} \tag{2.1}$$

где $t = \max\limits_{1 \leqslant L \leqslant M}(t_L + \theta_L)$ (условие 1).

Числа ν_1, ν_2, . . ., ν_n определяют множество W интервалов

$$\begin{gathered}w_{pL} = (\nu_p + z_L,\ \nu_p + z_L + \sigma_L),\\ p = \overline{1, n}, \quad L = \overline{1, M}.\end{gathered} \tag{2.2}$$

Будем говорить, что интервалы $w_{p_1L_1}$ и $w_{p_2L_2}$ $(p_1 < p_2)$ несравнимы, если

$$\nu_{p_2} - \nu_{p_1} \in (z_{L_1} - z_{L_2} - \sigma_{L_2} - \delta_{L_2L_1},\ z_{L_1} - z_{L_2} + \sigma_{L_1} + \delta_{L_1L_2}). \tag{2.3}$$

Так как для межоперационной транспортировки требований используется s операторов переноса, то множество W не может содержать более s попарно несравнимых интервалов (условие 2). Справедливо и обратное, т. е. если множество W содержит s попарно несравнимых интервалов, то для реализации процесса необходимо, по крайней мере, s операторов переноса (см. предыдущий параграф этой главы).

В случае, когда $s = 1$, условия 1 и 2 можно записать в виде

$$\begin{gathered} \nu_{p_2} - \nu_{p_1} \in B, \quad \nu_1 = 0, \\ 1 \leqslant p_1 < p_2 \leqslant n, \end{gathered} \tag{2.4}$$

где B — множество, образованное путем удаления интервалов (2.3) из интервала (t, ∞).

Набор $\nu_1, \nu_2, \ldots, \nu_n$ называется *расписанием s-операторного процесса* обслуживания однотипных требований (обозначение: $R\ \{\nu_1, \nu_2, \ldots, \nu_n\}$), если числа $\nu_1, \nu_2, \ldots$ $\ldots, \nu_n$ удовлетворяют условиям 1 и 2. Расписание, очевидно, полностью определяет календарные сроки обслуживания каждого требования каждым прибором. При этом общее время обслуживания требований выразится величиной $\nu_n + z_M$. Расписание $R\ \{\nu_1, \nu_2, \ldots, \nu_n\}$ называется *оптимальным* (обозначение: $R_{\text{опт}}\ \{\nu_1, \nu_2, \ldots, \nu_n\}$), если $\nu_n \leqslant \nu_n'$, где $R\ \{\nu_1', \nu_2', \ldots, \nu_n'\}$ — произвольное расписание.

Обозначим

$$\nu_n + z_M = T_{\text{опт}}(n),$$

если $R\{\nu_1, \nu_2, \ldots, \nu_n\}$ — оптимальное расписание.

Наряду с конечными расписаниями мы будем рассматривать расписания бесконечных s-операторных процессов обслуживания однотипных требований в системе. Расписанием бесконечного s-операторного процесса обслуживания однотипных требований в системе с известными параметрами называется последовательность чисел $\{\nu_p\}$, удовлетворяющих условиям 1 и 2 (обозначение: $R\ \{\nu_p\}$). Расписание $R\ \{\nu_p\}$ называется *периодическим* с периодом T, если последовательность $\nu_1, \nu_2, \ldots, \nu_h, \nu_1 + T, \ldots$ $\ldots, \nu_h + T, \nu_1 + 2T, \ldots$ — периодическая с тем же периодом (обозначение: $R\ \{\nu_1, \nu_2, \ldots, \nu_h;\ T\}$). *Плотностью* периодического расписания называется величина h/T. Оптимальным периодическим расписанием называется периодическое расписание с максимальной плотностью (обозначение: $R_{\text{опт}}\ \{\nu_1, \nu_2, \ldots, \nu_h;\ T\}$).

2.2. Оптимальные периодические расписания занимают особое место среди расписаний s-операторных процессов. При весьма общих предположениях относительно того, что следует понимать под оптимальным расписанием бесконечного s-операторного процесса, можно показать, что поиск таких расписаний естественным образом ограничивается классом периодических расписаний.

В этом параграфе мы покажем, что оптимальные расписания конечного s-операторного процесса можно аппроксимировать «отрезком» подходящей длины оптимального периодического расписания. Рассмотрим также вопросы существования оптимальных периодических расписаний и методы их построения.

Поскольку на практике рассматриваются только системы с рациональными параметрами и при подходящем выборе единицы измерения эти параметры можно считать целочисленными, то в дальнейшем мы будем рассматривать системы с целочисленными параметрами. Наблюдение за состоянием системы будем проводить также в целочисленные моменты времени $t = 0, 1, 2, \ldots$

Состояние системы в момент времени t полностью определяется заданием вектора $\mathfrak{U}(t) = \langle X(t), Y(t), V(t)\rangle$. При этом $X(t) = \langle x_1(t), \ldots, x_M(t)\rangle$ описывает загруженность приборов в момент времени t, т. е. если $x_L(t) = 0$, то L-й прибор в момент времени t свободен, если $x_L(t) > 0$, то L-й прибор в момент времени t занят обслуживанием некоторого требования и с начала обслуживания этого требования L-м прибором прошло $x_L(t)$ единиц времени. Вектор $Y(t) = \langle y_1^1(t), \ldots, y_1^s(t), \ldots, y_{M-1}^1(t), \ldots \ldots, y_{M-1}^s(t)\rangle$ описывает загруженность операторов переноса, т. е. если $y_L^r(t) > 0$, то в момент времени t оператор с номером r транспортирует обслуживаемое требование от L-го к $(L + 1)$-му прибору и с начала транспортировки этого требования до момента времени t прошло $y_L^r(t)$ единиц времени. Если в момент времени t оператор с номером r не транспортирует обслуживаемое требование от L-го к $(L + 1)$-му прибору, то $y_L^r(t) = 0$. Наконец, вектор $V(t) = \langle v_1(t), \ldots, v_s(t)\rangle$ описывает расположение (координаты) операторов переноса в момент времени t. Будем полагать, что если t — целое число, то компоненты вектора $V(t)$ могут принимать лишь конечное число значений.

Так как в этом случае число различных состояний системы в моменты времени $t = 0, 1, 2, \ldots$ при любом целочисленном расписании $R\{\nu_p\}$ не превосходит некоторой величины $\mathfrak{P}$ (так называемого числа допустимых состояний системы), существует такая подпоследовательность $\{\nu'_p\}$ последовательности целых чисел $\{\nu_p\}$, что состояния системы в моменты времени $\nu'_1, \nu'_2, \ldots$ одинаковы, т. е. описываются одним и тем же вектором: $\mathfrak{U}(\nu'_p) = \mathfrak{U}(\nu'_{p+1})$, $p = 1, 2, \ldots$

Теорема 2.1. *Среди периодических целочисленных расписаний s-операторного процесса обслуживания однотипных требований в системе с целочисленными параметрами существует оптимальное расписание.*

Доказательство. Пусть $R\{\nu_1, \ldots, \nu_l, \ldots, \nu_{l+r}, \ldots, \nu_h; T\}$ — оптимальное целочисленное расписание при заданном периоде T. При достаточно большом периоде $T \geqslant T_0$ (в качестве T_0 можно выбрать, например, величину $\mathfrak{P} \cdot z_M$) существуют такие числа ν_l и ν_{l+r}, что состояния системы в моменты времени ν_l и ν_{l+r} одинаковы. Если $\frac{r}{\nu_{l+r} - \nu_l} \geqslant \frac{h}{T}$, то можно построить такое расписание $R'\{\nu_l, \ldots, \nu_{l+r-1}; T'\}$, где $T' = \nu_{l+r} - \nu_l$, что $\frac{r}{T'} \geqslant \frac{h}{T}$ и $T' < T$. Если $\frac{r}{\nu_{l+r} - \nu_l} < \frac{h}{T}$, то можно построить такое расписание $R''\{\nu_1, \ldots, \nu_l, \nu'_{l+1}, \ldots, \nu'_{h-r}; T''\}$, где $T'' = T - T'$ и $\nu'_{l+k} = \nu_{l+r+k} - (\nu_{l+r} - \nu_l)$, $1 \leqslant k \leqslant n - l - r$, что $\frac{h-r}{T''} > \frac{h}{T}$ и $T'' < T$.

В обоих случаях периодическое расписание с периодом T может быть заменено периодическим расписанием с периодом, меньшим T, и с плотностью не ниже плотности исходного расписания. В заключение доказательства заметим, что плотность периодического расписания обслуживания однотипных требований в системе с заданными параметрами не превышает величины $1/t$.

2.3. Частным видом периодических расписаний является расписание так называемого *примитивного процесса* $R\{\nu; z_M\}$ и расписание циклического процесса, полученного в результате *совмещения* примитивных процессов

$R\{\nu_1, \nu_2, \ldots, \nu_h; z_M\}$. Числа $\nu_1, \nu_2, \ldots, \nu_h$ выбираются так, чтобы они принадлежали сегменту $[t, z_M - t]$ и удовлетворяли условиям типа условий 1 и 2.

Циклические расписания, полученные в результате совмещения примитивных расписаний, образуют подкласс периодических расписаний, который может не включать оптимальное расписание.

Рассмотрим, например, однооператорный бесконечный процесс обслуживания однотипных требований в системе со следующими параметрами: $(t_L) = (2^1/_2, 2, 2, 2, 0)$, $(\sigma_L) = (^1/_2, 1, 1, 1^1/_2)$, $(\theta_L) = (0)$, $(\delta_{L_1L_2}) = (0)$.

Очевидно, в данном случае не существует циклического процесса, совмещающего более двух примитивных процессов.

Иначе говоря, не существует циклического процесса, являющегося результатом совмещения примитивных процессов, плотность расписания которого превышает 0,16. В то же время существует допустимое периодическое расписание $R\{0; 5\}$ с плотностью 0, 2.

2.4. Связь между конечными и бесконечными расписаниями s-операторного процесса обслуживания требований в системах с целочисленными параметрами устанавливает следующая

Теорема 2.2. *Имеет место соотношение*

$$0 \leqslant T_{\text{пер}}(n) - T_{\text{опт}}(n) \leqslant (T/h - t)\,\mathfrak{P} + (1 - 1/h)\,T, \quad (2.5)$$

где $T_{\text{пер}}(n)$ — время обслуживания n требований при периодическом расписании $R_{\text{опт}}\{\nu_1, \nu_2, \ldots, \nu_h; T\}$.

Доказательство. Пусть $R_{\text{опт}}\{\nu_1, \ldots, \nu_{i_1}, \ldots, \nu_{i_2}, \ldots, \nu_{i_{l_0}}, \ldots, \nu_n\}$ — оптимальное расписание s-операторного процесса обслуживания n требований в системе с известными параметрами. Так как параметры системы целочисленны, набор чисел $\{[\nu_1], \ldots, [\nu_{i_1}], \ldots, [\nu_{i_2}], \ldots, [\nu_{i_{l_0}}], \ldots, [\nu_n]\}$ является расписанием и, поскольку $[\nu_p] \leqslant \nu_p$, оптимальным расписанием. Пусть в моменты времени $[\nu_{i_1}], [\nu_{i_2}], \ldots, [\nu_{i_{l_0}}]$ состояния системы одинаковы. Рассмотрим расписание $R_1\{\nu_1^{(1)}, \ldots, \nu_{r_1}^{(1)}\} = R\{[\nu_1], \ldots, [\nu_{i_1}], [\nu_{i_{l_0}+1}] - ([\nu_{i_{l_0}}] - [\nu_{i_1}]), \ldots, [\nu_n] - ([\nu_{i_{l_0}}] - [\nu_{i_1}])\}$. Положим $\nu_{r_1}^{(1)} + z_M = T_1$. Очевидно, $T_{\text{опт}}(n) = T_1 + ([\nu_{i_{l_0}}] - [\nu_{i_1}])$. Если в моменты времени $\nu_{j_1}^{(1)}, \ldots, \nu_{j_{l_1}}^{(1)}$ состояния системы одинаковы, рассмотрим расписание $R_2\{\nu_1^{(2)},\ldots,\nu_{r_2}^{(2)}\} = R\,\{\nu_1^{(1)},\ldots,\nu_{j_1}^{(1)},\nu_{j_{l_1+1}}^{(1)} - (\nu_{j_{l_1}}^{(1)}-\nu_{j_1}^{(1)}),\ldots,\ \nu_{r_1}^{(1)}-(\nu_{j_{l_1}}^{(1)}-\nu_{j_1}^{(1)})\}$. Очевидно, $T_1 = T_2 + (\nu_{j_{l_1}}^{(1)}-\nu_{j_1}^{(1)})$, где $T_2 = \nu_{r_2}^{(2)} + z_M$. В результате этих построений получаем такое расписание $R_u\{\nu_1^{(u)},\ldots,\nu_{r_u}^{(u)}\}$, что состояния системы в моменты времени $\nu_1^{(u)},\ldots,\nu_{r_u}^{(u)}$ различны и, следовательно, $r_u \leqslant \mathfrak{P}$ ($\mathfrak{P}$ — число допустимых состояний системы).

Имеем

$$T_{\text{опт}}(n) = T_u + ([\nu_{i_{l_0}}] - [\nu_{i_1}]) + (\nu_{j_{l_1}}^{(1)} - \nu_{j_1}^{(1)}) + \ldots$$
$$\ldots + (\nu_{u_{l_{u-1}}}^{(u-1)} - \nu_{u_1}^{(u-1)}) \geqslant t\,(r_u - 1) + \frac{T}{h}(n - r_u) + z_M,$$

$$T_{\text{пер}}(n) = T\left[\frac{n-1}{h}\right] + T_{\text{пер}}\left(n - h\left[\frac{n-1}{h}\right]\right) \leqslant$$
$$\leqslant T\left(\frac{n-1}{h} + 1\right) - t + z_M.$$

Следовательно,

$$T_{\text{пер}}(n) - T_{\text{опт}}(n) \leqslant \left(\frac{T}{h} - t\right) r_u + \left(1 - \frac{1}{h}\right) T \leqslant$$
$$\leqslant \left(\frac{T}{h} - t\right)\mathfrak{P} + \left(1 - \frac{1}{h}\right) T.$$

По определению $T_{\text{опт}}(n) \leqslant T_{\text{пер}}(n)$. Теорема доказана.

Из доказанной теоремы следует, что в качестве расписания $R_{\text{опт}}\{\nu_1, \nu_2, \ldots, \nu_n\}$ при достаточно большом значении n можно использовать отрезок оптимального периодического расписания соответствующей длины. В случае, когда число операторов достаточно для реализации любого расписания, очевидно, $T_{\text{опт}}(n) = T_{\text{пер}}(n)$.

Таким образом, поиск оптимального расписания (конечного или бесконечного) процесса обслуживания однотипных требований в системе с известными целочисленными параметрами можно ограничить рассмотрением периодических целочисленных расписаний $R\,\{\nu_1, \ldots, \nu_h;\ T\}$, где $\nu_1, \nu_2, \ldots, \nu_h, \nu_1 + T, \nu_2 + T, \ldots$ — целые числа, удовлетворяющие условиям 1 и 2. При этом числа $\nu_1, \nu_2, \ldots, \nu_h$ и T необходимо подобрать так, чтобы максимизировать величину h/T.

2.5. Покажем, что эту задачу можно решить методами линейного программирования.

Состояние системы называется *допустимым*, если существует такое целочисленное расписание $R\ \{\nu_p\}$, при котором система в некоторый момент времени t (t — целое число) будет находиться в этом состоянии. Состояние $\mathfrak{U}'$ называется состоянием, *смежным* допустимому состоянию $\mathfrak{U}$, если существует целочисленное расписание $R\ \{\nu_p\}$, при котором система переходит из состояния U в состояние $\mathfrak{U}'$ за единицу времени.

Рассмотрим граф, вершинам которого соответствуют допустимые состояния системы, и каждая вершина которого, соответствующая состоянию $\mathfrak{U}$, соединена дугами с вершинами, соответствующими состояниям, смежным состоянию $\mathfrak{U}$.

Если $\mathfrak{U}' = \langle (x_1', \ldots, x_M'), \ Y', \ V' \rangle$ — состояние, смежное состоянию $\mathfrak{U}$, и $x_1' = 1$, то дуге, соединяющей соответствующие вершины графа, припишем «массу», равную единице. Если $x_1' \neq 1$, то этой дуге припишем «массу», равную нулю. Длины всех дуг полагаем равными единице. Выберем произвольный контур в построенном таким образом графе. Определим длину и массу контура соответственно как сумму длин и масс его дуг. Плотностью контура назовем отношение массы контура к его длине.

Из доказательства теоремы следует, что задача нахождения оптимального периодического расписания сводится к задаче нахождения в описанном графе контура с наибольшей плотностью.

Последняя задача может быть сформулирована в терминах линейного программирования.

Действительно, пусть $(X, \vec{U})$ — некоторый ориентированный граф, каждой дуге $(i, j) \in \vec{U}$ которого приписаны длина l_{ij} и масса m_{ij}. Если γ — некоторый контур в графе $(X, \vec{U})$, то плотность этого контура

$$\rho(\gamma) = \sum_{(i, j) \in \gamma} m_{ij} \Big/ \sum_{(i, j) \in \gamma} l_{ij}. \tag{2.6}$$

Требуется найти в $(X, \vec{U})$ контур γ с наибольшей плотностью $\rho(\gamma)$.

Рассмотрим задачу линейного программирования, в которой необходимо найти $x_{ij} \geqslant 0, \quad (i, j) \in \vec{U}$, удовлетворяющие условиям

$$\sum_{(i,\ j)\in\vec{U}} x_{ij} = \sum_{(j,\ i)\in\vec{U}} x_{ji}, \quad i \in X, \tag{2.7}$$

$$\sum_{(i,\ j)\in\vec{U}} l_{ij}x_{ij} = 1, \tag{2.8}$$

и такие, что

$$\sum_{(i,\ j)\in\vec{U}} m_{ij}x_{ij} \to \max. \tag{2.9}$$

Если $x_{ij} > 0$, то дуга $(i,\ j)$ принадлежит искомому контуру γ^*. Если $x_{ij} = 0$, то дуга $(i,\ j)$ не принадлежит этому контуру.

Каждому контуру γ можно сопоставить значения x_{ij}, равные $\left(\sum\limits_{(i,\ j)\in\gamma} l_{ij}\right)^{-1}$ для всех $(i,\ j) \in \gamma$ и равные нулю для всех остальных дуг из $\vec{U}$. Эти значения x_{ij}, очевидно, удовлетворяют условиям (2.7), (2.8).

Можно показать также, что любой вектор (x_{ij}), удовлетворяющий условиям (2.7), (2.8), представим в виде выпуклой линейной комбинации векторов (x_{ij}), определяющих контуры в графе $(X,\ \vec{U})$.

Отметим также, что поиск оптимального по плотности контура можно ограничить рассмотрением элементарных контуров графа $(X,\ \vec{U})$. Действительно, если γ^* — составной контур, то представляя его в виде объединения двух контуров $\overset{*}{\gamma}_1$ и $\overset{*}{\gamma}_2$, получаем $\rho(\gamma^*) > \rho(\overset{*}{\gamma}_2)$, если $\rho(\gamma^*) < \rho(\overset{*}{\gamma}_1)$. Следовательно, $\rho(\gamma^*) = \rho(\overset{*}{\gamma}_1) \doteq \rho(\overset{*}{\gamma}_2)$.

2.6. Известные моменты начала обслуживания требований неоднозначно определяют перемещения операторов переноса и таким образом лишь частично описывают поведение системы. Действительно, задавая целые числа ν_1, $\nu_2, \ldots, \nu_n$, удовлетворяющие условиям 1 и 2, мы тем самым полностью определяем вектор $X(t)$ в любой момент времени t, что, вообще говоря, не справедливо для векторов $Y(t)$ и $V(t)$. В связи с возможной неоднозначностью определения векторов $Y(t)$ и $V(t)$ известным целочисленным расписанием $R\ \{\nu_1, \nu_2, \ldots, \nu_n\}$ возникает задача нахождения оптимального в том или ином смысле режима работы операторов переноса.

Пусть на плоскости задано $(\nu_n - \nu_1 + z_M + 1)$ параллельных прямых (рис. 8.2.1). Занумеруем их в порядке 0, 1, 2, . . ., k, . . ., $\nu_n - \nu_1 + z_M$. Рассмотрим граф, вершины которого расположены на этих прямых, причем на

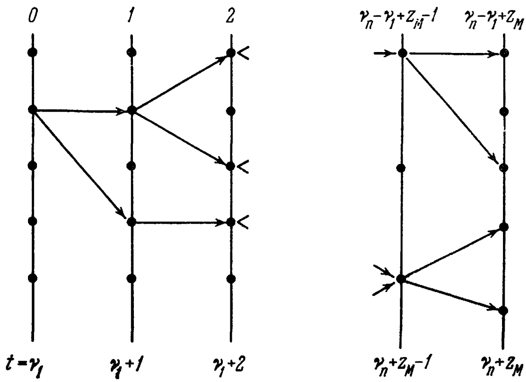

Рис. 8.2.1. Схема возможных изменений состояний операторов переноса.

k-й прямой расположены вершины, соответствующие допустимым состояниям $\mathfrak{U}(\nu_1 + k)$ системы при известном расписании $R\ \{\nu_1, \nu_2, \ldots, \nu_n\}$. Если $\mathfrak{U}(\nu_1 + k + 1)$ — состояние, смежное состоянию $\mathfrak{U}(\nu_1 + k)$, то соответствующие вершины графа соединим дугой. Переход $\mathfrak{U}(\nu_1 + k) \to$ $\to \mathfrak{U}(\nu_1 + k + 1)$ вызывает изменение в загрузке и месторасположении операторов переноса, которое можно охарактеризовать некоторой величиной, которую мы припишем соответствующей дуге графа и будем называть «длиной» этой дуги.

Очевидно, путь в этом графе, началом которого служит вершина, расположенная на 0-й прямой и соответствующая начальному состоянию системы $\mathfrak{U}(\nu_1)$, концом — вершина, расположенная на $(\nu_n - \nu_1 + z_M)$-й прямой, дает полную информацию о расположении загруженных и незагруженных операторов переноса в любой момент времени. Так как этот путь существует не для любого начального состояния системы, то следует дополнить рассматриваемый граф вершинами, соответствующими всевозможным начальным состояниям системы.

Таким образом, задача рационального использования операторов переноса при реализации заданного расписания сводится к задаче нахождения пути наименьшей длины вида

$$\mathfrak{U}(\nu_1) \to \mathfrak{U}(\nu_1 + 1) \to \ldots \to \mathfrak{U}(\nu_n + z_M)$$

в построенном графе. Последняя задача может быть решена известными методами. В каждом конкретном случае размер графа можно существенно уменьшить.

§ 3. Библиографическая справка

Аппарат описания и анализа многооператорных процессов развит в работах Д. А. Супруненко, Р. И. Тышкевич, В. С. Айзенштата и А. С. Метельского [1, 2, 114, 115, 116, 166, 167]. В [1, 2] рассматриваются вопросы совмещения примитивных процессов в системах с одинаковыми маршрутами обслуживания, в [114—116] эти результаты обобщаются на случай различных маршрутов.

В основу данной главы положены работы А. Ш. Блоха и В. С. Танаева [18, 173]. Вопросы асимптотического оптимального управления рассматривались И. В. Романовским [150]. В частности, в [150] содержится формулировка задачи 2.7—2.9 и аналог теоремы 2.2. Алгоритм определения ограниченного по длине пути с максимальным весом приведен в [157].

Метод последовательного восстановления связности описан в [172]. Определению оптимальных путей в графах типа рис. 8.2.1 посвящены работы Н. Н. Моисеева [123], И. Х. Сигала и В. А. Чебакова [158].

Вопросы реализуемости плана и использования методов линейного программирования для построения оптимальных расписаний рассмотрены Д. И. Коганом [80].

ЦИТИРОВАННАЯ ЛИТЕРАТУРА

1. Айзенштат В. С., Многооператорные циклические процессы, ДАН БССР 7, № 4 (1963), 224—227.
2. Айзенштат В. С., Совмещение примитивных циклических процессов, ДАН БССР 7, № 3 (1963), 148—151.
3. Акимов Г. П., Об одном алгоритме в задаче календарного планирования, Сб. «Матем. программирование», М., «Наука», 1966, 46—61.
4. Акоф Р., Сасиени М., Основы исследования операций, М., «Наука», 1971.
5. Алексеев А. С., Метод точечных преобразований в задаче составления оптимального графика ритмичного производства двух типов изделий на одном оборудовании, Изв. высш. учебн. заведений, Радиофизика 3, № 4 (1960), 706—715.
6. Ангуладзе М. Х., Ломинадзе М. М., Чанкотадзе В. В., Шартава В. К., К задаче оптимального распределения работ по параллельным местам в случае нелинейных функций потерь, Тр. проблем. лаб. автом. и выч. техн., Груз. политехн. ин-т, № 3, 1972, 117—123.
7. Ангуладзе М. Х., Об использовании метода статистического поиска с применением правил предпочтения в задачах составления расписаний, Сакартвелос ССР, Мецниеребата Академиис моамбе, Сообщ. АН Груз. ССР 65, № 2 (1972), 417—420.
8. Антипов В. И., Решение частной задачи календарного планирования методом сравнения состояний, Сб. «Системы распределения ресурсов на графах», М., Вычисл. центр АН СССР, 1970, 7—24.
9. Арайс Е. А., О задаче коммивояжера, Тр. Сиб. физ.-техн. ин-та при Томск. ун-те, вып. 51, 1970, 274—277.
10. Аронович А. Б., О выборе оптимальных комбинаций локальных правил календарного планирования, Экономика и матем. методы 6, № 4 (1970), 548—557.
11. Аронович А. Б., Об одной задаче календарного планирования, Экономика и матем. методы 4, № 3 (1968), 401—406.
12. Баркан С. А., Об одной модели двухстадийного производства, Изв. АН БССР, сер. физ.-техн. наук, № 2 (1969), 42—47.
13. Баркан С. А., Танаев В. С., О построении расписаний учебных занятий, Изв. АН БССР, сер. физ.-мат. наук, № 1 (1970), 76—82.
14. Барский А. Б., Две задачи оптимизации использования неоднородных вычислительных систем, Изв. АН СССР, Техн. кибернетика, № 4, (1968), 32—38.

15. Беллман Р., Динамическое программирование, М., ИЛ, 1960.
16. Беллман Р., Дрейфус С., Прикладные задачи динамического программирования, М., «Наука», 1965.
17. Берж К., Теория графов и ее применение, М., ИЛ, 1962.
18. Блох А. Ш., Танаев В. С., Многооператорные процессы, Изв. АН БССР, сер. физ.-мат. наук, № 2 (1966), 5—11.
19. Бондаренко А. В., Эвристические подходы к решению задач календарного планирования, Сб. «Автоматизир. системы упр. предприятиями. Тр. семинара», вып. 1, Киев, 1968, 47—65, Дискуссия, 71—73.
20. Булгаков В. А., Кузнецов И. Н., Детерминированная система с многократным обслуживанием, Сб. «Избр. тр. Всес. межвуз. симпоз. по прикл. матем. и кибернет.», Горький, 1967, М., «Наука», 1973, 146—150.
21. Бурдюк В. Я., О выборе последовательности загрузки станков, Экономика и матем. методы 6, № 1 (1970), 112—116.
22. Бурдюк В. Я., Одна экстремальная задача теории расписаний, Кибернетика, № 3 (1973), 103—108.
23. Бурдюк В. Я., О задаче m станков ($m \geqslant 2$), Кибернетика, № 3 (1969), 74—76.
24. Бурдюк В. Я., Стохастическая задача двух станков, Кибернетика, № 5 (1969), 110—114.
25. Бурдюк В. Я., Мазня Т. А., Об одной задаче упорядочения, Сб. «Автоматизир. системы упр. предприятиями. Тр. семинара», вып. 2, Киев, 1969, 88—98.
26. Бурдюк В. Я., Шкурба В. В., Теория расписаний. Задачи и методы решений, Кибернетика, № 1 (1971), 89—102.
27. Бурдюк Т. А., Упорядочение работ для станков равной производительности, Изв. АН СССР, Техн. кибернетика, № 1 (1972), 32—36.
28. Бурков В. Н., Ловецкий С. Е., Методы решения экстремальных комбинаторных задач (обзор), Изв. АН СССР, Техн. кибернетика, № 4 (1968), 82—93.
29. Бурков В. Н., Ловецкий С. Е., Методы решения экстремальных задач комбинаторного типа (обзор), Автоматика и телемеханика, № 11 (1968), 68—93.
30. Бурков В. Н., Соколов В. Б., Оптимальное размещение информационных массивов в памяти на магнитных лентах для случая двунаправленного поиска, Автоматика и телемеханика, № 4 (1969), 107—117.
31. Бусленко Н. П., Математическое моделирование производственных процессов М., «Наука», 1964.
32. Васильев В. В., Додонов А. Г., Левина А. И., Об одном методе решения задачи коммивояжера, Тр. семинара по методам матем. моделирования и теории электр. цепей, Ин-т кибернет. АН УССР, вып. 9, 1971, 58—67.
33. Ведута Н. И., Сафроненко В. А., Математическая модель календарного планирования серийного производства, Экономика и матем. методы 6, № 6 (1970), 894—909.
34. Венкова Н. Б., Тушкина Т. А., Шахбазян К. В., О расписании, минимизирующем число процессоров

однородной вычислительной системы, Изв. АН СССР, Техн. кибернетика, № 5 (1971), 115—117.

35. Власюк Б. А., Задача оптимального расписания при параллельно-последовательном соединении машин, Изв. АН СССР, Техн. кибернетика, № 4 (1966), 67—71.

36. Власюк Б. А., Задача оптимального расписания при наличии переналадок, Экономика и матем. методы 3, № 3 (1967), 415—419.

37. Власюк Б. А., Оптимальное расписание обработки деталей на трех последовательных машинах, Изв. АН СССР, Техн. кибернетика, № 4 (1967), 79—84.

38. Волынский Э. И., Красин Л. А., Оптимальная расстановка исполнителей на заключительном этапе работ, Изв. АН СССР, Механиз. и автоматиз. упр., Научн.-произв. сб., № 2, 1973, 16—18.

39. Воронов А. А., Исследование операций и управление, М., «Наука», 1970.

40. Выгнан П. К., Кузин Б. И., Алгоритм построения календарного графика работы оборудования на участках с разнонаправленными маршрутами обработки, Сб. «Примен. матем. в экон.», вып. 5, Л., Ленингр. ун-т, 1969, 68—78.

41. Габович Е., Задача коммивояжера. I. Тр. Вычисл. центра, Тартуск. ун-т, вып. 19, 1970, 52—96.

42. Габович Е., Малая задача коммивояжера, Тр. Вычисл. центра, Тартуск. ун-т, вып. 19, 1970, 27—51.

43. Габович Е., Чиж А., Ялас А., О задаче коммивояжера на узкие места, Тр. вычисл. центра, Тартуск. ун-т, вып. 22, 1971, 3—24.

44. Гельтман Л. С., Гусман С. Я., О решении задачи нахождения оптимального маршрута, Тр. Н.-и. ин-та упр. выч. машин и систем, вып. 2, 1968, 51—55.

45. Гиффлер Б., Алгебры календарного планирования и их применение при формулировании моделей общих систем, Сб. «Календарн. планирование», М., «Прогресс», 1966, 62—83.

46. Гиффлер Б., Томпсон Д., Ван-Нессе В., Опыт вычислений с применением линейного алгоритма и алгоритма Монте-Карло для решения задач календарного планирования на производстве, Сб. «Календарн. планирование», М., «Прогресс», 1966, 42—61.

47. Глебов Н. И., Алгоритм составления оптимального расписания для двух работ, Сб. «Управляемые системы», вып. 1, Новосибирск, «Наука», 1968, 14—20.

48. Глебов Н. И., О верхней оценке длины оптимального расписания для двух работ, Сб. «Пробл. кибернетики», вып. 20, М., «Наука», 1968, 225—229.

49. Глебов Н. И., Перепелица В. А., О нижней и верхней оценках для одной задачи теории расписаний, Сб. «Исслед. по кибернетике», М., «Сов. радио», 1970, 11—17.

50. Глебский Ю. В., Шевченко В. Н., О составлении оптимального расписания работы на n станках, Тр. по вопр. применения ЭВМ в нар. хоз-ве, Горький, 1964, 31—34.

51. Глебский Ю. В., Шевченко В. Н., О составлении оптимального плана работ, Научн. тр. Моск. инж.-экон. ин-та, вып. 21, 1964, 318—324.
52. Глубочанский А. Д., Об одном расширении класса функций штрафов в задачах составления расписаний, Кибернетика, № 1 (1971), 103—104.
53. Голенко Д. И., Тарнопольский Ю. Я., Оптимизация календарных планов методами направленного поиска, Кибернетика, № 6 (1970), 138—144.
54. Гордон В. С., Детерминированные одностадийные системы обслуживания с прерываниями процесса обслуживания, Сб. «Вычислит. техн. в машиностроении», 1973, июнь, 30—38.
55. Гордон В. С., Танаев В. С., Детерминированная система обслуживания с одним прибором и ступенчатыми функциями штрафа, Сб. «Вычислит. техн. в машиностроении», 1971, сентябрь, 3—8.
56. Гордон В. С., Танаев В. С., Прерывания в детерминированных системах с параллельными приборами и неодновременным поступлением требований на обслуживание, Материалы семинара ИТК БАН и ИТК АН БССР «Оптимизация систем сбора, передачи и обработки аналоговой и дискретной информации в локальных ИВС», Минск, 1973, 36—50.
57. Гордон В. С., Танаев В. С., Директивные сроки в однофазных детерминированных системах обслуживания, Материалы семинара ИТК БАН и ИТК АН БССР «Оптимизация систем сбора, передачи и обработки аналоговой и дискретной информации в локальных ИВС», Минск, 1973, 51—58.
58. Гордон В. С., Танаев В. С., Детерминированные системы обслуживания с одним прибором, древовидным упорядочением требований и экспоненциальными функциями штрафа, Сб. «Вычислит. техн. в машиностроении», 1973, июнь, 3—10.
59. Данциг Д. Б., Гоффман А. Дж., Теорема Дилворта о частично упорядоченных множествах, Сб. «Линейные неравенства и смежн. вопр.», М., ИЛ, 1959, 311—317.
60. Джексон Д. Р., Очереди с динамическим правилом приоритета, Сб. «Календарное планирование», М., 1966, 357—377.
61. Драхлин Е. Х., Якимов Р. М., Об одном методе определения оптимальной последовательности непрерывного процесса обработки, Сб. научн. тр. Пермск. ин-та, № 21, 1966, 55—58.
62. Драхлин Е. Х., Якимов Р. М., Анализ свойств оптимального варианта последовательности обработки изделий, Сб. научн. тр. Пермск. ин-та, № 21, 1966, 59—61.
63. Думлер С. А., Управление производством и кибернетика, М., «Машиностроение», 1969.
64. Журженко С. Л., О методе решения одного класса задач теории расписаний, Изв. АН СССР, Техн. кибернетика, № 1 (1968), 59—67.
65. Журженко С. Л., Задачи распределения одностадийного производства (обзор литературы), Сб. «Моделиров. экон. процессов», вып. 2, М., МГУ, 1968.

66. З а к Ю. А., Алгоритмы решения задач «n коммивояжеров», Кибернетика, № 1 (1972), 99—106.
67. З а к Ю. А., Об одной задаче определения оптимальной последовательности переналадок оборудования, Кибернетика, № 6 (1968), 86—92.
68. З а к Ю. А., О некоторых задачах определения оптимальной последовательности переналадок оборудования, Сб. «Оперативн. упр. произв.», М., «Наука», 1971, 119—128.
69. З а к Ю. А., Определение порядка выполнения независимых операций на параллельных машинах, Изв. АН СССР, Техн. кибернетика, № 2 (1969), 15—20.
70. З а к Ю. А., М е л ь н и к И. М., Определение оптимальной последовательности переналадок, Сб. «Применение матем. методов в экон. исслед. и планирования. Тр. семинара», вып. 1, Киев, 1967, 10—19.
71. З а л е с с к и й А. Е., Сведение некоторых комбинаторных задач к целочисленному линейному программированию, Изв. АН БССР, сер. физ.-мат. наук, № 3 (1965), 24—28.
72. И о ф ф е Э. Г., Об одном алгоритме расчета календарных расписаний для производственного участка, Экономика и матем. методы 7, № 6 (1971), 876—882.
73. И о ф ф е Э. Г., Алгоритм для определения всех оптимальных расписаний в двухоперационной задаче Джонсона, Автоматика и телемеханика, № 7 (1973), 95—101.
74. И р и к о в В. А., Некоторые задачи упорядочения, Изв. АН СССР, Техн. кибернетика, № 4 (1970), 38—42.
75. К а з а н ц е в Э. Н., Математическая модель графика выпуска изделий, удовлетворяющего требованию комплектации сборки, Изв. АН БССР, сер. физ.-мат. наук, № 1 (1968), 54—60.
76. К а т а е в В. А., Оптимальная очередь обработки деталей на произвольном количестве станков (теория расписаний), Научн. тр. Н.-и. ин-та управл. выч. машин и систем, вып. 1, 1967, 57—68.
77. К и с л и ц ы н С. С., Конечные частично упорядоченные множества и соответствующие им множества перестановок, Матем. заметки 4, № 5 (1968), 511—518.
78. К и т и к М. Г., К вопросу «сужения» ветвления в задаче одного станка, Кибернетика, № 1 (1972), 145—147.
79. К л а д о в Г. К., Л и в ш и ц Э. М., О задаче минимизации суммы штрафов при составлении расписания, Кибернетика, № 6 (1968), 99—100.
80. К о г а н Д. И., D-множества, Δ-множества и неразрешимые задачи дискретного управления, Изв. высш. учебн. заведений, Радиофизика, 15, № 3 (1972), 358—364.
81. К о н в е й Р., Д ж о н с о н Б., М а к с в е л л У., Экспериментальные исследования распределения работ в соответствии с фиксированными правилами очередности их выполнения, Сб. «Применение статистических методов в производстве», М., 1963, 212—232.
82. К о н в е й Р., М а к с в е л л У., Календарное планирование в условиях сети очередей с дисциплиной по кратчайшей

операции, Сб. «Календарное планирование», М., 1966, 321—347.
83. Корбут А. А., Финкельштейн Ю. Ю., Дискретное программирование, М., «Наука», 1969.
84. Коробков В. К., Кричевский Р. Е., Некоторые алгоритмы для решения задачи «коммивояжера», Сб. «Матем. модели и методы оптимальн. планир.», Новосибирск, «Наука», 1966, 106—108.
85. Кофман А., Дебазей Г., Сетевые методы планирования и их применение, М., «Прогресс», 1968.
86. Кочнев Л. В., Оптимизация системы приоритета в задачах расписания, Сб. «Кибернетика и вуз», вып. 3, Томск, Томск. ун-т, 1970, 114—122.
87. Кузин Б. И., Тютюкин В. К., Задача о коммивояжере с ограничением на передвижение, ее алгорифм и экономические приложения, Сб. «Применение матем. в экон.», вып. 6, Л., Ленингр. ун-т, 1970, 53—67.
88. Кузин Б. И., Тютюкин В. К., Обобщенная «задача коммивояжера», ее алгоритм и применение к определению порядка запуска предметов на переменно поточных линиях. Сб. «Применение матем. в экон.», вып. 5, Л., Ленингр. ун-т, 1969, 56—67.
89. Кукса А. И., Задачи упорядочения в графах, Сб. «Алгоритмизация произ. процессов. Тр. семинара», вып. 1, Киев, 1968, 56—71.
90. Кукса А. И., Расписания на графах и независимые множества графов, Кибернетика, № 4 (1970), 92—95.
91. Лазебник А. И., Хранович И. Л., Решение обобщенной задачи коммивояжера методом ветвей и границ, Экономика и матем. методы 9, № 2 (1973), 563—564.
92. Ламбин Н. В., Танаев В. С., О бесконтурной ориентации смешанных графов, ДАН БССР 14, № 9 (1970), 780—781.
93. Левин Г. М., Танаев В. С., Об одном классе комбинаторных задач оптимизации, Изв. АН БССР, сер. физ.-мат. наук № 5 (1968), 30—35.
94. Левин Г. М., Танаев В. С., К теории оптимизации на множествах перестановок, ДАН БССР 14, № 7 (1970), 588—590.
95. Левнер Е. В., Сетевой подход к задачам теории расписаний, Сб. «Исслед. по дискретн. матем.», М., «Наука», 1973, 135—150.
96. Левнер Е. В., Оптимальное планирование обработки объектов на ряде машин, Автоматика и телемеханика, № 12 (1969), 94—100.
97. Левнер Е. В., Об одной задаче календарного планирования, сводимой к задаче о кратчайшем пути, Тр. 4-й зимней школы по матем. программированию и смеж. вопросам, вып. 2, М., 1971, 93—98.
98. Леонтьев В. А., Построение на заданном множестве точек гамильтонова цикла, близкого по длине к наикратчай-

шему, Сб. «Актуальн. вопр. техн. кибернетики», М., «Наука», 1972, 244—248.

99. Лепешинский Н. А., О классификации $2 \times n$ задач Джонсона — Беллмана, Изв. АН БССР, сер. физ.-мат. наук, № 1 (1967), 18—21.

100. Лепешинский Н. А., К вопросу упорядочения обработки деталей, Изв. АН БССР, сер. физ.-мат. наук, № 2 (1966), 31—35.

101. Лепешинский Н. А., Одна задача теории расписаний и максимум суммы линейных форм на множестве перестановок, Изв. АН БССР, сер. физ.-мат. наук, № 3 (1967), 51—54.

102. Лепешинский Н. А., Об одной задаче теории расписаний, Изв. АН БССР, сер. физ.-мат. наук, № 3 (1966), 90—96.

103. Лившиц Э. М., Последовательность операций при изготовлении сложной детали, Автоматика и телемеханика, № 11 (1968), 94—95.

104. Лившиц Э. М., Минимизация максимального штрафа в задаче одного станка, Тр. 1-й зимней школы по матем. программированию в г. Дрогобыче, М., 1969, 474—475.

105. Лившиц Э. М., Пленкова В. А., Рублинецкий В. И., Рандомизация локально-оптимальных алгоритмов решения дискретных задач, Сб. научн. тр. физ.-техн. ин-та низк. температур, АН УССР, вып. 1, 1969, 5—11.

106. Лившиц Э. М., Рублинецкий В. И., О сравнительной сложности некоторых задач дискретной оптимизации, Сб. «Вычисл. матем. и вычисл. техн.», вып. 3, Харьков, 1972, 78—85.

107. Линский В. С., Корнев М. Д., Составление оптимальных расписаний для параллельно действующих процессоров, Изв. АН СССР, Техн. кибернетика, № 3 (1972), 160—167.

108. Лобановский Н. М., Об оптимальном планировании последовательности выполнения работ на одной или двух машинах, Сб. «Кибернетика и управление», М., «Наука», 1967, 85—92.

109. Логинов Э. А., Замечание к алгоритму Джонсона, Экономика и матем. методы 9, № 2 (1973), 359—361.

110. Лурье А. Л., О некоторых задачах календарного планирования, Сб. «Пробл. кибернетики», М., «Наука», вып. 7, 1962, 201—208.

111. Лурье А. Л., О математических методах решения задач на оптимум при планировании социалистического хозяйства, М., «Наука», 1964.

112. Лыткин И. П., Об одной задаче календарного планирования, Экономика и матем. методы 7, № 2 (1971), 286—289.

113. Матюшков Л. П., Танаев В. С., Программные генераторы допустимых расписаний, Сб. «Вычислительная техника в машиностроении», Минск, 1967, июль, 35—48; 1968, февраль, 12—28.

114. Метельский А. С., О некоторых периодических процессах, ДАН БССР 7, № 9 (1963), 584—587.

115. Метельский А. С., Периодические процессы, ДАН БССР, 9, № 12 (1965), 788—790.

116. Метельский А. С., К системе периодических процессов, ДАН БССР 7, № 11 (1963), 724—728.
117. Мимори С., Теория расписаний, Дэнки гаккай дзасси, J. Inst. Elec. Eng. Jap. 92, № 11 (1972), 1088—1092.
118. Мироносецкий Н. Б., Экономико-математические методы календарного планирования, Новосибирск, «Наука», 1974.
119. Михалевич В. С., Последовательные алгоритмы оптимизации и их применение. I, II, Кибернетика, № 12 (1965), 85—89.
120. Михайлевич В. С., Ермольев Ю. М., Шкурба В. В., Шор Н. З., Сложные системы и решение экстремальных задач, Кибернетика, № 5 (1967), 29—37.
121. Михалевич В. С., Шкурба В. В., Последовательные схемы оптимизации в задачах упорядочения выполнения работ, Кибернетика, № 2 (1966), 34—40.
122. Михалевич В. С., Шор Н. З., Численное решение многовариантных задач по методу последовательного анализа вариантов. Научн.-метод. материалы экон.-матем. семинара, вып. 1, М., ЛЭММ АН СССР, 1962.
123. Моисеев Н. Н., Методы динамического программирования в теории оптимальных управлений, ЖВМ и МФ, ч. I, 4, № 3 (1964), 485—494; ч. II, 5, № 1 (1965), 44—56.
124. Моисеев Н. Н., Численные методы теории оптимальных управлений, использующие вариации в пространстве состояний, Кибернетика, № 3 (1966), 1—129.
125. Мудров В. И., Один способ решения задачи коммивояжера с помощью целочисленного линейного программирования, ЖВМ и МФ 3, № 6 (1963), 1137—1139.
126. Мудров В. И., Определение гамильтоновых путей кратчайшей длины в полном графе методами целочисленного линейного программирования, Изв. АН СССР, Техн. кибернетика, № 2 (1965), 3—8.
127. Никитин А. В., Применение динамического программирования для решения некоторых задач упорядочения, Тр. Моск. энерг. ин-та, вып. 68, 1969, 165—169.
128. Николаев К. Г., Плужников Л. П., Алгоритм перестановок для построения допустимого расписания, Изв. АН СССР, Техн. кибернетика, № 3 (1969), 26—32.
129. Озерной В. М., Рябов Л. П., Эвристический метод оптимизации последовательности выполнения операций, Автоматика и телемеханика, № 12 (1967), 169—172.
130. Осколкова С. Е., Осколков И. О., Применение некоторых эвристических методов к решению задач календарного планирования (обзор), Автоматика и телемеханика, № 2 (1968), 177—184.
131. Панайоти Б. Н., Пьянзина Л. Я., Чебаков В. А., Минимизация числа прерываний в многопроцессорном расписании, Изв. АН СССР, Техн. кибернетика, № 4 (1971), 103—110.
132. Парамонов Ф. И., Математические методы расчета многономенклатурных потоков, М., «Машиностроение», 1964.

133. Парамонов Ф. И., Автоматизация управления групповыми поточными линиями, М., «Машиностроение», 1973.
134. Перепелица А. В., Асимптотический подход к решению некоторых экстремальных задач на графах, Сб. «Пробл. кибернетики», вып. 26, 1973, 291—314.
135. Перепелица В. А., Об одной задаче теории расписаний, Кибернетика, № 5 (1966), 75—78.
136. Перепелица В. А., К задаче нахождения минимального гамильтонова пути на графе со взвешенными дугами, Сб. «Методы управления большими системами», т. 2, Иркутск, 1970, 252—259.
137. Подчасова Т. П., Об оценках и выборе правил предпочтения в задачах календарного планирования, Сб. «Автоматизир. системы упр. предприятиями. Тр. семинара», вып. 1, Киев, 1968, 5—46, Дискуссия, 71—73.
138. Португал В. М., Решение задач календарного планирования с помощью правил предпочтения, Сб. «Прикл. матем. и кибернет. Материалы к Всес. межвуз. симпозиуму по прикл. матем. и кибернет.», Горький, 1967, 254—258.
139. Португал В. М., Исследования асимптотического поведения решений Джонсона, Кибернетика, № 1 (1971), 105—107.
140. Португал В. М., Об одной задаче теории расписаний, Кибернетика, № 1 (1972), 143—145.
141. Португал В. М., Применение комбинаторного метода для решения задачи составления расписаний, Кибернетика, № 4 (1967), 98—100.
142. Прилуцкий М. Х., Шевченко В. Н., Об одном подходе к составлению оптимального плана работ, Сб. «Исследования по кибернетике», М., «Сов. радио», 1970, 18—20.
143. Прокофьев О. Н., Выбор оптимальной очередности обслуживания, Изв. АН СССР, Техн. кибернетика, № 5 (1972), 104—107.
144. Пчелинцев Л. А., Об одном решении задачи коммивояжера, ЖВМ и МФ 6, № 3 (1966), 597—599.
145. Радев Е. И., О двух эвристических методах решения задачи коммивояжера, Тр. Моск. экон.-стат. ин-та, ч. I, 1972, 134—150.
146. Резниченко В. Н., Решение одной задачи теории расписаний эвристическим методом, Сб. «Алгоритмизация произ. процессов. Тр. семинара», вып. 1, Киев, 1968, 91—110.
147. Резниченко В. Н., Минимизация времени входа в ритмичную работу в задаче одного станка, Сб. «Системотехника», вып. 2, Киев, 1970, 80—86.
148. Резниченко В. Н., Шкурба В. В., Подчасова Т. П., Определение оптимальной программы запуска-выпуска деталей в условиях неустановившегося производственного процесса, Сб. «Автоматизир. системы упр. предприятиями. Тр. семинара», вып. 3, Киев, 1969, 3—33.
149. Романовский И. В., Задача о наивыгоднейшей круговой расстановке станков, Экономика и матем. методы 2, № 4 (1966), 578—581.

150. Романовский И. В., Асимптотика рекуррентных соотношений динамического программирования и оптимальное стационарное управление, ДАН СССР 157, № 6 (1964), 1303—1306.
151. Рубинштейн М. И., О симметричной задаче коммивояжера, Автоматика и телемеханика, № 9 (1971), 126—133.
152. Рублинецкий В. И., Об оценках точности процедур в задаче коммивояжера, Сб. «Вычислительная техника в машиностроении», вып. 4, 1973, 18—23.
153. Рыбников К. А., Введение в комбинаторный анализ, М., издательство МГУ, 1972.
154. Сафроненко В. А., Математическое и электронное моделирование задачи оптимального календарного планирования, Минск, «Наука и техника», 1972.
155. Сгурев В., Построяване на оптимално разписание при стационарни процеси, Изв. Ин-та техн. кибернетики, № 12 (1970), 29—51.
156. Семенов А. И., Португал В. М., Задачи теории расписаний в календарном планировании мелкосерийного производства, М., «Наука», 1972.
157. Сигал И. Х., Последовательный анализ вариантов в задаче о нахождении на графе ограниченного по длине пути с максимальным весом, Изв. АН СССР, Техн. кибернетика, № 6 (1970), 41—47.
158. Сигал И. Х., Чебаков В. А., Метод матричного перебора и его применение к одной задаче теории графов, ЖВМ и МФ 5, № 1 (1965), 148—150.
159. Сихарулидзе Г. Г., Об одном обобщении задачи коммивояжера, Автоматика и телемеханика, № 8 (1971), 116—123; № 10 (1971), 142—147.
160. Солих Р., Задача календарного планирования для циклически повторяющегося производства, ЖВМ и МФ 13, № 2 (1973), 326—342.
161. Сотсков Ю. Н., Танаев В. С., О перечислении ориентированных бесконтурных графов, порождаемых смешанным графом, Изв. АН БССР, сер. физ.-мат. наук, № 2 (1974), 16—21.
162. Сохранская В. С., Решение на ЭВМ задач параллельного упорядочения, Зап. научн. семинаров, Ленингр. отд. Мат. ин-та, АН СССР 35 (1973), 138—141.
163. Сприндкук В. Г., К теореме Дилворта о частично упорядоченных множествах, ДАН БССР 7, № 12 (1963), 803—804.
164. Стори А. Е., Вагнер Х. М., Опыт применения целочисленного программирования при расчете календарного плана для предприятий единичного и мелкосерийного производства, Сб. «Календарное планирование», М., «Прогресс», 1966, 241—256.
165. Супруненко Д. А., Айзенштат В. С., Лепешинский Н. А., Экстремальные значения функций на множествах перестановок, Тезисы докл. 1 Всес. конф. по исслед. операций, Минск, 1972, 61—64.

166. Супруненко Д. А., Айзенштат В. С., Метельский А. С., Многоэтапный процесс превращения, ДАН БССР 6, № 9 (1962), 541—544.
167. Супруненко Д. А., Тышкевич Р. И., Динамические отображения и один класс детерминированных машин, Кибернетика, № 2 (1965), 9—17.
168. Таланов В. А., Задача о переналадках, Уч. зап. Горьковск. ун-та, вып. 105, 1969, 9—12.
169. Таланов В. А., Математическая модель некоторых задач теории расписаний, Сб. «Вычислительная техника в машиностроении», 1972, декабрь, 12—16.
170. Танаев В. С., К теории расписаний, ДАН БССР 8, № 12 (1964), 792—794.
171. Танаев В. С., Прерывания в детерминированных системах обслуживания с параллельными идентичными приборами, Изв. АН БССР, сер. физ.-мат. наук № 6 (1973), 44—48.
172. Танаев В. С., Метод решения задач дискретного программирования, Экономика и матем. методы 4, вып. 5 (1968), 776—782.
173. Танаев В. С., К задаче составления расписания работы поточной линии с одним автооператором, Инж.-физ. ж. 7, № 3 (1964), 111—114.
174. Танаев В. С., Об одной задаче теории расписаний, Изв. АН БССР, сер. физ.-мат. наук № 4 (1964), 128—131.
175. Танаев В. С., О числе перестановок n частично упорядоченных элементов, ДАН БССР 11, № 3 (1967), 208.
176. Танаев В. С., Некоторые оптимизируемые функции одностадийного производства, ДАН БССР 9, № 1 (1965), 11—14.
177. Танаев В. С., Об одном классе задач теории расписаний, Тр. III Всес. совещания по автомат. упр. (техн. кибернет.), 1965, М., «Наука», 1967, 211—214.
178. Танаев В. С., Левин Г. М., Об оптимальном поведении систем с частично ограниченной памятью, Изв. АН БССР, сер. физ.-мат. наук, № 3 (1967), 82—88.
179. Телемтаев М. М., К вопросу о решении задачи о коммивояжере, Изв. АН СССР, Техн. кибернетика, № 6 (1972), 94—100.
180. Титов В. В., Об одном алгоритме решения задачи о коммивояжере, Сб. «Оптимальн. планирование», вып. 7, Новосибирск, «Наука», 1967, 107—111.
181. Тютюкин В. К., Об оптимальном однооперационном календарном графике со станками-дублерами. Сб. «Прим. матем. в эконом.», вып. 5, Л., Ленингр. ун-та, 1969, 79—91.
182. Тютюкин В. К., Об оптимизации календарной длительности обработки изделий при одинаковой последовательности запуска их на всех станках. Сб. «Оптимальн. планирование», вып. 16, Новосибирск, 1970, 89—97.
183. Тютюкин В. К., Нахождение оптимальных планов в многооперационных процессах обработки изделий методом «ветвей и границ», Сб. «Применение матем. в эконом.», вып. 6, Л., Ленингр. ун-т, 1970, 29—45.

184. Фейгин Л. И., О влиянии возмущений на расписание, Изв. АН СССР, Техн. кибернетика, № 1 (1967), 14—15.
185. Фейгин Л. И., Применение метода статистического моделирования для определения оптимальной частоты коррекции расписания, Сб. «Оперативн. упр. произв.», М., «Наука», 1971, 60—67.
186. Фейгин Л. И., Управление и прогнозирование в задачах теории расписаний при неполной информации, ДАН СССР 200, № 6 (1971), 1298—1301.
187. Фейгин Л. И., Об осуществимости процесса обработки объектов множеством машин, Изв. АН СССР, Техн. кибернетика, № 2 (1964), 24—29.
188. Фейгин Л. И., Оптимальное расписание для одной машины при неполной информации, Кибернетика, № 4 (1971), 149—151.
189. Фейгин Л. И., Общая задача теории расписаний при неполной информации. Автоматика и телемеханика, № 3 (1972), 110—116.
190. Фишбейн С. М., О вероятностном подходе к решению задачи очередностей, Кибернетика, № 6 (1971), 95—97.
191. Фишер Г., Томпсон Г. Л., Комбинации локальных правил календарного планирования применительно к самонастраивающимся на вероятностной основе программ для ЭВМ, Сб. «Календарное планирование», М., «Прогресс», 1966, 260—290.
192. Харди Г. Г., Литтлвуд Дж., Полиа Г., Неравенства, М., ИЛ, 1948.
193. Хенкин В. Э., Об одном вопросе теории расписаний ($2 \times M$ задача упорядочения), Кибернетика, № 6 (1966), 67—71.
194. Хенкин В. Э., О некоторых последовательностях перестановок, возникающих в теории расписаний, Сб. «Пробл. кибернетики», вып. 20, 1968, 231—240.
195. Чернова Г. В., Задача об определении порядка запуска деталей, обеспечивающего минимум пролеживания деталей, Сб. «Прим. матем. в экон.», вып. 5, Л., Ленингр. ун-т, 1969, 147—158.
196. Чернова Г. В., Задача определения последовательности обработки деталей, имеющих одинаковые технологические маршруты, Сб. «Применение матем. в экон.», вып. 4, Л., Ленингр. ун-т, 1967, 66—81.
197. Чернявский А. Л., Алгоритмы для решения комбинаторных задач, основанные на методе неявного перебора, Автоматика и телемеханика, №№ 2, 3 (1972), 98—108, 76—82.
198. Шахбазян К. В., Тушкина Т. А., Метод ветвей и границ для задачи параллельного упорядочения, Зап. научн. семинаров Ленингр. отд. Мат. ин-та АН СССР 35 (1973), 146—155.
199. Шевченко В. Н., Задача о равномерном распределении простоев (несколько смен), Экономика и матем. методы 3, № 4 (1967), 619—623.

200. Шевченко В. Н., Задача оптимального календарного планирования с ограничением на число рабочих, Изв. высш. учебн. заведений, Радиофизика 8, № 3 (1965), 635—637.
201. Шевченко В. Н., Задача составления оптимального расписания работы на n станках, Сб. «Пробл. кибернетики», вып. 18, 1967, 129—146.
202. Шкурба В. В., О задачах упорядочения, Кибернетика, № 2 (1967), 63—65.
203. Шкурба В. В., Теория расписаний, 1—2 (доклад), Киев, 1964.
204. Шкурба В. В., Интервалы очередности в задачах упорядочения, Кибернетика, № 2 (1970), 77—79.
205. Шкурба В. В., Вычислительные схемы решения задач теории расписаний, Кибернетика, № 3 (1965), 72—76.
206. Шкурба В. В., К решению задачи о круговой расстановке станков, Кибернетика, № 3 (1967), 44—46.
207. Шкурба В. В., Некоторые вопросы автоматизированного управления предприятиями, Кибернетика, № 5 (1967), 66—77.
208. Шкурба В. В., Автоматизированные системы управления предприятиями. Структура, функции, направления разработки, Сб. «Автоматизир. системы упр. предприятиями», ИК АН УССР, вып. 2, 1968.
209. Шкурба В. В., Подчасова Т. П., Об одном подходе к организации и планированию работы предприятий массового производства, Киев, 1963.
210. Шкурба В. В., Подчасова Т. П., Пеннер Н. Я., Савицкий В. Н., Исследование эффективности некоторых функций предпочтения в задачах календарного планирования, Сб. «Системотехника», Киев, 1971, 79—96.
211. Шкурба В. В., Подчасова Т. П., Пшичук А. Н., Тур Л. П., Задачи календарного планирования и методы их решения, Киев, «Наукова думка», 1966.
212. Шрейдер Ю. А., Равенство, сходство, порядок, М., «Наука», 1971.
213. Юдин Д. Б., Теория расписаний, В кн. «Энциклопедия современной техники», т. 3, М., 1964, 151—152.
214. Якимов Р. М., Об одном функциональном уравнении, описывающем оптимальную последовательность обработки изделий, Сб. научн. тр. Пермск. политехн. ин-та, № 21, 1966, 62—71.
215. Якимов Р. М., Оптимальный вариант работы одной автоматической линии, Сб. научн. тр. Пермск. политехн. ин-та, № 21, 1966, 72—74.
216. Agin N., Optimum seeking with branch and bound, Manag. Sci. 13, № 4 (1966), B176—B185.
217. Akers S. E., Jr., A graphical approach to production scheduling problems, Operat. Res. J. 4, № 2 (1956), 244—245.
218. Akers S. E., Jr., Friedman J., A non-numerical approach to production scheduling problems, Operat. Res. III (1955), 429—442.
219. Arthanari T. S., Mukhopadhyay A. C., On some sequencing problems. A note on a paper by W. Szwarc, Nav. Res. Log. Quart. 18, № 1 (1971), 135—138.

220. A s h o u r S., Sequencing theory, Lect. Notes Econ. and Math. Syst. 69 (1972).
221. A s h o u r S., A decomposition approach for the machine scheduling problem, Internat. J. Product. Res. 6, № 2 (1967), 109—122.
222. A s h o u r S., An experimental investigation and comparative evaluation of flow-shop scheduling techniques, Operat. Res. 18, № 3 (1970), 541—549.
223. B a g g a P. C., Two machines scheduling problem, Logist. Rev. 3, № 14 (1967), 27—37.
224. B a g g a P. C., Two machine scheduling problem when passing allowend, Metrika 20, № 1 (1973), 36—40.
225. B a g g a P. C., C h a k r a v e r t i N. K., Minimizing waiting cost over two stage production schedule, Z. angew. Math. und Mech. 49, № 5 (1969), 299—301.
226. B a k e r K. R., M a r t i n J. B., An experimental comparison of solution algorithms for the singlemachine tardiness problem, Nav. Res. Log. Quart. 21, № 1 (1974), 187—199.
227. B a k s h i M. S., A r o r a S. R., The sequencing problem, Manag. Sci. 16, № 4 (1969), B247—B263.
228. B a l a s E., A note on the branch-and-bound principle, Operat. Res. 16, № 2 (1968), 442—445.
229. B a l a s E., Machine sequencing via disjunctive graphs: an implicit enumeration algorithm, Operat. Res. 17, № 6 (1969), 941—957.
230. B a n e r j e e B. P., Single facility sequencing with random execution times, Operat. Res. 13, № 3 (1965), 358—364.
231. B a n k B., «Branch and Bound» Algorithmen fur zwei Reinhenfolgeprobleme, Math. Operationsforsch. und Statist. 1, № 3 (1970), 217—228.
232. B a n k B., S e i f f a r t E., Ein «Branch and Bound» Algorithmus eines Reihenfolgeproblems, Ekon.-mat. obzor 5, № 1 (1969), 22—29.
233. B a r a n - J a r o s z B., Metoda rozwiązywania pewnego zagadneenia sekwencyjnego, Prz. statyst. 20, № 2 (1973), 181—192.
234. B a r a n - J a r o s z B., G r a b o w s k i W., Wyznaczanie kolejności detali przechodzacych w procesie produkcji ciag operacji, Prz. statyst. 18, № 3—4 (1971), 365—376.
235. B e l l m a n R., Dynamic programming treatment of the travelling salesman problem, S. Assoc. Comput. Machinery 9, № 1 (1962), 61—63.
236. B e l l m a n R., Mathematical aspects of scheduling theory, J. Soc. Industr. and Appl. Math. 4, № 3 (1956), 168—205.
237. B e l l m a n R., G r o s s O., Some combinatorial problems arising in the theory of multistage processes, J. Soc. Industr. and Appl. Math. 2, № 3 (1954), 175—183.
238. B e l l m o r e M., M a l o n e J. C., Pathology of travelling-salesman subtour-elimination algorithms, Operat. Res. 19, № 2 (1971), 278—307.
239. B l a u R. A., N-job, one machine sequencing problems under uncertainty, Manag. Sci. 20, № 1 (1973), 101—109.

240. Bowman V. J., Permutation polyhedra, SIAM J. Appl. Math. 22, № 4 (1972), 580—589.
241. Bowman E. H., The schedule-sequencing problem, Operat. Res. 7, № 5 (1959), 621—624.
242. Bozóki G., Richard J. P., A branch-and-bound algorithm for the continions-process job-shop scheduling problem, AIIE Trans. 2, № 3 (1970), 246—252.
243. Brown R. G., Simulations to explore alternative sequencing rules, Nav. Res. Log. Quart. 15, № 2 (1968), 281—286.
244. Brown A. P., Lomnicki Z. A., Some applications of the «branch-and-bound» algorthm to the machine scheduling problem, Operat. Res. Quart. 17, № 2 (1966), 173—186.
245. Burlaga H., Minimalizacja ilości wykonawców operacji o ustalonym harmonogrie, Biul. WAT J. Dabrowskiego 21, № 12 (1972), 93—99.
246. Burstall R. M., A heuristic method for a jobscheduling problem, Operat. Res. Quart. 17, № 3 (1966), 291—304.
247. Buzacott J. A., Dutta S. K., Sequencing many jobs on a multipurpose facility, Nav. Res. Log. Quart. 18, № 1 (1971), 75—82.
248. Byrne E. R., Optimal scheduling of indivisible jobs on a sequence of machines without job passing, Dissert. Abstrs. B28, № 5 (1967), 2021.
249. Cerny M., Algoritmus rešeni okružnich dopravnich uloh, Ekon.-mat. obzor 1, № 4 (1965), 391—405.
250. Charlton J. M., Death C. C., A method of solution for general machine-scheduling problems, Operat. Res. 18, № 4 (1970), 689—707.
251. Christofides N., Eilon S., Algorithms for large scale travelling salesman problems, Operat. Res. Quart. 23, № 4 (1972), 511—518.
252. Contini B., Un modello di sequenze di produzione risolvibile con la program mazione linneare, Metroeconomica 20, № 1 (1968), 36—41.
253. Conway R. W., Johnson B. M., Maxwell W. L., An experimental investigation of priority dispatching, J. Industr. Engng. 11, № 3 (1960), 221—229.
254. Conway R. W., Maxwell W. L., Miller L. W., Theory of scheduling, New York, Addison — Wesley, 1967.
255. Conway R. W., Maxwell W. L., Oldziey J. W., Sequencing against due-dates, Proc. 4th. Int. Conf. Operat. Res., Boston, New York e. a. (1966), 599—618.
256. Croes G. A., A method for solving travelling-salelsman problems, Operat. Res. 6, № 6 (1958), 791—812.
257. Dantzig G. B., A machine-job scheduling model, Manag. Sci. 6, № 2 (1960), 191—196.
258. Dantzig G. B., Fulkerson D. R., Johnson S. M., On a linear-programming, combinatorial approach to the travelling-salesman problem, Operat. Res. 7, № 1 (1959), 58—66.
259. Dantzig G. B., Fulkerson D. R., Johnson S. M., Solution of a large-scale travelling-salesman problem, J. Operat. Res. Soc. Amer. 2, № 3 (1954), 393—410.

260. Day J. E., Hottenstein M. P., Reveew of sequencing research, Nav. Res. Log. Quart. **17**, № 1 (1970), 11—39.
261. Derman C., Klein M., Surveillance of multi-component systems: a stochastic travelling salesman's problem. Nav. Res. Log. Quart. **13**, № 2 (1966), 103—111.
262. Dessouky M. I., Margenthaler C. R., The one-machine sequencing problem with early starts and dates, AIIE Trans. **4**, № 3 (1972), 214—222.
263. Dilworth R. P., A decomposition for partially ordered sets, Ann. of Math. **51** (1950), 161—166.
264. Ducamp A., Un probleme d'affectation, Cahiers Centre etudes recn. operat. **8**, № 1 (1966), 69—72.
265. Dudek R. A., Chare P. M., Makespan sequencing on *m*-machines, J. Industr. Engng. **18**, № 1 (1967), 131—134.
266. Dudek R. A., Teuton O. F. J., Development of *M*-stage decision rule for scheduling *n* jobs through *M* machines, Operat. Res. **12**, № 3 (1964), 471—497.
267. Eastman W. L., Comments on a paper by Schild and Fredman, Manag. Sci. **11**, № 7 (1965), 754—755.
268. Eastman W. L., Even S., Isaacs I. M., Bounds for the optimal scheduling of *n* jobs on *m* processers, Manag. Sci. **11**, № 2 (1964), 268—279.
269. Elmagraby S. E., The sequencing of «related» jobs, Nav. Res. Log. Quart. **15**, № 1 (1968), 23—32.
270. Elmagraby S. E., The one machine sequencing problem with delay costs, J. Industr. Engng **19**, № 2 (1968), 105—108.
271. Elmagraby S. E., The machine sequencing problem-review and extensions, Nav. Res. Log. Quart. **15**, № 2 (1968), 205—232.
272. Emmons H., One-machine sequencing to minimize certain functions of job tardiness, Operat. Res. **17**, № 4 (1969), 701—715.
273. Fife D. W., Scheduling with random arrivals and linear loss functions, Manag. Sci. **11**, № 3 (1965), 429—437.
274. Fujii M., Kasami T., Ninomiya K., Optimal sequencing of two equivalent processors, SIAM J. Appl. Math. **17**, № 4 (1969), 784—789.
275. Fukunaga L., Tamura H., Haveda H., Simulating a production scheduling control systems, IEEE Trans. Automat. Control **12**, № 1 (1967), 102—103.
276. Gapp W., Mankekar P. S., Mitten L., Sequencing operations to minimize in-process inventory costs, Manag. Sci. **2**, № 3 (1965), 476—484.
277. Gaunt S., A non-computer method using search for resolving the travelling salesman problem, CORS Journal **6**, № 1 (1968), 44—54.
278. Gavett J. W., Three heuristic rules for sequencing jobs to a single production facility, Manag. Sci. **11**, № 8 (1965), 166—176.
279. Gere W. S. J., Heuristics in job shop scheduling, Manag. Sci. **13**, № 3 (1966), 167—190.

280. Giffler B., Schedule algebra: a progress report, Nav. Res. Log. Quart. 15, № 2 (1968), 255—280.
281. Giffler B., Thompson G. L., Algorithms for solving production-scheduling problems, Operat. Res. 8, № 4 (1960), 487—503.
282. Giglio R. J., Wagner H. M., Approximate solutions to the threemachine scheduling problem, Operat. Res. 12, № 2 (1964), 305—324.
283. Gilmore P. C., A solvable case of the travelling salesman problem, Canad. Math. Bull. 9, № 6 (1966), 743—744.
284. Gilmore P. C., Gomory R. E., Sequencing a one state-variable machine: a solvable case of the travelling salesman problem, Operat. Res. 12, № 5 (1964), 655—679.
285. Gilmore P. C., Gomory R. E., A solvable case of the travelling salesman problem. Proc. Nat. Acad. Sci. USA 51 (1964), 178—181.
286. Glassey C. R., Minimum change — over scheduling of several products on one machine, Operat. Res. 16, № 2 (1968), 342—352.
287. Gonzalez Mario J. Jr., Ramamoorthy C. V., Parallel task execution in a decentralised system, IEEE Trans. Comput. 21, № 12 (1972), 1310—1331.
288. Gotterer M. H., On the development of a supervisory sequencing routine, On Line Data Process, New York, N. Y., Inst. Electr. and Electron. Engrs., Inc. (1963), 22—32.
289. Goyal S. K., The tree development method for solving the travelling-salesman problem, Internat. J. Product. Res. 9, № 2 (1971), 239—246.
290. Greenberg H. H., A branch-bound solution to the general scheduling problem, Operat. Res. 16, № 2 (1968), 353—361.
291. Gupta Jatinder N. D., A general algorithm for the $n \times M$ flowshop scheduling problem, Internat. J. Product. Res. 7, № 3 (1969), 241—247.
292. Gupta Jatinder N. D., A functional heuristic algorithm for the flowshop scheduling problem. Operat. Res. Quart. 22, № 1 (1971), 39—47.
293. Gupta Jatinder N. D., An improved combinatorial algorithm for flowshopscheduling problem, Operator. Res. 19, № 7 (1971), 1753—1758.
294. Gupta Jatinder N. D., Maykut A. R., Flow-shop scheduling by heuristic decomposition, Internat. J. Product. Res. 11, № 2 (1973), 105—111.
295. Hardgrave W. W., Nemhauser G. L., A geometric model and a graphical algorithm for a sequencing problem, Operat. Res. 11, № 6 (1963), 889—900.
296. Heck H., Roberts S., A note on the extension of a result on scheduling with secondary criteria, Nav. Res. Log. Quart. 19, № 2 (1972), 403—405.
297. Held M., Karp R. M., The travelling-salesman problem and minimum spanning trees, Math. Program. 1, № 1 (1971), 6—25.

298. Held M., Karp R. M., A dynamic programming approach to sequencing problems, J. Soc. Industr. and Appl. Math. 10, № 1 (1962), 196—210.
299. Held M., Karp R. M., The travelling-salesman problem and minimum spanning trees, Operat. Res. 18, № 6 (1970), 1138—1162.
300. Heller J., Logemann G., An algorithm for construction and evaluation of feasible schedules, Manag. Sci. 8, № 2 (1962), 168—183.
301. Henry-Labordére A. L., The record balancing problem: a dynamic programming solution of a generalized traveling salesman problem, Rev. franç. inform. et rech. operat. 3, № B—2 (1969), 43—49.
302. Horn W. A., Minimizing average flow time with parallel machines, Operat. Res. 21, № 3 (1973), 846—847.
303. Horn W. A., Single-machine job sequencing with treelike precedence ordering and linear delay penalties, SIAM. J. Appl. Math. 23, № 2 (1972), 189—202.
304. Horn W. A., Some simple scheduling algorithms, Nav. Res. Log. Quart. 21, № 1 (1974), 177—185.
305. Hu T. C., Decomposition in traveling-salesman problems, Proc. 4th Int. Conf. Oper. Res., Boston, New York e. a. (1966), 1021—1027.
306. Hu T. C., Parallel sequencing and assembly line problems, Operat. Res. 9, № 6 (1961), 841—848.
307. Ibaraki T., Scheduling and integer programming, Сисутэму то сэйгё, Syst. and Contr. 17, № 5 (1973), 269—274.
308. Ignall E., Schrage L., Application of the branch and bound technique to some flowshop scheduling problems, Operat. Res. 13, № 3 (1965), 400—412.
309. Jackson J. R., Scheduling a production line to minimize maximum tardiness, Research Report 43, Manag. Sci. Res. Project, UCLA, Jan., 1955.
310. Jackson J. R., An extension of Johnson's results on job lot scheduling, Nav. Res. Log. Quart. 3, № 3 (1956), 201—203.
311. Jaeschke G., Vicinal sequecing problems, Operat. Res. 20, № 5 (1972), 984—992.
312. Jankowska-Zorychta Z., Modele uszeregowania prac na maszynach, Prz. statyst 20, № 1 (1973), 11—25.
313. Johnson S. M., Optimal two-and-three stage production schedules with set-up times included, Nav. Res. Log. Quart. 1, № 1 (1954), 61—68.
314. Johnson S. M., Discussion: sequencing n-jobs on two machines with arbitrary time lags, Manag. Sci. 5, № 3 (1959), 299—303.
315. Karg R. L., Thompson G. L., A heuristic approach to solving travelling salesman problem, Manag. Sci 10, № 2 (1964), 225—248.
316. Kirisu T., Flow shop sequencing problem with time lags, Technol. Repts. Osaka Univ. 22 (1972), 303—314.
317. Kise H., Scheduling and branch-and-bound method, Syst. and Contr. 17, № 5 (1973), 275—281.

318. Knödel W., Kürzeste Rundreise zwischen n Orten (Travelling salesman problem), Computing 3, № 2 (1968), 151—156.
319. Korte B., Über eine Klasse kombinatorischer Extremalprobleme. Ber. Ges. Math. und Datenverarb. № 57 (1972), 123—134.
320. Korth H., Losung eines Maschinenbelegungsproblems 9. Internat. Kolloq. Techn. Hochschul Ilmenan (1964), okon. Ind., 5—6.
321. Kurisu T., Two-machine sequencing problem with exponential processing times, Technol. Rept. Osaka Univ. 23, March (1973), 1—8.
322. Lanzenauer C. H. von, Himes R. S., A linear programming solution to the general sequencing problem, CORS Journal 8, № 2 (1970), 129—134.
323. Lasdon L., Solving multi-item scheduling problems, 20th Ann. ISA Conf. Proc., Los. Angeles, 1965, vol. 20, part. 3, Pittsburgh, Pa, 1/2, № 31 (1965), 1—4.
324. Lasdon L. S., Terjung R. S., An efficient algorithm for multi-item scheduling, Operat. Res. 19, № 4 (1971), 946—969.
325. Lawler E. L., Optimal sequencing of a single machine subject to precedence constraints, Manag. Sci. 19, № 5 (1973), 544—546.
326. Lawler E. L., On scheduling problems with deferral costs, Manag. Sci. 11, № 2 (1964), 280—288.
327. Lawler E. L., Moore J. M., A functional equation and its application to resource allocation and sequencing problems, Manag. Sci. 16, № 1 (1969), 77—84.
328. Lawler E. L., Wood D. E., Branch-and-bound methods: a survey, Operat. Res. 14, № 4 (1966), 699—719.
329. Ligtenberg E., Minimal cost sequencing of n-grouped and ordered jobs on m-machine, J. Industr. Engng. 17, № 4 (1966), 217—223.
330. Lin S., Kernighan B. W., A heuristic technique for solving a class of combinatorial optimization problems, Proc. 5th Annu. Princeton Conf. Inform. Sci. and Syst., 1971, Princeton, N. J. s. a., 210—213.
331. Litsios S., A resource allocation problem, Operat. Res. 13, № 6 (1965), 960—988.
332. Little J., Murty K., Sweeney D., Karel C., An algorithm for the travelling salesman problem, Operat. Res. 11, № 6 (1963), 972—989.
333. Lomnicki Z. A., A «branch-and-bound» algorithm for the exact solution of the three-machine scheduling problem, Operat. Res. Quart. 16, № 1 (1965), 89—100.
334. Mack C., The elimination-reduction method of solving the symmetric travelling salesman problem, New J. Statist. and Operat. Res. 3, № 3 (1967), 3—15.
335. Mack C., Further rules for the elimination-reduction method of solving the symmetric Travelling Salesman problem, New J. Statist. and Operat. Res. 5, № 2 (1969), 8—15.
336. Mack C., More improvements in the elimination-reduction method for the symmetric travelling salesman problem. New J. Statist. and Operat. Res. 6, № 3 (1970), 3—11.

337. Makino Toji, On a scheduling problem, J. Operat. Res. Soc. Japan 8, № 1 (1965), 32—44.
338. Manne A. S., On the job-shop scheduling problem, Operat. Res. 8, № 2 (1960), 219—223.
339. Maxwell W. L., On sequencing n-jobs on one machine to minimize the number of late jobs, Manag. Sci. 16, № 5 (1970), 295—297.
340. Maxwell W. L., Mehra M., Multiple-factor rules for sequencing with assembly constraints. Nav. Res. Log. Quart. 15, № 2 (1968), 241—254.
341. McMahon G. B., Optimal production schedules for flow shops. CORS Journal 7, № 2 (1969), 141—151.
342. McMahon G. B., Note on optimal production schedules. CORS Journal 7, № 2 (1969), 154—155.
343. McMahon G. B., Burton P. G., Flow-shop scheduling with the branch-and-bound method, Operat. Res. 15, № 3 (1967), 473—481.
344. McNaughton R., Scheduling with deadlines and loss functions, Manag. Sci. 6, № 1 (1959), 1—12.
345. Mellor P., Job shop scheduling, Product. Engr. 46, № 2 (1967), 82—87.
346. Mensch G., Generalization of Aker's two-demensional job shop scheduling model to n dimensions. SIAM J. Appl. Math. 18, № 2 (1970), 462—466.
347. Mensch G., Single-stage linear programming zero-one solutions to some travelling salesman type problems. INFOR. Can. J. Operat. Res. and Inform. Process 9, № 3 (1971), 282—292.
348. Mierlea V., Algoritm pentru ordonantarea a n lucrâri pe m masini cu timp interoperational minim la pornire si opire, Stud. si cerc. calc. ekon. si cibern. econ., № 3 (1972), 45—59.
349. Miller C. E., Tucker A., Zemlin R. A., Integer programming formution of traveling salesman problems, I. Assoc. Comput. Machinery 7, № 4 (1960), 326—329.
350. Mitten L. G., Sequencing n-jobs on two machines with arbitrary time lags, Manag. Sci. 5, № 3 (1959), 293—298.
351. Moore J. C., Vandegrift J. B., Linear programming technique for assigning work to machines, Machinery 96, № 2466 (1960), 383—385.
352. Moore J. M., An n-job, one machine sequencing algorithm for minimizing the number of late jobs, Manag. Sci. 15, № 1 (1968), 102—109.
353. Morton G., Land A. H., A contribution to the «travelling salesman» problem, J. Roy. Statist. Soc. № 2 (1955), 185—194.
354. Müller-Merbach H., Optimale Reihenfolgen (Ökon. und Unternehmensforsch., 15), Berlin — Heidelberg — New York, Springer, 1970, X, 2255.
355. Muntz R. R., Coffman E. A., Preemptive scheduling of real-time tasks on multiprocessors systems, J. Assoc. Comput. Mach. 17, № 2 (1970), 324—338.

356. Muntz R. R., Coffman E. A., Optimal preemptive scheduling on two-processor systems, IEEE Trans. Computers C-18, № 11 (1969), 1014—1020.
357. Murty Katta G., A fundamental problem in linear inequalities with applications to travelling salesman problem, Math. Progr. **2**, № 3 (1972), 296—308.
358. Murty Katta G., On the tours of a travelling salesman, SIAM J. Control **7**, № 1 (1969), 122—131.
359. Nabeshima I., Sequencing on two machines with start lag and stop lag, J. Operat. Res. Soc. Japan **5**, № 3 (1963), 97—101.
360. Nabeshima I., Unified treatment on single machine scheduling problems, Repts. Univ. Electro-Communs, **22**, № 2 (1971), 29—38.
361. Nabeshima I., Some extensions of the m-machine scheduling problem, J. Operat. Res. Soc. Japan **10**, № 1—2 (1967), 1—17.
362. Nabeshima I., The order of n-items processed on m — machines.1, J. Operat. Res. Soc. Japan **3**, № 4 (1961), 170—175.
363. Nabeshima I., The order of n-items processed on m-machines. 2, J. Operat. Res. Soc. Japan **4**, № 1 (1961), 1—8.
364. Nabeshima I., Computational solution to the m-machine scheduling problem, J. Operat. Res. Soc. Japan **7**, № 3 (1965), 93—103.
365. Németi L., Über die Durchführungsdauer eines Arbeitganges in der Ablaufplanung, Bull. math. Soc. sci. math. RSR **19**, № 3 (1969), 81—86.
366. Németi L., On the scheduling problem in the case of several machines of the same type, Mathematica (RSR) **13**, № 2 (1971), 251—261.
367. Németi L., Das Reihefolgeproblem in der Fertigung mit Pufferbestanden, Rev. roumain math. pures et appl. **13**, № 7 (1968), 1009—1016.
368. Németi L., Das Reihenfolgeproblem in der Fertigungsprogrammierung und Lenearplanung mit Logischen Bedingungen, Mathematica (RPR) **6**, № 1 (1964), 87—99.
369. Németi L., Rado F., Ein Wartezeitproblem in der programmierung der Produktion, Mathematica (RPR) **5**, № 1 (1963), 65—96.
370. Nemhauser G. L., Bellmore M., The travelling salesman problem: a survey, Operat. Res. **16**, № 3 (1968), 538—558.
371. Obruča A. K., Spanning tree manipulation and the travelling salesman problem, Comput. J. **10**, № 4 (1968), 374—377.
372. Okamura K., Yamashina H., Eastablishment of linear sequences, Mem. Fas. Engng. Kyorto Univ. **31**, № 3 (1969), 307—331.
373. Page E. S., An approach to the scheduling of jobs on machines, J. Roy. Statist. Soc. **B23**, № 2 (1961), 484—492.
374. Page E. S., On Monte Carlo methods in congestion problems 1. Searching for an optimum in discrete situations, Operat. Res. **13**, № 2 (1965), 291—299.

375. Page E. S., On the scheduling of jobs by computer, Comput. J. 5, № 3 (1962), 214—220.
376. Palmer D. S., Sequencing jobs through a multistage process in the minimum total time — a quick method of obtaining a near optimum, Operat, Res. Quart. 16, № 1 (1965), 101—107.
377. Parsons J. A., Matrix methods. J. Syst. Manag. 20, № 8 (1969), 38—40.
378. Petersen C. C., Solving sequencing problems through reordering operations, AIIE Trans. 5, № 1 (1973), 68—73.
379. Petick J., Ein Beitrag zum Rundreiserproblem, Wiss. Z. Hochsch. Archit. und Bauwesen Weimar 17, № 2 (1970), 193—196.
380. Piehler J., Ein Beitrag zum Reihenfolgerproblem, Unternehmensforschung 4, № 3 (1960), 138—142.
381. Pritsker A., Alau B., Watters L. J., Wolfe Ph. M., Multiprogect scheduling with limited resources: a zero-one programming approach, Manag. Sci. 16, № 1 (1969), 93—108.
382. Rau J. G., Minimizing a function of permutations of n integers, Operat. Res. 19, № 1 (1971), 237—240.
383. Reddi S. S., Ramamoorthy C. V., On the flow-shop sequencing problem with no wait in process, Operat. Res. Quart. 23, № 3 (1972), 323—331.
384. Reiter S., Sherman C., Discrete optimizing, J. Soc. Industr. and Appl. Math. 13, № 3 (1965), 864—889.
385. Roberts S. M., Flores B., An engineering approach to the travelling salesman problem, Manag. Sci. 13, № 3 (1966), 269—288.
386. Root J. G., Scheduling with deadlines and loss functions on *k* parallel machines, Manag. Sci. 11, № 3 (1965), 460—475.
387. Rothkopf M. H., Scheduling independent tasks on parallel processors, Manag. Sci. 12, № 5 (1966), 437—447.
388. Rothkopf M. H., Scheduling with random service times, Manag. Sci. 12, № 9 (1966), 707—713.
389. Roy B., Phisionomie et traitement des problemes d ordonnacement, Gestion 6, avr. (1963), 220—227.
390. Saksena L. P., Mathematical model of scheduling clients through welfare agencies, CORS Journal 8, № 3 (1970), 185—200.
391. Salah E. E., Mallik A. K., Nuttle H. L. W., The scheduling of lots on a single facility, AIIE Trans. 2, № 3 (1970), 203—213.
392. Salveson M. E., A computational technique for the scheduling problem, J. Industr. Engng. 13, № 1 (1962), 30—41.
393. Sanderova J., Matematičky model pro, rešeni jisteho sekvencniho problemu, Ekon-mat. obz. 7, № 1 (1971), 48—60.
394. Schild A., Fredman I. J., On scheduling tasks with associated linear loss functions, Manag. Sci. 7, № 3 (1961), 280—285.
395. Schild A., Fredman I. J., Scheduling tasks with deadlines and non-linear functions, Manag. Sci. 8, № 1 (1962), 73—81.

396. Schoch M., Ein Algorithmus zur exakten Bestimmung aller Optimallösungen des Rundreiseproblems, Wiss. Z. Techn. Hochschule Karl-Marx-Stadt 8, №№ 4—5 (1966), 247—255.
397. Schoch M., Ein Erwarterungsprinzip als Konzepzion zur Lösung kombinatorischer Optimierungsprobleme, Math. Operationforsch. und Statist. 1, № 4 (1970), 265—280.
398. Schrage L., Solving resource-constrained network problems by implicit enumeration nonpreemptive case, Operat. Res. 18, № 2 (1970), 263—278.
399. Schwartz E. S., A heuristic procedure for parallel sequencing with choice of machines, Manag. Sci. 10, № 4 (1964), 767—777.
400. Seiffart E., Exakte und approximative Losungsmoglichkeiten von Reihenfolgeproblemen, Elektron. Informationsverarb. und Kybernet. 2, № 3 (1966), 123—150.
401. Seiffart E., Ein Reihenfolgeproblem. Colloq. on applic. mathem. to econ. Budapest, 1963, Hung. Acad. Sci. (1965), 339—341.
402. Shapiro D. M., Algorithms for the solution of the optimal cost and bottleneck travelling salesman problems, Dissert. Abstrs. B27, № 2 (1966), 550—551.
403. Shwimer J., On the n-job, one-machine, sequence-independent scheduling problem with tardiness penalties. A branch-bound solution, Manag. Sci. 18, № 6 (1972), 301—313.
404. Sidney J. B., An extension of Moore's due date algorithm, Lect. Notes Econ. and Math. Syst. 86 (1973), 393—398.
405. Sisson R. L., Methods of sequencing in job shops (A review), Operat. Res. 7, № 1 (1959), 10—29.
406. Sisson R. L., Sequencing theory, Progr. Operat. Res., vol. 1, New York — London, John Wiley and Sons, Inc. (1961), 293—326.
407. Smith R. D., Dudek R. A., A general algorithm for solution of the n-job, m-machine sequencing problem of the flow shop, Operat. Res. 15, № 1 (1967), 71—82.
408. Smith R. D., Dudek R. A., A general algorithm for solution of the n-job, m-machine sequencing problem of the flow shop. Errata, Operat. Res. 17, № 4 (1969), 756.
409. Smith W. E., Various optimizers for single stage production, Nav. Res. Log. Quart. 3, №№ 1—2 (1956), 59—66.
410. Solich R., Задача календарного планирования для циклически повторяющегося производства, Pr. CO PAN, № 80 (1972).
411. Srinivasan V., A hybrid algorithm for the one machine sequencing problem to minimize total tardiness, Nav. Res. Log. Quart. 18, № 3 (1971), 317—327.
412. Srivastava S. S., Kumar Santosh, Garg R. C., Sen Prasenjit. Generalized travelling salesman problem trough n sets of nodes, CORS Journal 7, № 2 (1969), 97—101.
413. Sturm L. B. J. M., A simple optimality proof of Moore's sequencing algorithm, Manag. Sci. 17, № 1 (1970), 116—118.
414. Summer A., Optimal and sub-optimal algorithms for the assignment and travelling-salesman problems, Brit. Joint Computer Conf., London (1966), 9—15.

415. Sussmann B., Scheduling problems with interval disjunctions, Z. Operat. Res. Ser. A16, № 5 (1972), 165—178.
416. Szwarc W., O pewnym zagadnieniu kolejnosci. Cz. I, Prz. statyst. 9, № 4 (1962), 367—382.
417. Szwarc W., O pewnym zagadnieniu kolejnosci. Cz. II, Prz. statyst. 10, № 1 (1963), 139—154.
418. Szwarc W., On some sequencing problems, Nav. Res. Log. Quart. 15, № 2 (1968), 127—155.
419. Szwarc W., Elimination methods in the $m \times n$ sequencing problem, Nav. Res. Log. Quart. 18, № 3 (1971), 295—305.
420. Szwarc W., Solution of the Akers — Friedman scheduling problem, Operat. Res. 8, № 6 (1960), 782—788.
421. Takagi Masahide, Graphical solution of 2-job m-machine sequencing problem, Дэнки сикэнсё ихо, Bull. Elektrotechn. Lab. 25, № 8 (1961), 575—584.
422. Talwar P. P., A note on sequencing problem with uncertain job times, J. Operat. Res. Soc. Japan 9, №№ 3—4 (1967), 93—97.
423. Terno J., Algorithmen für das klassische Maschinenbelegungsproblem, Math. Operationsforsch. und Statist. 3, № 3 (1972), 195—201.
424. Tyagi M. S., The travelling-salesman problem, Math. Semin. 4, № 4 (1968), 168—176.
425. Usn S. G. B., Scheduling to minimize the number of late jobs when set-up and processing times are uncertain, Manag. Sci. 19, № 11 (1973), 1283—1288.
426. Webb M. H. J., Some methods of producing approximate solutions to travelling salesman problems with hundreds or thousands of cities, Operat. Res. Quart. 22, № 1 (1971), 49—66.
427. Weinberg F., Behandlung des verallgemeinerten Traveling-Salesman-Problems und seiner Anwendungen im Sequecing mit ganzzahliger Programmierung, Industr. Organis. 36, № 4 (1967), 127—133.
428. Werra D. de, Balanced schedules, INFOR. Can. J. Operat. Res. and Inform. Process. 9, № 3 (1971), 230—237.