

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ВОЛОГОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**Л.В. Усов, Р.А. Усова**

**СЕТЕВОЕ МОДЕЛИРОВАНИЕ  
СТРОИТЕЛЬНЫХ ПРОЦЕССОВ:**  
*прикладные модели и методы исследования  
строительных процессов*

*Утверждено редакционно-издательским советом университета  
в качестве учебного пособия*

Вологда  
2014

УДК 69. 003 (075.8)

ББК 65.31я73

У 74

**Рецензенты:**

заместитель директора по науке Вологодского филиала академии МУБиНТ,  
доктор экономических наук, профессор **Л.С. Усов**;  
начальник управления технической поддержки Вологодского отделения № 8638  
ОАО «Сбербанк России», кандидат технических наук, доцент **А.В. Тупицын**

**Усов, Л.В.**

У 74 **Сетевое моделирование строительных процессов: прикладные модели и методы исследования строительных процессов: учебное пособие / Л.В. Усов, Р.А. Усова. - Вологда: ВоГУ, 2014. - 114 с.**

В учебном пособии по дисциплине «Методы решения научно-технических задач в строительстве» рассмотрены сетевые методы и модели исследования строительных процессов. Пособие подготовлено для магистрантов направления 08.04.01 – Водоотведение и очистка сточных вод, однако материал учебного пособия может быть использован при изучении других дисциплин, таких как «Организация и технология производства», «Математическое моделирование производственных процессов», «Математические модели и методы производственно-технических систем», «Экономико-математические модели и методы» и т.п., студентами и магистрантами других профилей как академического, так и прикладного направлений подготовки, а также аспирантами и преподавателями других кафедр университета.

УДК 69. 003 (075.8)

ББК 65.31я73

© Усов Л.В., Усова Р.А., 2014

© ВоГУ, 2014

## Оглавление

ВВЕДЕНИЕ.....	4
1. СЕТЕВОЕ МОДЕЛИРОВАНИЕ КАК МЕТОД ИССЛЕДОВАНИЯ СТРУКТУРНО-ДИНАМИЧЕСКИХ ПРОЦЕССОВ В СТРОИТЕЛЬСТВЕ .....	6
1.1. Сетевая модель .....	6
1.1.1. Общие сведения и терминология.....	6
1.1.2. Задачи исследования строительных процессов на основе сетевых моделей.....	10
1.2. Основные понятия и определения сетевого планирования (сетевого графика).....	16
1.3. Правила построения сетевых графиков .....	24
1.4. Нумерация событий.....	30
2. РАСЧЕТ ВРЕМЕННЫХ ПАРАМЕТРОВ СЕТЕВОГО ГРАФИКА .....	37
2.1. Расчет ранних характеристик сетевого графика .....	37
2.2. Нахождение критического пути по ранним характеристикам .....	42
2.3. Расчет поздних характеристик и резервов времени .....	44
2.4. Методы реализации алгоритма расчета $T_j^{(p)}$ и $T_j^{(n)}$ .....	50
2.4.1. Расчет временных параметров сетевого графика на самом графике .....	50
2.4.2. Расчет временных параметров сетевого графика в матричной форме .....	53
2.4.3. Расчет временных характеристик сетевого графика в форме таблицы .....	56
2.5. Метод потенциалов.....	58
2.6. Алгоритм выявления циклов .....	60
3. РАСЧЕТ ОПТИМАЛЬНОГО ИСПОЛЬЗОВАНИЯ РЕСУРСОВ .....	66
3.1. Общие замечания .....	66
3.2. Постановка задачи при постоянных интенсивностях .....	67
3.3. Алгоритм решения задачи .....	68
3.4. Оптимизация сетевого графика по стоимости .....	75
4. ДРУГИЕ ПРИЛОЖЕНИЯ СЕТЕВОГО МОДЕЛИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ В ВОДОСНАБЖЕНИИ И ВОДООТВЕДЕНИИ .....	78
4.1. Построение кратчайшей связывающей сети водоснабжения.....	78
4.2. Оптимизация очередности строительства связывающей сети методом Джонсона.....	84
4.3. Оптимизация очередности строительства связывающей сети методом ПСП.....	88
4.4. Задача о максимальном потоке или о пропускной способности сети.....	92
4.5. Концепция критерия финансово-экономических результатов и методика оценки материальных затрат предприятия .....	100
4.5.1 Концепция критерия финансово-экономических результатов .....	100
4.5.2 Методика оценки материальных затрат предприятия .....	104
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	113

## ВВЕДЕНИЕ

Характерной особенностью современной эпохи является необычайно бурное развитие науки и техники. Основная тенденция этого развития заключается в переходе от отдельных устройств к созданию больших систем или сложных комплексов. Ясно, что методы решения научно-технических задач в строительстве в области проектирования, планирования и управления созданием таких систем или комплексов должны соответствовать уровню требований и сложности задач создаваемого объекта.

Это возможно только тогда, когда практика проектирования, планирования и управления будет базироваться на научной основе, на современных методах решения научно-технических задач в строительстве, когда сами методы планирования будут развиваться и совершенствоваться вместе с развитием объекта исследования, когда арсенал таких методов будет пополняться новыми эффективными методами, приемами и способами решения научно-технических задач.

Иными словами как практика проектирования, планирования и управления в строительстве, так и подходы и методы решения научно-технических задач в строительстве должны представлять гибкую динамическую систему.

Одним из таких подходов, который пригоден для проектирования, планирования и управления именно больших систем или сложных комплексов, является сетевое моделирование и планирование строительных процессов. Многие специалисты считают методы сетевого моделирования и планирования крупнейшим достижением науки об организации и управлении. Опыт показывает, что методы сетевого моделирования и планирования позволяют резко улучшить качество планирования и координации совместных действий множества исполнителей, занятых выполнением некоторого набора работ. Результат деятельности отдельных исполнителей в этом случае немало зависит от четкой слаженной работы всех участников.

Настоящее пособие посвящено изложению элементарных основ метода сетевого моделирования строительных процессов.

В данном учебном пособии изложены основные, фундаментальные идеи сетевых моделей и методов, которые, с одной стороны, нашли более широкое применение в практике под общим названием « Методы сетевого планирования и управления», и, с другой стороны, включены в программы подготовки инженеров и экономистов, освоение которых дает возможность самостоятельного изучения других сетевых моделей и методов, которые не нашли отражения в данном пособии (альтернативные и вероятностные сети, сети Петри, сетевые системы типа Циклон и др.).

Пособие ориентировано в основном на студентов и магистров строительного и экономического профилей, но может быть использовано и студентами других специальностей вуза, а также аспирантами и преподавателями.

Отзывы и замечания просим направлять по адресу:

г. Вологда

ул. Гагарина, 81<sup>а</sup> каб. 413

Кафедра экономической теории, учета и анализа.

# 1. СЕТЕВОЕ МОДЕЛИРОВАНИЕ КАК МЕТОД ИССЛЕДОВАНИЯ СТРУКТУРНО-ДИНАМИЧЕСКИХ ПРОЦЕССОВ В СТРОИТЕЛЬСТВЕ

## 1.1. СЕТЕВАЯ МОДЕЛЬ

### *1.1.1. Общие сведения и терминология*

Моделирование как метод научного познания в теории познания науке и практике известен давно. Исследование поведения сложных технологических процессов и систем на моделях принято называть их моделированием. Под моделью следует понимать некоторый образ реального процесса или объекта, свойства которого правильно отражают свойства объекта моделирования с точки зрения достижения цели исследования. К исследованию функционирования реальных объектов – систем и процессов на моделях – прибегают всегда, когда изучение их свойств невозможно иначе, либо дает множество полезной, в том числе наглядной информации, например, изучение земли с помощью глобуса или строения атома на его модели. Но особую ценность для науки и практики представляют математические модели исследования поведения технологических процессов и сложных систем, так как они позволяют привлечь к анализу весь арсенал современной математики, информационных систем и компьютерной техники.

Сеть как инструмент моделирования и анализа функционирования сложных технических систем и технологических процессов ученым и инженерам также известна давно. Достаточно указать на широко используемые в инженерной практике методы анализа электрических схем и цепей. Но эти методы не применялись для исследования строительных процессов и анализа функционирования организационно-экономических систем. Конечно, к середине прошлого столетия уже был накоплен значительный опыт по научной организации управления производством, руководствуясь которым инженеры и администрация фирмы имели возможность способствовать улучшению организации планирования и управления фирмой. Но универсального метода моделирования, исследования и анализа структуры и динамики функционирования сложных технико-технологических и организационно-экономических процессов и систем не существовало.

Такой метод был создан только в середине 50-х годов 20-го столетия специалистами Американской ассоциации управления (АМА – American Management Association) и применен впервые для координации деятельности многих фирм и управления работами по программе «Поларис». В основу инструментария для моделирования взаимосвязей и анализа взаимовлияния отдельных элементов системы, а также для координации комплекса работ и

различных видов деятельности в целях наиболее успешного выполнения программы и была положена сеть.

Математической основой анализа сетевых моделей явились идеи и методы теории графов, математической статистики и исследования операций, которые к тому времени были уже достаточно развиты и нашли применение в анализе организационно-экономических систем, а потому ученым, инженерам и экономистам-практикам были понятны. Правда, при взаимосвязанных комплексах работ обычно использовались ленточные графики Генри Лоуренса Ганта. Однако подход, предложенный АМА, не только не повторял старое, а, наоборот, содержал много нового. Возникший из потребностей практики, он представлял сочетание новых идей, практического опыта и широкого использования современной математики. В результате этого сетевые модели и методы для моделирования и анализа организационно-экономических систем и технологических процессов оказались, во-первых, принципиально новыми, во-вторых, универсальными, в-третьих, обеспечивающими возможность автоматизации процессов разработки и анализа функционирования сложных систем на базе использования компьютерной техники, в-четвертых, сравнительно простыми и достаточно эффективными в практическом применении. Поэтому не случайно то, что после возникновения и применения сетевых методов в США по данному направлению за 2-3 года было опубликовано свыше 400 сборников, научных статей, трудов и диссертаций.

В нашей стране первые публикации, посвященные содержанию, развитию и применению сетевых моделей и методов, появились в 1963 -1966 гг. [1-5,9]. В этих работах были изложены все основные идеи анализа сетей, разработанные к тому времени. После указанных публикаций, как в нашей стране, так и за рубежом появилось множество работ, посвященных изложению, развитию, применению сетевого метода моделирования и анализа сложных систем вплоть до сетей Петри [6,10,12-17]. Более того, новые методы, основанные на теории графов и аналитических методах их исследования, позволили поставить и решать еще множество актуальных задач в строительстве, таких как оптимизация организации технологических процессов поточного строительства, рационализации затрат времени и денежных средств на создание физической системы звеньев, связывающих множество пунктов в единую сеть, расчет максимального потока в сети и т.д.

С появлением в 90-е годы образовательных стандартов в системе высшего образования и включением в них курса «Математические методы исследования операций» сетевые модели и методы были включены в состав данного курса. Наряду с этим в ряде отдельных дисциплин, касающихся организации производства, сетевое планирование и сетевые графики входят в эти дисциплины отдельной темой. С переходом на многоступенчатую подготовку

специалистов, включающую и магистерскую подготовку в вузах, математическое моделирование технологических процессов в строительстве вообще, и сетевое моделирование их, в частности, становится неотъемлемой частью магистерской подготовки и освоения основ курса «Методы решения научно-технических задач в строительстве».

В основу сетевого моделирования положен системный, а точнее системно-структурный подход. Надо признать, что единого определения понятия «система» в науке до сих пор нет. Существуют лишь различные формулировки это понятия, отражающие позиции отдельных авторов относительно изучаемых ими систем и их свойств. Однако в основном под системой понимается нечто единое целое, состоящее из отдельных элементов, которые находятся в определенных отношениях между собой, а потому взаимосвязаны и оказывают друг на друга взаимовлияние. Связи и отношения между элементами системы образуют ее структуру, которая определяет главные свойства системы. Значит, изменение только структуры системы при тех же свойствах элементов приводит к другой системе с иными свойствами. Величина системы определяется количеством элементов системы, а сложность ее - количеством взаимосвязей между элементами и числом состояний, в которых они могут находиться, и их неоднородностью, а также возможностями исследователя учитывать и обобщать их свойства, чтобы характеризовать свойства системы в целом. Системно-структурный подход сетевого моделирования строительных процессов предполагает моделирование структуры, когда моделируемый технологический процесс или иной объект представляют сложную систему.

На основе учета особенностей отображаемой структуры объекта сетевое моделирование позволило разработать эффективные методы анализа и оптимизации сложных строительных процессов. Более того, как выше было отмечено, оно позволяет реализовать на практике системный подход, применять математические методы и информационные технологии при исследовании динамики сложных процессов. В результате этого не только удастся синхронизировать работу отдельных исполнителей и частей системы, но и резко повысить эффективность взаимодействия всего комплекса, как единой системы, функционирование которой нацелено на достижение общей конечной цели.

Основными областями сетевого моделирования являются: проектирование и осуществление строительных процессов при возведении зданий и сооружений, различных коммуникаций и комплексов объектов.

Четкого определения сетевого моделирования до сих пор еще нет. Однако в литературе и практике прочно закрепилось определение основного раздела сетевого моделирования: сетевого планирования, причем не как оторванного метода, а как цельной системы сетевого планирования и управления (СПУ), предназначенной для решения определенного круга научно-

технических задач в строительстве. Одним из определений системы СПУ является следующее [1].

Система сетевого планирования и управления является комплексом расчетных методов, организационных мероприятий и контрольных приемов. Она обеспечивает средства динамического и скоординированного представления и анализа сложных производственных и непроизводственных программ. Конечным продуктом действия системы являются: выявление и мобилизация резервов времени и материальных ресурсов, скрытых процессов, осуществление управления программой по принципу «ведущего звена» с прогнозированием и предупреждением возможных сбоев в ходе выполнения программы, улучшение технических показателей планируемой программы; повышение эффективности управления в целом при четком распределении ответственности между руководителями разного уровня и ответственными исполнителями.

Следует заметить, что методы сетевого моделирования не обеспечивают, вообще говоря, математически оптимального решения задач, но они позволяют значительно улучшить показатели работы за счет дополнительного изыскания резервов времени и материальных ресурсов.

Важной особенностью сетевого моделирования является то, что оно широко использует вычислительную технику. Разработаны методы расчета сетей как вручную, так и на ЭВМ. При использовании ЭВМ, сетевое моделирование представляет автоматизированную систему планирования и управления сложными производственными программами. Автоматизация процесса планирования и управления состоит из двух этапов. На первом этапе производится построение сетевого плана и его анализ, т. е. выявляются резервы времени и материальных ресурсов, а также минимизируется время выполнения проекта при заданном количестве наличных ресурсов или минимизируется стоимость при заданном времени. Второй этап характеризуется оперативным управлением ходом выполнения проекта, т. е. производится систематическое сравнение реального хода выполнения проекта во времени с заранее построенным сетевым планом и оперативное устранение помех, возникающих в ходе реализации проекта.

Сетевое моделирование обеспечивает следующие возможности:

1. Графически отобразить состав и структуру проектируемого строительного процесса, выявить важнейшие операции моделируемого процесса и установить взаимосвязи между ними, а также распределить функции и обязанности между исполнителями технологических операций.
2. Использовать опыт и знания компетентных специалистов-практиков, как при разработке сложных проектов, так и в процессе их реализации.
3. Анализировать динамику реализации проекта, вскрывать узкие места и находить резервы средств и времени для повышения эффективности проекта.

4. Концентрировать контроль за ходом реализации проекта на наиболее важных направлениях, не отвлекаясь на контроль второстепенных звеньев.
5. Обеспечивать наглядное, удобное для восприятия, отображение структуры как процесса в целом, так и отдельных его частей.
6. Получать количественные оценки хода реализации проекта, прогнозов ожидаемых состояний в будущем, с учетом фактического хода процесса целенаправленно корректировать ранее принятые решения и предвидеть возможные последствия таких корректировок.
7. Упростить систему отчетности, а при смене руководства проектом обеспечить непрерывность управления посредством быстрой адаптации и включение в работу новых лиц.

Основное внимание в следующих главах пособия уделяется математическим аспектам и практическим вычислительным методам сетевого моделирования.

### ***1.1.2. Задачи исследования строительных процессов на основе сетевых моделей***

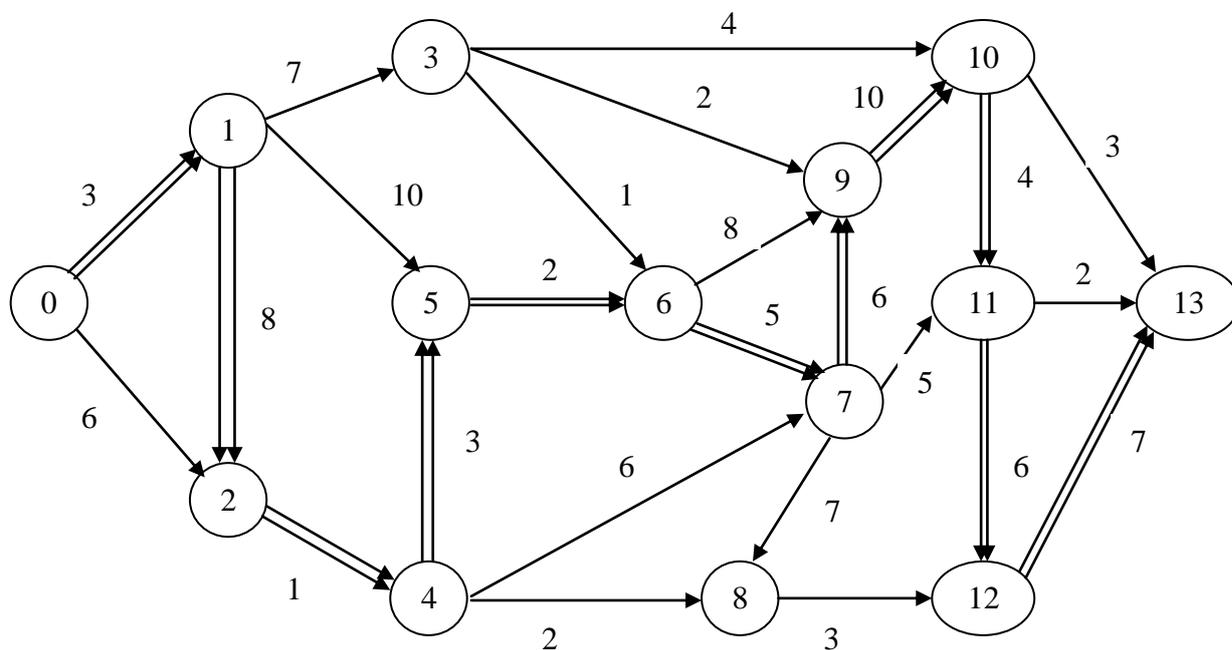
*А) Проектирование строительных процессов на основе модели сетевого графика.*

Одним наиболее известных методов планирования и управления в строительстве, использующих основы сетевого моделирования, является сетевое планирование и управление на основе сетевого графика.

При разработке и реализации больших систем или сложных комплексов (проектов) возникает необходимость производства множества операций или работ. Так, при создании какого-либо объекта строительства мы можем выделить взаимосвязанную последовательность ряда процессов: рытье котлована, возведение фундаментов и стен, монтажные, отделочные и другие работы. По технологии проведение одних работ во времени должно выполняться последовательно, а другие могут производиться одновременно, параллельно. Для планирования подобных процессов используются ленточные диаграммы Ганта или план - графики. Сетевой метод по характеру близок к широко применяемому на практике методу сквозного технологического планирования при запуске нового изделия в производство. Он учитывает не только последовательность проведения операций, но и их параллельность, что не учитывается в диаграммах Ганта.

Таким образом, весь комплекс работ или процессов при сетевом планировании представляется графически уже не в виде отдельных ленточных диаграмм, а объединен в один график, который представляет сеть логически взаимосвязанных между собой работ или процессов данного комплекса. Сеть

состоит из узлов-событий, соединенных стрелками. Стрелки отображают отдельные операции или работы, узлы-события – моменты начала или завершения работ. Иными словами, при сетевом планировании составляется стрелочная диаграмма - логическая сеть проекта, как показано на рис. 1



*Рис. 1. Сеть проекта или сетевой график  
Цифры в кружках – номера событий;  
цифры на стрелках – время выполнения соответствующей работы;  
стрелки – отдельные работы данного комплекса*

Важным преимуществом сетевой модели является высокая наглядность, интерпретируемость ее и простота анализа. Глядя на диаграмму, мы замечаем взаимозависимости, а, следовательно, и взаимное влияние между выполняемыми работами. Так, из диаграммы, приведенной на рис. 1, мы видим, что работа 6-9 не может быть начата до тех пор, пока не будут закончены работы 3-6 и 5-6. Работа 5-6, в свою очередь, не может быть начата, пока не будут закончены работы 1-5 и 4-5, а работа 4-5 не начнется, пока не будет выполнена работа 2-4. Таким образом, хотя у нас и нет прямой связи между событиями (2) и (6), однако от свершения события (2) существенно зависит свершение события (6).

На каждой стрелке проставляется время, необходимое для выполнения определенной операции или работы. Время свершения события считается равным нулю, т.е. события свершаются мгновенно. Ясно, что событие может свершиться только в том случае, когда все подходящие к нему стрелки-работы выполнены, т.е. стрелки-работы, исходящие из данного события, не могут быть начаты до тех пор, пока все стрелки-работы, входящие в событие, не бу-

дуг закончены. Таким образом, в логической сети плана разработки реализуется одна единственная логическая операция «И» (и то, и другое). *А сама логическая сеть и есть сетевая модель комплекса операций или работ.* Свершение события по времени совпадает с работой, оканчивающейся последней из всех работ, входящих в данное событие. Время свершения события в то же время есть время начала работ, исходящих из него.

Анализируя далее сеть, мы всегда можем указать последовательность, которая определит время выполнения всего комплекса работ. Такая последовательность всегда существует и называется критической последовательностью, или *критическим путем*. На рисунке 1 критический путь включает следующую последовательность работ: 0-1, 1-2, 2-4, 4-5, 5-6, 6-7, 7-9, 9-10, 10-11, 11-12, 12-13. Он проходит последовательно через события (0)–(1)–(2)–(4)–(5)–(6)–(7)–(9)–(10)–(11)–(12)–(13). Это максимальный путь из всех путей, которые ведут из события с номером 0 в событие с номером 13. Следовательно, все другие пути короче критического, а поэтому имеют запас или резерв времени по сравнению с критическим. Отсюда справедливо следующее заключение: для сокращения срока разработки проекта или выполнения комплекса работ, прежде всего, необходимо сократить критический путь, ибо он лимитирует срок всей разработки проекта. Сокращение сроков выполнения работ, не лежащих на критическом пути, не может сократить срок завершения разработки. Так, в нашем примере сокращение срока выполнения работы 6-9 никак не влияет на сокращение завершения комплекса работ в целом, т.е. не сократит срок свершения события (13), в то время как сокращение времени выполнения работы 6-7 немедленно сокращает общий срок завершения всех работ. На практике часто для сокращения общего срока выполнения комплекса работ сокращают пропорционально время выполнения каждой работы. Это почти всегда связано с затратой некоторых ресурсов: трудовых, денежных и т.д. Как показал сетевой анализ, нет необходимости привлекать эти ресурсы, тем более, что они не способствуют сокращению общего срока разработки.

Наличие критического пути существенно упрощает процесс управления выполнением плана. Руководству проекта нет необходимости вести повседневный контроль над всеми работами. Достаточно строго контролировать работы, лежащие на критическом пути, чтобы не было срывов. Время завершения работ, не лежащих на критическом пути, может варьировать в пределах своего резерва или запаса.

Двигаясь далее в этом направлении, легко понять, что для сокращения критического пути необходимы некоторые ресурсы, прежде всего ресурсы труда и денежные средства. Однако часть этих ресурсов можно получить за счет внутренних резервов, за счет перераспределения ресурсов между отдельными работами внутри комплекса. Так, с работ, не лежащих на критическом

пути и имеющих некоторый запас времени, можно снять часть ресурсов в пользу критического пути, за счет уменьшения этого запаса. Поступая, таким образом, мы за счет рационального распределения внутренних ресурсов получим наименьший критический путь. Иными словами, мы при заданном объеме ресурсов выполним весь комплекс работ за минимальное время. Можно поступить и наоборот: при заданном сроке выполнить весь комплекс работ с минимальными затратами ресурсов. Однако минимизация критического пути или затрат в конкретных условиях требует применения методов оптимальных решений: линейного и динамического программирования и других [3,4,7-12] и ПЭВМ.

Сетевое моделирование в своем развитии прошло ряд этапов.

Этап первый: рассматривается сетевой план только по временным характеристикам, так называемая система PERT–ТАИМ. На этом этапе ищутся возможности максимального сокращения критического пути без анализа затрат материальных и денежных ресурсов. Этот этап в строительстве уже широко применяется. Более того, многие строительные организации уже перешли на второй более совершенный этап планирования.

Этап второй: осуществляется анализ времени и затрат материальных ресурсов, необходимых для выполнения проекта, одновременно. Этап оптимизации материальных ресурсов на основе сетевого графика.

Третий этап: осуществляется оптимизация времени и денежных затрат на выполнение проекта. При этом решается либо задача минимизации стоимости при заданном сроке выполнения всех работ комплекса, либо задача минимизации срока выполнения всех работ комплекса при фиксированной стоимости реализации проекта.

И, наконец, четвертый этап: строительная организация, как правило, выполняет одновременно не один проект, а несколько взаимосвязанных комплексов работ (проектный институт разрабатывает несколько проектов, строительный трест строит сразу несколько объектов и т.д.). Но все программы, все проекты расходуют ресурсы при своем выполнении из единого ограниченного источника. На этом этапе возникает необходимость проводить одновременный анализ нескольких сетевых моделей. Оптимальное использование ограниченных ресурсов по сетевой модели одновременно в нескольких сетевых планах в строительстве называется системой сетевого планирования комплекса объектов.

#### *Б) Задача о кратчайшей связывающей сети*

При проектировании линейных сооружений, связывающих некоторое число пунктов в единую сеть: сеть шоссейных дорог, линий электропередач или сеть водоснабжения и водоотведения и т.п., возникают многие сложные многовариантные технико-экономические задачи, связанные с оптимизацией

затрат времени и финансовых средств. Как величина капитальных вложений на строительство линейных сооружений, так и затраты при их эксплуатации в значительной мере зависят от суммарной протяжённости всей сети. Чем меньше протяжённость сети между определённым числом узлов, тем меньше капитальных вложений при прочих равных условиях потребуется на строительство линейных сооружений сети. Сети меньшей протяжённости при их эксплуатации потребуют меньше текущих затрат.

Поэтому первая важная задача проектировщика связать заданные пункты, например, водоснабжения и водоотведения сетью наименьшей протяжённости в конкретных условиях. Множество возможных вариантов сетей, которые связывают всю совокупность пунктов или узлов, подлежащих соединению в единую сеть водоснабжения или водоотведения, обусловлено тем обстоятельством, что заданный объём жидкости между различными узлами сети можно передавать различными путями. А определение варианта кратчайшей длины сети, связывающей все узлы в единую сеть, не очевидно и поэтому представляет предмет оптимизации.

*В) Задача о максимальном потоке или о пропускной способности сети.*

Для любой коммуникационной сети пропускная способность соответствующего потока каждого участка, звена и сети в целом ограничена. А в системах водоснабжения и особенно водоотведения одной из особенностей организации и планирования производственных процессов является неравномерность поступления потоков жидкости в разные периоды эксплуатации соответствующих сетей.

В связи с этим расчёт мощности (пропускной способности) труб в проектах должен проводиться с учетом данных вплоть до почасового максимального и минимального поступления жидкости. Существующие методы расчёта оборудования основаны на определении некоего усредненного соотношения между нагрузкой и пропускной способностью.

Однако, это не исключает необходимости определения максимально возможной пропускной способности сети. Ведь в периоды резкого увеличения поступления, например, ливневых стоков канализационные сети могут не справиться с возлагаемыми на них задачами, что может привести к аварийным ситуациям вплоть до чрезвычайных. Поэтому важно знать максимальную пропускную способность сети и уж потом соотносить ожидаемые затраты и результаты. Подобного рода задачи возникают и при определении пропускной способности водопроводной сети городов. В периоды предпраздничных и праздничных дней, массовых отпусков и т. п. потребление воды резко возрастает. Возрастает и нагрузка на сети.

Следовательно, опять возникает необходимость решения задачи по определению часов максимального и минимального водопотребления. Более то-

го, в ряде случаев водопроводная или водоотводящая сеть может быть представлена в форме последовательных участков, некоторые из которых составляют сами по себе относительно сложную схему. Поэтому даже в пределах этого сложного участка может быть найдено несколько вариантов схемы подключения. И, конечно, интересен вариант с максимальной пропускной способностью данного участка сети. Наконец, общая задача заключается в отыскании такой схемы соединения участков сети, при которой не нарушаются соединения внутри участка и в то же время обеспечивается возможность максимального прохождения жидкости через участок. Данная задача представляет соответствующую сетевую модель, а определение максимального потока в условиях сложных схем представляет собой весьма не простую задачу.

*Г) Задача оптимизации возведения группы строительных объектов при поточной организации строительства*

При разработке проектов организации строительства (ПОС) крупных промышленных комплексов возникает задача определения очередности строительства зданий комплекса, сгруппированных в поток. При различных вариантах очередности возведения объектов получаются разные сроки продолжительности строительства всей группы сетей из-за непропорциональности (асинхронности) трудоемкостей отдельных видов работ на разных объектах. Возникают простои двух видов.

1. Простои рабочих бригад в ожидании ими фронта работ, когда последующая бригада на предыдущем этапе в запланированной последовательности строительства объектов уже выполнила весь объем работ и готова приступить к выполнению своих работ на данной стадии строительства, но для нее еще не подготовлен фронт работ.

2. К простоям второго вида отнесем простои сетей (перерывы в их строительстве): фронт работ для последующей бригады здесь создан, но бригада занята еще выполнением работ на другом этапе.

На практике выбор эффективной очередности строительства зданий комплекса осуществляется путем сопоставления многочисленных вариантов. Но поскольку задача относится к комбинаторным задачам, то выбор варианта с наименьшей продолжительностью строительства таким методом затруднен, а при большом числе этапов строительства поиск оптимального варианта становится практически невозможным из-за слишком большого числа возможных вариантов и высокой трудоемкости оценки их эффективности. Уже при семи этапах строительства количество возможных вариантов очередности их возведения составляет 5040, а при десяти – более трех миллионов (3628800).

Таким образом, перед разработчиками ПОС возникает дилемма: либо выбирать очередность строительства сетей традиционными методами путем рассмотрения небольшого числа вариантов, что ведет к увеличению продол-

жительности строительства комплекса по сравнению с оптимальной и к отвлечению на это время капвложений и других ресурсов, либо находить методы поиска оптимальной последовательности без перебора всего множества возможных вариантов. Такие задачи также могут быть представлены сетевой моделью.

В настоящее время разработаны точные методы, основанные на идеях динамического программирования и методах ветвей и границ, обеспечивающие оптимальное решение задачи, но они требуют от разработчика ПОС высокой квалификации в области методов оптимизации и связаны с высокой трудоемкостью расчетов и с применением компьютеров.

Поэтому наряду с точными методами развивались менее трудоемкие, но весьма эффективные приближенные методы решения задачи [10]. В главе 4 рассмотрены методы сетевого моделирования и решения подобных задач.

## 1.2. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ СЕТЕВОГО ПЛАНИРОВАНИЯ (СЕТЕВОГО ГРАФИКА)

Прежде чем излагать содержание изучаемого предмета, всегда необходимо ознакомиться с его языком, той терминологией, которой приходится оперировать в ходе изложения. Сетевое планирование выработало свой язык, свои понятия, свою терминологию. Цель данного раздела и состоит в ознакомлении читателя с основными понятиями, которыми оперирует сетевое планирование. Некоторые понятия негласно мы уже ввели при изложении сути сетевого планирования. Например, такие понятия, как критический путь, резерв времени, событие, работа и др. Здесь мы попытаемся определить их более четко и ввести ряд других весьма важных понятий сетевого планирования.

Основным математическим аппаратом сетевого планирования является теория графов. Более того, сетевое моделирование можно рассматривать как частный случай теории графов, как ее экономическое приложение. Поэтому введем несколько понятий из теории графов [3].

*Графом* будем называть множество точек (вершин)  $\{P_0, P_1, \dots, P_n\}$  и множество ориентированных дуг  $\{(P_i, P_j)\}$ , соединяющих некоторые пары этих точек; при этом дуга  $(P_i, P_j)$  имеет начало в  $P_i$ , и конец в  $P_j$ . На чертеже дугу  $(P_i, P_j)$  обозначают в виде направленного отрезка. Вершины графа часто изображают кружком и нумеруют их. Номер вершины ставят либо внутри кружка (рис. 1), либо обозначают знаком « $P$ » (пункт, точка) с индексом внизу, обозначающим номер вершины.

*Дуги*, исходящие (берущие начало) из точки  $P_i$ , назовем выходящими из  $P_i$ ; а дуги, конец которых в  $P_j$ , – входящими в  $P_j$ .

Граф, в котором существует всего одна точка  $P_0$ , не имеющая входящих дуг, и лишь одна точка  $P_n$ , не имеющая выходящих дуг, и каждой дуге которого приписано одно или несколько чисел, будем называть *сетью*. Примером сети является диаграмма, изображенная на рис. 1.

Последовательность дуг  $(P_{i1}, P_{j2}), (P_{i2}, P_{j3}), (P_{ik-1}, P_{jk}), (P_{ik}, P_{jk+1})$ , в которой конец каждой предыдущей дуги совпадает с началом последующей, называется *путем*  $(P_{j1}, P_{j2}, \dots, P_{jk+1})$  из вершины  $P_{j1}$  в вершину  $P_{jk+1}$ .

Числа, приписанные дугам (речь идет о случае, когда каждой дуге приписано всего одно число), будем называть их *длинами*. *Длиной пути* будем называть сумму длин дуг, составляющих путь.

Все эти понятия теории графов легко можно увидеть на диаграмме, изображенной на рис. 1, и тем самым проследить тесную связь теории графов с сетевым планированием. Аналогичную связь теории графов можно установить и с другими задачами сетевого моделирования.

Остановимся более подробно на связи теории графов и сетевого планирования. Сетевая модель проекта является обычным ориентированным графом, или как ее называют в сетевом планировании – сетевым графиком.

Несмотря на то, что теория графов по существу является математическим аппаратом сетевого планирования, само сетевое планирование развивалось из практики планирования и управления сложными проектами, обособленно от теории графов, а поэтому выработало свою терминологию, несколько отличную от терминологии теории графов. Рассмотрим основные понятия, выработанные и употребляемые в сетевом планировании:

Первым и основополагающим понятием в сетевом планировании является понятие сетевого графика.

*Сетевой график* (сетевая модель) есть наглядное изображение проекта (комплекса работ) в виде графа, отображающее технологическую взаимосвязь между отдельными работами. Другими словами, сетевой график представляет собой графическую модель проекта. Чтобы построить такую модель, необходима информация о проекте, содержащая перечень всех работ проекта, последовательность их выполнения и продолжительность (время, необходимое для выполнения) каждой работы.

Вершины графа, представляющего сетевой график, называются *событиями*.

*Событие* – это результат работы. Событие в сетевом графике изображается кружком, внутри которого ставится номер данного события. Событие представляет собой промежуточный или окончательный результат одной или нескольких работ. Так, событие (1) на рис. 1 представляет результат одной работы 0-1, а событие (9) результат трех работ 3-9, 6-9, 7-9. Говорят, что собы-

тие свершилось, если выполнены все работы, входящие в него. Как только событие свершилось, сразу же могут быть начаты работы, выходящие из данного события. Пока событие не свершилось, работы, выходящие из него, не могут быть начаты. Если в событие входит несколько работ, то мы уже указали, что оно свершится лишь тогда, когда будут выполнены все работы, входящие в него. Следовательно, событие свершится, как только будет выполнена самая последняя по времени работа, входящая в данное событие. В самом деле, раз выполнена последняя работа, то это значит, что все остальные работы выполнены раньше, а следовательно, выполнением последней работы и завершается выполнение всех работ, входящих в событие.

Ясно, что нередки случаи, когда событие есть результат нескольких работ и в то же время – момент начала многих работ. Этот случай имеет место у события (6) на рис. 1. В него входят 2 работы: 3-6 и 5-6 и из него выходят 2 работы 6-9 и 6-7.

Итак, мы определили событие как результат завершения одной или нескольких работ, как момент этого завершения. Поэтому само событие не имеет продолжительности во времени. Мы полагаем, что события сетевого графика свершаются мгновенно.

Сетевой график, как правило, имеет единственное начальное событие, определяющее начало проекта, т.е. момент, с которого начинают выполняться работы данного проекта. На графике начальное событие обычно изображается самым левым. На рис. 1 начальным событием является (0). Каждый сетевой график имеет также единственное конечное событие, отражающее момент готовности проекта, т.е. момент выполнения всего комплекса работ, входящих в данный проект. Если свершилось конечное событие, то это значит, что выполнены все работы данного проекта. Характерным признаком начального события является то, что в него не входит ни одной работы, а все работы, с ним связанные, лишь исходят из него. Относительно конечного события приходится утверждать противное, т. е. все работы, с ним связанные, входят в него, а из него не выходит ни одна работа. На рис. 1 конечным событием является событие (13). Конечное событие на сетевом графике является, как правило, самым правым. Таким образом, течение времени идет от начального события к конечному, т.е. слева направо, что на графике и отражено направлением стрелки.

*Работа.* Термин работа мы уже невольно ввели в наш язык при изложении понятия «событие». Иначе и нельзя. Событие и работа на графике тесно связаны. Под работой в сетевом графике понимается, в отличие от события, не момент, а процесс. В большинстве случаев под работой понимается некоторая реальная деятельность, направленная на выполнение проекта. На сетевом графике работы изображаются стрелками. (На графе работы соответствуют дугам

графа). Один конец стрелки лежит в одном событии, а другой – в другом. Острые стрелки обозначает конец работы, а противоположный конец стрелки – ее начало. Всякая работа исходит из одного события (начинается после свершения данного события) и входит в другое. Таким образом, работа связывает два события. Иногда говорят, что работа переводит одно событие в другое (предшествующее в последующее). Сама же работа на сетевом графике определяется предшествующим и последующим событием. Таким образом, предшествующее событие определяет момент начала, а последующее – момент завершения данной работы. Работы на графике подразделяются на реальные работы, работы – ожидание и фиктивные работы.

*Реальная работа* – это работа, потребляющая время и ресурсы (возведение стен, рытье под фундаменты и т. д.). Реальная работа на сетевом графике изображается сплошной стрелкой и над ней указано время, необходимое для ее выполнения, т.е. ее *продолжительность*. (Иногда кроме продолжительности на работах-стрелках проставляются интенсивности использования ресурсов и стоимости выполнения).

*Работа-ожидание* – это работа, которая потребляет только время и не потребляет никаких других ресурсов. Такие работы на сетевом графике изображаются пунктирными стрелками с указанием под ними времени ожидания. Следует указать, что ожидание возникает тогда, когда после свершения события работа, следующая за этим событием, не может быть сразу начата. Необходимо выдержать некоторое время, прежде чем ее начать. Примером этого в строительстве может служить выдержка бетона, т.е. время ожидания, пока бетон успеет застыть.

*Фиктивная работа* – это работа, не потребляющая ни времени, ни ресурсов. Фиктивная работа определяет лишь зависимость возможности свершения одного события от факта свершения другого. Фиктивные работы на сетевом графике изображаются пунктирными стрелками. Появление фиктивных работ будет более понятно, когда мы рассмотрим ниже правила построения сетей.

Введя понятие «события» и «работы», можно сказать, что сетевой график есть множество событий, связанное множеством работ. Он представляет собой наглядную модель реализуемого проекта. При планировании по ленточным графикам (традиционным способом) невозможно отразить все зависимости между работами. На практике нередко были случаи, когда в ходе выполнения проекта вдруг обнаруживалось, что какую-то плановую работу нельзя начать, так как не закончены некоторые другие работы. Сетевой график это исключает.

Рассмотренных выше понятий «событие» и «работа» вполне достаточно, чтобы по изложенным ниже правилам построить сетевой график. После того

как сетевой график построен, для каждой работы определяется ее продолжительность. *Продолжительность работы* (или длина дуги графа) *есть время, необходимое для ее выполнения*. На графике продолжительности указываются над (или под) соответствующими стрелками-работами.

Главным моментом при построении сетевого графика является определение технологической последовательности выполнения работ данного проекта. Другими словами, важно наметить путь последовательного выполнения работ.

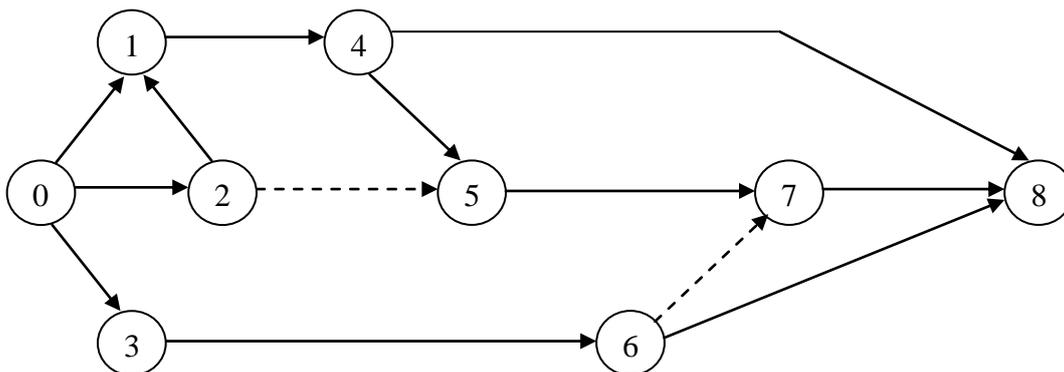


Рис. 2. Пример сетевого графика

*Путем* в сетевом графике называется непрерывная последовательность работ, связывающая любые два события. Путь может быть записан последовательностью работ или событий. На рис. 2 от события 0 до события 7 идут 4 пути: 0-1, 1-4, 4-5, 5-7; 0-2, 2-5, 5-7; 0-2, 2-1, 1-4, 4-5, 5-7 и 0-3; 3-6; 6-7. Это мы записали все пути от события 0 до события 7 по работам. Эти же пути можно записать и как последовательность событий: 0-1-4-5-7; 0-2-5-7; 0-2-1-4-5-7 и 0-3-6-7. Практически наибольшую ценность представляют пути, идущие из начального события в конечное. Это своего рода звенья, цепочки работ, которые проходят от начала выполнения проекта до его завершения. Задача здесь заключается в выявлении ведущего звена, лимитирующего и определяющего срок завершения проекта. Чтобы выявить это ведущее звено, необходимо знать длину каждого пути, связывающего начальное событие с конечным. *Под длиной пути* из события  $P_i$  в событие  $P_j$  понимается сумма продолжительностей работ, составляющих данный путь, т.е. понимается продолжительность выполнения всей последовательности работ, составляющих этот путь.

Нетрудно видеть, что из начального события в конечное идет, как правило, не один, а несколько путей и число их конечно. Каждый из этих путей имеет свою длину. Длины разных путей, вообще говоря, различны. *Путь наи-*

*большей длины называется критическим.* Значение критического пути в сетевом планировании очень велико. Именно этот путь определяет продолжительность выполнения всего комплекса работ, т.е. проекта. В самом деле, раз критический путь самый длинный из всех путей, идущих из начального события в конечном, то это значит, что если выполнены все работы, лежащие на критическом пути, то, следовательно, должны быть выполнены и все работы, лежащие на всех других путях. Другими словами, последнее событие свершится лишь после того, как будут выполнены все работы, входящие в него. Но самой последней будет выполнена работа критического пути. Следовательно, критический путь и лимитирует срок завершения выполнения всего комплекса работ. Он является самым напряженным путем при реализации проекта. Все остальные пути считаются ненапряженными и поэтому не влияют на срок завершения проекта. Все остальные пути короче критического и поэтому они имеют запас времени по сравнению с критическим. Таким образом, работы ненапряженных путей имеют резервы времени. Итак, следующим важным понятием в сетевом планировании является понятие «резерва времени» и другие понятия, с ним связанные (сроки начала и окончания работ и свершения событий).

Выше мы уже указывали, что событие считается свершившимся, как только будут выполнены все работы, входящие в него, или, что то же самое, событие считается свершившимся, как только будет выполнена последняя по времени из работ, входящих в данное событие. Отсюда появляется понятие «срока свершения события».

*Под сроком свершения события* понимается дата свершения события. Различают два срока свершения события. *Самый ранний срок* – это дата, возможного свершения события в результате выполнения всех предшествующих ему работ. Он определяется путем наибольшей длины от начального события до данного. *Самый поздний срок* – это дата возможного свершения события, при котором общий срок выполнения всего комплекса работ не нарушается. Другими словами, самый поздний срок определяет самую позднюю дату свершения данного события, которая не изменит самый ранний срок свершения конечного события.

Таким образом, срок свершения события определяет начало работ, входящих из него. *Самый ранний срок начала работы* – это длина наибольшего пути от начального события до момента начала данной работы. Очевидно, что самый ранний срок начала работы совпадает с самым ранним сроком свершения события, из которого выходит данная работа. Всякая работа на сетевом графике имеет конечную продолжительность. Следовательно, если к моменту начала работы прибавить ее продолжительность, т.е. время, необходимое для ее выполнения, то получим *срок окончания работы*. *Самый ранний срок окончания работы* – это длина наибольшего пути от начального события до мо-

мента окончания данной работы. Кроме самого раннего начала и окончания работы в сетевом планировании вводится понятие самого позднего начала и окончания. Так же как и для событий, *самым поздним сроком* начала и окончания работы является тот, который не задерживает раннего срока свершения конечного события.

Определим теперь *резервы времени*. Для каждой работы сетевого графика различают два резерва времени: полный (или общий) и частный.

*Полный (общий) резерв времени* – это количество времени, на которое можно перенести начало работы или увеличить ее продолжительность без изменения срока завершения проекта, т.е. без изменения срока свершения конечного события. Он определяется разностью позднего и раннего сроков начала или окончания работы.

*Частный резерв времени* – это количество времени, на которое можно перенести начало работы или увеличить ее продолжительность, не изменяя раннего срока свершения события, в которое данная работа входит, т.е. не изменяя раннего срока начала последующих работ. Определяется он разностью раннего срока свершения события, в которое входит данная работа, и раннего срока окончания данной работы.

**Примечание.** Вообще говоря, кроме полного и частного резерва у работ может существовать так называемый независимый (или свободный) резерв времени. *Независимый резерв времени* работы – это максимально допустимое увеличение продолжительности этой работы (или опоздание момента ее начала) при условии, что все работы, входящие в событие  $P_i$ , закончатся в самое позднее допустимое время, а все работы, исходящие из  $P_j$ , начнутся в самое раннее допустимое время. Следовательно, использование независимого резерва времени у работы не может повлиять на резервы других работ.

*Сроки начала и окончания работ, сроки свершения событий, резервы времени и критический путь являются основными расчетными характеристиками сетевого графика*, особенно на первом этапе сетевого планирования. Вторая глава настоящего пособия почти целиком посвящена методам расчета указанных характеристик.

Кроме рассмотренных выше понятий сетевого графика, полезно уяснить смысл еще таких понятий, как директивный срок, подкритические пути и цикл. Остановимся кратко на них.

Любой проект, любая производственная программа не являются изолированными, не существуют лишь как самоцель. Всегда есть другие программы, для которых, собственно говоря, и необходима данная программа. Другими словами, от срока завершения этой программы, этого проекта зависит выполнение других, по отношению к рассматриваемой, внешних программ. Дан-

ная программа органически входит в другие программы как часть, как отдельное звено в единой цепи. А поэтому, чтобы не срывались другие программы из-за несвоевременного выполнения данной, необходимо более или менее строго указать обязательный срок завершения данной программы. Этот срок, эта календарная дата и называется *директивным сроком* завершения проекта. Заметим, что директивные сроки устанавливаются людьми и поэтому вполне возможно, что эта дата установлена не объективно. Сетевое планирование предлагает одну из методик обоснования директивных сроков – проводит строгий анализ времени, необходимого для завершения данной программы и заданного директивного срока.

Выше мы указывали, что критический путь лимитирует время завершения проекта. Все остальные пути короче критического, а поэтому имеют резерв времени. Чтобы сократить время завершения проекта, надо, прежде всего, сократить критический путь. Это можно сделать за счет перераспределения ресурсов с ненапряженных путей на критический. За счет такого перераспределения критический путь сократится, а ненапряженный путь, с которого сняли часть ресурсов, увеличится. Нужно, чтобы он увеличился в пределах резерва времени. После сокращения данный путь может опять остаться критическим. Но нередко критический путь пройдет по другой последовательности работ, так как существуют пути, которые отличаются по длине незначительно от критического. *Все пути, которые отличаются от критического не более чем на некоторую заданную величину, называются подкритическими.* При расчете сетевого графика важно знать не только критический путь, но и все подкритические пути, хотя бы для того, что с них нельзя снимать ресурсы в пользу критического пути, хоть они и имеют запас времени. Наоборот, наряду с критическим путем, надо стремиться сократить и подкритические пути.

Заметим, что выполнение данной работы не зависит от выполнения самой себя, т.е. на графике не должно быть циклов. *Под циклом понимается путь, выходящий из некоторого события  $P_i$  и входящий, т.е. завершающийся в этом же событии.* На рис. 3 путь  $P_1, P_3, P_4, P_1$  исходит из события  $P_1$  и входит в событие  $P_1$ . Этот путь и есть цикл. Если сетевой график построен верно, то циклов не должно быть. Прежде чем проводить расчеты сетевого графика, необходимо выявить и устранить все циклы. Алгоритм выявления циклов рассмотрен во второй главе.

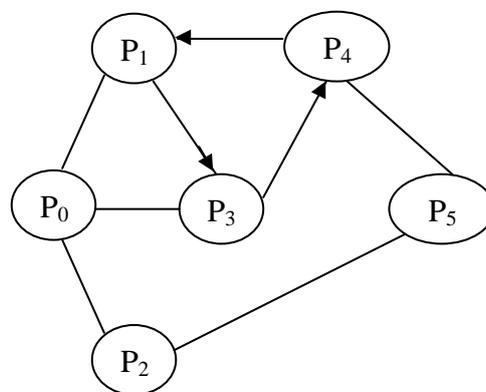


Рис. 3. Сетевой график с циклом

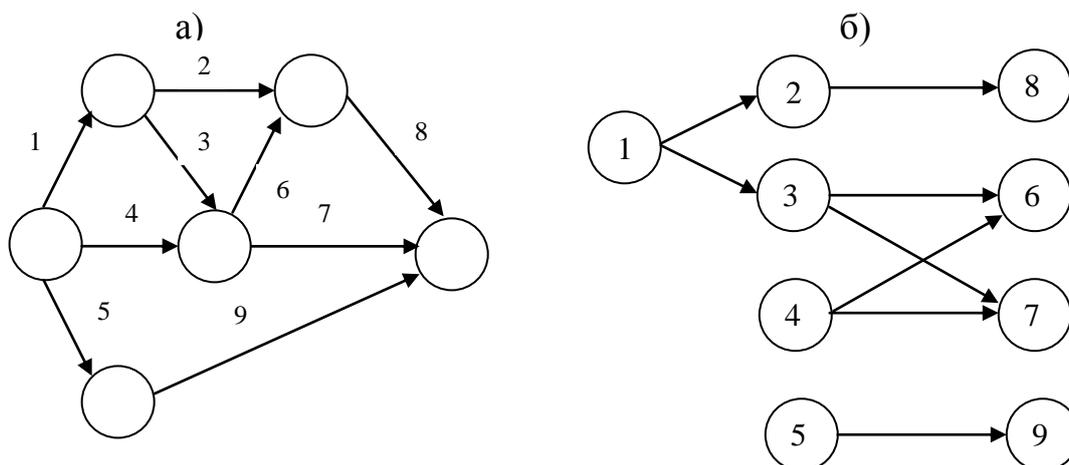


Рис. 4. Различные способы изображения сетевых графиков:  
 а) сетевой график, построенный в событиях  
 (цифры на стрелках указывают номера работ);  
 б) двойственный сетевой график  
 (цифры в кружках указывают номера работ,  
 а стрелки – связи между работами)

В заключение остановимся на понятии – «двойственный сетевой график». Сетевой график, в котором кружки-вершины графа изображают работы, а дуги, их соединяющие, указывают зависимость между этими работами (рис. 4), называется двойственным по отношению к предыдущему графику или графиком без событий. Преимуществом сетевого графика без событий является то, что для его построения не надо знать правил, излагаемых в разделе 1.3. Такой график легче увязать с шифрами работ действующих нормативных справочников. Ниже рассматриваются только сетевые графики с событиями.

### 1.3. ПРАВИЛА ПОСТРОЕНИЯ СЕТЕВЫХ ГРАФИКОВ

Прежде чем приступить к практическому построению сетей, необходимо хорошо усвоить правила их построения.

1. Направление стрелок в сетевом графике изображают слева направо.

2. В сетевом графике надо стремиться к минимальному количеству пересечений стрелок. Для этого надо возможно большее число работ изображать горизонтальными линиями.

3. Все события сетевого графика, кроме конечного, должны иметь хотя бы одну исходящую работу. Если в сети появился «тупик», то это значит, что 1) либо имеется ошибка в построении графика, 2) либо работы, входящие в такое событие, не влияют на сроки выполнения работ данного комплекса. В последнем случае событие надо соединить с конечным фиктивной работой.

В сетевом графике не должно быть событий, кроме начального, в кото-

рые не входит ни одна работа. Наличие подобных событий указывает на ошибку построения. Эти события следует соединить зависимостями с начальным событием. Правильно построенная сеть должна быть замкнутой.

4. При нумерации событий разным вершинам сети следует присваивать разные номера, ибо каждая конкретная работа должна быть определена на сетевом графике однозначно, только ей присущей парой событий.

5. В сетевом графике не должно быть циклов. Наличие циклов свидетельствует об ошибке построения сети.

6. Если ряд работ может начаться после частичного завершения предшествующей работы  $a$ , то работу  $a$  необходимо представить как сумму ее частей  $a_1, a_2, a_3$ , разделенных соответствующими событиями. Это позволяет отображать детализацию. Правило 6 приобретает особое значение при организации поточного выполнения работ, так как в этом случае необходимо разбить общий фронт работ на отдельные участки.

7. Если несколько работ выходят из одного события и заканчиваются в другом, то для их различия надо ввести фиктивные работы и дополнительные события. Это позволяет наглядно показать тот факт, что рассматриваемые работы протекают независимо друг от друга, но последующие работы могут быть начаты лишь после полного завершения всех данных работ.

8. Если в событие входят несколько работ, а из события выходит одна работа, зависящая лишь от части входящих работ, то, чтобы соблюсти предельную ясность при построении сетевого графика, вводят дополнительное событие и фиктивную связь. После чего все работы изображаются в технологической последовательности.

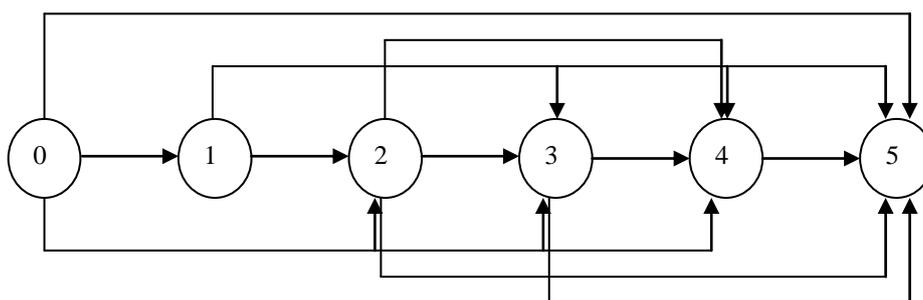
9. Нередко в отдельные события сетевого графика входят внешние по отношению к данной сети работы. Работы, исходящие из данного события, зависят от таких внешних работ. Внешними работами могут быть некоторые виды хозяйственной или правовой деятельности (например, разрешение на строительство, кредит и т.п.), но чаще всего – это поставка материалов, оборудования и т.д. Такие внешние связи в сетевом графике принято называть *поставками*. Поставки изображаются сплошными стрелками. Если продолжительность стрелки-поставки можно определить по времени, то эта работа привязывается своим началом к соответствующему событию и при расчете сетевого графика учитывается. Если же продолжительность стрелки-поставки определить трудно или вообще невозможно, то она при расчете графика не учитывается, а лишь фиксируется момент времени (на графике события), к которому она должна быть завершена, ибо в противном случае нельзя начинать работы, исходящие из данного события. Начало поставки не фиксируется. При введении поставок в сетевой график надо четко определить работы, которые

зависят от данной поставки, чтобы не зависимые от поставки работы на графике не связывались с поставкой.

10. В процессе составления сетевого графика с целью увязки отдельных узлов появляется много дополнительных работ и событий, которые просто необходимы в процессе составления графика. После того как все необходимые взаимосвязи работ проекта установлены, следует весь график внимательно проверить и исключить все зависимости, которые большой роли не играют. Это позволит упростить график. Желательно сократить число событий сети до минимума, не нарушая при этом перечисленные выше правила построения сетевого графика.

11. На основе одних и тех же данных для различного уровня руководства и исполнителей необходимы разные сетевые графики. В одних случаях (для более низких уровней руководства) необходимы более детализированные сетевые графики, а в других (для более высокого ранга руководства) - более укрупненные графики, значит, в одних случаях приходится разделять работу на составляющие ее процессы, в других - объединять ряд процессов в одну работу. Укрупнение необходимо и для иных целей.

Приведенные правила позволяют построить сеть проекта с учетом поставок и других условий, необходимых для его выполнения. Сетевой график должен представлять собой связный ориентированный граф без циклов, имеющий  $n + 1$  вершину. В сетевом графике с  $n + 1$  событием из начального события не может исходить, а в конечное событие – входить стрелок больше, чем  $n$ . В промежуточное событие входит стрелок не больше, чем номер события, а выходит – не больше, чем разность номеров конечного и данного событий (рис. 5).



*Рис. 5. График с наибольшим количеством связей между событиями*

Формы реализации правил построения сетевого графика:

**Правила 1, 2, 4, 10** очевидны.

**Правило 3** – рис. 6

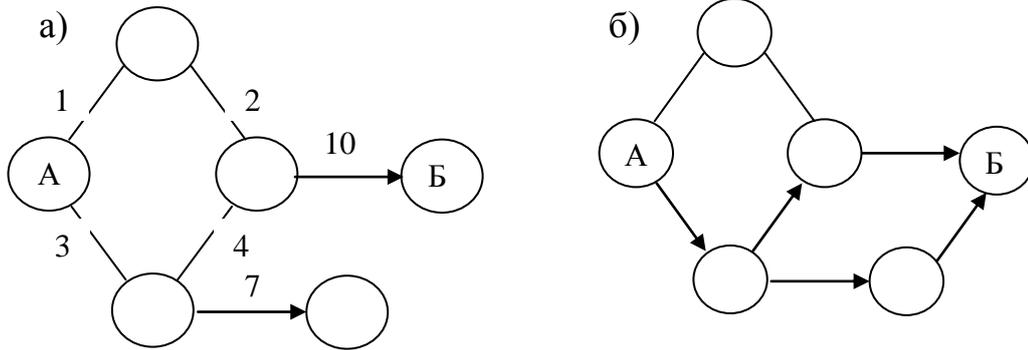


Рис. 6. Реализация правила 3: а) неправильно; б) правильно

**Правило 5**

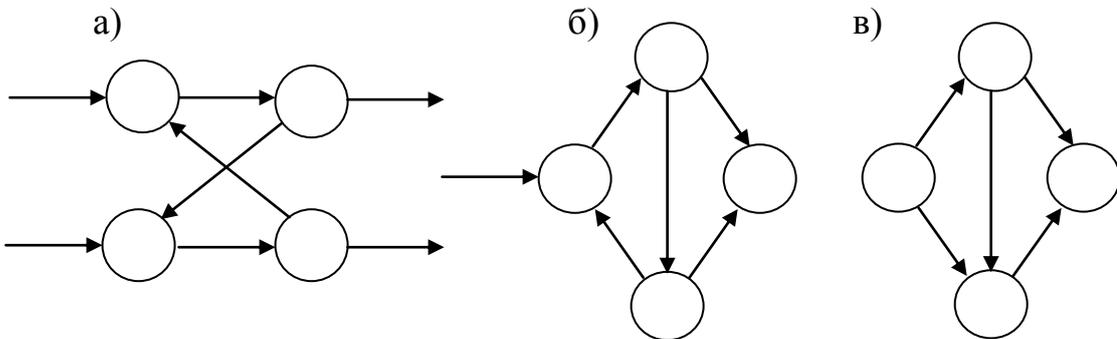


Рис. 7 Реализация правила 5: а) и б) – неправильно; в) правильно

**Правило 6**

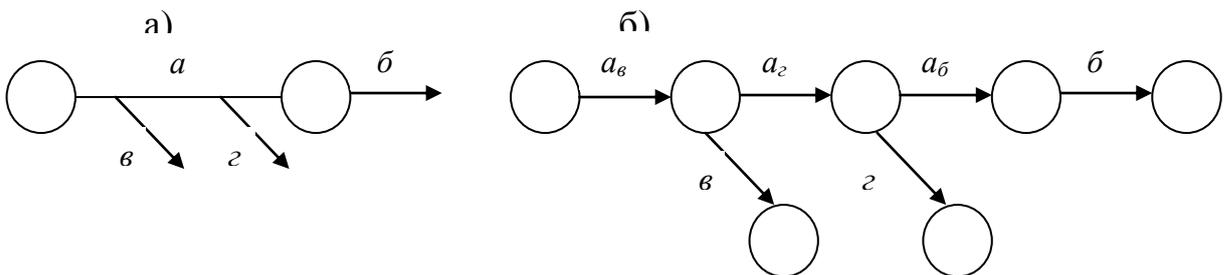
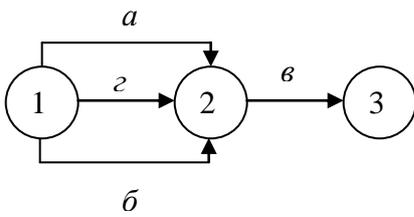
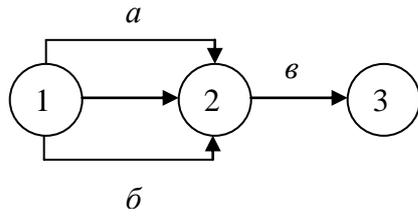


Рис. 8. Реализация правила 6: а) неправильно; б) правильно

**Правило 7**

а)



б)

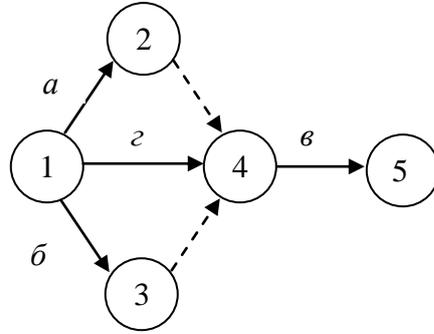
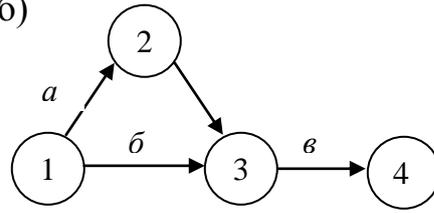
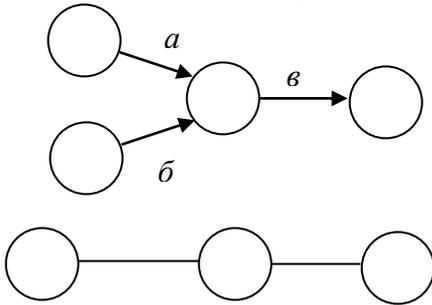


Рис. 9. Реализация правила 7:

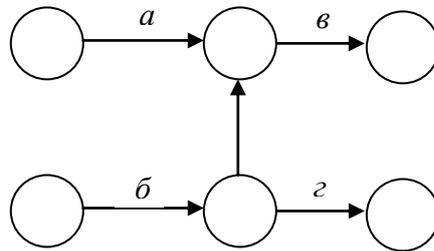
а) неправильное изображение параллельных работ; б) правильное

**Правило 8**

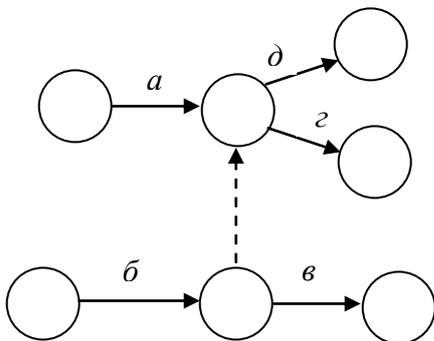
а)



б)



в)



г)

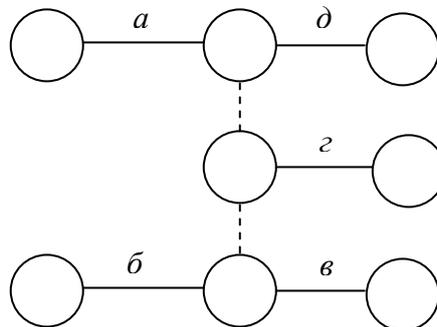


Рис. 10. Реализация правила 8:

а) отдельные условия; б) их правильное сочетание;  
в) неправильно; г) правильно

### Правило 9

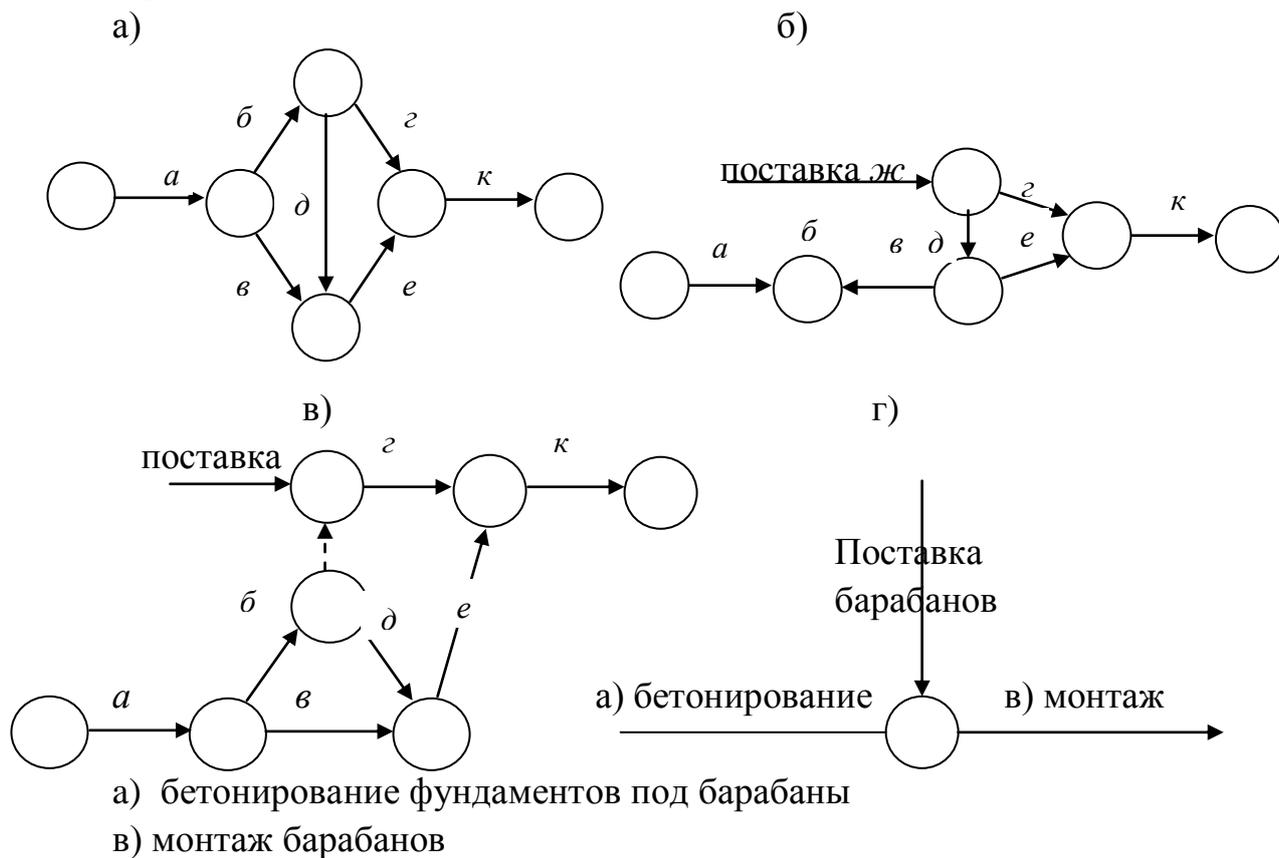


Рис. 11. Реализация правила 9:

а) график без учета поставки; б) график с учетом поставки; при этом б) — неправильно, в) правильно; г) поставка

### Правило 10

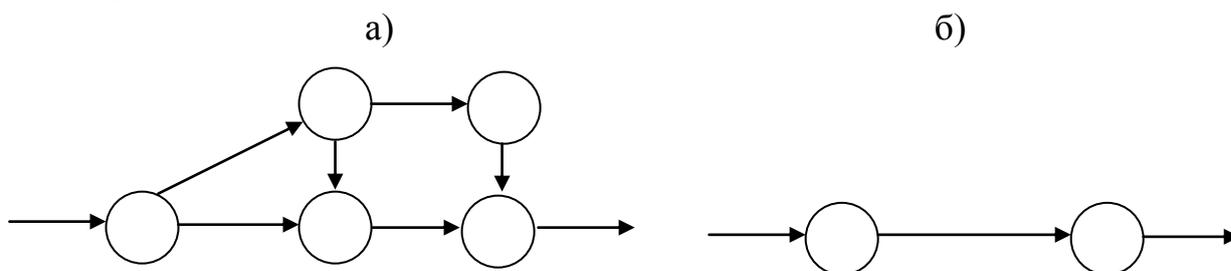


Рис. 12. Реализация правила 11:

а) детализированный сетевой график; б) укрупненный сетевой график

## 1.4. НУМЕРАЦИЯ СОБЫТИЙ

Рассмотрим методы нумерации событий сетевого графика, которые пригодны и для определения других важных параметров сетевого графика.

**Алгоритм 1.** Предположим, что нам дан сетевой график проекта. Необходимо перенумеровать все события графика так, чтобы для любой работы  $(P_i, P_j)$  выполнялось неравенство  $i < j$ . Соблюдение условия  $i < j$  во многих практических задачах дает большое удобство.

Если каждое событие  $P_j$  соединено со всеми остальными  $n$  дугами, часть этих дуг входит в  $P_j$ , а другая часть выходит из него, то, присваивая каждому событию номер, равный числу входящих дуг, номера событий будут отвечать указанным требованиям единственным образом.

**Алгоритм 2.** (Метод вычеркивания дуг).

Алгоритм состоит из следующих операций: события сети разбиваются по рангам, а нумерация событий ведется последовательно по мере возрастания рангов. События одного ранга нумеруются произвольно.

Как разбить события по рангам? В правильно составленном сетевом графике имеется только одно начальное и только одно конечное событие.

1 шаг. Выбираем на сетевом графике событие, которое не имеет входящих стрелок-работ, и присваиваем ему нулевой ранг. Начальное событие является событием нулевого ранга, и оно только одно, его номер равен нулю, т.е. получили начальное событие  $P_0$ .

2 шаг. Вычеркнем все работы, выходящие из события нулевого ранга.

3 шаг. После второго шага у нас обязательно появится одно или несколько событий, которые не будут иметь входящих работ. Присвоим этим событиям ранг = 1. Таким образом, все работы, входящие в события первого ранга, исходят из события нулевого ранга.

4 шаг. Если событий первого ранга несколько, то нумеруем их в произвольном порядке:  $P_1, P_2, \dots, P_k$ .

5 шаг. Вычеркиваем все работы, выходящие из событий первого ранга, и присваиваем всем событиям, которые после этого не будут иметь входящих работ, ранг = 2. Ясно, что любое событие второго ранга имеет входящие работы лишь из событий нулевого или первого ранга. Обобщая это и далее, можно сказать, что всякое событие ранга  $m$  имеет входящие работы лишь из событий, ранги которых меньше, чем  $m$ . В пределах ранга события нумеруем в произвольном порядке. Причем номер первого из них на единицу больше самого большого номера событий предыдущего ранга.

В сетевом графике число работ и событий – конечное, поэтому через конечное число шагов все события будут единственным образом разбиты по рангам и занумерованы. Очевидно, что для любой работы  $(P_i, P_j)$  в этом случае  $i < j$ .

**Примечание.** Если в процессе упорядочения событий по рангам окажется, что после вычеркивания всех работ, выходящих из событий  $(k - 1)$  ран-

га, не окажется ни одного события, которое не имеет входящих работ, то в сетевом графике имеется цикл.

Поясним, как используется изложенный алгоритм на примере (рис. 13).

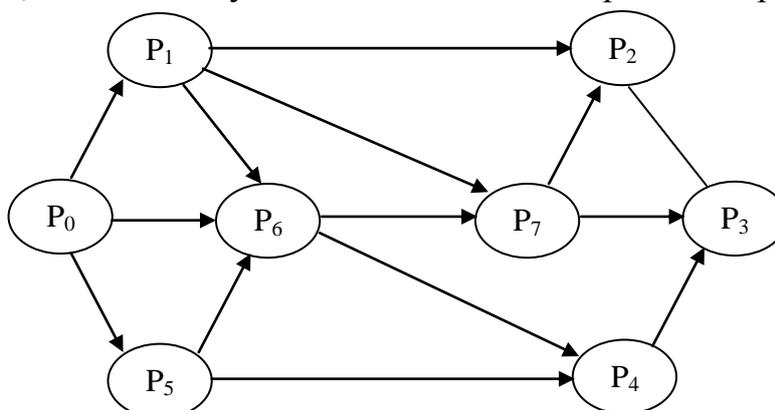


Рис. 13. Сетевой график с произвольной нумерацией событий

На данном примере события занумерованы в произвольном порядке. Перенумеруем события данного сетевого графика по алгоритму 2.

1. Разобьем события по рангам. Присвоим событию  $P_0$  нулевой ранг.

Вычеркнем выходящие из него работы  $(P_0, P_1)$ ,  $(P_0, P_6)$ ,  $(P_0, P_5)$ . После этого события  $P_1$  и  $P_5$  не будут иметь входящих работ. Таким образом, события  $P_1$  и  $P_5$  являются событиями первого ранга. Вычеркнем теперь все работы, выходящие из событий первого ранга, т.е. работы  $(P_1, P_2)$ ,  $(P_1, P_6)$ ,  $(P_1, P_7)$ ,  $(P_5, P_6)$ ,  $(P_5, P_4)$ . Теперь событие  $P_6$  не имеет входящих работ. Это будет событие второго ранга. Аналогично получаем, что события  $P_4$  и  $P_7$  – события третьего ранга,  $P_2$  – событие четвертого ранга, а событие  $P_3$  – событие пятого ранга.

Таким образом, события сетевого графика разбиты по рангам (табл. 1а).

Таблица 1а

Распределение событий сетевого графика, изображенного на рис.13, по рангам	
№ ранга	№№ событий
0	$P_0$
1	$P_1, P_5$
2	$P_6$
3	$P_4, P_7$
4	$P_2$
5	$P_3$

Таблица 1б

Соответствие старых номеров сетевого графика (рис. 13) новым	
Новый номер	Старый номер
$P_0$	$P_0$
$P_1$	$P_1$
$P_2$	$P_5$
$P_3$	$P_6$
$P_4$	$P_4$
$P_5$	$P_7$
$P_6$	$P_2$
$P_7$	$P_3$

2. Перенумеруем события по порядку возрастания рангов. События одного ранга равноценны, им номера присваиваем в произвольном порядке, но начинаем с номера, на единицу больше, чем номер наибольшего события предшествующего ранга. Новая нумерация приведена в табл. 1б.

В нашем примере  $P_0$  и в новой нумерации имеет номер 0. Событию  $P_1$  присваиваем номер 1,  $P_5 - 2$ ,  $P_4 - 3$ ,  $P_7 - 4$ ,  $P_3 - 5$ .

По-новому пронумерованный сетевой график приведен на рис.14.

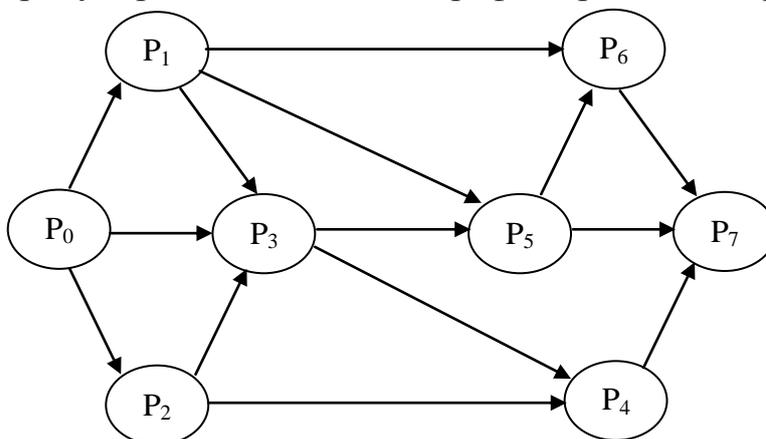


Рис. 14. Сетевой график с упорядоченной нумерацией событий

### Алгоритм 3 (Основной алгоритм).

Изложенный выше алгоритм 2 весьма прост в употреблении. Однако когда сеть имеет большое количество работ и событий, несмотря на всю простоту и наглядность алгоритма 2, использовать его все же становится затруднительно. В этих случаях удобно пользоваться алгоритмом Форда [4]. Алгоритм Форда является основным не только для задачи нумерации событий, но и для целого ряда излагаемых ниже задач.

Алгоритм 3 состоит из 2 этапов (предварительного и общего).

1 этап (предварительный).

Каждому событию  $P_j$  ставим в соответствие некоторое число  $\lambda_j^{(0)} = 0$ . Аналогично, каждой работе  $Y_{ij} = I$  (число  $Y_{ij} = I$  не меняется в процессе решения задачи по данному алгоритму).

2 этап (общий).

Все события сетевого графика просматриваем в порядке возрастания номеров, т.е.  $P_0, P_1, P_2, \dots, P_n$  (или в любом другом порядке).

При просмотре заменяем числа  $\lambda_j^{(k-1)}$ , которые получились на предыдущем шаге, на новые  $\lambda_j^{(k)}$  по формуле:

$$\lambda_j^{(k)} = \max_i \left\{ \lambda_i^{(p)} + Y_{ij} \right\} \quad j = 1, 2, \dots, n, \quad \lambda_0^{(k)} = 0(I),$$

где  $P = \begin{cases} k_i, & \text{если данное событие } P_i \text{ на } k\text{-м шаге уже просмотрено;} \\ k-1, & \text{если событие } P_i \text{ на } k\text{-м шаге не просматривалось.} \end{cases}$

Второй этап необходимо повторять до тех пор, пока на очередном  $k$ -м шаге все  $\lambda_j^{(k)}$  останутся неизменными, т.е. когда будет получено равенство:

$$\lambda_j^* = \lambda_j^{(k)} = \lambda_j^{(k-1)}; \quad j = 1, 2, \dots, n.$$

Число  $\lambda_j^* = \lambda_j^{(k)} = \lambda_j^{(k-1)}$  и будет являться рангом события  $P_i$ .

Чтобы в этом убедиться, достаточно понять смысл величин  $\lambda_j^{(k)}$ , полученных на  $k$ -м шаге. Можно показать, что число  $\lambda_j^{(k)}$  равно максимальному количеству работ цепочки, входящей в  $P_i$  и обнаруженной на  $k$ -м шаге.

Покажем практически, как работает данный алгоритм на примере того же сетевого графика, который мы использовали для иллюстрации алгоритма 2.

1 этап. Полагаем  $\lambda_0^{(0)} = \lambda_1^{(0)} = \dots = \lambda_7^{(0)}$ , и для всех  $(P_i, P_j) Y_{ij} = 1$ .

2 этап.

1 шаг. Просматриваем все события сетевого графика  $P_0, P_1, \dots, P_7$ .

Так как в событие  $P_1$  входит всего одна работа  $(P_0, P_1)$ , то  $\lambda_1^{(1)} = \lambda_0^{(0)} + Y_{01} = 1$   $\lambda_0^{(1)} = 0$ .

Аналогично, поскольку в событие  $P_2$  входят 2 работы  $(P_1, P_2)$ ,  $(P_7, P_2)$ , то:  $\lambda_2^{(1)} = \max\{\lambda_1^{(1)} + Y_{12}; \lambda_7^{(0)} + Y_{72}\} = \max\{2, 1\} = 2$ .

По этой же формуле вычисляем и все остальные  $\lambda_j^{(1)}$ :

$$\lambda_3^{(1)} = \max\{\lambda_2^{(1)} + Y_{23}; \lambda_7^{(0)} + Y_{73}; \lambda_4^{(0)} + Y_{43}\} = \max\{3, 1, 1\} = 3;$$

$$\lambda_4^{(1)} = \max\{\lambda_5^{(0)} + Y_{54}; \lambda_6^{(0)} + Y_{64}\} = \max\{1, 1\} = 1;$$

$$\lambda_5^{(1)} = \lambda_0^{(0)} + Y_{05} = 1;$$

$$\lambda_6^{(1)} = \max\{\lambda_4^{(1)} + Y_{46}; \lambda_0^{(0)} + Y_{06}; \lambda_5^{(1)} + Y_{56}\} = \max\{2, 1, 2\} = 2;$$

$$\lambda_7^{(1)} = \max\{\lambda_1^{(1)} + Y_{17}; \lambda_6^{(1)} + Y_{67}\} = \max\{2, 3\} = 3.$$

$\lambda_j^{(1)} \neq \lambda_j^{(0)}$  для всех  $j = 1, 2, \dots, 7$ , то делаем аналогично второй шаг.

2 шаг.

$$\lambda_0^{(2)} = 0;$$

$$\lambda_1^{(2)} = \lambda_0^{(2)} + Y_{01} = 1;$$

$$\lambda_2^{(2)} = \max\{\lambda_1^{(2)} + Y_{12}; \lambda_7^{(1)} + Y_{72}\} = \max\{2, 4\} = 4$$

$$\lambda_3^{(2)} = \max\{\lambda_2^{(2)} + Y_{23}; \lambda_7^{(1)} + Y_{73}; \lambda_4^{(1)} + Y_{43}\} = \max\{5, 4, 2\} = 5;$$

$$\lambda_4^{(2)} = \max\{\lambda_5^{(1)} + Y_{54}; \lambda_6^{(1)} + Y_{64}\} = \max\{2, 3\} = 3;$$

$$\lambda_5^{(2)} = \lambda_0^{(2)} + Y_{05} = 1;$$

$$\lambda_6^{(2)} = \max\{\lambda_1^{(2)} + Y_{16}; \lambda_0^{(2)} + Y_{06}; \lambda_5^{(2)} + Y_{56}\} = \max\{2, 1, 2\} = 2;$$

$$\lambda_7^{(2)} = \max\{\lambda_1^{(2)} + Y_{17}; \lambda_6^{(2)} + Y_{67}\} = \max\{2, 3\} = 3.$$

Опять  $\lambda_j^{(2)} \neq \lambda_j^{(1)}$  для всех  $j = 1, 2, \dots, 7$  выполняем третий шаг:

$$\lambda_0^{(3)} = 0; \quad \lambda_1^{(3)} = 1; \quad \lambda_2^{(3)} = 4; \quad \lambda_3^{(3)} = 5; \quad \lambda_4^{(3)} = 3;$$

$$\lambda_5^{(3)} = 1; \quad \lambda_6^{(3)} = 2; \quad \lambda_7^{(3)} = 3$$

Так как  $\lambda_0^{(3)} = \lambda_0^{(2)}$ ;  $\lambda_1^{(3)} = \lambda_1^{(2)}$ ;  $\lambda_2^{(3)} = \lambda_2^{(2)}$ ;  $\lambda_3^{(3)} = \lambda_3^{(2)}$ ;  $\lambda_4^{(3)} = \lambda_4^{(2)}$ ;

$$\lambda_5^{(3)} = \lambda_5^{(2)}; \quad \lambda_6^{(3)} = \lambda_6^{(2)}; \quad \lambda_7^{(3)} = \lambda_7^{(2)},$$

то получено необходимое закрепление вершин по рангам. Напомним, что величина  $\lambda_j$  показывает номер ранга события  $P_j$ . Таким образом, получено следующее упорядочение событий по рангам:

События нулевого ранга  $P_0$ .

События первого ранга  $P_1$  и  $P_5$ .

События второго ранга  $P_6$ .

События третьего ранга  $P_4$  и  $P_7$ .

События четвертого ранга  $P_2$ .

События пятого ранга  $P_3$ .

Теперь осталось занумеровать события в порядке возрастания рангов, нумеруя события одного ранга произвольно, т. е.

$$P_0 - P_0 \quad P_4 - P_4$$

$$P_1 - P_1 \quad P_5 - P_7$$

$$P_2 - P_5 \quad P_6 - P_2$$

$$P_3 - P_6 \quad P_7 - P_3$$

Если сравнить данную нумерацию событий с той, что мы получили по алгоритму 2, то увидим, что они совпадают. В дальнейшем условимся под пронумерованным сетевым графиком понимать график, у которого для любой работы  $(P_i, P_j)$  справедливо неравенство  $i < j$ .

Основной алгоритм можно запрограммировать и реализовать на ЭВМ. Ранг любого события сетевого графика можно определить при помощи данного алгоритма и в том случае, если информация о графике представлена в таблице списком работ  $(P_i, P_j)$  и их продолжительностей вида таблицы 2.

**Вид таблицы с необходимой информацией о сетевом графике**

Работа $(P_i, P_j)$		Продолжительность $t_{ij}$
начальное событие $P_i$	конечное событие $P_j$	
$P_0$	$P_s$	$t_{os}$
$\vdots$	$\vdots$	$\vdots$
$P_i$	$P_j$	$t_{ij}$
$\vdots$	$\vdots$	$\vdots$

Реализуется алгоритм и в этом случае так же: каждому событию ставится в соответствие число  $\lambda_j^{(0)} = 0$ , а каждой работе  $(P_i, P_j)$  – число  $Y_{ij} = 1$ . Различие лишь в том, что просмотр ведется уже по работам, а поэтому величина  $\lambda'_j$ , соответствующая событию  $P_j$ , в которое входит работа  $(P_i, P_j)$ , вычисляется по формуле:

$$\lambda'_j = \max[\lambda_j, \lambda_i + y_{ij}].$$

Здесь числа  $\lambda_j$  и  $\lambda_i$  – это вычисленные ранее числа. Как только просмотрен весь список работ, данный шаг заканчивается и начинается новый. Алгоритм заканчивается после того, как окажется, что после очередного шага все  $\lambda_j$  не изменились.

Рассмотрим более подробно вычислительную схему:

1. Составляется таблица вида 3.

Как видно из таблицы, в левом столбце записаны все события сетевого графика, а в остальных значения  $\lambda_j^{(0)}$ ,  $\lambda_j^{(1)}$  и т.д., которые получаются соответственно после первого, второго и т. д. шагов.

**Вид таблицы для расчета временных характеристик сетевого графика**

События	$\lambda^{(0)}$	$\lambda^{(1)}$	$\lambda^{(2)}$	...
$P_0$	0			
$P_1$	0			
$\vdots$	$\vdots$			
$P_i$	0			
$\vdots$	$\vdots$			
$P_j$	0			
$\vdots$	$\vdots$			
$P_n$	0			

2. Во всех строках столбца  $\lambda_j^{(0)}$  проставляются нули.

3. Просматриваем таблицу 2 в порядке записи работ сверху вниз. При

просмотре работы  $(P_i, P_j)$  в таблице 3 исправляем значение  $\lambda_j$ . Для чего находим в строке  $P_i$  таблицы 3 записанное и не вычеркнутое там  $\lambda_i$  и прибавляем к нему единицу, т.е. получаем  $\lambda'_j = \lambda_i + Y_{ij} = \lambda_i + 1$ .

3а). Находим в строке  $P_j$  записанное и не вычеркнутое  $\lambda_j$  и сравниваем его с  $\lambda'_j$ . Если окажется, что  $\lambda'_j \geq \lambda_j$ , то  $\lambda_j$  зачеркиваем и в столбце  $\lambda$ , который соответствует данному шагу в строке  $P_j$ ; записываем величину  $\lambda'_j$ . Если же окажется, что  $\lambda'_j < \lambda_j$ , то в строке  $P_j$  ничего не меняем.

4. Действия пункта 3 производим со всеми работами таблицы 2.

5. После того как полностью выполнен некоторый  $k$ -й шаг, вычеркиваем нуль в клетке  $(P_0, \lambda^{(k-1)})$  и записываем его в клетку  $(P_0, \lambda^{(k)})$ .

**Примечание.** При практических вычислениях лучше не зачеркивать, а как-нибудь запоминать  $\lambda'_j = \max\{\lambda_j; \lambda_i + Y_{ij}\}$  и стирать  $\lambda_j$ .

Проиллюстрируем практическую реализацию алгоритма на том же примере. Исходные данные приведены в табл. 4.

Таблица 4

Работа $(P_i, P_j)$		$t_{ij}$	Работа $(P_i, P_j)$		$t_{ij}$
$P_i$	$P_j$		$P_i$	$P_j$	
$P_0$	$P_1$	5	$P_4$	$P_3$	6
$P_0$	$P_6$	8	$P_5$	$P_6$	5
$P_0$	$P_5$	5	$P_5$	$P_4$	2
$P_1$	$P_2$	7	$P_6$	$P_7$	6
$P_1$	$P_7$	2	$P_6$	$P_4$	1
$P_1$	$P_6$	4	$P_7$	$P_2$	4
$P_2$	$P_3$	1	$P_7$	$P_3$	7

### Решение

Составляем таблицу 5.

Таблица 5

$P_i$	$\lambda^{(0)}$	$\lambda^{(1)}$	$\lambda^{(2)}$	$\lambda^{(3)}$
$P_0$	0	0	0	0
$P_1$	0	1	1	1
$P_2$	0	4	4	4
$P_3$	0	4	5	5
$P_4$	0	3	3	3
$P_5$	0	1	1	1
$P_6$	0	2	2	2
$P_7$	0	3	3	3

Так как столбцы  $\lambda^{(2)}$  и  $\lambda^{(3)}$  совпадают, то процесс распределения событий по рангам закончен. В результате получаем  $P_0$  – событие нулевого ранга,  $P_1$  и  $P_5$  – первого,  $P_6$  – второго,  $P_4$  и  $P_7$  – третьего,  $P_2$  – 4-го и  $P_3$  – пятого ранга.

Если теперь сравнить полученное распределение событий по рангам с предыдущим, то увидим, что они совпадают. Нумерацию событий производим как и раньше, т.е.

$$\begin{array}{ll} P_0 - P_0 & P_4 - P_4 \\ P_1 - P_1 & P_5 - P_7 \\ P_2 - P_5 & P_6 - P_2 \\ P_3 - P_6 & P_7 - P_3. \end{array}$$

## 2. РАСЧЕТ ВРЕМЕННЫХ ПАРАМЕТРОВ СЕТЕВОГО ГРАФИКА

### 2.1. РАСЧЕТ РАННИХ ХАРАКТЕРИСТИК СЕТЕВОГО ГРАФИКА

Важной является задача отыскания ранних временных характеристик сетевого графика  $(T_j^{(p)}, T_n^{(p)}, T_i^{(p)}, t_{ij}^{(p)}, S_{ij}^{(r)})$  и выявление критического пути от начального события  $P_0$  до конечного –  $P_n$ . Очевидно, что вычисление всех ранних характеристик сетевого графика, вообще говоря, сводится к отысканию моментов  $T_j^{(p)}$  ( $j = 1, 2, \dots, n$ ). Величина  $T_j^{(p)}$  равна длине самого длинного из всех путей, которые ведут из  $P_0$  в  $P_j$ . А так как событие  $P_n$  определяет момент завершения всех работ данного проекта, то  $T_n^{(p)}$ , равное длине критического пути при  $T_0^{(p)} = 0$ , показывает минимальное время, необходимое для выполнения всего проекта.

Данный раздел посвящен алгоритмам расчета моментов  $T_j^{(p)}$  для  $j = 1, 2, \dots, n$ , и если  $T_0^{(p)} = 0$ .

Заметим, что в настоящее время существует несколько алгоритмов расчета временных характеристик и критического пути, незначительно отличающихся по своей сути друг от друга. Основные из них мы и рассмотрим ниже.

$$\left\{ \begin{array}{l} T_1^{(p)} = t_{01}, \\ T_2^{(p)} = \max\{T_1^{(p)} + t_{12}, t_{02}\}, \\ T_3^{(p)} = \max\{T_2^{(p)} + t_{23}; T_1^{(p)} + t_{13}, t_{03}\}, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ T_j^{(p)} = \max\{T_{j-1}^{(p)} + t_{j-1,j}; \dots; T_1^{(p)} + t_{1j}, t_{0j}\}, \\ \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ T_n^{(p)} = T_n \cdot \max\{T_{n-1}^{(p)} + t_{n-1,n}; \dots; T_1^{(p)} + t_{1n}, t_{0n}\} \end{array} \right. \quad (*)$$

**Алгоритм 1.** Этот алгоритм построен на принципе оптимальности динамического программирования. Суть алгоритма хорошо видна из выражения (\*). По формулам (\*) мы всегда найдем моменты  $T_j^{(p)}$ . Однако, момент  $T_j^{(p)}$  показывает не только самый ранний момент свершения события  $P_j$ , а также и самый ранний срок начала всех работ, выходящих из события  $P_j$ . Прибавив к этому сроку время, необходимое для выполнения данной работы, мы получим самый ранний срок окончания работы.

Например, для работы  $(P_i, P_j)$  ранний срок окончания может быть записан по следующей формуле:

$$t_{ij}^{(p)} = T_i^{(p)} + t_{ij}, \quad (2.1)$$

а частный резерв – по формуле:

$$S_{ij}^{(r)} = T_j^{(p)} - t_{ij}^{(p)}. \quad (2.2)$$

**Пример.** Вернемся к сетевому графику, изображенному на рис. 1. Напомним, что числа в кружках – это номера событий, а числа на стрелках – значения  $t_{ij}$ . Воспользуемся формулами (\*).

$$\begin{aligned} T_1^{(p)} &= t_{01} = 3, \\ T_2^{(p)} &= \max\{T_1^{(p)} + t_{12}, t_{02}\} = T_1^{(p)} + t_{12} = 11, \\ T_3^{(p)} &= T_1^{(p)} + t_{13} = 10, \\ T_4^{(p)} &= T_2^{(p)} + t_{24} = 12, \\ T_5^{(p)} &= \max\{T_1^{(p)} + t_{15}, T_4^{(p)} + t_{45}\} = T_4^{(p)} + t_{45} = 15, \\ T_6^{(p)} &= \max\{T_3^{(p)} + t_{36}, T_5^{(p)} + t_{56}\} = T_5^{(p)} + t_{56} = 17, \end{aligned}$$

$$\begin{aligned}
T_7^{(p)} &= \max\{T_4^{(p)} + t_{47}, T_6^{(p)} + t_{67}\} = T_6^{(p)} + t_{67} = 22, \\
T_8^{(p)} &= \max\{T_4^{(p)} + t_{48}, T_7^{(p)} + t_{78}\} = T_7^{(p)} + t_{78} = 29, \\
T_9^{(p)} &= \max\{T_3^{(p)} + t_{39}, T_6^{(p)} + t_{69}, T_7^{(p)} + t_{79}\} = T_7^{(p)} + t_{79} = 28, \\
T_{10}^{(p)} &= \max\{T_3^{(p)} + t_{310}, T_9^{(p)} + t_{910}\} = T_9^{(p)} + t_{910} = 38, \\
T_{11}^{(p)} &= \max\{T_7^{(p)} + t_{711}, T_{10}^{(p)} + t_{1011}\} = T_{10}^{(p)} + t_{1011} = 42, \\
T_{12}^{(p)} &= \max\{T_8^{(p)} + t_{812}, T_{11}^{(p)} + t_{1112}\} = T_{11}^{(p)} + t_{1112} = 48, \\
T_{13}^{(p)} &= \max\{T_{10}^{(p)} + t_{1013}, T_{11}^{(p)} + t_{1113}, T_{12}^{(p)} + t_{1213}\} = T_{12}^{(p)} + t_{1213} = 55.
\end{aligned}$$

Таким образом, все работы сетевого графика, изображенного на рис. 1, будут выполнены не раньше, чем через  $T_{13}^{(p)} = 55$  единиц времени. В результате реализации алгоритма 1 мы получили для каждого события  $P_j$  моменты  $T_j^{(p)}$ . Все остальные ранние характеристики и частный запас времени могут быть получены по формулам (2.1) и (2.2) соответственно. Эти расчеты предлагается выполнить читателю в качестве упражнения.

**Алгоритм 2.** Для вычисления ранних характеристик и частичного резерва времени работ сетевого графика можно воспользоваться и основным алгоритмом, который изложен в первой главе для упорядочения событий по рангам. Алгоритм 2 также состоит из двух шагов, как основной алгоритм нумерации событий, но с той лишь разницей, что величины  $Y_{ij}$  здесь равны не единице, а  $t_{ij}$ . Поэтому всюду здесь мы вместо чисел  $Y_{ij}$  будем писать  $t_{ij}$ . В остальном алгоритмы совпадают. Рассмотрим алгоритм 2. Пусть мы уже имеем пронумерованный сетевой график.

*Предварительный шаг.* Каждому событию  $P_j$  сетевого графика, как и раньше, поставим в соответствие число  $\lambda = 0$ , а каждой работе  $(P_i, P_j)$  – ее продолжительность  $t_{ij}$ .

*Общий шаг.* Весь сетевой график просматриваем по событиям последовательно от  $P_0$  к  $P_n$  и вычисляем новые значения  $\lambda'_j$  чисел  $\lambda_j$  по формуле:

$$\lambda'_j = \max\{\lambda'_i + t_{ij}\}, \quad j = 1, 2, \dots, n; \quad \lambda'_0 = \lambda_0 = 0. \quad (2.3)$$

Так же как и раньше, в основном алгоритме, процесс расчета заканчиваем, как только на  $k$ -м шаге все  $\lambda_j^{(k)} = \lambda_j^{(k-1)}$ . В случае пронумерованного сетевого графика уже на втором шаге не может измениться ни одно  $\lambda_j$ , т.е. все

$\lambda_j^{(2)} = \lambda'_j$ . Это можно легко показать. Следовательно, если сетевой график пронумерован, то достаточно сделать всего один шаг, и мы получим величины  $\lambda'_j = \lambda_j^{(2)} = \lambda_j^{(1)}$ .

Заметим, что, как и в основном алгоритме первой главы, каждое из чисел  $\lambda'_j$ ; ( $j = 1, 2, \dots, n$ ) равно пути максимальной продолжительности, который исходит из события  $P_0$  и заканчивается в событии  $P_j$ . А по определению самого раннего срока свершения события  $P_j$ , получаем:

$$T_j^{(p)} = \lambda'_j = \lambda_j^{(1)}. \quad (2.4)$$

Таким образом, алгоритм 2 позволяет найти моменты  $T_j^{(p)}$ , а по формулам (2.1) и (2.2) найдем необходимые нам ранние характеристики и частный запас времени.

**Замечание.** Если в ходе расчета ранних характеристик сетевого графика по изложенным выше алгоритмам записывать около каждого события  $P_j$  не только  $T_j^{(p)}$  (или  $\lambda'_j$ ), но и номера  $i_1, i_2, \dots, i_t$  тех событий  $P_i$ , для которых в формуле (2.4) был достигнут максимум, то по этим пометкам можно легко найти критический путь.

**Пример.** Пусть мы имеем пронумерованный сетевой график (рис/ 15). На каждой стрелке-работе проставлено число, равное продолжительности этой работы. Применим алгоритм 2 для вычисления моментов  $T_j^{(p)}$  данного сетевого графика.

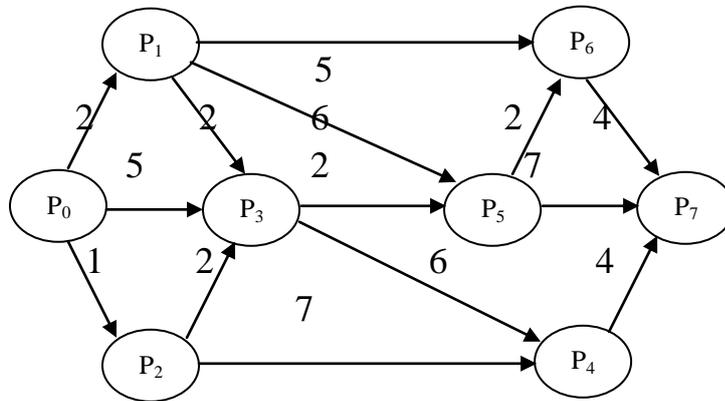


Рис. 15.

**Предварительный шаг.** Ставим в соответствие каждому событию  $P_j$  число  $\lambda_j = 0$ .

Первый шаг. Вычислим по формуле (2.3) все  $\lambda'_j = \lambda_j^{(1)} = T_j^{(p)}$ , просматривая график последовательно от события  $P_0$  к событию  $P_7$ .

$$T_0^{(p)} = \lambda'_0 = \lambda_0 = 0,$$

$$T_1^{(p)} = \lambda'_1 = \lambda_0 + t_{01} = 2,$$

$$T_2^{(p)} = \lambda'_2 = \lambda'_0 + t_{01} = 2,$$

$$T_3^{(p)} = \lambda'_3 = \max\{\lambda'_0 + t_{03}; \lambda'_1 + t_{13}; \lambda'_2 + t_{23}\} = \max\{5, 4, 3\} = 5,$$

$$T_4^{(p)} = \lambda'_4 = \max\{\lambda'_2 + t_{24}; \lambda'_3 + t_{34}\} = \max\{8, 11\} = 11,$$

$$T_5^{(p)} = \lambda'_5 = \max\{\lambda'_3 + t_{35}; \lambda'_1 + t_{15}\} = \max\{7, 8\} = 8,$$

$$T_6^{(p)} = \lambda'_6 = \max\{\lambda'_5 + t_{56}; \lambda'_1 + t_{16}\} = \max\{10, 7\} = 10,$$

$$T_7^{(p)} = \lambda'_7 = \max\{\lambda'_4 + t_{47}; \lambda'_5 + t_{57}; \lambda'_6 + t_{67}\} = \max\{15, 15, 14\} = 15.$$

Следовательно, все работы данного сетевого графика будут выполнены не раньше, чем через 15 единиц времени.

Отметим, что алгоритм 2 можно использовать и в том случае, когда сетевой график не пронумерован или задан таблицей работ вида таблицы 2. В этих случаях в предварительном шаге по-прежнему полагаем все  $\lambda_j = 0$ . В общем шаге поступаем различно.

В первом случае (график не пронумерован) вычисляем все  $\lambda_j^{(k)}$  на  $k$ -м шаге по формуле:

$$\lambda_j^{(k)} = \max_i \left[ \lambda_i^{(p)} + t_{ij} \right], \quad (2.5)$$

где  $P = k$ , если событие  $P_i$  уже просматривалось на  $k$ -м шаге и  $P = (k - 1)$  в противном случае.

Процесс заканчиваем, как только на  $k$ -м шаге все  $\lambda_j^{(k)} = \lambda_j^{(k-1)}$ , т.е. как только очередной  $k$ -й шаг не изменил всех предыдущих шагов.

Во втором случае (график задан таблицей работ), вычисляем величины  $\lambda'_j$  по формуле:

$$\lambda'_j = \max \left[ \lambda_j; \lambda_i + t_{ij} \right], \quad (2.6)$$

где  $\lambda_i$  и  $\lambda_j$  ранее полученные числа. Все вычисления производятся так же, как и при определении ранга событий, в предыдущей главе. Процесс вычисления заканчиваем опять, как только все  $\lambda_j^{(k)} = \lambda_j^{(k-1)}$  на  $k$ -м шаге.

Очевидно, что в обоих последних случаях число шагов, вообще говоря, больше одного, но не больше ранга конечного события. Моменты  $T_j^{(p)}$  определяются по формулам:

$$T_j^{(p)} = \lambda_j^* \quad j = 0, 1, \dots, n; \quad T_j^{(p)} = \max_j \left[ \lambda_j^* \right], \quad (2.7)$$

где  $\lambda_j^*$  – значение  $\lambda'_j$  на последнем шаге.

## 2.2. НАХОЖДЕНИЕ КРИТИЧЕСКОГО ПУТИ ПО РАННИМ ХАРАКТЕРИСТИКАМ

Из алгоритма 2 ясно, что для любой работы  $(P_i, P_j)$  справедливо неравенство  $T_j^{(p)} - T_i^{(p)} \geq t_{ij}$ . Можно показать, что для любой работы  $(P_i, P_j)$ , которая принадлежит критическому пути, имеет место равенство:

$$T_j^{(p)} - T_i^{(p)} - t_{ij} = 0. \quad (2.8)$$

Другими словами, *из всех путей, соединяющих событие  $P_0$  с событием  $P_n$ , критическим является тот, и только тот путь, для всех работ которого имеет место равенство (2.8).*

Докажем это. Пусть мы имеем некоторый путь  $(P_0, P_{i_1}, P_{i_2}, \dots, P_{i_k}, P_n)$ , для каждой работы которого выполняется равенство (2.8). Очевидно, что длина этого пути равна  $t_{0i_1} + t_{i_1i_2} + \dots + t_{i_k n}$  и согласно (2.8) она же равна:

$$T_{i_1}^{(p)} - T_0^{(p)} + T_{i_2}^{(p)} - T_{i_1}^{(p)} + \dots + T_n^{(p)} - T_{i_k}^{(p)} = T_n^{(p)} - T_0^{(p)},$$

$$\text{так как } t_{ij} = T_j^{(p)} - T_i^{(p)},$$

но так как  $T_0^{(p)} = 0$ , то:

$$\begin{aligned} t_{0i_1} + t_{i_1i_2} + \dots + t_{i_k n} &= T_{i_1}^{(p)} - T_0^{(p)} + T_{i_2}^{(p)} - T_{i_1}^{(p)} + \dots + \\ &+ T_n^{(p)} - T_{i_k}^{(p)} = T_n^{(p)} - T_0^{(p)} = T_n^{(p)}. \end{aligned}$$

Но  $T_n^{(p)}$ , по определению, равно длине критического пути. Следовательно, рассматриваемый путь является критическим. Таким образом, мы показали, что если для всех работ некоторого пути, соединяющего  $P_0$  с  $P_n$ , выполнено равенство (2.8), то оно является критическим.

Покажем теперь обратное, т.е., если путь является критическим, то для всех его работ справедливо равенство (2.8).

Пусть мы имеем критический путь  $(P_0, P_{j_1}, P_{j_2}, \dots, P_{j_m}, P_n)$  и пусть равенство (2.8) не выполнено для некоторой работы этого пути  $(P_{j_k}, P_{j(k+1)})$ .

Но так как согласно алгоритму 2 для любой работы сетевого графика справедливо неравенство  $t_{ij} \leq T_j^{(p)} - T_i^{(p)}$ , то для нашей работы, очевидно, справедливо неравенство:

$$t_{j_k j(k+1)} < T_{j(k+1)}^{(p)} - T_{j_k}^{(p)}.$$

А раз это так, то для длины рассматриваемого пути будет справедливо неравенство:

$$t_{0j_1} + t_{j_1j_2} + \dots + t_{j_m n} < T_{j_1}^{(p)} - T_0^{(p)} + T_{j_2}^{(p)} - T_{j_1}^{(p)} + \dots + \\ + T_n^{(p)} - T_{j_m}^{(p)} = T_n^{(p)} - T_0^{(p)} = T_n^{(p)}.$$

Мы пришли к противоречию. Наш путь не является самым длинным, а по предположению он критический. Следовательно, нельзя допускать, что хотя бы для одной работы критического пути не выполнено равенство (2.8) и т.д.

Изложенными выше доказательствами мы обосновали метод нахождения критического пути по ранним характеристикам.

Алгоритм состоит в следующем.

1. Для всех работ, входящих в конечное событие  $P_n$ , проверяют равенство (2.8) и выделяют из них те, для которых это равенство справедливо.

2. Рассматривают события, из которых выходят выделенные работы. Опять для всех входящих работ проверяют условие (2.8) и выделяют те, для которых оно выполняется.

3. Процесс этот продолжают до тех пор, пока не попадем в событие  $P_0$ . Ясно, что в сетевом графике может существовать и не один критический путь. Данный алгоритм позволяет найти все критические пути. Если же мы делаем пометки, о которых говорилось в предыдущем разделе, то надо двигаться от события  $P_n$  к событию  $P_0$  по пометкам.

Заметим, что если сетевой график не является упорядоченным или информация о нем задана списком работ в виде таблицы 2, то критический путь можно определить также по пометкам. Начинать надо теперь только не с события  $P_n$ , а с некоторого события  $P_k$ , для которого:

$$T_k^{(p)} = \max_i \left\{ T_j^{(p)} \right\} = T_{кр}.$$

Очевидно, что событие  $P_k$  всегда является конечным в сетевом графике. В этом случае критическим путем, соединяющим событие  $P_0$  с событием  $P_k$ , является тот и только тот путь, для всех работ которого выполняется условие:

$$T_j^{(p)} - T_i^{(p)} - t_{ij} = 0.$$

В заключение данного раздела сделаем еще один интересный вывод. Согласно формуле (2.2), частный резерв работы  $(P_i, P_j)$  равен:

$$S_{ij}^{(r)} = T_i^{(p)} - t_{ij}^{(p)}.$$

Подставив вместо  $t_{ij}^{(p)}$  его значение из (2.1) получаем:

$$S_{ij}^{(r)} = T_j^{(p)} - T_i^{(p)} - t_{ij}.$$

Если теперь сравнить правую часть этого равенства с условием (2.8), то видно, что *частный резерв любой работы критического пути равен нулю*, т.е.

$$S_{ij}^{(r)} = T_j^{(p)} - T_i^{(p)} - t_{ij} = 0. \quad (2.9)$$

### 2.3. РАСЧЕТ ПОЗДНИХ ХАРАКТЕРИСТИК И РЕЗЕРВОВ ВРЕМЕНИ

В первом разделе настоящей главы мы рассмотрели алгоритмы расчета ранних характеристик сетевого графика. Ранние характеристики являются весьма важными при управлении ходом работ по сетевому графику. Величины  $T_j^{(p)}$  говорят о том, что событие  $P_j$  не может свершиться раньше чем через  $T_j^{(p)}$ . Однако, это вовсе не значит, что оно не может свершиться позднее момента  $T_j^{(p)}$ . Это возможно в том случае, когда максимальная длина пути из  $P_j$  и  $P_n$  меньше разности  $T_n^{(p)} - T_j^{(p)}$ , т.е. когда свершение  $P_j$  позднее срока  $T_j^{(p)}$  не нарушает раннего срока свершения события  $P_n$ . Очевидно, что поздние характеристики являются важным дополнением к ранним характеристикам при управлении ходом работ по сетевому графику. Моменты  $T_j^{(p)}$  отражают самый поздний срок свершения события  $P_j$ , который не нарушает срока завершения всех работ проекта.

Выше мы изложили два алгоритма отыскания ранних характеристик. Каждому из этих алгоритмов соответствует свой алгоритм отыскания поздних характеристик. Теперь рассмотрим алгоритмы расчета поздних характеристик сетевого графика.

**Алгоритм 1.** Так как для конечного события  $P_n$   $T_n^{(p)} = T_n^{(n)} = T_n$ , то вычисление поздних характеристик производят, двигаясь от события  $P_n$  к событию  $P_0$ . Алгоритм 1 вычисления поздних характеристик, точнее моментов  $T_j^{(p)}$ , отличается от алгоритма 1 расчета моментов  $T_j^{(p)}$  тем, что все действия производят в обратном порядке, т. е. расчет начинаем с конечного события  $P_n$ , движемся от  $P_n$  к  $P_0$  и на каждом шаге выбираем минимум из разностей  $T_j^{(n)} - t_{ij}$ , схема алгоритма опять использует принцип оптимальности и хорошо видна из формул (\*\*).

$$\left\{ \begin{array}{l} T_{n-1}^{(n)} - T_n^{(n)} - t_{n-1,n}, \\ T_{n-2}^{(n)} = \min \left\{ T_n^{(n)} - t_{n-2,n}; T_{n-1}^{(n)} - t_{n-2,n-1} \right\}, \\ \dots \\ T_i^{(n)} = \min \left\{ T_n^{(n)} - t_{in}; T_{n-2}^{(n)} - t_{in-1}; \dots; T_j^{(n)} - t_{ij} \right\}, \\ \dots \\ \left\{ T_0^{(n)} = \min \left\{ T_n^{(n)} - t_{0n}; T_{n-1}^{(n)} - t_{0n-1}; \dots; T_1^{(n)} - t_{01} \right\} \right\}. \end{array} \right. \quad (**)$$

**Пример.** Найдем при помощи данного алгоритма поздние характеристики сетевого графика, изображенного на рис. 1.

$$\begin{aligned} T_{13}^{(n)} &= T_{13}^{(p)} = 55, \\ T_{12}^{(n)} &= T_{13}^{(n)} - t_{12,13} = 48, \\ T_{11}^{(n)} &= \min \left[ T_{13}^{(n)} - t_{1113}; T_{12}^{(n)} - t_{1112} \right] = T_{12}^{(n)} - t_{1112} = 42, \\ T_{10}^{(n)} &= \min \left[ T_{13}^{(n)} - t_{1013}; T_{11}^{(n)} - t_{1011} \right] = T_{11}^{(n)} - t_{1011} = 38, \\ T_9^{(n)} &= T_{10}^{(n)} - t_{910} = 28, \\ T_8^{(n)} &= T_{12}^{(n)} - t_{812} = 45, \\ T_7^{(n)} &= \min \left[ T_{11}^{(n)} - t_{711}; T_9^{(n)} - t_{79}; T_8^{(n)} - t_{78} \right] = T_9^{(n)} - t_{79} = 22, \\ T_6^{(n)} &= \min \left[ T_9^{(n)} - t_{69}; T_7^{(n)} - t_{67} \right] = T_6^{(n)} - t_{69} = 17, \\ T_5^{(n)} &= T_6^{(n)} - t_{56} = 15, \\ T_4^{(n)} &= \min \left[ T_8^{(n)} - t_{48}; T_7^{(n)} - t_{78}; T_5^{(n)} - t_{45} \right] = 12, \\ T_3^{(n)} &= \min \left[ T_{10}^{(n)} - t_{310}; T_9^{(n)} - t_{39}; T_6^{(n)} - t_{36} \right] = T_6^{(n)} - t_{36} = 16, \\ T_2^{(n)} &= T_4^{(n)} - t_{24} = 11, \\ T_1^{(n)} &= \min \left[ T_5^{(n)} - t_{15}; T_3^{(n)} - t_{13}; T_2^{(n)} - t_{12} \right] = T_2^{(n)} - t_{12} = 3. \end{aligned}$$

**Алгоритм 2.** Для отыскания поздних характеристик, а точнее моментов  $T_j^{(n)}$  также можно использовать и алгоритм 2, который, вообще говоря, мы описали в первом разделе. Алгоритм состоит в следующем:

1. Вычисляем от каждого события  $P_j$  ( $j = n-1, n-2, \dots, 1, 0$ ) макси-

мальные длины путей до конечного события  $P_n$ . Практически это выглядит следующим образом: в пронумерованном сетевом графике каждому событию  $P_j$  ставим в соответствие число  $\mu_j = 0$ .

2. Затем просматриваем сетевой график от  $P_n$  к  $P_0$  и вычисляем новое значение  $\mu'_j$  по формуле:

$$\mu'_j = \max_k [\mu'_j + t_{jk}], \quad (2.10)$$

$$\mu'_j = \mu_n = 0 \quad j = n-1, n-2, \dots, 1, 0.$$

В этом случае моменты  $T_j^{(n)}$  вычисляются по формуле:

$$T_j^{(n)} = T_n^{(n)} - \mu'_j \quad (\text{для } j = n-1, n-2, \dots, 1, 0). \quad (2.11)$$

Следует заметить, что если график не является пронумерованным, то как и раньше, положив все  $\mu_j = 0$ , на  $k$ -м шаге вычислим  $\mu_j^{(k)}$  по формуле:

$$\mu_j^{(k)} = \max_s [\mu_s^{(p)} + t_{js}] \quad (j = 0, 1, \dots, n), \quad (2.12)$$

где  $p = k$ , если на  $k$ -м шаге событие уже просмотрено, и  $p = (k-1)$  в противном случае.

Если же информация о графике задана списком работ в виде таблицы 2, то сначала полагаем все  $\mu_j = 0$ , а затем по ходу просмотра таблицы вычисляем  $\mu'_j = 0$  по формуле:

$$\mu'_j = \max [\mu_j; \mu_s + t_{js}]. \quad (2.13)$$

Снова, как и раньше, процесс вычислений по данному алгоритму прекращаем как только на  $k$ -м шаге получим, что:

$$\mu_j^{(k)} = \mu_j^{(k-1)} = \mu_j^{(*)}.$$

Моменты  $T_j^{(n)}$  вычисляются по формуле:

$$T_j^{(n)} = T_n^{(n)} - \mu_j^*. \quad (2.14)$$

Отметим, что число шагов не превышает максимального ранга.

**Пример.** Выше мы рассмотрели пример применения алгоритма 2 для подсчета моментов  $T_j^{(p)}$ . Вычисляем данным алгоритмом моменты  $T_j^{(n)}$  для того же примера.

Согласно алгоритму 2 расчет произведен в следующем порядке:

1. Полагаем  $T_7^{(n)} = T_7^{(p)} = T_{kp} = 15$ .
2. Каждому событию сетевого графика ставим в соответствие  $\mu_j = 0$ , т.е.  $\mu_0 = \mu_1 = \dots = \mu_7 = 0$ .
3. Начиная с конечного события, вычисляем  $P_7$ , по  $\mu'_j$  формуле (2.10)

для каждого события, т.е.  $\mu'_7, \mu'_6, \dots, \mu'_1, \mu'_0$

$$\mu'_7 = \mu_7 = 0,$$

$$\mu'_6 = \mu'_7 + t_{67} = 4,$$

$$\mu'_5 = \max[\mu'_6 + t_{56}; \mu'_7 + t_{57}] = \max[6, 7] = 7,$$

$$\mu'_4 = \mu'_7 + t_{47} = 4,$$

$$\mu'_3 = \max[\mu'_5 + t_{35}; \mu'_4 + t_{34}] = \max[9, 10] = 10,$$

$$\mu'_2 = \max[\mu'_4 + t_{24}; \mu'_3 + t_{23}] = \max[11, 12] = 12,$$

$$\mu'_1 = \max[\mu'_6 + t_{16}; \mu'_5 + t_{15} + \mu'_3 + t_{13}] = \max[9, 13, 12] = 13,$$

$$\mu'_0 = \max[\mu'_3 + t_{03}; \mu'_2 + t_{02} + \mu'_1 + t_{01}] = \max[15, 13, 15] = 15.$$

4. Согласно формуле (2.11), моменты  $T_j^{(n)}$  равны:

$$T_7^{(n)} = T_7^{(p)} = 15,$$

$$T_6^{(n)} = T_7^{(n)} - \mu'_6 = 15 - 4 = 11,$$

$$T_5^{(n)} = T_7^{(n)} - \mu'_5 = 15 - 7 = 8,$$

$$T_4^{(n)} = 15 - 4 = 11,$$

$$T_3^{(n)} = 15 - 10 = 5,$$

$$T_2^{(n)} = 15 - 12 = 3,$$

$$T_1^{(n)} = 15 - 13 = 2,$$

$$T_0^{(n)} = 15 - 15 = 0.$$

Мы рассмотрели алгоритмы расчета моментов  $T_j^{(p)}$  и  $T_j^{(n)}$ . Оказывается, если для каждого события  $P_j$  известны моменты  $T_j^{(p)}$  и  $T_j^{(n)}$ , то можно легко найти критический путь по следующей теореме:

**Теорема.** Для того, чтобы событие  $P_j$  принадлежало критическому пути, необходимо и достаточно, чтобы соблюдалось равенство:

$$T_j^{(n)} = T_j^{(p)} \quad (2.15)$$

или, что то же самое,  $T_j^{(n)} - T_j^{(p)} = 0$ .

В самом деле, по определению:

$$T_j^{(n)} = T_j^{(p)} - \mu_j^*. \quad (2.16)$$

Если (2.16) подставить в (2.15), то получим:

$$T_n^{(p)} - \mu_j^* = T_j^{(p)} \text{ или } T_n^{(p)} = T_j^{(p)} + \mu_j^*, \quad (2.17)$$

но  $T_j^{(p)}$  – это длина максимального пути из  $P_0$  в  $P_j$ , а  $\mu_j^*$  – длина максимального пути от  $P_j$  до  $P_n$ . Откуда и делаем заключение, что  $T_j^{(p)} + \mu_j^*$  – длина максимального пути из  $P_0$  в  $P_n$ , проходящего через  $P_j$ , т.е. критического.

А поэтому событие  $P_j$  принадлежит критическому пути.

Это мы сформулировали условие, при котором событие  $P_j$  принадлежит критическому пути.

Нас, безусловно, будет интересовать не только принадлежность события критическому пути, но и принадлежность той или иной работы критическому пути.

Теорема. Работа  $(P_i P_j)$  принадлежит критическому пути тогда и только тогда, когда

$$T_j^{(n)} - T_j^{(p)} - t_{ij} = 0. \quad (2.18)$$

Доказательство условия (2.18) довольно просто. Так как согласно (2.14)  $T_j^{(n)} = T_n^{(n)} - \mu_j^*$ , где  $T_n^{(n)} = T_n^{(p)}$ , то (2.18) можно записать:

$$T_j^{(n)} - T_i^{(p)} - t_{ij} = T_n^{(p)} - (T_i^{(p)} + t_{ij} + \mu_j^*). \quad (2.19)$$

Но так как  $T_j^{(p)}$  равно максимальной длине пути из  $P_0$  в  $P_j$ , а  $\mu_j^*$  – максимальной длине пути от  $P_j$  до  $P_n$ , то  $T_i^{(p)} + \mu_j^* + t_{ij}$  – есть максимальная длина пути от  $P_0$  до  $P_n$ .

Следовательно, если работа  $(P_i P_j)$  принадлежит критическому пути, то условие (2.18) выполнено. Можно показать и обратное, т.е. если условие (2.18) или, что то же самое (2.19), выполнено, то работа  $(P_i P_j)$  принадлежит критическому пути. Пусть мы имеем (2.19), из него получаем:

$$T_i^{(p)} + t_{ij} + \mu_j^* = T_n^{(p)}.$$

Это равенство нам и показывает, что максимальная длина пути, проходящего через работу  $(P_i P_j)$ , равна  $T_n^{(p)}$ , а поэтому это путь критический. Что и требовалось доказать.

Работы, лежащие на критическом пути, будем называть критическими. Все остальные работы – некритическими. Очевидно, что для всякой некритической работы:

$$T_i^{(n)} - T_j^{(p)} - t_{ij} > 0.$$

Условие (2.17) позволяет нам найти все критические работы, а, следова-

тельно, и все критические пути.

Чтобы найти критический путь, надо для каждой работы сетевого графика проверить условие (2.18) и выбрать те из них, которым это условие удовлетворяет. Затем из этих работ надо построить непрерывный путь, который связывает  $P_0$  с  $P_n$ . Таких путей, вообще говоря, несколько. Лучше двигаться от  $P_n$  к  $P_0$ .

Выше мы рассмотрели порядок расчета моментов и условия принадлежности события и работы критическому  $T_j^{(n)}$  пути, т.е. указали условие, благодаря которому можно найти все критические работы и все критические пути. Все это весьма важно при практическом использовании сетевого графика. Однако не менее важной является характеристика резерва времени работы. В первом разделе мы показали, как можно найти частный резерв времени. Более ценным для практики является полный резерв времени работы. Он показывает срок, на сколько единиц времени можно либо увеличить продолжительность работы  $(P_i P_j)$ , либо начать выполнять позднее момента  $(P_i P_j)$ , не повлияв на срок завершения проекта.

Для работы  $(P_i P_j)$  полный резерв равен:

$$S_{ij}^{(n)} = T_j^{(n)} - T_i^{(p)} - t_{ij}. \quad (2.20)$$

Очевидно, что все критические работы имеют  $S_{ij}^{(n)} = 0$ , а полный резерв всех некритических работ  $S_{ij}^{(n)} > 0$ .

В заключение отметим, что кроме рассмотренных выше частного и полного резервов, полезно знать *независимый* резерв времени работы  $(P_i P_j)$ . Он вычисляется по формуле:

$$S_{ij}^{(n)} = \max \left[ 0; T_j^{(p)} - T_i^{(n)} - t_{ij} \right]. \quad (2.21)$$

$S_{ij}^{(n)}$  показывает максимально допустимое количество времени у работы  $(P_i P_j)$ , которое можно использовать либо для увеличения ее продолжительности, либо для запаздывания начала ее выполнения при условии, что все работы, входящие в  $P_i$ , будут закончены к моменту  $T_i^{(n)}$ , а все работы, выходящие из  $P_j$ , будут начаты в момент  $T_j^{(p)}$ .

Следовательно, использование независимого резерва времени на работе  $(P_i P_j)$  никоим образом не изменит резервы времени других работ.

## 2.4. МЕТОДЫ РЕАЛИЗАЦИИ АЛГОРИТМА РАСЧЕТА $T_j^{(p)}$ И $T_j^{(n)}$

Все временные параметры сетевого графика мы намеренно разбили на две группы: ранние характеристики и поздние характеристики и указали два алгоритма их расчета, которые, кстати сказать, по своей сути имеют много общего.

Следует заметить, что в практике, когда ведут расчеты временных параметров сетевого графика вручную или на ЭВМ, не разделяют алгоритмы расчета на расчет ранних и поздних характеристик как нечто совершенно различное. Обычно сначала рассчитывают  $T_j^{(p)}$ , а затем  $T_j^{(n)}$  тем или иным способом. Ниже мы будем понимать *под прямым ходом расчета сетевого графика расчет момента  $T_j^{(p)}$* , ибо этот расчет ведется от  $P_0$  к  $P_n$ , как бы по направлению стрелок-работ. Расчет же моментов  $T_j^{(n)}$  ведется от  $P_n$  к  $P_0$ , т.е. против направления стрелок-работ. Мы его будем называть *обратным ходом расчета сетевого графика*. Произведя прямой, а затем обратный ход расчета сетевого графика, мы, вообще говоря, уже можем получить все временные характеристики.

Однако просчет сетевого графика прямым и обратным ходом по одному и тому же алгоритму может быть реализован в различной форме. В разных условиях оказывается наиболее удобной та или иная форма расчета временных параметров.

### 2.4.1. Расчет временных параметров сетевого графика на самом графике

Эта форма расчета более удобна при ручном счете и для небольших графиков.

Каждое событие, обозначенное кружком на сетевом графике, делят на 4 части-сектора.

В верхнем секторе записывают номер данного события, в левом –  $T_j^{(p)}$ , в – правом  $T_j^{(n)}$ , а в нижнем – номера тех событий, через которые идет путь от  $P_0$  к  $P_j$  максимальной продолжительности.

Расчет обычно ведут в следующей последовательности:

1. Нумеруют события сетевого графика и номера событий записывают в верхний сектор. График становится пронумерованным.

2. В левом секторе события  $P_0$  записывают  $T_0^{(p)} = 0$ .

3. По формуле, скажем  $T_j^{(p)} = \max_i [T_i^{(p)} + t_{ij}]$   $j = 1, 2, \dots, n$ , при  $T_0^{(p)} = 0$

вычисляют моменты  $T_j^{(p)}$  для каждого  $P_j$ .

4. Полагают  $T_n^{(n)} = T_n^{(p)} = T_n$ .

5. Определяют моменты  $T_j^{(n)}$  по одному из изложенных алгоритмов (обратный ход расчета сетевого графика), например по формуле

$$T_j^{(n)} = \min_s [T_s^{(n)} - t_{js}], \quad T_n^{(n)} = T_n^{(p)}$$

для всех  $j = n-1, n-2, \dots, 1, 0$ .

6. Для расчета резервов времени информацию либо снимают с графика и рассчитывают отдельно, либо просто она видна непосредственно на графике.

**Пример.** Пусть имеем пронумерованный сетевой график (рис. 17). Номера событий проставлены в верхнем секторе, и в левый сектор события  $P_0$  записано  $T_0^{(p)} = 0$ . Согласно формуле

$$T_j^{(p)} = \max_i [T_i^{(p)} + t_{ij}] \quad j = 1, 2, \dots, n$$

вычислим для каждого  $P_j$  моменты  $T_j^{(p)}$  и проставим их в левых секторах соответствующих событий.

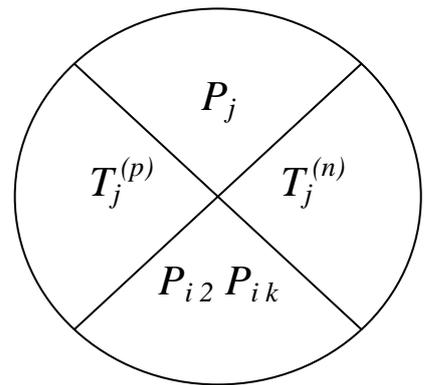


Рис. 16

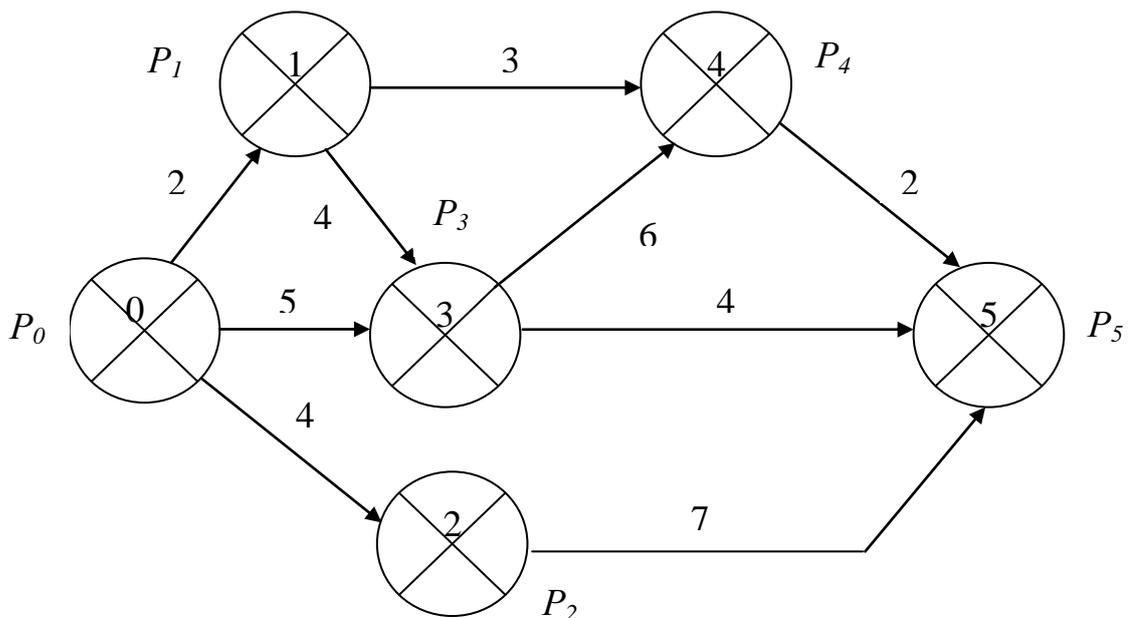


Рис. 17

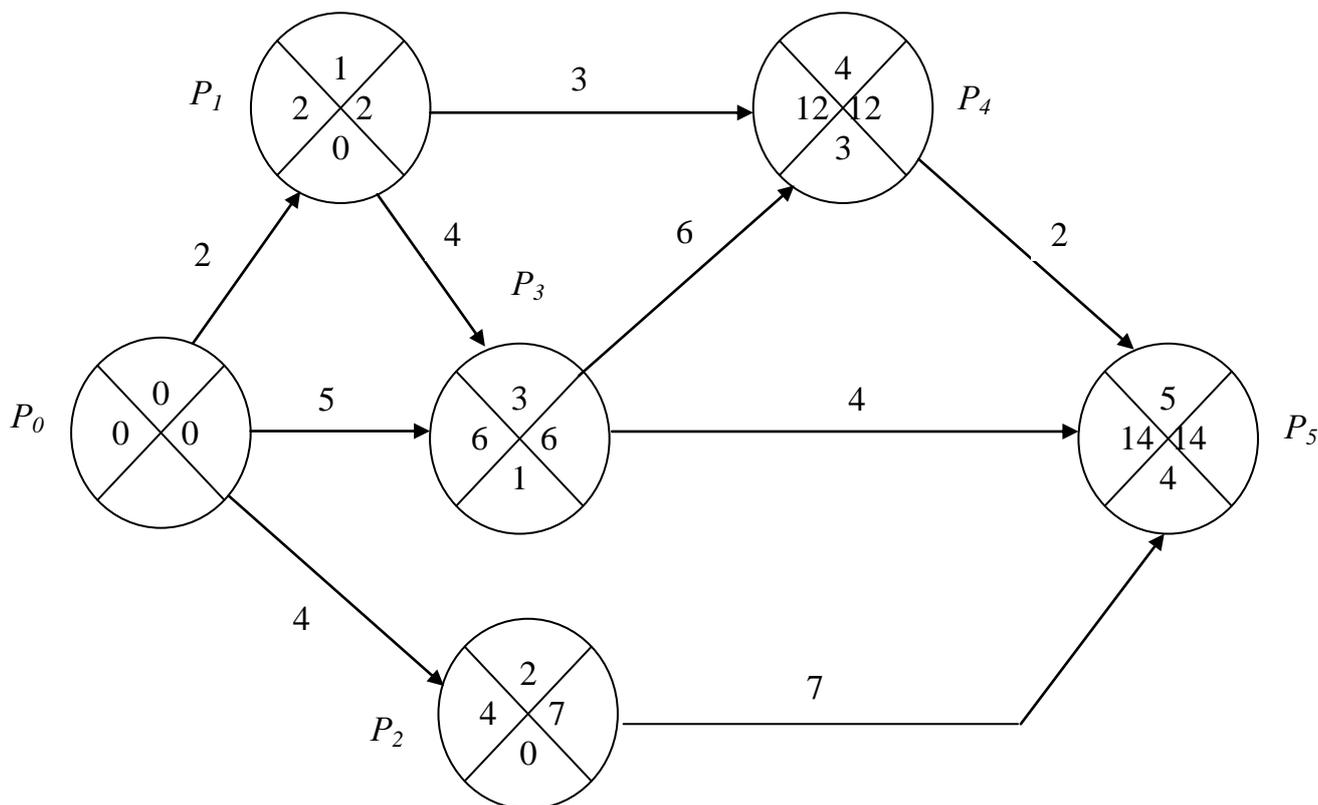


Рис. 18

В нижнем секторе проставлены номера предшествующих событий, через которые прошел максимальный путь от  $P_0$  до  $P_j$ . Так, цифра 4, стоящая в нижнем секторе события  $P_5$ , означает, что максимальный путь от  $P_0$  к  $P_5$  прошел через событие  $P_4$ . По цифрам, проставленным в нижних секторах событий, можно легко отыскать критический путь. Для нашего графика это можно сделать так.

Начнем строить критический путь с конечного события  $P_5$ . Видим, что самый длинный путь, связывающий события  $P_0$  и  $P_5$  проходит через событие  $P_4$  (это нам показывает цифра «4» в нижнем секторе события  $P_5$ ). Значит, чтобы найти критический путь, мы должны найти событие, через которое проходит критический путь от  $P_0$  до  $P_4$ . А это нам показывает цифра «3», стоящая в нижнем секторе события  $P_4$ . Поступая так и далее, т.е. двигаясь по событиям, номера которых указаны в нижнем секторе последующих событий, мы получаем последовательность событий, через которые проходит критический путь. А тем самым и находим его. В нашем примере критический путь проходит через события  $P_5, P_4, P_3, P_1, P_0$  или, что то же самое,  $P_0, P_1, P_3, P_4, P_5$ .

На графике критический путь обычно выделяется либо жирной линией, либо двойной линией, либо линией другого цвета.

Рассчитаем теперь моменты  $T_j^{(n)}$ . Для этого сначала полагаем

$T_5^{(n)} = T_5^{(p)} = 14$  и затем по формуле  $T_j^{(n)} = \min_s [T_s^{(n)} - t_{js}]$  ( $j = n-1, n-2, \dots, 1, 0$ )

для каждого события  $P_j$  вычисляем самый поздний момент свершения величины  $T_j^{(n)}$ , записываем в правый сектор события  $P_j$ .

В результате расчета нашего графика имеем:

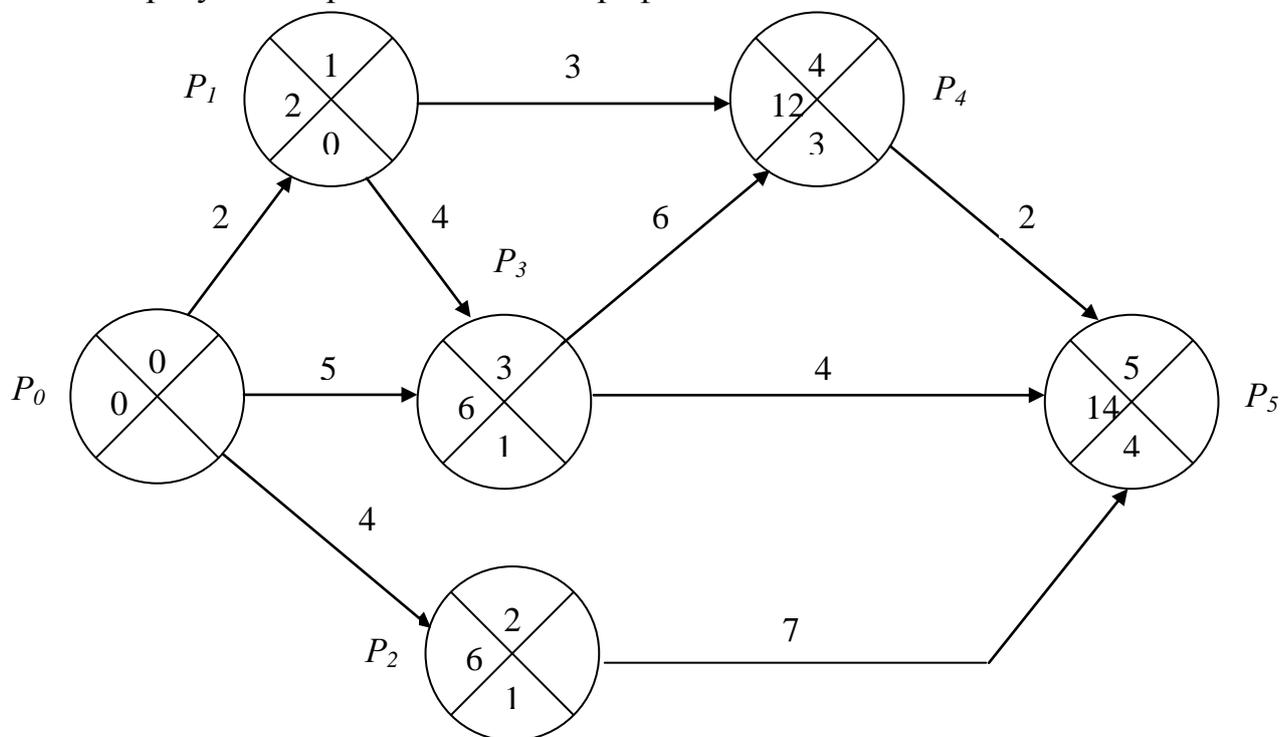


Рис. 19

Вычислив  $T_j^{(p)}$  и  $T_j^{(n)}$  для каждого  $P_j$  сетевого графика, можно не только построить критический путь, но и вычислить резервы времени либо отдельно в таблице, либо непосредственно на графике. Это делается довольно просто и предоставляется выполнить читателю.

#### 2.4.2. Расчет временных параметров сетевого графика в матричной форме

Расчет временных параметров сетевого графика в матричной форме производится следующим образом или в следующем порядке.

1. Составляется таблица вида 6.

В каждой клетке  $(P_i P_j)$  таблицы записана продолжительность работы  $(P_i P_j)$ . Очевидно, что если сетевой график пронумерован, т.е. для каждой работы  $(P_i P_j)$  выполнено  $i < j$ , то все числа  $t_{ij}$  матрицы будут находиться

выше главной диагонали, отмеченной крестиками. В этом случае, пользуясь, скажем алгоритмом 2, за один шаг прямым ходом расчета получим все  $T_j^{(p)}$ , а затем обратным ходом расчета получим  $T_j^{(n)}$ .

Практически расчет моментов  $T_j^{(p)}$  производится так. В каждую строку самого левого столбца ( $T_j^{(p)}$ ) таблицы 6 пишем  $\lambda_i = 0$ . Затем, положив  $\lambda'_0 = \lambda_0 = 0$ , берем столбец  $P_1$  и вычисляем  $T_1^{(p)} = \lambda'_0$  по формуле  $T_1^{(p)} = \max_i [T_i^{(p)} + t_{i1}]$ . Другими словами, складываем числа самого левого столбца с соответствующими числами столбца  $P_1$ , выбираем среди них максимальное и записываем в строку  $P_1$  в столбец  $T_j^{(p)}$ , а стоящий там ноль зачеркнем. После того как мы рассчитали таким образом  $\lambda_1 = T_1^{(p)}$ , переходим к столбцу  $P_2$  и повторяем расчет до тех пор, пока не будет заполнен весь столбец  $T_j^{(p)}$ .

Затем переходим к расчету  $T_j^{(n)}$ . Моменты  $T_j^{(n)}$  рассчитываются, аналогично, но все делается наоборот. А именно, если при определении моментов  $T_j^{(p)}$  расчет ведут от  $P_0$  к  $P_n$ , то при определении  $T_j^{(n)}$  – от  $P_n$  к  $P_0$ . Для этого полагают, что  $\lambda'_n = T_n^{(n)} = T_n$ . Далее моменты  $T_j^{(n)}$  определяют по формуле:  $T_j^{(n)} = \min_{k>j} [T_k^{(n)} - t_{jk}]$ .

Таблица 6

**Общий вид таблицы для расчета временных параметров сетевого графика**

$T_j^{(p)}$	$P_j$	$P_0$	$P_1$	...	$P_i$	...	$P_j$	...	$P_n$
	$P_0$	×	$t_{01}$	...	$t_{0i}$	...	$t_{0j}$	...	$t_{0n}$
	$P_1$		×	...	$t_{1i}$	...	$t_{1j}$	...	$t_{1n}$
	⋮			×	⋮	...	⋮	...	⋮
	$P_i$		$t_{i1}$	...	×	...	$t_{ij}$	...	$t_{in}$
	⋮		⋮	...	⋮	×	⋮	...	⋮
	$P_j$		$t_{j1}$	...	$t_{ji}$	...	×	...	$t_{jn}$
	⋮		⋮	...	⋮	...	⋮	×	⋮
	$P_n$		$t_{n1}$	...	$t_{ni}$	...	$t_{nj}$	...	×
$T_j^{(n)}$									
$T_j^{(p)}$									
$T_j^{(n)} - T_j^{(p)}$									

Получив величины  $T_j^{(p)}$  и  $T_j^{(n)}$ , в последнюю строку матрицы, записывают разность  $T_j^{(n)} - T_j^{(p)}$ , которая определяет принадлежность события  $P_j$  критическому пути. Из этих же характеристик можно получить и резервы времени.

**Пример 3**

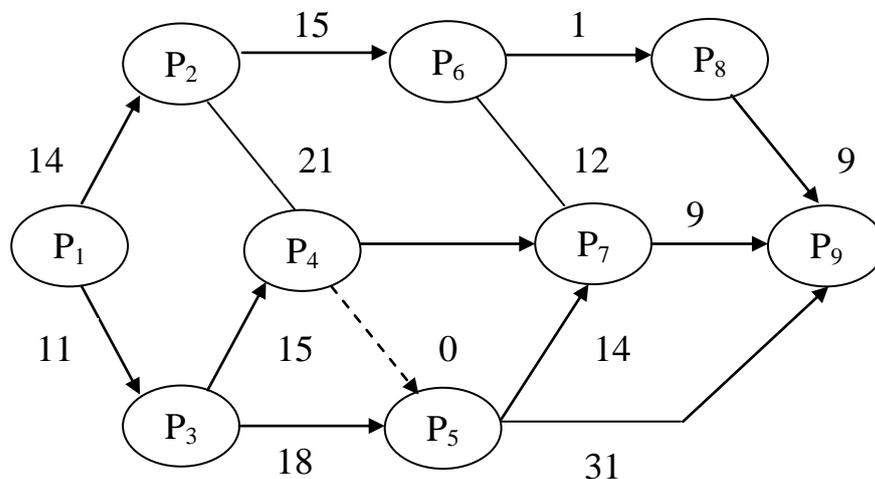


Рис. 20

Шаг 1. Строим таблицу вида 6.

Шаг 2. Во все строки столбца  $T_j^{(p)}$  таблицы записываем нули.

Шаг 3. Складываем столбец  $T_j^{(p)}$  со столбцом  $P_j$ , находим максимальную сумму и записываем ее в столбец  $T_j^{(p)}$  в строку  $P_j$ . Расчет надо начинать с начального события и переходить последовательно от события к событию. Если сетевой график пронумерован, то достаточно провести расчет от  $P_0$  до  $P_n$  один раз. В нашем примере расчет происходит так:  $\lambda'_1 = \lambda_1 = 0$  записываем в первую строку столбца  $T_j^{(p)}$  и зачеркиваем стоящий в ней нуль. Далее,  $\lambda'_2 = \lambda'_1 + 14 = 14$  во вторую строку столбца  $T_j^{(p)}$  пишем число 14 и зачеркиваем стоящий в ней нуль.  $\lambda'_3 = \lambda'_1 + 11 = 11$ . Аналогично, записали число 11 в третью строку столбца  $T_j^{(p)}$  и зачеркнули нуль.

$$\lambda'_4 = \max[\lambda'_2 + 21, \lambda'_3 + 15] = \max[35, 26] = 35.$$

В четвертую строку столбца  $T_j^{(p)}$  записали число 35. Таким же образом рассчитаны все числа столбца  $T_j^{(p)}$ .

Шаг 4. Записываем столбец  $T_j^{(p)}$  в строку  $T_j^{(p)}$ .

Шаг 5. Полагаем  $T_9^{(n)} = T_9^{(p)} = 66$  и вычисляем все остальные  $T_j^{(n)}$ . Сначала вычисляем  $T_8^{(n)}$ . Берем строку  $P_8$  и вычитаем из строки  $T_j^{(n)}$ , в которую

можно было также записать нули во все столбцы. В случае пронумерованного графика это не обязательно. Важно лишь проводить вычисления последовательно от конечного события к начальному.

$$T_8^{(n)} = T_9^{(n)} - 9 = 57.$$

Число 57 записываем в столбец  $P_8$  строки  $T_j^{(n)}$ .  $T_7^{(n)} = T_9^{(n)} - 9 = 57$ . Здесь мы из строки  $T_j^{(n)}$  вычитали строку  $P_7$ .

$$T_6^{(n)} = \min \left[ T_8^{(n)} - 1, T_7^{(n)} - 12 \right] \min [56, 45] = 45.$$

Записываем «45» в шестой столбец строки  $T_j^{(n)}$ . Аналогично вычисляем все другие числа строки  $T_j^{(n)}$ .

Шаг 6. Из строки  $T_j^{(n)}$  вычитаем строку  $T_j^{(p)}$  и определяем события критического пути. Согласно теореме, чтобы событие  $P_j$  принадлежало критическому пути, необходимо и достаточно, чтобы  $T_j^{(n)} = T_j^{(p)}$  или  $T_j^{(n)} - T_j^{(p)} = 0$ . В нашем примере критический путь проходит через события  $P_1, P_2, P_4, P_5, P_9$ .

### 2.4.3. Расчет временных характеристик графика в форме таблицы

Расчет моментов  $T_j^{(p)}$ ,  $T_j^{(n)}$  и всех других временных характеристик производится по вышеизложенным алгоритмам. Особенность расчета в таблицах состоит в том, что здесь все рассчитывается не на событие, а на работу. В таблице добавляют столбец для независимого резерва работы. Расчет сетевого графика в таблицах можно проводить в следующем порядке:

**Шаг 1.** Определяем раннее начало и раннее окончание каждой работы  $(P_i P_j)$ , т. е. заполняем столбцы 4 и 5. Для этого:

а) полагаем раннее начало всех работ, выходящих из начального события, равным нулю, ибо  $T_0^{(p)} = 0$ ;

б) рассматриваем работы, выходящие из начального события, и вычисляем их раннее окончание. Раннее окончание любой работы равно раннему началу плюс продолжительность;

в) в столбце 2 берем следующую работу;

г) смотрим в столбце 1, сколько работ ей предшествует. Находим эти работы. Очевидно, что номер начального события данной работы должен совпадать с номером конечного события любой предшествующей работы. По этому признаку и находим все работы, предшествующие данной;

д) в столбце 5 (раннее окончание) выбираем максимальное значение этих работ и записываем в столбец 4 в строку данной работы;

е) вычисляем раннее окончание данной работы. Для этого к записанному только что значению в столбце 4 (раннему началу) прибавляем продолжительность этой работы из столбца 3 и записываем в столбец 5;

ж) пункты в)-е) повторяем, пока не заполним столбцы 4 и 5 полностью.

Таблица 7

**Расчет временных характеристик сетевого графика**

Частный резерв: $T_j^{(p)} - T_i^{(p)} - t_{ij}$	9	
Полный резерв времени у работы $(P_i P_j): T_j^{(n)} - T_j^{(p)} - t_{ij}$	8	
Позднее окончание работы $(P_i P_j): T_j^{(n)} + t_{ij}$	7	
Позднее начало работы: $T_i^{(n)}$	6	
Раннее окончание работы $(P_i P_j): T_i^{(p)} + t_{ij}$	5	
Раннее начало работы: $T_i^{(p)}$	4	
Продолжительность работы $(P_i P_j): t_{ij}$	3	
Список работ $(P_i P_j)$	2	
Количество предшествующих работ	1	

Очевидно, что максимальное значение столбца 5 равно длине критического пути  $T_{кр}$ .

**Шаг 2.** Определяем позднее начало и позднее окончание работ  $(P_i P_j)$ , т.е. заполняем столбцы 6 и 7. Для этого:

а) максимальное значение столбца 5  $= T_{кр}$  перепишем в столбец 7 (позднее окончание) в строки всех тех работ, которые входят в конечное событие  $P_n$ ;

б) из только что записанного значения в столбец 7 вычитаем продолжительность рассматриваемых работ, стоящую в столбце 3, и результат записываем в столбец 6 (позднее начало) в строку данной работы;

в) продолжаем расчет поздних характеристик (заполнение столбцов 6 и 7) от конца к началу, т.е. от событий с большими номерами к событиям с меньшими номерами;

г) рассматриваем в столбце 2 очередную работу, затем находим все работы, у которых начальное событие равно конечному событию данной работы и из столбца 6 берем минимальное значение, соответствующее рассматриваемым работам, т.е. стоящее в рассматриваемых строках. Это минимальное значение записываем в столбец 7 в строку данной работы;

д) из только что записанного значения столбца 7 вычитаем продолжительность данной работы и результат записываем в столбец 6 в строку данной работы;

е) Пункты г)-д) повторяем, пока не заполним полностью столбцы 6 и 7, т.е. пока не вычислим поздние характеристики всех работ.

**Шаг 3.** Вычисляем полные резервы всех работ, т.е. заполняем столбец 8. Из значений столбца 6 вычитаем соответствующие значения столбца 4 (или из значений столбца 7 вычитаем соответствующие значения столбца 5).

**Шаг 4.** Вычисляем частные резервы работ, т.е. заполняем столбец 9. Для этого обращаемся к столбцам 4 и 5.

а) в столбце 2 выбираем работу  $(P_i P_j)$  и смотрим номер события, в которое она входит;

б) в столбце 2 отыскиваем все работы, которые выходят из данного события;

в) в столбце 4 среди значений этих работ берем максимум и из него вычитаем значение столбца 5 данной работы;

г) очевидно, что для работ, входящих в конечное событие, значения столбца 5 вычитаются из максимального значения этого же столбца и записываются в столбец 9. Находят критический путь также, как и раньше.

## 2.5. МЕТОД ПОТЕНЦИАЛОВ

Расчет ранних временных характеристик в методе потенциалов осуществляется одним из изложенных выше алгоритмов. Обычно это выполняется прямым ходом расчета сетевого графика, т. е. от  $P_0$  к  $P_n$ .

Обратный ход расчета сетевого графика от  $P_n$  к  $P_0$  в методе потенциалов отличается от изложенных выше алгоритмов, поэтому приведем здесь лишь обратный ход расчета.

В методе потенциалов конечному событию сетевого графика  $P_n$  задают потенциал, равный нулю. Потенциалы остальных событий определяются обратным ходом расчета сетевого графика по формуле:

$$\Pi_i = \max \{ \Pi_j + t_{ij} \}, \quad (2.22)$$

где  $\Pi_j$  – потенциал события  $P_j$ .

Из формулы (2.22) видно, что обратный счет в методе потенциалов аналогичен определению моментов  $T_j^{(p)}$ , т.е. прямому счету. Потенциал  $\Pi_j$  события  $P_j$  показывает максимальную длину пути от  $P_n$  до  $P_j$ . Максимум, (2.22), говорит о том, что если из события  $P_i$  выходят несколько работ, то для определения  $\Pi_j$  надо брать максимальную из сумм  $[\Pi_j + t_{ij}]$ .

В чем преимущества расчёта сетевого графика по методу потенциалов? В ходе выполнения проекта возникают отклонения от характеристик ранее рассчитанного графика. Гибкость сетевого планирования в том и состоит, что дальнейшее выполнение проекта осуществляется с учетом изменений, учет которых требует пересчета оставшейся части сетевого графика. Но информация о ходе выполнения проекта не изменяет потенциалов. Поэтому расчет по методу потенциалов упрощает пересчет графика заново. Эта особенность метода потенциалов важна при распределении ресурсов по работам сетевого графика, ибо алгоритм расчета сетевого графика по ресурсам требует многократного пересчета временных параметров графика.

Укажем теперь порядок определения поздних характеристик и резервов времени сетевого графика в методе потенциалов. Моменты  $T_j^{(n)}$  определяются как разность длины критического пути и потенциала события  $P_j$ , т. е. по формуле:

$$T_j^{(n)} = T_{kp} - \Pi_j. \quad (2.23)$$

Полный резерв времени работы  $(P_i P_j)$  вычисляется по формуле:

$$S_{ij}^n = T_{kp} - (\Pi_j + T_j^{(p)} + t_{ij}), \quad (2.24)$$

частный – по формуле:

$$S_{ij}^{(r)} = T_j^{(p)} - (T_i^{(p)} + t_{ij}) = T_j^{(p)} - T_i^{(p)} - t_{ij} \quad (2.25)$$

и независимый:

$$S_{ij}^{(n)} = T_j^{(p)} + \Pi_j - (T_{kp} + t_{ij}). \quad (2.26)$$

Критический путь можно определить либо по ранним характеристикам, либо при помощи резервов времени. Зная моменты  $T_j^{(p)}$  и  $T_j^{(n)}$  для всех  $j = 0, 1, 2, \dots, n$ , легко определить все остальные характеристики.

### Пример

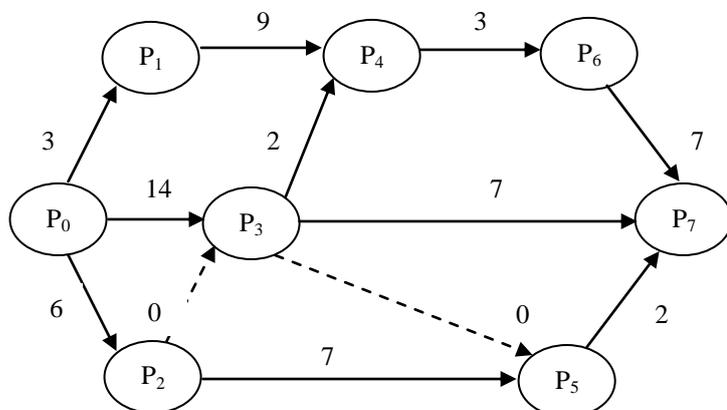


Рис. 21

Читателю предлагается рассчитать данный сетевой график методом по-

тенциалов. Ответ к предложенному расчету приведен в таблице 8.

Таблица 8

**Сроки свершения событий сетевого графика (рис. 21)**

Моменты событий	Номера событий							
	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$T_j^{(p)}$	0	3	6	14	16	14	19	26
$\Pi_j$	26	19	12	12	10	2	7	0
$T_j^{(n)}$	0	7	14	14	16	24	16	26

Таблица 9

**Временные параметры сетевого графика (рис. 21)**

Работы		Продолжи- тельность	Начало		Окончание		Резерв	
начальное событие	конечное событие		раннее	позднее	раннее	позднее	$S_{ij}^{(n)}$	$S_{ij}^{(r)}$
$P_0$	$P_1$	3	0	4	3	7	4	0
$P_0$	$P_2$	6	0	8	6	14	8	0
$P_0$	$P_3$	14	0	0	14	14	0	0
$P_1$	$P_4$	9	16	7	16	16	4	4
$P_2$	$P_3$	0	6	14	14	14	8	8
$P_2$	$P_5$	7	6	17	24	24	11	1
$P_3$	$P_4$	2	14	14	16	16	0	0
$P_3$	$P_5$	0	14	24	24	24	10	0
$P_3$	$P_7$	7	14	19	26	26	5	5
$P_4$	$P_6$	3	16	16	19	19	0	0
$P_5$	$P_7$	2	14	24	26	26	10	10
$P_6$	$P_7$	7	19	19	26	26	0	0

**2.6. АЛГОРИТМ ВЫЯВЛЕНИЯ ЦИКЛОВ**

Выше было установлено, что в сетевом графике не должно быть циклов. Наличие циклов свидетельствует об ошибке, возникшей при построении сетевого графика. Кроме того, изложенные выше алгоритмы определения временных параметров сетевого графика пригодны лишь в том случае, если сетевой график не содержит циклов. Если сетевой график небольшой, то можно сравнительно легко обнаружить и устранить цикл.

Однако при составлении больших сетевых графиков, содержащих тысячи и десятки тысяч событий, гораздо труднее избежать ошибки. И поэтому в таких сетевых графиках появление циклов вероятнее, а заметить их гораздо

сложнее. Ясно, что если график содержит циклы, то все расчеты по изложенным выше алгоритмам становятся бессмысленными. В этом можно убедиться, просчитав для примера небольшой сетевой график, содержащий хотя бы один цикл.

Возникает вопрос, нельзя ли выявить все циклы сетевого графика каким-либо методом, прежде чем проводить его расчет. Положительный ответ на этот вопрос дает излагаемый ниже алгоритм. Заметим, что если сетевой график невелик, то все циклы можно выявить любым методом нумерации событий, например, методом вычеркивания стрелок-работ, который изложен выше. Но если сетевой график содержит тысячи событий, то такое выявление циклов вручную практически невозможно или довольно затруднено. Алгоритм же, приводимый ниже, сравнительно легко позволяет выявить все события, входящие в тот или иной цикл. Если же сетевой график не содержит циклов, то данный алгоритм для каждого события  $P_j$  вычисляет момент  $T_j^{(p)}$ . Перейдем к изложению алгоритма.

**Алгоритм.** Алгоритм состоит из двух этапов.

Этап I. Каждому событию  $P_j$  сетевого графика ставим в соответствие два числа  $\mu_j$  и  $\lambda_j$ . Эти числа в процессе расчета по данному алгоритму постоянно меняются. Каждой работе  $(P_i, P_j)$  поставлена в соответствие ее продолжительность  $t_{ij}$ . Поясним, что означают числа  $\mu_j$  и  $\lambda_j$ . Как и раньше,  $\lambda_j$  означает обнаруженную максимальную длину пути от начального события  $P_0$  до рассматриваемого события  $P_j$ . В начале полагаем все  $\lambda_j = 0$ . Число же  $\mu_j$  показывает количество еще не рассмотренных работ, которые входят в событие  $P_j$ . Очевидно, что в начале  $\mu_j$  равно общему количеству работ, входящих в событие  $P_j$ . Итак, на первом этапе каждой работы  $(P_i, P_j)$  поставлено в соответствие число  $t_{ij}$ , равное продолжительности этой работы, а каждому событию  $P_j$  – два числа:  $\lambda_j = 0$  и  $\mu_j$ , равное количеству всех работ, входящих в  $P_j$ .

Этап II (соответствует общему шагу в других алгоритмах). Этот этап состоит из двух шагов, которые, как правило, приходится повторять несколько раз.

**Шаг 1.** Берем конечное событие  $P_n$ . Ясно, что  $M_n \neq 0$ . Из события  $P_n$  пытаемся построить какой-либо путь в некоторое событие  $P_j$ , у которого  $\mu_j = 0$ . Поскольку  $P_n$  – есть конечное событие, то ясно, что движение от события к событию идет по работам в направлении, противоположном направлению стрелки. Рассмотрим, как можно построить путь от  $P_n$  до  $P_j$ . Это мож-

но сделать, например, так:

а) берем какую-либо еще не рассмотренную работу, которая входит в  $P_n$  (допустим  $(P_{j_1}, P_n)$ ) и отмечаем ее знаком тильда ( $\sim$ );

б) уменьшаем число  $\mu_n$  на единицу;

в) отмечаем событие  $P_n$  знаком звездочка (\*);

г) переходим к событию  $P_{j_1}$ . При этом, если  $\mu_{j_1} = 0$ , то шаг 1 закончен, и переходим к шагу 2. Если же  $\mu_{j_1} \neq 0$ , то переходим к пункту а) первого шага, но вместо события  $P_n$  оперируем событием  $P_{j_1}$ . Пункты а)-г) повторяем.

В результате этих повторений мы либо найдем такое событие  $P_i$ , у которого  $\mu_j = 0$ . В этом случае переходим к шагу 2. Либо придем к какому-нибудь событию  $P_{j_k}$ , которое уже отмечено звездочкой. Это значит, что событие  $P_{j_k}$  уже встречалось в нашей цепочке, что как раз и свидетельствует о наличии цикла. Последовательность отмеченных событий составляет цикл.

Если на сетевом графике обнаружен цикл, то одна из стрелок-работ, входящая в цикл, из графика удаляется. После этого опять каждому событию  $P_j$  ставится в соответствие число  $\mu_j$ , равное количеству работ, входящих в  $P_j$ , кроме  $\mu_j = 0$  тех событий  $P_j$ , которые еще не были отмечены, а все пометки событий и работ стираются. Теперь шаг первый повторяется снова с пункта а).

**Шаг 2.** В соответствии с вышеизложенным, к шагу 2 мы переходим, как только находим такое событие  $P_i$ , у которого  $\mu_j = 0$ . Шаг два состоит из следующих операций:

1) выбираем событие  $P_i$  и по отмеченным работам будем двигаться в направлении стрелок к конечному событию  $P_n$ . Для этого найдем отмеченную работу, которая исходит из события  $P_i$  и по ней перейдем в событие  $P_{j_k}$ , вычисляя при этом  $\lambda'_{j_k}$  по формуле:

$$\lambda'_{j_k} = \max[\lambda_{j_k}; \lambda_i + t_{ij_k}];$$

2) заменим величину  $\lambda_{j_k}$  на  $\lambda'_{j_k}$ . Если окажется, что  $\mu_{j_k} = 0$ , то это значит  $\lambda'_{j_k} = \lambda^*_{j_k} = T^{(p)}_{j_k}$ ;

3) сотрем отметку события  $P_{j_k}$  и отметки всех работ, которые входят в  $P_{j_k}$ . Перейдем к операции 1), только вместо события  $P_i$  теперь будем оперировать событием  $P_{j_k}$ . Операции 1)-3) повторяем. При этом, если окажется, что  $\mu_{j_k} \neq 0$ , то от события  $P_{j_k}$  будем достраивать путь уже по нерассмотрен-

ным работам и против направления стрелок. Другими словами, в этом случае переходим к шагу 1, считая  $P_{jk}$  конечным. Стирать отметки событий и работ не надо.

Процесс расчета по данному алгоритму заканчиваем, если не выявлен цикл сразу, как только все числа  $\mu_j = 0$ . В этом случае получаем все  $\lambda_j = \lambda_j^* = T_j^{(p)}$ .

Если же выявлен цикл, то продолжаем выполнять операции изложенного алгоритма, но не вычисляя величин  $\lambda_j'$ . Все операции в этом случае прекращаем, как только получим, что все  $\mu_j = 0$ .

**Примечание.** Можно показать, что данный алгоритм позволяет вычислять и моменты  $T_j^{(n)}$ , а, следовательно, и находить все критические пути. Для этого необходимо лишь в процессе расчета делать соответствующие пометки, как и в алгоритме 2 отыскания моментов  $T_j^{(n)}$ .

**Пример 3.** Пусть мы имеем сетевой график.

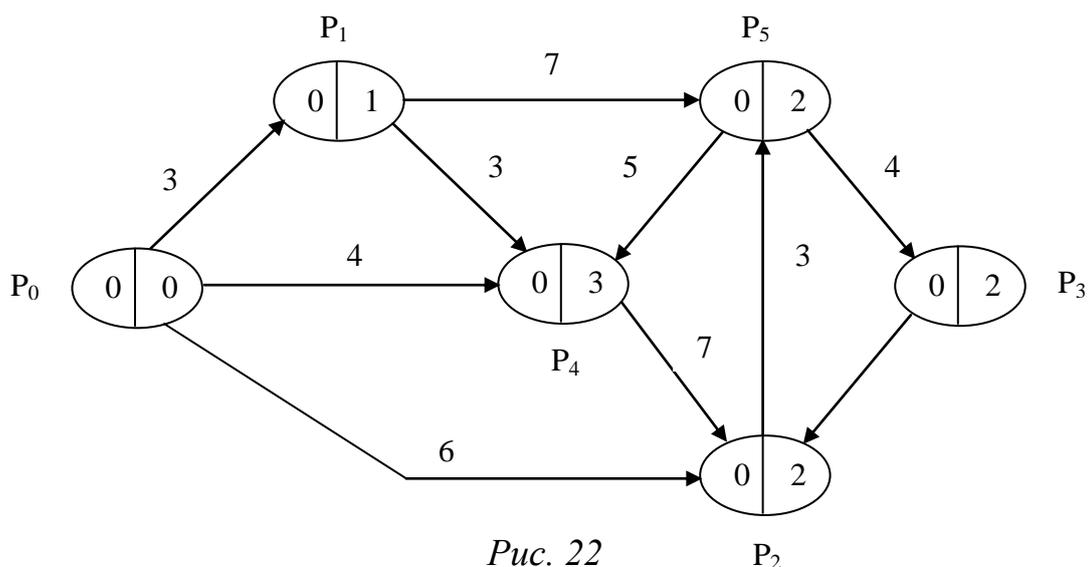


Рис. 22

Проиллюстрируем изложенный алгоритм, проводя вычисления прямо на сетевом графике. Каждый кружок-событие мы разбили пополам и в левую половину записали числа  $\lambda_j = 0$ , а в правую —  $\mu_j$ .

**Шаг 1.** Берем конечное событие  $P_3$ . Имеем  $\mu_3 = 2 \neq 0$ . В  $P_3$  входят две работы  $(P_2P_3)$  и  $(P_5P_3)$ . Обе работы еще не отмечены. Выбираем любую из них, например  $(P_2P_3)$  и отмечаем ее, число  $\mu_3$  уменьшаем на единицу, т. е.  $\mu_3 = 1$ , а событие  $P_3$  отмечаем (\*).

Теперь выполняем те же операции с событием  $P_2$ . Имеем  $\mu_2 = 2 \neq 0$ . В  $P_2$  входят две работы  $(P_0P_2)$  и  $(P_4P_2)$ . Обе они не отмечены. Отмечаем одну из них, скажем  $(P_0P_2)$ ,  $\mu_2$  уменьшаем на единицу и отмечаем  $P_2$ .

Переходим к  $P_0$ . Имеем  $\mu_0 = 0$ . Шаг 1 закончен. Переходим к шагу 2.

**Шаг 2.** Выбираем событие  $P_0$ . У него  $\mu_0 = 0$ . По отмеченной работе  $(P_0P_2)$  из  $P_0$  движемся в  $P_2$  по направлению стрелки. Вычисляем  $\lambda'_2 = \max[\lambda_2; \lambda_0 + t_{02}]$ . Получаем  $\lambda'_2 = \max[0, 6] = 6$ .

В левую половину события  $P_2$  вместо нуля записываем число шесть. Ход расчетов виден из графика. Итак, мы из  $P_0$  пришли в  $P_2^*$ . Но поскольку  $\mu_2 = 1 \neq 0$  (число в правой половине события  $P_2$ ), то переходим к шагу 1, приняв за конечное событие теперь  $P_2$ , т.е. шаг 1 начинаем с события  $P_2$ .

В событие  $P_2$  входит не отмеченная работа  $(P_4P_2)$ . Отмечаем ее: уменьшаем  $\mu_2$  на единицу, т.е.  $\mu_2 = 0$  и отмечаем событие  $P_2$ . В результате переходим к событию  $P_4$ . Имеем  $\mu_4 = 3 \neq 0$ . Одну из входящих в  $P_4$  работ отмечаем. Мы отметили  $(P_1P_4)$ . Уменьшаем  $\mu_4$  на единицу и отмечаем  $P_4$ . Переходим к  $P_1$ , отмечаем, например,  $(P_0P_1)$ , уменьшаем на единицу  $\mu_1$ , отмечаем  $P_1^*$  и попадаем в  $P_0$ . Так как  $\mu_0 = 0$ , опять переходим к шагу 2.

По отмеченной работе  $(P_0P_1)$  возвращаемся в  $P_1$  и записываем в левую часть  $P_1$   $\lambda'_1 = 3$ , ибо  $\lambda'_3 = \max[\lambda_1; \lambda_0 + t_{01}] = \max[0, 3] = 3$ .

Поскольку  $\mu_1 = 0$ , то  $\lambda'_1 = \lambda^*_1 = T^{(p)}_1 = 3$ . Стираем отметку с работы  $(P_0P_1)$  и с события  $P_1$ .

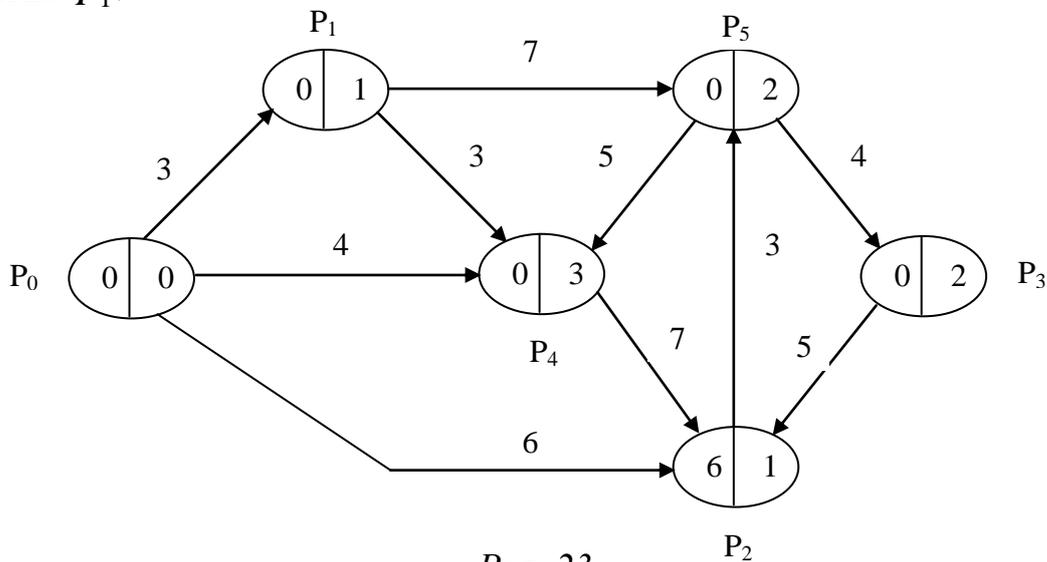
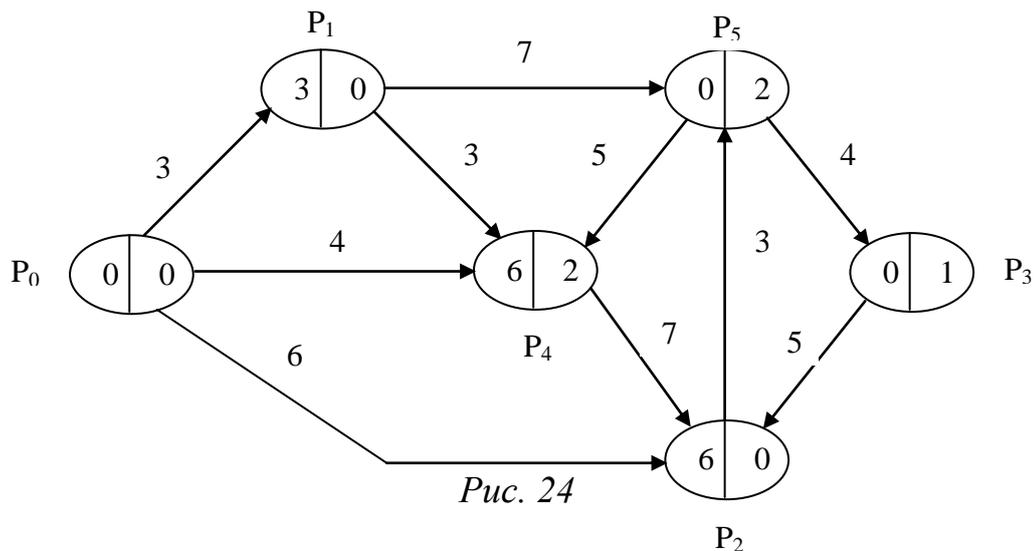


Рис. 23

Теперь аналогично из  $P_1$  по работе  $(P_1P_4)$  переходим в  $P_4$  и вычисляем:

$$\lambda'_4 = \max[\lambda_4; \lambda_1 + t_{14}] = \max[0, 3 + 3] = 6.$$

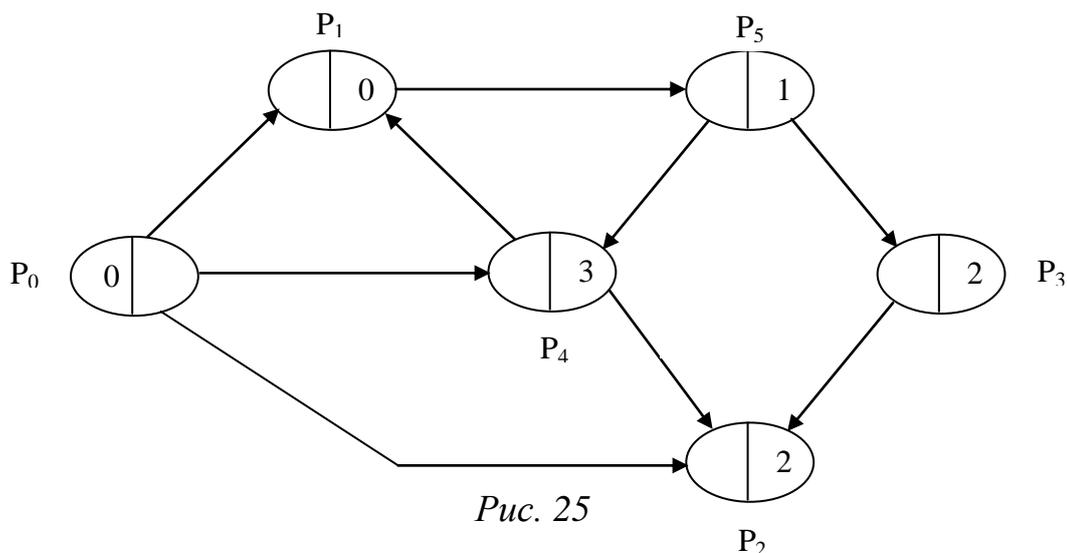
Приводимый ниже рис. 24 хорошо иллюстрирует результат описанных действий.



Итак, мы попали в событие  $P_4$ , у которого  $\mu_4 = 2 \neq 0$ .

Поэтому опять переходим к шагу 1. В событие  $P_4$  входят две не отмеченные работы  $(P_0P_4)$  и  $(P_5P_4)$ . Отметим одну из них, скажем  $(P_5P_4)$ , и вычтем из  $\mu_4$  единицу. По отмеченной работе, двигаясь против направления стрелки, попадаем в событие  $P_5$ . Имеем  $\mu_5 = 2 \neq 0$ . В  $P_5$  входят две не отмеченные работы  $(P_1P_5)$  и  $(P_2P_5)$ . Одну из них опять отмечаем, к примеру  $(P_5P_2)$ . Уменьшаем  $M_5$  на единицу и отмечаем  $P_5$ .

По отмеченной работе  $(P_5P_1)$  переходим в  $P_2$ . Но  $P_1$  уже отмечено, следовательно, выявлен цикл  $P_1P_4P_5P_1$ . Согласно алгоритму, все действия надо начинать сначала, предварительно выкинув из цикла одну из работ, например  $(P_1P_5)$ . В результате получаем новый сетевой график следующего вида.



К этому графику и применяем алгоритмы выявления циклов.

Алгоритм будет закончен, как только все  $\mu_j$  окажутся равными нулю и не будет обнаружено циклов.

### 3. РАСЧЕТ ОПТИМАЛЬНОГО ИСПОЛЬЗОВАНИЯ РЕСУРСОВ

#### 3.1. ОБЩИЕ ЗАМЕЧАНИЯ

Во второй главе мы рассмотрели алгоритмы расчета временных параметров сетевого графика. Как известно, сетевой график представляет собой модель выполнения некоторого комплекса мероприятий, работ или, другими словами, сетевую модель проекта.

Очевидно, что выполнение любой реальной работы требует расхода тех или иных ресурсов (трудовых, материальных, финансовых и т.д.), количество которых, вообще говоря, ограничено. При расчете временных параметров сетевого графика по алгоритмам второй главы, имеющиеся в наличии ресурсы (оборудование, рабочая сила и т.п.) не учитывались или, можно сказать, считались неограниченными.

Однако, поскольку количество ресурсов ограничено, то из-за нехватки ресурсов на тех или иных работах может увеличиться время выполнения проекта. Возникает задача оптимального использования наличных ограниченных ресурсов. Необходимо найти такое распределение ресурсов по работам, которое минимизирует время выполнения проекта или найти такое распределение ресурсов по работам, которое при заданном сроке выполнения проекта минимизирует расход самих ресурсов. Для инженерной и экономической практики планирования и управления строительными процессами задача оптимального распределения ресурсов по сетевым моделям является одной из важнейших задач, так как от успешного решения этой задачи зависит эффективность строительства и широкое внедрение этих моделей в практику планирования и управления. Поэтому методы решения подобных задач, безусловно, относятся к важным методам решения научно-технических задач в строительстве.

К сожалению, до сих пор не найдено полное решение указанных задач. Удовлетворительные результаты – математические модели и методы их решения – достигнуты пока лишь для задач минимизации прямых расходов, связанных с выполнением проекта. Заметим, что все виды ресурсов можно разбить на два класса. К первому классу мы будем относить такие ресурсы, которые, если они поступают и с течением времени не расходуются, то где-то (скажем на складе) накапливаются. Примерами таких ресурсов могут служить сырье и материалы (сталь, уголь, нефть и т.д.). Ко второму классу относятся ресурсы, которые с течением времени не накапливаются, несмотря на то что они не расходовались. Например, неиспользованное располагаемое время работы рабочих, машин, механизмов и т.п.

Удовлетворительные результаты достигнуты пока лишь для задач распределения ресурсов первого класса. Что касается задач оптимального рас-

пределения ресурсов второго класса, то для них не созданы даже подобные модели. Очевидно, что создание и исследование таких моделей сейчас чрезвычайно важно. Более того, до сих пор нет и эффективных алгоритмов решения указанных задач, которые бы позволяли найти истинно оптимальное решение. Разработаны лишь эвристические, приближенные методы. Общая идея эвристических методов заключается в том, что по тем или иным правилам все работы проекта (проектов) пытаются упорядочить (устанавливаются приоритеты работ или очередности их производства) и затем уже производится распределение ресурсов в соответствии с установленным порядком. Приводимый в данной главе алгоритм использует именно эту идею. Ниже будем рассматривать методы решения задач оптимального распределения ресурсов второго класса (например, рабочей силы).

### 3.2. ПОСТАНОВКА ЗАДАЧИ ПРИ ПОСТОЯННЫХ ИНТЕНСИВНОСТЯХ

Пусть нам задан сетевой график некоторого проекта. Все работы проекта могут быть выполнены различными ресурсами. Ежедневный наличный запас  $k$ -го ресурса известен и равен  $A_k(t)$   $k = 1, 2, \dots, n$ . Для каждой работы  $(P_i P_j)$  задана постоянная интенсивность  $r_{ij}^{(k)}$  расхода  $k$ -го вида ресурса. Под интенсивностью расхода ресурса мы будем понимать количество  $k$ -го вида ресурса, которое потребляется на соответствующей работе в единицу времени (день, неделю, месяц и т.д.). По каждой работе  $(P_i P_j)$  известен ее объем в ресурсоединицах (например, в человеко-днях).

При заданном объеме и постоянной интенсивности получена продолжительность  $t_{ij}$  выполнения работы  $(P_i P_j)$ .

Требуется распределить наличные ресурсы по работам так, чтобы общий расход  $k$ -го вида ресурса на все работы в любую единицу времени (день, неделю) не превосходил наличного запаса, а это распределение ресурсов по работам позволяло бы выполнить весь проект за минимальное время. Фактически задача заключается в определении сроков начала и окончания каждой работы таким образом, чтобы выполнение проекта за минимальное время ни в один день не требовало ресурсов больше, чем их имеется в наличии.

Для упрощения решения поставленной задачи примем следующие допущения.

1. Проект выполняется одним видом ресурса, т.е. все работы потребляют для своего выполнения один и тот же вид ресурса ( $n = 1$ ). Примером этому может служить либо некоторый комплексный ресурс, либо, скажем, рабочая сила.

2. Наличный запас ресурса в любой момент времени постоянен и, следовательно, с течением времени не изменяется (количество рабочих в бригаде, цехе и т. д.). Это значит, что  $A(t) = A$ .

3. Интенсивность потребления ресурса  $r_{ij}$  задана и постоянна для каждой работы, т.е. с течением времени не изменяется.

4. Под объемом работы  $(P_i P_j)$  будем понимать величину  $W_{ij} = r_{ij} t_{ij}$ . Общая потребность ресурса за все время выполнения проекта равна  $\frac{1}{A} \sum_{(P_i P_j)} W_{ij}$ . Очевидно, что эта сумма берется по всем работам, которые входят в данный проект.

5. Так как  $A(t) = A$ , то очевидно, что проект не может быть выполнен раньше, чем за время  $\frac{1}{A} \cdot \sum_{(P_i P_j)} W_{ij}$ , т.е. эта величина указывает нижнюю гра-

ницу времени, необходимого для выполнения проекта ( $T_n$ ). Как видим, время  $T_n$  зависит от наличного количества ресурсов. С другой стороны, мы знаем, что  $T_n$  не может быть меньше величины  $T_{kp}$ , вычисляемого без учета ограничения по ресурсам (вычисленного при неограниченных ресурсах). Соединяя все рассуждения воедино; приходим к выводу, что время

$$T_n \geq \max \left\{ T_{kp} \left( \frac{1}{A} \sum_{(P_i P_j)} W_{ij} \right) \right\}.$$

### 3.3. АЛГОРИТМ РЕШЕНИЯ ЗАДАЧИ

Опишем приближенный алгоритм решения следующей задачи: найти оптимальное распределение ресурсов по работам, позволяющее выполнить проект за минимальное время при условии, что ежедневное потребление ресурса не превосходит его наличного запаса. Фактически задача сводится к определению времени начала и окончания каждой работы при тех же условиях. Предполагаем, что все описанные выше допущения имеют место.

Алгоритм состоит из 2 этапов.

На первом этапе составляется такой план выполнения работ, который позволяет ежедневно расходовать ресурсы в количестве, не превосходящем наличный запас. На втором этапе производится уплотнение полученного на первом этапе графика выполнения работ, т.е. производится возможная сдвигка работ на более короткий промежуток времени, не выходя ежедневно за пределы наличного запаса ресурсов.

## Первый этап

Первый этап расчета состоит из двух основных шагов: предварительного и общего.

### *Предварительный шаг*

Этот шаг состоит из ряда последовательных операций.

1. Согласно имеющемуся сетевому графику проекта составляется линейная диаграмма. В этой диаграмме начало всех работ графика устанавливается на самый ранний срок, т.е. начало работы  $(P_i P_j)$  совпадает с моментом  $T_i^{(p)}$  наступления события  $P_i$ .

2. Определяется время выполнения проекта  $T_n = T_{кр}$ , критический путь и полные резервы времени всех работ сетевого графика одним из алгоритмов главы 2.

3. Линейная диаграмма проекта построена в масштабе времени на координатной плоскости. Время откладывается обычно по абсциссе. Начало и конец каждой работы проектируется на временную ось. Очевидно, что самая левая проекция совпадает с моментом начала всех работ, выходящих из начального события  $P_0$ . Обозначим эту проекцию через  $t_0 = T_0^{(p)} = 0$ . Следующую за ней проекцию обозначим через  $t_1$ .

4. В промежутках времени  $(t_0 t_1)$  просматриваем все работы, которые по плану должны выполняться в этом промежутке времени. Другими словами, рассматриваем все работы, которые расположены на диаграмме над промежутком  $(t_0 t_1)$ . Очевидно, что это будут работы  $(P_0 P_j)$ , которые начинаются в событии  $P_0$  и имеют продолжительность  $t_{0j} \geq t_1$ .

5. Работы, расположенные над промежутком  $(t_0 t_1)$ , нумеруются в порядке возрастания их полных резервов времени, т.е. работы упорядочиваются, устанавливается приоритет работ. При этом, если работы имеют одинаковый резерв времени, то присваиваем им номера в порядке убывания их интенсивностей. Если же и интенсивности работ окажутся равными, то можно устанавливать приоритет в порядке возрастания отношения  $r_{ij} / t_{ij}$  или же проще, в произвольном порядке. Одним словом, по каким-либо правилам надо все работы, лежащие над промежутком  $(t_0 t_1)$ , занумеровать.

6. Интенсивности использования ресурсов суммируются в порядке возрастания номеров работ до тех пор, пока получаемая сумма не превзойдет наличного запаса ресурса, указанного для промежутка  $(t_0 t_1)$ .

7. Пока получаемая сумма не превзошла наличного запаса ресурса, работы остаются в прежнем положении. Но как только для какой-то работы

окажется, что после прибавления интенсивности расхода ресурса на той работе полученная сумма превосходит наличный запас ресурса, заданный для промежутка  $(t_0 t_1)$ , то начало этой работы сдвигаем вправо на момент  $t_1$ , т.е. раннее начало этой работы уже устанавливается не на момент  $t_0 = T_0^{(p)}$ , а на некоторый момент  $t_1$ . То же самое делаем со всеми работами, которые еще не рассматривались и лежат над промежутком  $(t_0 t_1)$ , т.е. те работы, номера которых больше номеров рассмотренных работ. Очевидно, что их интенсивности использования ресурса не вошли в полученную сумму потребления ресурса в промежутке  $(t_0 t_1)$ . После этого переходим к общему шагу.

### *Общий шаг*

Предположим, что уже выполнено  $k$ -шагов алгоритма, т.е. рассмотрены работы, расположенные над промежутком  $(t_0 t_k)$ . Суммарный расход ресурса, необходимый для выполнения работ, расположенных над промежутком  $(t_0 t_k)$ , не превышает в любой точке  $(t_0 t_k)$  наличного запаса. Переходим к  $(k+1)$  шагу, где  $k \geq 1$ . Этот переход осуществляется последовательностью следующих операций:

1. Ранним моментом начала оставшейся части проекта, которая состоит из всех работ, расположенных над промежутком  $(t_k, T_n)$ , считаем момент  $t_k$ .

2. Раннее начало каждой, не рассмотренной ранее работы  $(P_i P_j)$ , устанавливаем на момент времени  $t_k$ . Те работы, которые лежат над промежутком  $(t_k, T_n)$ , но начаты раньше (левее) момента  $t_k$ , разумеется, не сдвигаем.

3. С оставшейся частью проекта повторяем операции (3)-(7) предварительного шага. При этом для всех работ, расположенных правее  $t_k$  (не оконченных к моменту  $t_k$ ), по алгоритмам главы 2 определяются новые полные резервы времени.

После того как выполнено проектирование начал и концов всех работ, расположенных над промежутком  $(t_k, T_n)$ , самая левая проекция совпадает с  $t_k$ . Следующую за ней проекцию обозначим через  $t_{k+1}$ .

4. Переходим к пункту 4 предварительного шага, т.е. рассматриваем все работы, расположенные над промежутком  $(t_k, t_{k+1})$ , и выполняем последовательно операции (4)-(7) предварительного шага.

5. Первый этап излагаемого алгоритма закончен полностью, если рассмотрены все работы проекта. В результате получается, что в любой момент времени на промежутке  $(t_0 T_n)$  общий расход ресурса не превосходит его наличного запаса. Получаемое при этом время выполнения проекта  $T_n$ , как было указано выше, конечно, не является минимальным, но, как правило, пред-

ставляет хорошее приближение к минимуму. Фактически на первом этапе строится некоторое допустимое решение поставленной задачи. Причем, это допустимое решение близко к оптимальному. В отдельных случаях практика может вполне устроить даже это допустимое решение.

### **Примечания**

1. На каждом новом промежутке  $(t_k, t_{k+1})$  алгоритм предполагает выявление резервов времени всех работ, расположенных над этим промежутком. Это можно получить одним из алгоритмов отыскания временных характеристик, описанных в главе 2. В случае больших сетей наиболее приемлемым для этой цели является метод потенциалов.

2. В зависимости от конкретных условий задачи установление приоритета работ (нумерация работ), не лежащих над рассматриваемым промежутком, можно осуществлять одним из следующих двух способов.

*а) работы проекта не допускают перерыва в их выполнении.*

В этом случае сначала нумеруются те работы, которые начаты до рассматриваемого промежутка  $(t_k, t_{k+1})$ , т.е. левее момента  $t_k$ , но не закончены к этому моменту. Для каждой начатой работы вычисляется разность между ее новым полным резервом времени и длиной отрезка, который соединяет начало работы с моментом  $t_{k+1}$ . Номера работам присваиваются в порядке возрастания разностей. Если же разности нескольких работ окажутся одинаковыми, то такие работы нумеруем в порядке убывания их интенсивностей и т.д.

После того как пронумерованы все работы, начатые раньше, нумеруем все остальные работы, которые принадлежат  $(t_k, t_{k+1})$ , как и в предварительном шаге. Ясно, что критические работы нумеруются сразу же, как только занумерованы все ранее начатые работы.

*б) работы проекта допускают перерыв в своем выполнении.*

В этом случае нумерация работ осуществляется в порядке возрастания их полных резервов времени, т.е. в точности так же, как и в предварительном шаге. При этом, если промежутку  $(t_k, t_{k+1})$  принадлежит начатая ранее  $t_k$  работа, то часть этой работы, расположенная правее  $t_k$ , рассматривается как самостоятельная работа.

### **Второй этап (уплотнение ресурса)**

Как уже отмечалось выше, на первом этапе строится допустимое решение задачи, т.е. такое решение, которое в каждый момент времени  $t_k \in [t_0, T_n]$  обеспечивает расход ресурса, не превосходящий наличного запаса. Ясно, что время  $T_n$  выполнения проекта, которое получилось в результате выполнения первого этапа, может оказаться и не минимальным. Иногда можно сократить это время за счет так называемого «уплотнения» ресурса.

Суть «уплотнения» состоит в том, что окончательную линейную диаграмму, которая получилась после завершения первого этапа алгоритма, просматривают с конца к началу и все работы сдвигают *слева направо*, если это допускается сетевым графиком и наличием ресурса в рассматриваемый промежуток времени.

На втором этапе опять в зависимости от конкретных условий задачи возможны два случая:

- 1) работы сетевого графика допускают перерыв в их выполнении;
- 2) работы сетевого графика не допускают перерыва в своем выполнении.

Изложение второго этапа алгоритма начнем с того случая, когда работы сетевого графика не допускают перерыва в их выполнении.

В этом случае сдвиг работ слева направо выполняется в следующем порядке.

1. Рассматриваем полученную на первом этапе диаграмму и проектируем опять начало и конец каждой работы на весь временный промежуток  $(T_0, T_n)$ . Выбор промежутков начинаем теперь с конца, т.е. на промежутке  $(T_0, T_n)$  берем самый правый промежуток  $(t_{m-1}, t_m = T_n)$ . Легко понять, что  $t_m$  есть самая правая проекция и поэтому она совпадает с  $T_n$ , а  $t_{m-1}$  – ближайшая к  $T_n$  проекция, расположенная левее  $T_n$ .

2. Подсчитываем общее количество потребного на этом промежутке ресурса. Если окажется, что количество израсходованного ресурса меньше, чем наличный запас, то надо попытаться произвести уплотнение. Для этого надо выполнить последовательно следующие операции.

3. Рассматриваем предпоследний промежуток  $(t_{m-2}, t_{m-1})$ . Выявляем в этом промежутке те работы, которые позволяют, согласно сетевому графику, сдвиг вправо. Это будут все те работы  $(P_i P_j)$ , которые заканчиваются в момент  $t_{m-1}$ , т.е.  $t_{ij}^{(p)} = t_m - 1$ . Эти работы опять нумеруем по какому-либо правилу (например, в порядке возрастания их интенсивностей).

4. Начинаем сдвигать занумерованные работы из промежутка  $(t_{m-2}, t_{m-1})$  в последний промежуток  $(t_{m-1}, t_m = T_n)$  в порядке их номеров таким образом, чтобы их раннее окончание совпало с моментом  $t_m = T_n$ , т.е. сдвигка происходит не по ранним началам, как это было на первом этапе алгоритма, а по окончаниям. При этом, как только работу  $(P_i P_j)$  сдвинем в промежуток  $(t_{m-1}, t_m = T_n)$ , обязательно прибавим к общей сумме расхода ресурса на этом промежутке интенсивность сдвинутой работы  $(P_i P_j)$ .

5. Сдвиг работ из промежутка  $(t_{m-2}, t_{m-1})$  производим до тех пор, пока получаемая сумма расхода ресурса на  $(t_{m-1}, t_m = T_n)$  не превзойдет наличный

запас. В последнем случае сдвиг прекращен, а работу, которая обеспечивает превышение расхода ресурса над наличным запасом, не сдвигаем.

6. После полного просмотра промежутка  $(t_{m-2}, t_{m-1})$  может оказаться, что часть работ этого промежутка сдвинута вправо. В остальной части диаграммы, кроме последнего промежутка, старые проекции могут оказаться нарушенными. Поэтому проектируем снова начала и концы всех оставшихся работ проекта на промежуток  $(T_0, t_{m-1})$ . Самой правой проекцией теперь будет  $t_{m-1}$ , а самым правым промежутком –  $(t'_{m-2}, t_{m-1})$ , где  $t'_{m-2}$  – ближайшая слева к  $t_{m-1}$  проекция. Очевидно, что необязательно  $t'_{m-2}$  равно  $t_{m-2}$ . Ясно, что если на промежуток  $(t_{m-1}, t_m)$  не была сдвинута ни одна работа, то нового проектирования выполнять не требуется, а поэтому  $t'_{m-2} = t_{m-2}$ .

7. После нового проектирования опять рассматриваем все работы, лежащие над предпоследним промежутком, т.е. работы, заканчивающиеся в  $t'_{m-2}$ . Выбираем среди этих работ те, которые сетевой график позволяет сдвигать вправо, и присваиваем им соответствующие номера, как и раньше.

8. Пронумерованные работы в порядке их номеров сдвигаем вправо на промежуток  $(t'_{m-2}, t_{m-1})$ . При этом, если какую-либо работу можно сдвинуть не только в промежуток  $(t'_{m-2}, t_{m-1})$ , но и дальше, т.е. в промежуток  $(t_{m-1}, t_m)$ , то работа сдвигается так, чтобы ее конец совпадал с моментом  $t_m = T_n$  (если конечно наличие ресурса на  $(t_{m-1}, t_m)$  позволяет выполнить этот сдвиг).

9. Процесс проектирования начал и концов работ и сдвижку их вправо описанным выше образом продолжаем до тех пор, пока не просмотрим все работы и все промежутки сетевого графика.

Скорее всего, в результате выполнения второго этапа начало выполнения проекта сместится вправо относительно точки  $T_0$ .

Но так как окончание проекта не изменялось, т.е. равно  $T_n$ , то тем самым уменьшится время выполнения всего проекта, ибо можно начать выполнять проект позднее, не изменив момент его завершения  $T_n$ .

Другими словами, смещение начала выполнения проекта вправо относительно точки  $T_0$  дает возможность либо начать проект в момент  $T_0$  и закончить его раньше момента  $T_n$ , либо закончить в момент  $T_n$ , но начать позднее момента  $T_0$ .

Второй этап алгоритма можно повторить несколько раз, за счет этого можно иногда еще более уплотнить график.

Рассмотрим случай, когда выполнение работы допускает перерыв. Уплотнение ресурса в этом случае может быть более эффективным, так как обязательно всякий раз сдвигать работы лишь соседнего промежутка, а можно

сдвигать вправо часть работы, расположенной в любом промежутке времени, если только это допустимо сетевым графиком и наличием ресурса. Последовательность всех операций та же, что и в предыдущем случае.

1. Проектируем начала и концы всех работ на промежутков  $(T_0, T_n)$ .
2. Подсчитываем расход ресурса в промежутке  $(t_{m-1}, t_m = T_n)$ .
3. Если наличие ресурса позволяет сделать в этом промежутке уплотнение, то находим все работы, входящие в  $P_n$ , и нумеруем их, как и раньше.
4. Определяем, допустим ли перенос части работы  $(P_i P_j)$  на промежуток  $(t_{m-1}, t_m)$ .
5. Если перенос возможен, то переносим часть рассматриваемой работы на  $(t_m, t_{m-1})$ , сдвигая одновременно вправо и начало этой работы на длину отрезка  $(t_m - t_{m-1})$ .
6. После того как мы рассмотрели все пронумерованные работы, переходим к  $(t_{m-2}, t_{m-1})$ . При этом, если окажется, что хотя бы одна работа или ее часть сдвинута вправо, то вновь проектируем все работы на  $(T_0, t_{m-1})$ .
7. Снова нумеруем все работы или оторванные части, которые заканчиваются в  $P_n$  или  $P_j$ , если все выходящие из  $P_j$  работы начинаются не раньше момента  $t_{m-1}$ .

Описанный процесс продолжаем до полного просмотра всей диаграммы.

Изложенный алгоритм легко можно приспособить и в том случае, когда проект выполняется  $n$  видами ресурсов и  $k$  исполнителями. В качестве упражнения читателю предлагается рассчитать по изложенному алгоритму следующий сетевой график для данного вида ресурса (рис. 26).

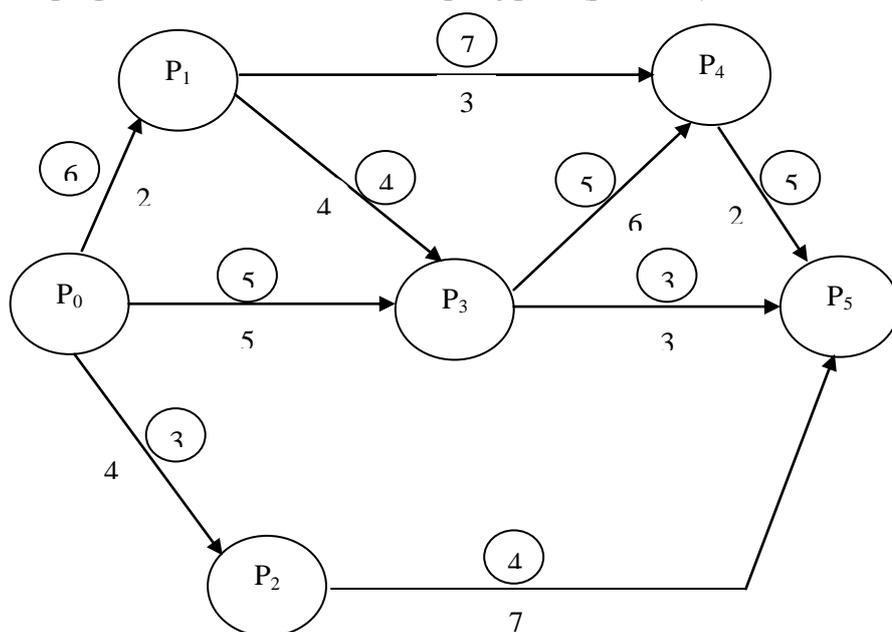


Рис. 26

Здесь числа в кружках, стоящие на стрелках, обозначают интенсивности использования ресурса, а числа под стрелками – соответствующие продолжительности работ.

В заключение следует отметить, что в настоящее время имеются подобные алгоритмы распределения ресурсов по работам сетевого графика и в случае переменных интенсивностей. Задачу распределения ресурсов по работам сетевого графика можно свести к задаче линейного программирования, однако, если количество работ и событий в графике велико, то методы линейного программирования практически трудно реализуемы. В случае расчета сетевого графика по стоимости, в принципе, также можно свести этот расчет к решению задачи параметрического программирования. Необходимо знать лишь зависимость время-стоимость, что, вообще говоря, не всегда просто можно определить.

### 3.4. ОПТИМИЗАЦИЯ СЕТЕВОГО ГРАФИКА ПО СТОИМОСТИ

Обычно эта задача формулируется так: при заданном сроке завершения строительства объекта требуется найти такое распределение средств по работам сетевого графика, которое давало бы минимальные суммарные затраты на выполнение всех работ данного графика. При этом предполагается, что зависимость время-стоимость известна для каждой работы. Ее удобно представлять либо графически, либо выражать в виде формулы. В наиболее простом случае, когда зависимость время - стоимость для каждой работы сетевого графика является линейной. Оптимизация сетевого графика по стоимости формулируется как задача линейного программирования. В случае нелинейной зависимости задачу оптимизации сетевого графика по стоимости приходится либо искусственно сводить к задаче линейного программирования, либо формулировать ее как некую нелинейную задачу.

Смысл зависимости время-стоимость таков: при времени выполнения данной работы  $t_1$  необходимы затраты в размере  $C_2$ , а при  $t_2$ , где  $t_2 > t_1$ , затраты соответственно составят  $C_1 < C_2$ . Тогда величина  $C_{ij}$  для работы  $P_{ij}$ , равная отношению приращения затрат  $(C_2 - C_1)$  к величине сокращения продолжительности выполнения работы  $(t_2 - t_1)$  показывает, на сколько снижаются затраты с увеличением времени выполнения данной работы на 1 день по сравнению с минимальной продолжительностью выполнения данной работы. В задаче линейного программирования данную величину следует брать со знаком минус, как выражающую экономию затрат на единицу прироста времени выполнения данной работы. Можно рассуждать и иначе, а именно; величина  $C_{ij}$  есть коэффициент наращивания затрат, который показывает размер прироста затрат, вызываемого сокращением продолжительности работы

$P_{ij}$  на 1 день по сравнению с максимально возможной продолжительностью выполнения данной работы. В этом случае коэффициент наращивания затрат берем со знаком плюс. Эта величина определяется для каждой работы сетевого графика.

Кроме  $C_{ij}$  введен еще ряд обозначений:

$T_{ij}$  и  $t'_{ij}$  – соответственно наибольшая и наименьшая продолжительность;

$t_{ij}$  – искомая продолжительность работы;

$T_j^p$  – ранний момент свершения событий;

$T_{дир}$  – директивный срок.

При таких обозначениях задача оптимизации сетевого графика по стоимости формулируется как следующая задача линейного программирования. Определить минимум затрат на выполнение всего комплекса работ сетевого графика

$$f(t) = \sum_i \sum_j C_{ij} t_{ij},$$

где  $\sum_i \sum_j$  берется по всем работам сетевого графика при условии, что

$$t_i^p + t_{ij} \leq t_j^p \quad \text{для всех } i, j;$$

$$t_{ij} \leq T_{ij} \quad \text{для всех } i, j;$$

$$t_{ij} \geq t'_{ij} \geq 0 \quad \text{для всех } i, j;$$

$$T_i^p - T_l^p \leq T_{дир}.$$

Как видим, данная сетевая модель представляет модель линейного программирования и может быть решена симплекс-методом, который широко представлен в интернет и информационных технологиях. Однако с учетом специфики модели, задача эта может быть решена и более эффективными способами теории потоков.

Решение задачи состоит из следующих этапов.

I этап. Строится сетевой график разработки.

II этап. Определяются для каждой работы минимальная продолжительность  $t'_{ij}$  и соответствующие ей затраты  $C'_{ij}$ , максимальная продолжительность  $T_{ij}$  и соответствующие ей затраты  $C_{ij}$ , а также директивный срок разработки  $T_{дир}$ .

III этап. Предполагается, что продолжительность каждой работы равна ее максимальной величине  $t_{ij} = T_{ij}$ . Значит, данная разработка будет выполнена с наименьшими затратами.

IV этап. Рассчитываются временные параметры сетевого графика, и если  $T_{кр} \leq T_{дир}$ , то задача решена. Если же  $T_{кр} > T_{дир}$ , то переходят к пятому этапу.

V этап. Выбираются работы критического пути, и для каждой из них определяются зависимость время-стоимость и соответствующее ей  $C_{ij}$  (графически или любым другим способом).

VI этап. Из полученных  $C_{ij}$  критических работ выбирается наименьшая. Пусть  $C_{i_l j_k} = \min(C_{ij})$ , т.е. наименьшее отношение соответствует работе  $P_{i_l j_k}$ . Это значит, что сокращение критического пути на 1 день на работе  $P_{i_l j_k}$  удорожает разработку в наименьшей степени по сравнению с другими работами критического пути.

VII этап. Продолжительность работы  $P_{i_l j_k}$  минимальная, т. е.

$t_{i_l j_k} = t'_{i_l j_k}$ . В этом случае возможно одно из следующих неравенств:

$$1. T_{i_l j_k} - t'_{i_l j_k} < T_{кр} - T_{дир} . \quad (3.1)$$

$$2. T_{i_l j_k} - t'_{i_l j_k} \geq T_{кр} - T_{дир} . \quad (3.2)$$

В случае неравенства 3.1 задача решена. А при наличии неравенства (3.2) переходят к VIII этапу.

VIII этап. Проверяется изменение критического пути. Если критический путь изменился, то вычисление повторяют с V этапа, а если нет, то переходят к этапу IX.

IX этап. Исключается из числа критических работа  $P_{i_l j_k}$  и переходят к этапу VI.

При продолжении процесса через некоторое число повторений находится решение, удовлетворяющее условию  $T_{кр} \leq T_{дир}$ . Оно и является оптимальным.

Практические расчеты реальных сетевых графиков вручную по изложенному методу целесообразнее производить в таблицах.

## 4. ДРУГИЕ ПРИЛОЖЕНИЯ СЕТЕВОГО МОДЕЛИРОВАНИЯ ТЕХНОЛОГИЧЕСКИХ ПРОЦЕССОВ В ВОДОСНАБЖЕНИИ И ВОДООТВЕДЕНИИ

### 4.1. ПОСТРОЕНИЕ КРАТЧАЙШЕЙ СВЯЗЫВАЮЩЕЙ СЕТИ ВОДОСНАБЖЕНИЯ

Выше мы указывали, что при проектировании линейных сооружений сетей водоснабжения и водоотведения возникают сложные многовариантные технико-экономические задачи, связанные с оптимизацией затрат времени и финансовых средств. Капитальные вложения на строительство линейных сооружений и затраты при их эксплуатации существенно зависят от суммарной протяжённости всей сети. Чем меньше общая протяжённость сети, тем меньше капитальных вложений потребуется на строительство линейных сооружений сети, а сети меньшей протяжённости при их эксплуатации потребуют меньше текущих затрат. Поэтому задача проектировщика связать заданные пункты водоснабжения и водоотведения сетью наименьшей протяженности в конкретных условиях. Рассмотрим простейший алгоритм определения кратчайшей связывающей все узлы сети на условном примере.

Пусть имеются  $n$  пунктов  $A_1, A_2, \dots, A_n$ , расстояния между которыми известны и их удобно представить квадратной симметричной матрицей  $\|a\|$  (таблица 10).

Таблица 10

Пункты	$A_1$	$A_2$	...	$A_{n-1}$	$A_n$
$A_1$		$a_{12}$	...	$a_{1(n-1)}$	$a_{1n}$
$A_2$	$a_{21}$		...	$a_{2(n-1)}$	$a_{2n}$
...	...	...	...	...	...
$A_n$	$a_{n1}$	$a_{n2}$	...	$a_{n(n-1)}$	$a_{nn}$

**Пункты требуется связать прямолинейными отрезками, образующими звенья единой сети, так, чтобы суммарная длина всей сети была минимальной.**

Сеть должна обладать следующим свойством: для каждого пункта существует путь, соединяющий его с любым другим пунктом.

Эта задача всегда имеет оптимальное решение среди огромного множества допустимых решений, но для ее решения нельзя использовать широко известные методы математического программирования. В более простом случае без использования промежуточных точек вне данных пунктов для решения задачи целесообразно использовать методы сетевого моделирования.

Прежде чем приступить к описанию одного из способов решения данной задачи, *введём некоторые определения.*

1. Точку назовём изолированной, если она не соединена звеном ни с какой другой точкой.

2. Часть точек, связанных звеньями, назовём фрагментом.

3. Расстоянием точки от фрагмента будем считать минимальное расстояние от этой точки до любой точки фрагмента.

### Суть задачи и алгоритма ее решения покажем на условном примере

Требуется спроектировать трассу водоотводящей сети для населённых пунктов А,Б,С,Д,Е,Ф,Г. Расположение пунктов на карте показано на рисунке 27. Расстояния между пунктами в км приведены в таблице 11.

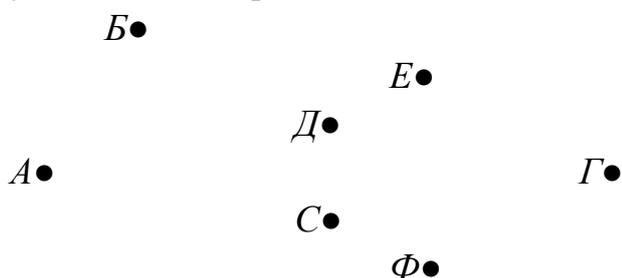


Рис. 27

В основу алгоритма решения данной задачи можно положить следующие **постулаты здравого смысла:**

1) каждая изолированная точка соединяется звеном с ближайшей точкой или фрагментом;

2) каждый фрагмент соединяется с ближайшей изолированной точкой или фрагментом.

3) построение связывающей все узлы сети считается законченным, если все точки будут соединены в единую сеть.

Подобного рода задачи возникают при проектировании не только водопроводной или водоотводящей сети, но и линий электропередач, сетей газопроводов или шоссейных дорог и так далее.

Таблица 11

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х	19	16	17	25	36	32
Б	19	Х	18	15	27	28	39
С	16	18	Х	5	11	17	23
Д	17	15	5	Х	7	18	26
Е	25	27	11	7	Х	13	21
Ф	36	28	17	18	13	Х	33
Г	32	39	23	26	21	35	Х

В силу симметричности значений таблицы относительно главной диагонали для решения задачи достаточно использовать только нижнюю часть таблицы 11, которая представлена в таблице 12

Таблица 12

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х						
Б	19	Х					
С	16	18	Х				
Д	17	15	<b>[5]</b>	Х			
Е	25	27	11	7	Х		
Ф	36	28	17	18	13	Х	
Г	32	39	23	26	21	35	Х

### Процесс решения состоит в следующих действиях

**Шаг 1.** В таблице 12 выбираем элемент с минимальным значением. Это элемент строки Д и столбца С, равный 5 км. Пункты С и Д соединяем звеном сети (рис. 28), а элемент  $a$  в таблице 12 выделяем жирным шрифтом.

**Шаг 2.** Из невыделенных элементов в таблице 12 опять выбираем элемент с минимальным значением. Он находится в строке Е и столбце Д, равный 7 км. Выделим его жирным шрифтом и преобразуем таблицу 12 в таблицу 13.

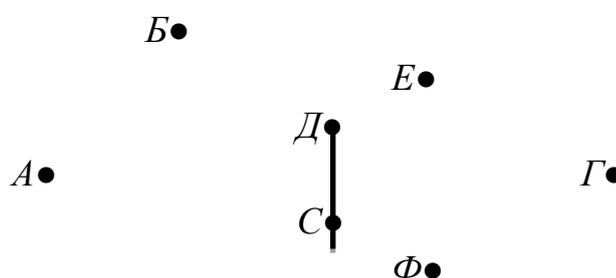


Рис. 28

Пункты Д и Е соединяем звеном сети (рис. 29), получим фрагмент, состоящий из пунктов С, Д и Е, точки которого связаны между собой. Пункты С и Е связаны сетью фрагмента, следовательно они не должны соединяться непосредственно, поэтому элемент строки Е и столбца С ( $a_{EC}$ ) (или клетка ЕС) в таблице 13 зачёркивается.

Таблица 13

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х						
Б	19	Х					
С	16	18	Х				
Д	17	15	[5]	Х			
Е	25	27	<del>11</del>	<b>[7]</b>	Х		
Ф	36	28	17	18	13	Х	
Г	32	39	23	26	21	35	Х

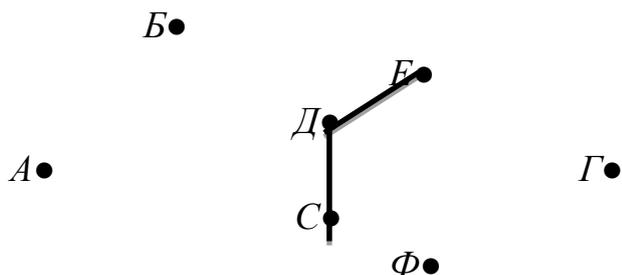


Рис. 29

**Шаг 3.** Повторяем шаг 2, но минимальный элемент выбираем из числа невыделенных и незачёркнутых элементов таблицы. Таким элементом является элемент  $a_{DB}$ . Этот элемент выделяется (см. табл. 14).

Таблица 14

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х						
Б	19	Х					
С	16	18	Х				
Д	17	<b>[15]</b>	[5]	Х			
Е	25	27	11	[7]	Х		
Ф	36	28	17	18	13	Х	
Г	32	39	23	26	21	35	Х

Пункты  $B$  и  $D$  соединяются друг с другом (рисунок 30).

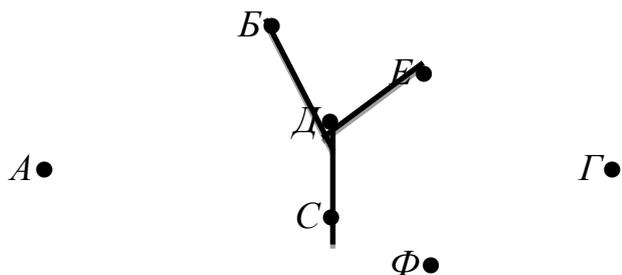


Рис. 30

Как видно из рис. 30, в фрагменте из пунктов  $B$ ,  $C$ ,  $D$  и  $E$  имеются транзитные связи  $B$  с  $C$  и  $E$ . Прямые связи между этими пунктами не нужны, соответствующие клетки должны быть зачёркнуты. Получим таблицу 15, минимальным элементом в которой является  $a_{CA}=16$ . В таблице 15 этот элемент выделяется. Пункты  $A$  и  $C$  соединяются (рис. 31). Поскольку имеется транзитная связь пункта  $A$  с пунктами  $B$ ,  $D$  и  $E$  и его прямые связи с этими пунктами нецелесообразны, элементы  $a_{BA}$ ,  $a_{DA}$ ,  $a_{EA}$  зачёркиваются.

Таблица 15

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х						
Б	19	Х					
С	[16]	18	Х				
Д	17	[15]	[5]	Х			
Е	25	27	11	[7]	Х		
Ф	36	28	17	18	13	Х	
Г	32	39	23	26	21	35	Х

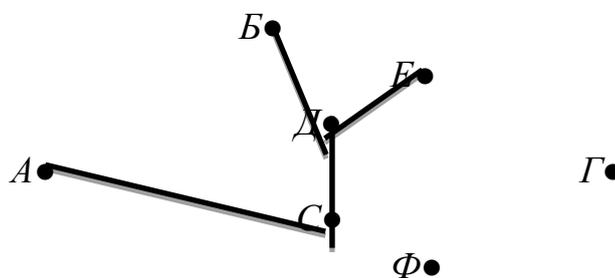


Рис. 31

Среди невыделенных и незачёркнутых элементов минимальным является  $a_{\Phi E} = 13$ . В таблице 16 этот элемент выделен. После соединения пунктов Е и Ф вид сети показан на рис.32.

Таблица 16

Пункты	А	Б	С	Д	Е	Ф	Г
А	Х						
Б	19	Х					
С	[16]	18	Х				
Д	17	[15]	[5]	Х			
Е	25	27	11	[7]	Х		
Ф	36	28	17	18	[13]	Х	
Г	32	39	23	26	21	35	Х

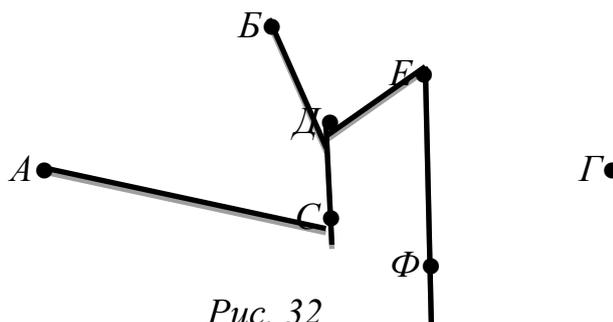


Рис. 32

Так как пункт F не должен иметь прямые связи с пунктами A, B, C и D, элементы  $a_{FA}$ ,  $a_{FB}$ ,  $a_{FC}$  и  $a_{FD}$  зачёркиваются. Теперь в табл. 17 минимальным элементом является  $a_{FE}=21$ . Соединяя пункты E и F, получаем сеть, представленную на рисунке 33, связывающую все семь пунктов. Элемент EG выделяется, а все другие элементы перечёркиваются. Матрица, соответствующая окончательному решению, показана в таблице 17.

Таблица 17

Пункты	A	B	C	Д	Е	Ф	Г
A	X						
B	19	X					
C	[16]	18	X				
Д	17	[15]	[5]	X			
Е	25	27	11	[7]	X		
Ф	36	28	17	18	[13]	X	
Г	32	39	23	26	[21]	35	X

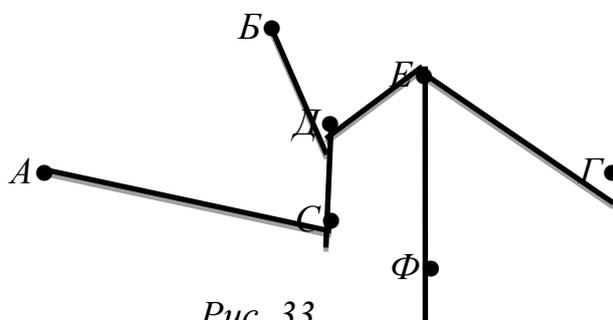


Рис. 33

Общая протяжённость сети (Z) равна сумме выделенных элементов в последней таблице 17.

$$Z = AC + BD + CD + DE + EF + EG = 16 + 15 + 5 + 7 + 13 + 21 = 77 \text{ км.}$$

## 4.2. ОПТИМИЗАЦИЯ ОЧЕРЕДНОСТИ СТРОИТЕЛЬСТВА СВЯЗЫВАЮЩЕЙ СЕТИ МЕТОДОМ ДЖОНСОНА [12]

В работе [12] С.М. Джонсон рассматривает многооперационную задачу, сформулированную Р. Беллманом следующим образом:

«Имеется  $n$  изделий, которые должны пройти обработку сначала на одной операции (на одном станке), а затем - на второй. Каждую операцию можно выполнить только на одном станке. Одновременно на одном станке может обрабатываться только одно изделие. Возьмем  $2n$  констант  $A_i$  и  $B_{ii}=1,2, \dots, n$ . Все они положительны. Пусть  $A_i$  сумма подготовительно-заключительного времени и оперативного времени для  $i$ -го изделия на первом станке, а  $B_i$  – на втором. Ищем оптимальный вариант плана запуска изделий в производство, имея в виду минимизировать суммарную длительность их обработки». При  $n$  изделиях существует  $n!$  вариантов последовательностей, число которых с ростом  $n$  быстро возрастает. При 5 изделиях оно равно 120, а при 10 изделиях уже 3628800 вариантов. На практике число изделий может быть значительно больше 10.

С.М. Джонсону удалось обосновать простой алгоритм оптимизации запуска в обработку  $n$  изделий, если они проходят последовательно две или при особых условиях три технологических операции. Следуя С.М. Джонсону, изложим суть алгоритма для двух- и трехоперационного календарного плана, а затем используем его при решении наших проблем.

### 1. Двухоперационный календарный план

Алгоритм основан на следующих доказанных в работе утверждениях.

**Лемма 1** [11, с. 33]. Последовательность запуска изделий в производство на любом станке может быть изменена так, что она совпадает с последовательностью на другом станке, без увеличения времени выполнения плана. Размещая по шкале времени трудоемкости операций  $A$  и  $B$  таким образом, чтобы график удовлетворял исходным условиям, то есть чтобы начало операции  $B_j$  было справа от конца соответствующего ему  $A_j$ , автор получает следующее соотношение  $\min(A_j, B_{j+1}) < \min(A_{j+1}, B_j)$  (\*), которое и дает возможность сформулировать простое правило выбора решения, позволяющее составить в данном случае оптимальный календарный план с минимальной календарной длительностью выполнения всей совокупности намечаемых работ.

Алгоритм Джонсона состоит из следующих шагов [12, с. 36-37]:

1. Записать  $A$  и  $B$  в колонки таблицы, в первый столбец которой записаны номера изделий, во второй - длительности их обработки на первом станке (величины  $A_i$ ), в третий - длительности их обработки на втором станке (величины  $B_i$ ).

2. Просмотреть все длительности обработки изделий на обоих станках и найти среди них наименьшую.

3. Если она относится к первому станку (то есть  $A_i$ ), расположить соответствующее изделие как можно левее в ряду последовательности обработки изделий.

4. Если она относится ко второму станку (то есть  $B_i$ ), расположить соответствующее изделие можно правее в ряду последовательности обработки изделий.

5. Вычеркнуть строку, относящуюся к этому изделию. В результате чего получим преобразованную таблицу.

6. Повторить шаги 2-5 в отношении преобразованной таблицы, двигаясь с обоих концов к середине последовательности запуска изделий в обработку.

7. Если попадаются равные числа, то для определенности изделие с меньшим номером располагается первым, в случае равенства  $A_i$  и  $B_i$  порядок изделий устанавливается в соответствии с  $A$ .

**Пример.** По плану застройки микрорайона необходимо запроектировать прокладку семи участков канализационной сети, на каждом из которых необходимо выполнить земляные и монтажные работы.

Продолжительность выполнения работ указана в таблице 18.

Таблице 18

**Продолжительность выполнения работ на участках**

Участок	Продолжительность работы (сут)	
	Земляные работы	Монтажные работы
А	10	11
Б	13	18
С	19	10
Г	14	12
Д	11	13
Е	16	24
Ф	11	9

Определить последовательность прокладки сетей на участках, обеспечивающую минимум затрат времени на выполнение всего комплекса работ. По алгоритму Беллмана - Джонсона получаем следующее решение (табл. 19):

Таблица 19

**Оптимальная очередность участков производства работ**

№ работы	1	2	3	4	5	6	7
участок	А	Е	Б	Д	Г	С	Ф

Получена последовательность работ с наименьшим сроком выполнения всех работ на всех 7 участках, равным 121 дню. Определение раннего срока начала и окончания каждого вида работ на каждом участке и продолжительности выполнения всего комплекса работ можно рассчитать на основе сетевого графика. Производство работ на участках следует отобразить в оптимальной последовательности, полагая время перехода спецподразделений с участка на участок равным нулю, либо реальным затратам времени на перебазировку. Пример построения подобного сетевого графика и продолжительность выполнения всего комплекса работ на основе расчёта асинхронностей изложена в пункте 4.3.

Предлагаем студентам проверить, является ли оптимальным, согласно таблице 19, решение следующей задачи: определить последовательность из 20 участков, на которых запроектировано производство земляных и монтажных работ.

Продолжительность выполнения работ приведена в таблице 20.

Таблица 20

**Продолжительность выполнения работ на участках**

Участок	Продолжительность работы (сут)	
	Земляные работы	Монтажные работы
А	10	11
Б	13	18
В	19	10
Г	14	8
Д	11	10
Е	16	24
Ж	17	19
З	9	10
И	4	6
К	18	17
Л	20	22
М	21	25
Н	12	14
О	13	16
П	25	24
Р	27	26
С	13	12
Т	8	9
У	7	6
Ф	5	4

Таблица 21

**Очередность участков производства работ**

№ ра- боты	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
уча- сток	И	Т	З	А	Н	Б	О	Е	Ж	Л	М	Р	П	К	С	Д	В	Г	У	Ф

Являются ли 314 дней наименьшим временем выполнения работ?

## 2. Трехоперационный календарный план

Если для прокладки водопроводной или канализационной сети нужно выполнить три вида работ А, В и С на каждом участке, то задача уже не столь проста, как при двух работах. Джонсону удалось доказать следующую теорему.

**Теорема** [12, с. 40]. Если  $\min A_i$  больше или равен  $\max B_i$ ,  $i=1,2, \dots, n$ , то оптимальный трехоперационный план получается по следующему правилу:

Изделие  $i$  предшествует изделию  $j$ , если выполняется условие (\*\*).

$$\min(A_i + B_j, C_j + B_j) < \min(A_j + B_j, C_i + B_i) (**).$$

Если имеет место равенство в соотношении (\*\*), то оба изделия индифферентны и допускают любое их размещение, совместимое с последовательностью, установленной на основании неравенств.

Если  $(A_i + B_i)$  заменить одним столбцом со значениями нового столбца **A**, величины которого как и в двухоперационной задаче обозначать просто  $A_i$ , а  $(C_j + B_j)$  на новый столбец **B** с величинами  $B_j$ , то для случаев, когда выполняется одно из приведенных ниже соотношений (\*\*\*), общее решение задачи находится аналогично двухоперационной задаче:

$$\min A_i \text{ больше или равен } \max B_i \text{ или } \min C_i \text{ больше или равен } \max B_j (***)$$

Короче, если столбцы А и В сложить и результат записать в новую таблицу в столбец, соответствующий первой операции новой таблицы (т.е. вместо двух столбцов А и В записать в новую таблицу один столбец (А+В), а столбцы В и С тоже сложить и результат записать в новую таблицу в столбец, соответствующий второй операции (во второй столбец таблицы (В+С)), то получим преобразованную таблицу, аналогичную двухоперационной задаче. Если хотя бы одно из сформулированных выше соотношений выполняется, то алгоритм Джонсона для двухоперационного плана обеспечивает оптимальную последовательность и при трехоперационном плане, но примененный к преобразованной таблице.

### Пример

Требуется проложить канализационную сеть из 5 участков, на которых надо выполнить последовательно три вида работ, продолжительности выполнения которых указаны в таблице 22.

Таблица 22

**Продолжительность выполнения работ**

Участок	Продолжительность работы(сут )		
	Разбивка на оси	Разработка грунта	Укладка труб
А	3	7	9
Б	2	5	10
В	4	8	12
Г	6	9	11

Д	5	6	13
---	---	---	----

Складываем продолжительности смежных работ и результат заносим в таблицу 23.

Таблица 23

#### Общая продолжительность попарно смежных работ

Участок	Продолжительность работы (сут)	
	Разбивка на оси + разработка грунта	Разработка грунта + укладка труб
А	10	16
Б	7	15
В	12	20
Г	15	20
Д	11	19

Используя алгоритм Джонсона, получаем следующую последовательность:

Таблица 24

#### Оптимальная последовательность участков для производства работ

№ работы	1	2	3	4	5
Участок	Б	А	Д	В	Г

Наименьшее время выполнения работ составит 60 дней.

### 4.3. ОПТИМИЗАЦИЯ ОЧЕРЕДНОСТИ СТРОИТЕЛЬСТВА СВЯЗЫВАЮЩЕЙ СЕТИ МЕТОДОМ ПСП

Для определения оптимальной очередности строительства объектов сети, когда число работ более двух, разработаны точные методы, основанные на идеях динамического программирования и метода ветвей и границ, обеспечивающие оптимальное решение задачи, но они требуют от разработчика высокой квалификации в области методов оптимизации и связаны с высокой трудоемкостью расчетов и с применением компьютеров. Практические работники часто не располагают знаниями, средствами и временем поиска точного оптимального решения.

Поэтому, наряду с точными методами, развивались и менее трудоемкие, но весьма эффективные приближенные методы решения данной задачи [10, 14]. Эти методы обеспечивают возможность решения серьезных задач даже вручную. Определение рациональной очередности возведения объектов одним из таких методов [10] рассмотрим на примере.

**Постановка задачи.** В районе строительства проектируется возвести «К»=6 участков сети водоводов. При производстве работ на каждом участке

сети необходимо выполнить последовательно «N»=4 видов работ. Каждый вид работ выполняется специальным подразделением постоянной мощности. Продолжительность  $T_{kp}$  выполнения всего объема  $p$ -го вида работ на  $k$ -м объекте задана и приведена ниже в таблице 25. Требуется определить последовательность возведения сетей на участках, которая обеспечивает наименьшие сроки строительства всего комплекса водоводов. Для эффективной последовательности рассчитать время начала и окончания работ на каждом объекте и общую продолжительность строительства комплекса в целом.

Простой алгоритм субоптимального решения данной задачи рассмотрим сначала в общем виде, а потом применим его к конкретным расчетам на основе данных таблицы 25.

### Алгоритм решения задачи

**Шаг 1.** Построим матрицу  $T = \{T_{kp}\}$ .

**Шаг 2.** Для каждого объекта с номером «к» рассчитаем оценку  $C_k$  по следующим формулам:

$$C_k = \sum_{p=1}^{\frac{N-1}{2}} \left( \frac{N+1}{2} - p \right) \cdot T_{kp} + \sum_{p=\frac{N+3}{2}}^N \left( \frac{N+5}{2} - p \right) \cdot T_{kp} \quad (4.1)$$

-для нечетного N;

$$C_k = \sum_{p=1}^{N/2} \left( \frac{N+2}{2} - p \right) \cdot T_{kp} + \sum_{p=\frac{N+2}{2}}^N \left( \frac{N}{2} - p \right) \cdot T_{kp} \quad (4.2)$$

-для четных N.

**Шаг 3.** Упорядочим систему оценок по возрастанию величины  $C_k$  и получим последовательность  $C$  оценок  $C_k$  вида

$$C = \{C_{k1}, C_{k2}, \dots, C_{kk}\} . \quad (4.3)$$

**Шаг 4.** В соответствии с (4.3) выпишем индексы:

$$K_1, K_2, \dots, K_{kk} . \quad (4.4)$$

**Шаг 5.** Вычислим продолжительность строительства  $T_{k_k}$  последнего объекта с номером  $k$  в последовательности (4.4) по формуле

$$T_{k_k} = \sum_{p=1}^N T_{k_k p} \quad (4.5)$$

$T_{k_k p}$  - продолжительность выполнения  $p$ -го вида работ на том же объекте.

**Шаг 6.** Для каждой пары объектов последовательности (4.4) рассчитаем асинхронности  $a_{k_{k+1}}$  по формуле

$$a_{\kappa_{k+1}} = \max \left[ \sum_{p=1}^N T_{\kappa p} - \sum_{p=0}^{N-1} T_{\kappa+1,p} \right], \quad (4.6)$$

где  $a_{\kappa_{k+1}}$  - асинхронность между смежными объектами последовательности (4.4), имеющими номера  $k$  и  $k+1$  соответственно при  $T_{\kappa+1,0} = 0$ .

**Шаг 7.** Вычислим наименьшую продолжительность строительства комплекса «К» объектов  $T_{\Pi}$  в последовательности (4.4) по формуле

$$T_{\Pi} = \sum_{\kappa=\kappa_1}^{\kappa_{k-1}} a_{\kappa_{k+1}} + T_{\kappa_{\kappa}}. \quad (4.7)$$

**Шаг 8.** На основе (4.4) и данных матрицы «Т» можно построить и рассчитать сетевой график строительства комплекса. Из этих расчетов получим моменты времени начала и окончания (ранние и поздние) строительства каждого объекта комплекса и выполнения каждого вида работ на каждом объекте, а также резервы времени. А привязав расчеты к календарю, получим ту же информацию в реальных сроках. Все указанные расчеты выполняются известными методами сетевого планирования.

Покажем теперь работу данного алгоритма решения задачи на основании таблицы 25.

Таблица 25

**Продолжительность выполнения работ на объектах (в днях)**

Вид работ / Объект	Рытье котлована	Монтаж трубопровода	Монтаж оборудования	Обратная засыпка
А	$T_{11}=28$	$T_{12}=25$	$T_{13}=4$	$T_{14}=21$
Б	$T_{21}=26$	$T_{22}=12$	$T_{23}=5$	$T_{24}=19$
С	$T_{31}=20$	$T_{32}=8$	$T_{33}=2$	$T_{34}=13$
Д	$T_{41}=23$	$T_{42}=10$	$T_{43}=3$	$T_{44}=16$
Е	$T_{51}=25$	$T_{52}=14$	$T_{53}=6$	$T_{54}=18$
Ф	$T_{61}=35$	$T_{62}=24$	$T_{63}=7$	$T_{64}=28$

Необходимые расчеты выполним в два этапа. Сначала определим оптимальную (субоптимальную) очередность участков, на которых следует выполнять работы, и минимальную продолжительность их производства на всех участках, а затем построим сетевой график для оптимальной последовательности работ по участкам и рассчитаем ранние сроки начала и окончания выполнения каждой работы на каждом участке.

**Этап 1.** Определим последовательность строительства сетей, которая обеспечивает минимальную продолжительность строительства всего комплекса водоводов. При шести объектах строительства существует 720 вариантов последовательностей их возведения, требуется выбрать ту из них, которой

соответствует минимальная продолжительность строительства комплекса из 6 участков сети. Для решения этой проблемы применим изложенный алгоритм.

**Шаг 1.** Строим матрицу  $T$ . Поскольку таблица 25 - это и есть матрица  $T$ , то отдельно выписывать ее не будем.

**Шаг 2.** Для каждого объекта с номером « $k$ » рассчитаем оценки  $C_k$  по формуле (4.2), так как количество видов работ есть нечетное число. Поскольку  $p=6$ , то формула (4.2) для нашей задачи примет вид:

$$C_k = \sum_{p=1}^2 (3-p) \cdot T_{kp} + \sum_{p=3}^4 (2-p) \cdot T_{kp} \quad (4.1')$$

$$C_1 = 28 \cdot 2 + 1 \cdot 25 - 1 \cdot 4 - 2 \cdot 21 = 35 \text{ дней}$$

$$C_2 = 26 \cdot 2 + 12 \cdot 1 - 5 \cdot 1 - 19 \cdot 2 = 21 \text{ дней}$$

$$C_3 = 20 \cdot 2 + 8 \cdot 1 - 2 \cdot 1 - 13 \cdot 2 = 20 \text{ дней}$$

$$C_4 = 23 \cdot 2 + 10 \cdot 1 - 3 \cdot 1 - 16 \cdot 2 = 21 \text{ дней}$$

$$C_5 = 25 \cdot 2 + 14 \cdot 1 - 6 \cdot 1 - 18 \cdot 2 = 22 \text{ дней}$$

$$C_6 = 35 \cdot 2 + 24 \cdot 1 - 7 \cdot 1 - 28 \cdot 2 = 31 \text{ дней}$$

**Шаг 3.** Упорядочим эти оценки в днях по возрастанию и получим последовательность  $C = C_3; C_2; C_4; C_5; C_6; C_1$ .

**Шаг 4.** Выпишем индексы оценок  $C_k$  в последовательности  $C$  и получим следующую оптимальную очередность строительства наших зданий 3-2-4-5-6-1 и последовательности объектов  $C - Б - Д - Е - Ф - А$ .

**Шаг 5.** Рассчитаем продолжительность строительства объекта  $A$ , т.е. последнего объекта в оптимальной последовательности (4.4), по формуле (4.5). Для этого просто просуммируем продолжительность выполнения работ на данном объекте.

$$T_B = 28 + 25 + 4 + 21 = 78.$$

Следовательно, строительство здания «А» будет продолжаться 78 дней.

**Шаг 6.** Рассчитаем асинхронности между парами смежных объектов по формуле (4.6).

Результаты расчетов асинхронностей приведены ниже:

$$a_{CB} = \max \{ 20 - 0 = 20; 28 - 23 = 5; 30 - 33 = -3; 43 - 36 = 7;$$

$$a_{БД} = \max ( 26 - 0 = 26; 38 - 23 = 15; 43 - 33 = 10; 62 - 36 = 26;$$

$$a_{ДЕ} = \max ( 23 - 0 = 23; 33 - 25 = 8; 36 - 39 = -3; 52 - 45 = 7;$$

$$a_{ЕФ} = \max ( 25 - 0 = 25; 39 - 35 = 4; 45 - 59 = -14; 63 - 66 = -3;$$

$$a_{ФА} = \max ( 35 - 0 = 35; 59 - 28 = 31; 66 - 53 = 13; 94 - 57 = 37.$$

**Шаг 7.** Вычислим общую продолжительность строительства наших объектов  $T_{\Pi}$  - в оптимальной последовательности по формуле (4.7).

$$T_{\Pi} = a_{CB} + a_{BD} + a_{DE} + a_{E\Phi} + a_{\Phi A} + T_A = 20 + 26 + 23 + 25 + 37 + 78 = 209.$$

Итак, минимальная продолжительность строительства водопроводных сетей составит 209 дней, если их строить по оптимальному варианту в последовательности С - Б - Д - Е - Ф - А (по варианту 1).

**Этап 2.** Строим сетевой график для варианта 1 и выполним расчет ранних характеристик этого графика.

Сетевой график реализации варианта 1 (рис.34) и таблица 26 с показателями расчета ранних характеристик сетевого графика в днях для варианта 1 приведены ниже. Из этих документов видно, что минимальный срок строительства всего комплекса равен 209 дней, что подтверждает правильность оценки этого срока на седьмом шаге изложенного выше алгоритма.

Любая другая последовательность строительства данных объектов приведет к увеличению общей продолжительности строительства комплекса. В данном примере в этом можно убедиться непосредственной проверкой. Например, продолжительность А - Б - С - Д - Е - Ф составит 214 дней.

#### **4.4. ЗАДАЧА О МАКСИМАЛЬНОМ ПОТОКЕ ИЛИ О ПРОПУСКНОЙ СПОСОБНОСТИ СЕТИ**

Одной из особенностей организации и планирования производственных процессов в системах водоснабжения и особенно водоотведения является неравномерность поступления потоков жидкости в разные периоды эксплуатации соответствующих сетей.

В связи с этим расчёт мощности (пропускной способности) труб в проектах должен проводиться с учетом данных часов максимального и минимального поступления жидкости. Однако, если рассчитать мощность оборудования и пропускную способность каналов на максимально возможную нагрузку, то это приведёт к значительному снижению коэффициентов использования оборудования, что, естественно, тоже нежелательно. Поэтому существующие методы расчёта оборудования и основаны на определении некоего усредненного соотношения между нагрузкой и пропускной способностью.

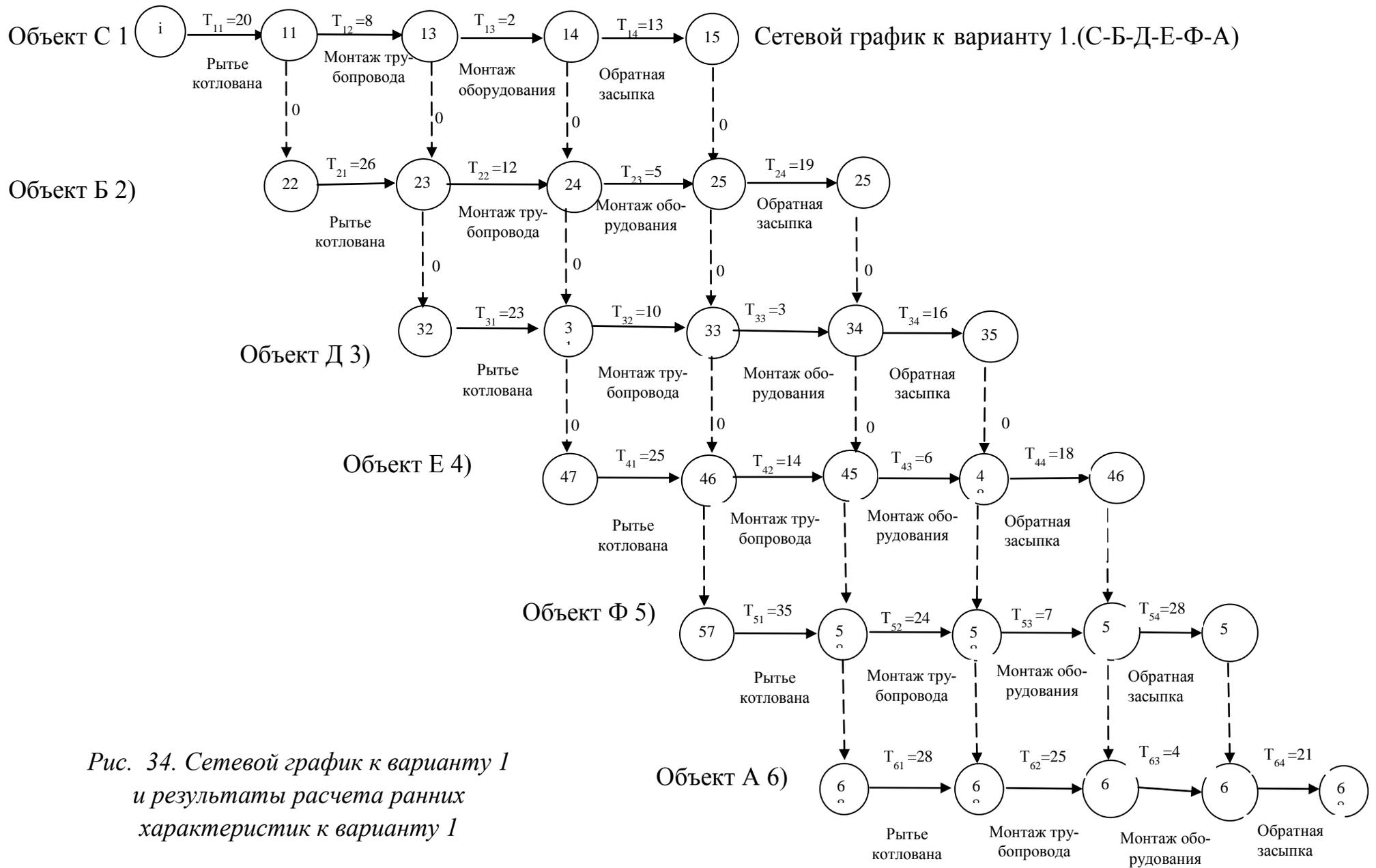


Рис. 34. Сетевой график к варианту 1 и результаты расчета ранних характеристик к варианту 1

Таблица 26

**Результаты расчета ранних характеристик сетевого графика в днях для варианта 1**

Работы	Объект	Начало работы	Конец работы	Объект	Начало работы	Конец работы	Объект	Начало работы	Конец работы	Объект	Начало работы	Конец работы	Объект	Начало работы	Конец работы	Объект	Начало работы	Конец работы
Рытье котлована	С	0	20	Б	20	46	Д	46	69	Е	69	94	Ф	94	129	А	129	157
Монтаж трубопровода	С	20	28	Б	46	58	Д	69	79	Е	94	108	Ф	129	153	А	157	182
Монтаж оборудования	С	28	30	Б	58	63	Д	79	82	Е	108	114	Ф	153	160	А	182	186
Обратная засыпка	С	30	43	Б	63	82	Д	82	98	Е	132	132	Ф	160	188	А	186	209

Это, однако, не исключает необходимость определения максимально возможной пропускной способности сети. Ведь в периоды резкого увеличения поступления, например, ливневых стоков канализационные сети могут не справляться с возлагаемыми на них задачами. А это может приводить к многим аварийным ситуациям вплоть до чрезвычайных. Поэтому важно знать максимальную пропускную способность сети и уж потом соотносить ожидаемые затраты и результаты. Подобного рода задачи возникают и при определении пропускной способности водопроводной сети городов. В периоды предпраздничных и праздничных дней, массовых отпусков и т. п. потребление воды резко возрастает. Возрастает и нагрузка на сети. Следовательно, опять возникает необходимость решения задачи по определению часов максимального и минимального водопотребления. Более того, в ряде случаев водопроводная или водоотводящая сеть может быть представлена в форме последовательных участков, некоторые из которых составляют сами по себе относительно сложную схему. Поэтому даже в пределах этого сложного участка может быть найдено несколько вариантов схемы подключения. И, конечно, интересен вариант с максимальной пропускной способностью данного участка сети. Наконец, общая задача заключается в отыскании такой схемы соединения участков сети, при которой не нарушаются соединения внутри участка и в то же время обеспечивается возможность максимального прохождения жидкости через данный участок.

Определение максимального потока в условиях сложных схем представляет собой трудную задачу, математическая модель которой в принципе может быть сформулирована в терминах общей задачи линейного программирования, а ее решение найдено общими методами, например симплексным методом или методом ветвей и границ. Однако решение подобного типа задач этими методами связано с проведением большого числа трудоёмких расчётов и требует много времени.

Значительно проще эти задачи решаются с помощью сетевого моделирования и специальных более простых алгоритмов.

Рассмотрим всю последовательность расчётов на примере сети, состоящей из восьми узлов сети ( $P_0, P_1, P_2, P_3, P_4, P_5, P_6$  и  $P_7$ ). Взаимная связь узлов друг с дру-

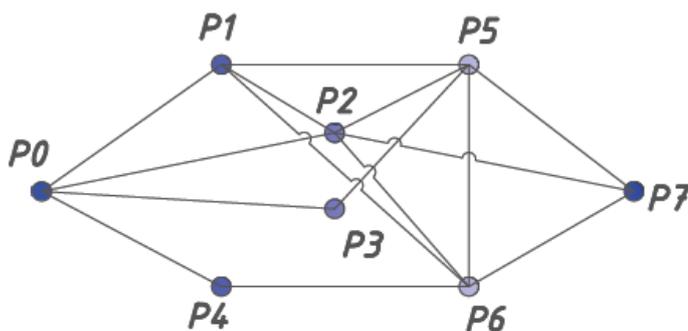


Рис. 35

гом и пропускная способность отдельных звеньев в единицу времени показаны на рис. 35.

Требуется определить максимальный поток из  $P_0$  в  $P_7$ , используя для этого все возможные участки сети.

Условия задачи оформляются таблицей, форма которой показана ниже (табл. 27).

Таблица 27

Узлы	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_0$		$a_{01}$	$a_{02}$	$a_{03}$	$a_{04}$	$a_{05}$	$a_{06}$	$a_{07}$
$P_1$	$a_{10}$		$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$	$a_{17}$
$P_2$	$a_{20}$	$a_{21}$		$a_{23}$	$a_{24}$	$a_{25}$	$a_{26}$	$a_{27}$
$P_3$	$a_{30}$	$a_{31}$	$a_{32}$		$a_{34}$	$a_{35}$	$a_{36}$	$a_{37}$
$P_4$	$a_{40}$	$a_{41}$	$a_{42}$	$a_{43}$		$a_{45}$	$a_{46}$	$a_{47}$
$P_5$	$a_{50}$	$a_{51}$	$a_{52}$	$a_{53}$	$a_{54}$		$a_{56}$	$a_{57}$
$P_6$	$a_{60}$	$a_{61}$	$a_{63}$	$a_{63}$	$a_{64}$	$a_{65}$		$a_{67}$
$P_7$	$a_{70}$	$a_{71}$	$a_{72}$	$a_{73}$	$a_{74}$	$a_{75}$	$a_{76}$	

В клетках таблицы  $a_{ij}$  означает пропускную способность звена между узлами  $P_i$  и  $P_j$ , а  $a_{ji}$  – пропускную способность звена в обратном направлении (из  $P_j$  в  $P_i$ ). Если  $a_{ij} > 0$ , а  $a_{ji} = 0$ , то в клетку  $a_{ji}$  следует записать нуль. Если  $a_{ij} = 0$  и  $a_{ji} = 0$ , то клетки  $ij$  и  $ji$  не заполняются.

Пользуясь данными рис. 35, заполним исходную таблицу (табл. 28).

Таблица 28

Узлы	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
0		3	4 <sup>-</sup>	3	2			
$P_1$	2		3			2	2	
$P_2$	3 <sup>+</sup>	2				3	4 <sup>-</sup>	3
$P_3$	3				1	4		
$P_4$	5			1			2	
$P_5$		3	3	0				4
$P_6$		2	4 <sup>+</sup>		1	2		4 <sup>-</sup>
$P_7$			3			5	4 <sup>-</sup>	

Выберем произвольно один из возможных путей из узла  $P_0$  в узел  $P_7$ . Допустим, что этот путь —  $(P_0, P_2, P_6, P_7)$ . Тогда  $a_{02}$ ,  $a_{26}$  и  $a_{67}$  отмечаем знаками « $\leftarrow$ », а  $a_{20}$ ,  $a_{62}$  и  $a_{76}$  — знаками « $+$ ».

Определяем пропускную способность выбранного пути  $O_i$ :

$$O_i = \min |a_{02}, a_{26}, a_{67}| = \min (4; 4; 4) = 4.$$

Вычитая  $O_i$  из  $a_{02}$ ,  $a_{26}$  и  $a_{67}$  и прибавляя к  $a_{20}$ ,  $a_{62}$  и  $a_{76}$ , получаем новую таблицу (табл. 29).

Таблица 29

Узлы	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>
P <sub>0</sub>		3 <sup>-</sup>	0	3	2			
P <sub>1</sub>	2 <sup>+</sup>		3			2 <sup>-</sup>	2	
P <sub>2</sub>	7	2				3	0	3
P <sub>3</sub>	3				1	4		
P <sub>4</sub>	5			1			2	
P <sub>5</sub>		3 <sup>+</sup>	3	0			0	4 <sup>-</sup>
P <sub>6</sub>		2	8		1	2		0
P <sub>7</sub>			3			5 <sup>+</sup>	8	

Выбираем новый путь из P<sub>0</sub> к P<sub>7</sub>. Таким путём может быть (P<sub>0</sub>, P<sub>1</sub>, P<sub>6</sub>, P<sub>7</sub>). Отмечаем a<sub>01</sub>, a<sub>15</sub> и a<sub>57</sub> знаком «—» и a<sub>10</sub>, a<sub>51</sub> и a<sub>75</sub> знаком «+».

Определяем пропускную способность пути O<sub>2</sub> = min {a<sub>01</sub>, a<sub>15</sub>, a<sub>67</sub>} = min (3; 2; 4) = 2.

Вычитая O<sub>2</sub> из элементов таблицы, отмеченных знаками (—), и прибавляя к элементам, отмеченным знаком (+), составим новую таблицу (табл. 30).

Таблица 30

Узлы	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>
P <sub>0</sub>		1	0	3 <sup>-</sup>	2			
P <sub>1</sub>	4		3			0	2	
P <sub>2</sub>	7	2				3	0	3
P <sub>3</sub>	3 <sup>+</sup>				1	4 <sup>-</sup>		
P <sub>4</sub>	5			1			2	
P <sub>5</sub>		5	3	0 <sup>+</sup>			0	2 <sup>-</sup>
P <sub>6</sub>		2	8		1	2		0
P <sub>7</sub>			3			7 <sup>+</sup>	8	

По таблице 30 выбираем путь (P<sub>0</sub>, P<sub>3</sub>, P<sub>5</sub>, P<sub>7</sub>). O<sub>3</sub> = min [a<sub>03</sub>; a<sub>35</sub>; a<sub>67</sub>] = min (3; 4; 2) = 2. Затем O<sub>3</sub> вычитаем из a<sub>03</sub>, a<sub>35</sub> и a<sub>57</sub> и добавляем к a<sub>30</sub>, a<sub>53</sub> и a<sub>75</sub>. Составим новую таблицу (табл. 31).

Таблица 31

Узлы	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>
P <sub>0</sub>		1 <sup>-</sup>	0	1	2			
P <sub>1</sub>	4 <sup>+</sup>		3 <sup>-</sup>			0	2	
P <sub>2</sub>	7	2 <sup>+</sup>				3	0	3 <sup>-</sup>
P <sub>3</sub>	5				1	2		
P <sub>4</sub>	5			1			2	
P <sub>5</sub>		5	3	2			0	0
P <sub>6</sub>		2	8		1	2		0
P <sub>7</sub>			3 <sup>+</sup>			9	8	

Рассмотрим путь  $(P_0; P_1; P_2; P_7)$ .  $O_4 = \min(a_{01}; a_{12}; a_{27}) = \min|1; 3; 3| = 1$ .  
 Переходим к новой таблице (табл. 32).

Таблица 32

Узлы	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_0$		0	0	$1^-$	2			
$P_1$	5		2			0	2	
$P_2$	7	3				$3^+$	0	$2^-$
$P_3$	$5^+$				1	$2^-$		
$P_4$	5			1			2	
$P_5$		5	$3^-$	$2^+$			0	0
$P_6$		2	8		1	2		0
$P_7$			$4^+$			9	8	

Имеется путь  $(P_0; P_3; P_5; P_2; P_7)$ .  $O_5 = \min|a_{03}; a_{35}; a_{52}; a_{27}| = \min(2; 3; 2) = 1$ . Составим следующую таблицу (табл. 33).

Таблица 33

Узлы	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_0$		0	0	0	$2^-$			
$P_1$	5		2			0	2	
$P_2$	7	3				4	$0^+$	$1^-$
$P_3$	6				1	1		
$P_4$	$5^+$			1			$2^-$	
$P_5$		5	2	3			0	0
$P_6$		2	$8^-$		$1^+$	2		0
$P_7$			$5^+$			9	8	

Имеется путь  $(P_0; P_3; P_5; P_2; P_7)$ .  $O_5 = \min(a_{04}; a_{46}; a_{62}; a_{27}) = \min(2; 2; 8; 1) = 1$ .  
 Составим новую таблицу (табл. 34).

Таблица 34

Узлы	$P_0$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$
$P_0$		0	0	0	1			
$P_1$	5		2			0	2	
$P_2$	7	3				4	1	0
$P_3$	6				1	1		
$P_4$	6			1			1	
$P_5$		5	2	3			0	0
$P_6$		2	7		2	2		0
$P_7$			6			9	8	

В столбце  $P_7$  имеются только нули. Это говорит о невозможности построения нового пути из  $P_0$  к  $P_7$ .

Вычитая из элементов первой таблицы элементы последней, получим итоговую таблицу со значениями  $x_{ij}$ , определяющими максимально возможный поток.

В итоговую таблицу записываются только положительные разности (см. табл. 35).

Таблица 35

Узлы	P <sub>0</sub>	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>
P <sub>0</sub>		3	4	3	1			
P <sub>1</sub>			1			2		
P <sub>2</sub>							3	3
P <sub>3</sub>						3		
P <sub>4</sub>							1	
P <sub>5</sub>			1					4
P <sub>6</sub>								4
P <sub>7</sub>								

Задача решена. Максимальный поток равен  $x_{01} + x_{02} + x_{03} + x_{04} = 3 + 4 + 3 + 1 = 11$  (из строки P<sub>0</sub>) или  $x_{27} + x_{57} + x_{67} = 3 + 4 + 4 = 11$  (из столбца P<sub>7</sub>).

Величина потока, исходящего из начального узла (P<sub>0</sub>), совпадает с величиной потока, поступающего в конечный узел (P<sub>7</sub>).

Для любого другого узла поступающий различными путями поток численно равен исходящему потоку из этого же узла (сумма цифр одноименных строки и столбца любого промежуточного узла равны между собой). В этом легко убедиться из рис.36 и табл. 35, отражающих схему прохождения максимального потока согласно полученному решению. Для всех звеньев сети  $a_{ij} \geq X_{ij}$ .

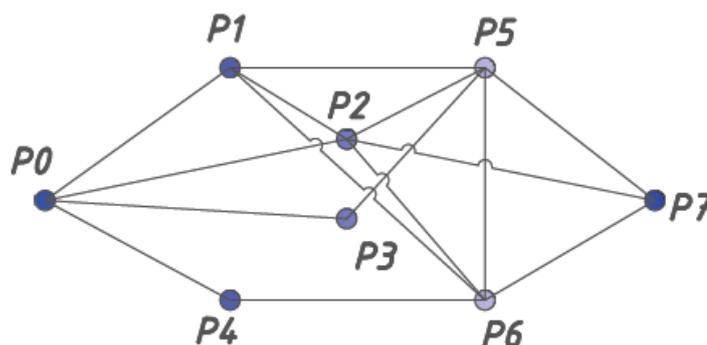


Рис. 36

## 4.5. КОНЦЕПЦИЯ КРИТЕРИЯ ФИНАНСОВО-ЭКОНОМИЧЕСКИХ РЕЗУЛЬТАТОВ И МЕТОДИКА ОЦЕНКИ МАТЕРИАЛЬНЫХ ЗАТРАТ ПРЕДПРИЯТИЯ<sup>1</sup>

### 4.5.1. Концепция критерия финансово-экономических результатов

Строительство относится к инвестиционным, материалоемким отраслям и видам деятельности. Поэтому оценка эффективности строительной организации и оценка материальных затрат должны быть одним из приоритетных направлений разработки и принятия решений инженерно-экономическими службами организации. Значит, концепции критерия, методы и алгоритмы соответствующей оценки должны входить в арсенал современных методов решения научно-технических задач как в строительстве вообще, так и на предприятиях водоснабжения и водоотведения, в частности.

Рассмотрим наиболее важные, на наш взгляд, проблемы относительно оценки эффективности строительного производства.

Финансово-экономический результат деятельности предприятия интересует ряд участников его распределения, к которым, прежде всего, относятся собственники капитала (акционеры), работники предприятия, кредиторы, государство и т.п. С точки зрения системного подхода любую организацию можно считать устойчиво эффективно работающей только в том случае, когда реализуются общие цели всех заинтересованных участников производства, а совокупность целей является целевым критерием для всей системы. Наличие в системе конфликтующих целей и целевых ориентиров - один из основных факторов нестабильности производственной организации [22]. Так, на предприятиях присутствуют, как правило, различные группы экономических интересов, связанные с различными субъектами: собственниками, менеджерами, работниками. Это затрудняет эффективное управление предприятием и обеспечение его устойчивости. Поэтому в теории менеджмента одно из центральных мест занимает координация разнонаправленных целей в измерении результата участников хозяйственной деятельности и нахождение баланса между индивидуальными целями участников и общей целью.

По вопросу иерархии интересов участников хозяйственной деятельности в литературе существует два подхода: *американский и евразийский*.

В соответствии с американским подходом только собственники организации и высшие менеджеры, выполняющие их волю, имеют право принимать ответственные управленческие решения. Влияние других сторон проявляется косвенно. В моделях, построенных по американскому принципу, основным показателем экономического результата является чистая прибыль, которая отражает интересы его собственников как «конечных получателей интересов».

---

<sup>1</sup> Параграф подготовлен совместно с к.э.н. Д.Л. Усовым.

Интересы других участников учитываются как вынужденные, или как выплаты, уменьшающие стоимостный интерес акционеров. Очевидно, что такой узконаправленный «учет» интересов участников, реально создающих стоимость хозяйственной деятельности, содержит в себе множество антистимулов повышения эффективности деятельности фирмы. Так, увеличение финансового результата - чистой прибыли для акционеров - может быть достигнуто за счет доходов других участников-претендентов: например, выплата минимальной зарплаты работникам или неуплата налогов. Но такая политика может давать лишь кратковременный результат. К сожалению, в России сейчас при формировании отчета о финансовом результате компании базовым показателем текущего финансового результата хозяйствования продолжает оставаться прибыль, т.е. доминирует узкий подход главенства прибыли для акционеров, главенства интересов собственников и игнорирование интересов непосредственных работников, незаинтересованность в труде и повышении его эффективности.

С позиции долгосрочной перспективы максимальный устойчивый результат может быть достигнут только при наиболее полном соблюдении интересов всех участников хозяйственной деятельности, что и предусматривает евразийский подход. В развитых экономиках Европы и Азии менеджеры придают большее значение всей стоимости, на которую претендуют и другие заинтересованные стороны, в особенности рабочая сила. В этих странах максимальное повышение стоимости в интересах частного акционера считается близорукой, неэффективной, упрощенной и даже антиобщественной политикой [22].

С точки зрения полного учета интересов всех участников таким показателем выступает выручка от реализации продукта. Именно она является финансовым источником удовлетворения интересов всех участников процесса. Но выручка не может характеризовать эффективность деятельности фирмы, так как включает в себя материализованные результаты деятельности других предприятий в форме материальных затрат. В то же время критерий эффективности должен аккумулировать в себе финансовые интересы системообразующих участников процесса производства: наемных работников и собственников. Их экономические интересы выражены в величине вновь созданной стоимости. С помощью применяемых средств производства наемные работники непосредственно создают новую стоимость. Собственники же определяют способы и методы ее создания.

Остальные носители интересов участвуют в формировании или распределении стоимости опосредованно и потому являются внешними участниками (контрагентами). Это относится и к государству. Свой финансовый интерес оно определяет через систему налогового законодательства и реализует его в форме принудительного изъятия части вновь созданной стоимости через сис-

тему корпоративных налогов и налогов на доходы физических лиц. Значит добавленная стоимость отражает результат труда коллектива предприятия, полученный в процессе производства сверх затрат и распределяемый между трудовым коллективом, собственником и обществом в целом. Показатель вновь созданной (добавленной) стоимости потенциально несет в себе пока что не используемую в экономической теории и практике возможность надежно отслеживать основные результаты деятельности предприятий и направлять их деятельность в нужное обществу русло [22].

Вновь создаваемая стоимость может и должна определяться на уровне государства, государственно-территориальных образований, но и отдельных хозяйствующих субъектов, их объединений.

Вновь созданная (добавленная) стоимость представляет собой величину

$$ДС = B - M, \quad (4.8)$$

где ДС – добавленная стоимость;

$B$  – выручка от реализации произведенной продукции, услуг;

$M$  – полные материальные затраты на производство и реализацию произведенной продукции и услуг, представляющие прошлый овеществленный труд (потребленные материалы, сырье и комплектующие изделия, энергия, затраты транспорта, амортизация основных средств, услуги сторонних организаций и др.).

Ориентация на добавленную стоимость ведет к снижению рисков производства при управлении отдельно взятой организацией. Покажем это на примере формул оценки эффективности производственной программы производства (количества продукции или услуг):

$$N_{\Pi} = \frac{C_t}{W - C_p}, \quad (4.9)$$

$$N_{ДС} = \frac{M_t}{W - M_p}, \quad (4.10)$$

где  $N_{\Pi}$ ,  $N_{ДС}$  – оптимальная производственная программа соответственно по критерию прибыли и добавленной стоимости;

$W$  – выручка от реализации продукции (услуг);

$C_t$  – общая сумма постоянных затрат на единицу продукции (услуг);

$C_p$  – общая сумма переменных затрат на единицу продукции (услуг).

$M_t$  – сумма овеществленных (приобретенных) постоянных затрат на единицу продукции (услуг);

$M_p$  – сумма овеществленных (приобретенных) переменных затрат на единицу продукции (услуг).

Если сравнить формулу для определения производственной программы самоокупаемости по критерию вновь созданной стоимости (4.10) с формулой (4.9), которая позволяет решить ту же задачу, но по критерию прибыли, то можно обнаружить следующее. Во-первых, по своей конструкции и структуре обе формулы одинаковы. Во-вторых, производственная программа самоокупаемости по критерию вновь созданной стоимости  $K_{ДС}$  всегда будет меньше аналогичной программы по критерию прибыли предприятия  $K_{П}$ . Это объясняется тем, что числитель у первого критерия меньше, чем у второго, ибо постоянные затраты прошлого труда представляют только какую-то часть полных постоянных затрат. Кроме того, знаменатель у первого критерия больше знаменателя второго по той же причине. Следовательно, и устойчивость производства в случае, когда организация работает по критерию вновь созданной стоимости, будет значительно выше, чем при оценке по критерию прибыли.

Можно математически это доказать [22]. Возьмем отношение производственных программ самоокупаемости по двум рассматриваемым критериям и проанализируем полученный результат:

$$\frac{N_{П}}{N_{ДС}} = \frac{C_t(W - M_p)}{M_t(W - C_p)} = \frac{C_t}{M_t} \cdot \frac{W - M_p}{W - C_p} = K_3 \cdot K_D > 1, \quad (4.11)$$

где  $K_3$  – коэффициент, учитывающий структуру постоянных затрат:

$$K_3 = \frac{C_t}{M_t} > 1;$$

$K_D$  – коэффициент, учитывающий структуру маржинального дохода:

$$K_D = \frac{W - M_p}{W - C_p} > 1.$$

Оба коэффициента всегда больше единицы, а их произведение больше каждого из сомножителей. Следовательно, поскольку  $N_{ДС} < N_{П}$ , устойчивость деятельности, построенной по критерию добавленной стоимости, всегда выше, а риск убыточности всегда меньше, чем аналогичные характеристики у производства, которое функционирует по критерию прибыли.

Таким образом, добавленная стоимость как показатель, учитывающий интересы основных участников воспроизводства – собственников, наемных работников и государства - приобретает важное значение для социального прогресса [22], а, следовательно, и расчет материальных затрат становится актуальным.

#### 4.5.2 Методика оценки материальных затрат предприятия

Итак, важнейшим показателем эффективности предприятия является объем добавленной стоимости, представляющий разность между выручкой (стоимостным объемом производства) и материальными затратами. Как видим, определение объема добавленной стоимости предприятия основано на оценке материальных затрат, которые можно получить по официальным данным учета и отчетности. Систему учета материальных затрат предприятия, к примеру, для двух видов продукции с точки зрения сетевого моделирования можно отобразить следующей сетью с обратными связями (рис.37).

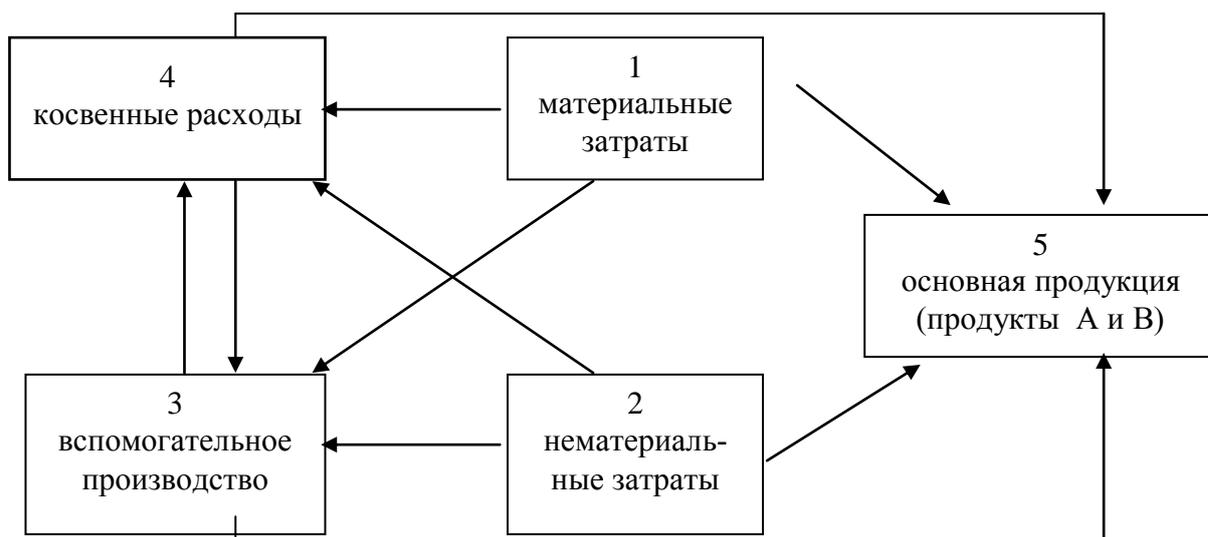


Рис. 37. Сеть бухгалтерского учета производственных затрат

Компоненты 1 и 2 представляют собой укрупненные элементы затрат. В состав материальных затрат входят как затраты на покупные сырье, материалы, полуфабрикаты, топливо, энергию, транспортные и другие материальные услуги, так и амортизация основных фондов. В нематериальные затраты входят заработная плата с отчислениями на социальное страхование и разные денежные расходы.

Компоненты 3 и 4 – промежуточные: они отображают формирование и перераспределение первичных затрат. При оценке материальных затрат по предприятию в целом следует суммировать все выходные потоки сети, т.е. подсчитать все выходные обороты компонента 1: Применительно к рисунку 37 получим

$M = x_{13} + x_{14} + x_{15}$ , или в общем виде по формуле (4.12), объем добавленной стоимости по формуле (4.13).

$$M = \sum_j x_{1j}. \quad (4.12)$$

$$ДС = В - М, \quad (4.13)$$

где В – выручка;

М – материальные затраты;

ДС – добавленная стоимость.

Себестоимость продукции калькулируется по статьям калькуляции, которые представлены на сети в виде входных потоков, отражающих входные обороты компонента 5. Два оборота включают только один вид затрат; прямые материальные или прямые нематериальные затраты, другие два - комплексные, включающие одновременно и материальные и нематериальные затраты. Это услуги вспомогательного производства и косвенные расходы. Чтобы распределить все материальные затраты по каждому виду продукции, необходимо их выделить из состава комплексных затрат. Как это можно сделать?

Рассмотрим один из способов.

**Шаг 1.** Составим таблицу затрат, в которой отражаются все обороты сети. Для сети, изображенной на рис. 37, имеем табл. 36.

Таблица 36

**Элементы производственных затрат по входным оборотам**

Вид затрат	Потребители затрат			
	Промежуточные компоненты		5.Основная продукция	
	3. Вспомогательное производство	4. Косвенные расходы	продуктА	продукт В
<b>По элементам</b>				
1.Материальные	$x_{13}$	$x_{14}$	$x_{15}^A$	$x_{15}^B$
2. Нематериальные	$x_{23}$	$x_{24}$	$x_{25}^A$	$x_{25}^B$
<b>Комплексные</b>				
3.Услуги вспомогательного производства	— $x_{43}$	$x_{34}$ —	$x_{35}^A$ $x_{45}^A$	— $x_{45}^B$
4. Косвенные расходы				
Всего	$\sum_i x_{i3}$	$\sum_i x_{i4}$	$\sum_i x_{i5}^A$	$\sum_i x_{i5}^B$

**Шаг 2.** Исчислить коэффициенты структуры затрат промежуточных компонентов по формуле:

$$c_{ij} = \frac{x_{ij}}{\sum_i x_{ij}}, \quad (4.14)$$

где  $c_{ij}$  – доля затрат вида  $i$  в общей сумме затрат по промежуточному компоненту  $j$ .

$x_{ij}$  – сумма затрат вида по промежуточному компоненту  $j$ ;

$\sum_i x_{ij}$  – общая сумма затрат по промежуточному компоненту  $j$ .

Исчисленные по формуле (4.14) коэффициенты делятся на две группы: коэффициенты затрат по элементам и коэффициенты комплексных затрат. Исходя из этого строятся две коэффициентные матрицы: матрица коэффициентов затрат по элементам промежуточных компонентов 3 и 4:

$$C = \begin{bmatrix} c_{13} & c_{14} \\ c_{23} & c_{24} \end{bmatrix};$$

матрица коэффициентов комплексных затрат по промежуточным компонентам 3 и 4:

$$A = \begin{bmatrix} 0 & c_{34} \\ c_{43} & 0 \end{bmatrix}.$$

**Шаг 3.** Из таблицы производственных затрат следует выделить в виде самостоятельной матрицы суммы подлежащих к расшифровке комплексных затрат, отнесенных на себестоимость продуктов **A** и **B**:

$$X = \begin{bmatrix} x_{35}^A & 0 \\ x_{45}^A & x_{45}^B \end{bmatrix}.$$

**Шаг 4.** Расшифровка сумм матрицы  $X$  на материальные и нематериальные затраты производится по формуле:

$$M = C \cdot (E - A)^{-1} X^*, \quad (4.15)$$

где  $M$  – матрица результата расшифровки комплексных расходов;

$C$  – матрица коэффициентов затрат по элементам промежуточных компонентов;

$E$  – единичная матрица;

$A$  – матрица коэффициентов комплексных затрат по промежуточным компонентам;

$(E - A)^{-1}$  – обратная матрица от  $(E - A)$ ;

$X$  – матрица сумм комплексных затрат, отнесенных на себестоимость продуктов.

Формула (4.15) применима при любом количестве промежуточных, входных и выходных компонентов, поскольку размер матриц не ограничен.

**Шаг 5.** Теперь определим, сколько единиц компонентов 1 и 2 в отдельности содержит каждая единица выходных оборотов системы. Для этого рассчитаем матрицу коэффициентов полных затрат внесистемных ресурсов:

$$D = C \cdot (E - A)^{-1},$$

где  $C$  – матрица коэффициентов прямых затрат внесистемных ресурсов;  
 $(E - A)^{-1}$  – матрица коэффициентов полных затрат системы.

Элемент матрицы  $D$  - т. е. коэффициент  $d_{is}$ , выражает полный расход внесистемных ресурсов вида  $i$  на единицу конечного продукта вида  $s$ .

В нашей системе конечным продуктом являются выходные компоненты 3 и 4, отражаемые в матрице  $X$ . Состав выходных оборотов системы в разрезе входящих компонентов (элементы матрицы  $M$ ) определяется умножением матрицы  $D$  на матрицу  $X$ , или

$$M = C \cdot (E - A)^{-1} X.$$

Исходя из приведенного конкретного содержания матриц  $A$ ,  $C$  и  $X$  формула (4.15) в нашем примере примет вид:

$$\begin{bmatrix} m_{15}^A & m_{15}^B \\ m_{25}^A & m_{25}^B \end{bmatrix} = \begin{bmatrix} c_{13} & c_{14} \\ c_{23} & c_{24} \end{bmatrix} \cdot \left[ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & c_{34} \\ c_{43} & 0 \end{pmatrix} \right]^{-1} \cdot \begin{bmatrix} x_{35}^A & 0 \\ x_{45}^A & x_{45}^B \end{bmatrix},$$

где  $m_{15}^A$  и  $m_{15}^B$  – материальные затраты в составе комплексных затрат по продуктам  $A$  и  $B$ ;

$m_{25}^A$  и  $m_{25}^B$  – нематериальные затраты в составе комплексных затрат по продуктам  $A$  и  $B$ .

Проиллюстрируем применение формулы (4.15) на числовых данных о производственных затратах промышленного предприятия, приведенных в табл. 37.

Таблица 37

### Обороты по элементам и структура затрат

Вид затрат	Потребители затрат						Всего тыс. руб.
	Промежуточные компоненты				5. Основная продукция, тыс. руб.		
	3. Вспомогательное производство		4. Косвенные расходы		Продукт А	Продукт В	
	тыс. руб.	$c_{i3}$	тыс. руб.	$c_{i4}$			
<b>По элементам</b>							
1. Материальные.....	12	0,6	8	0,27	60	40	120
2. Нематериальные.....	2	0,1	12	0,4	20	20	64
<b>Комплексные</b>							
3. Услуги вспомогательного производства	-	-	10	0,33	10	-	20
4. Косвенные расходы	6	0,3	-	-	10	14	30
Всего.....	20	1,0	30	1,00	100	84	234

Матрицы А, С и Х имеют следующее содержание:

1) матрица коэффициентов затрат по элементам промежуточных компонентов 3 и 4:

$$C = \begin{bmatrix} 0,6 & 0,27 \\ 0,1 & 0,4 \end{bmatrix};$$

2) матрица коэффициентов комплексных затрат по промежуточным компонентам 3 и 4:

$$A = \begin{bmatrix} 0 & 0,33 \\ 0,3 & 0 \end{bmatrix};$$

3) матрица сумм комплексных затрат, отнесенных на себестоимость продуктов А и В:

$$X = \begin{bmatrix} 10 & 0 \\ 10 & 14 \end{bmatrix};$$

Расшифровка комплексных затрат, отнесенных на себестоимость продуктов А и В, на материальные и нематериальные производится по формуле (4.15):

$$\begin{aligned} \begin{bmatrix} m_{15}^A & m_{15}^B \\ m_{25}^A & m_{25}^B \end{bmatrix} &= \begin{bmatrix} 0,6 & 0,27 \\ 0,1 & 0,4 \end{bmatrix} \cdot \left[ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 0 & 0,33 \\ 0,3 & 0 \end{pmatrix} \right]^{-1} \cdot \begin{bmatrix} 10 & 0 \\ 10 & 14 \end{bmatrix} = \\ &= \begin{bmatrix} 0,6 & 0,27 \\ 0,1 & 0,4 \end{bmatrix} \cdot \begin{bmatrix} 1,11 & 0,37 \\ 0,33 & 1,11 \end{bmatrix} \cdot \begin{bmatrix} 10 & 0 \\ 10 & 14 \end{bmatrix} = \begin{bmatrix} 12,7 & 7,3 \\ 7,3 & 6,7 \end{bmatrix}. \end{aligned}$$

В сумме комплексных затрат, отнесенных на себестоимость продукта А (10+10 = 20 тыс. руб.), материальные затраты составляют 12,7 и нематериальные - 7,3 тыс. руб. В сумме комплексных затрат, отнесенных на себестоимость продукта В (14 тыс. руб.), материальные затраты составляют 7,3 и нематериальные - 6,7 тыс. руб.

Следовательно, все материальные затраты по продукту А равны 60+12,7 = 72,7 тыс. руб.; по продукту В - 40 + 7,3 = 47,3 тыс. руб.; все нематериальные затраты по продукту А равны 20 + 7,3 = 27,3 тыс. руб., по продукту В - 30 + 6,7 = 36,7 тыс. руб.

Решение данной задачи на ПЭВМ удобно в среде Excel, так как однажды составленная программа действий позволяет решать подобные задачи при других данных, т.е. автоматизировать расчеты.

**Пример решения данной задачи в среде Excel**

Вид затрат	Потребители затрат						Всего тыс. руб.
	Промежуточные компоненты				5. Основная продукция, тыс. руб.		
	3.Вспомогательное производство		4.Косвенные расходы				
тыс. руб.	3	тыс. руб.	4	Продукт А	Продукт В		
<b>По элементам</b>							
1.Материальные.....	15,00	0,60	10,00	0,20	80,00	50,00	155,00
2. Нематериальные.....	5,00	0,20	15,00	0,30	50,00	30,00	100,00
<b>Комплексные</b>							
3.Услуги вспомогательного производства	-	-	20,00	0,40	25,00	15,00	60,00
4. Косвенные расходы	5,00	0,20	5,00	0,10	15,00	10,00	35,00
Всего	25,00	1,00	50,00	1,00	170,00	105,00	350,00

109

Затраты	Продукт	
	А	В
1.Материальные.....	92,84	57,60
2. Нематериальные.....	53,14	31,79
Всего	145,98	89,39

Материальные затраты	Продукт	
	А	В
Прямые	80,00	50,00
От услуг вспомогательного производства	15,00	9,00
От косвенных расходов	6,60	4,40
Всего	101,6	63,4

Нематериальные затраты	Продукт	
	А	В
Прямые	50,00	30,00
От услуг вспомогательного производства	5,00	3,00
От косвенных расходов	5,70	3,80
Всего	60,7	36,8

110

С	0,60	0,20
	0,20	0,30

А	-	0,40
	0,20	5,00

Х	25,00	15,00
	15,00	10,00

Е	1	0
	0	1

Е-А	1	-0,40
	-	0,20 -4

(Е-А) <sup>-1</sup>	0,98	-0,10
	-0,05	-0,25

$$\begin{array}{|c|c|} \hline mA15 & mB15 \\ \hline mA25 & mB25 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 0,58 & -0,11 \\ \hline 0,18 & -0,09 \\ \hline \end{array} * \begin{array}{|c|c|} \hline 25,00 & 15 \\ \hline 15 & 10 \\ \hline \end{array} = \begin{array}{|c|c|} \hline 12,84 & 7,60 \\ \hline 3,14 & 1,79 \\ \hline \end{array}$$

## Обозначения параметров сетевого графика

$P_i$  – событие с номером  $i$ .

$P_j$  – событие с номером  $j$ .

$P_i P_j$  – работа, начинающаяся в  $P_i$  и входящая в  $P_j$ .

$t_{ij}$  – время, необходимое для выполнения  $(P_i P_j)$  (продолжительность работы  $(P_i P_j)$ ).

$T_j$  – момент свершения события.

$T_j^{(p)}$  – самый ранний момент свершения события  $P_j$  (самое раннее время начала всех работ, выходящих из события  $P_j$ ).

$T_j^{(n)}$  – самое позднее допустимое время (момент) свершения события  $P_j$ .

$T_{kp} = T_n^{(p)} = T_n^{(n)}$  – длина критического пути, если  $T_j^{(p)} = 0$ .

$T_i^{(p)}$  – самый ранний момент начала работы  $(P_i P_j)$ .

$T_i^{(n)}$  – самый поздний допустимый момент начала работы  $(P_i P_j)$ .

$t_{ij}^{(p)}$  – самый ранний срок окончания работы  $(P_i P_j)$ . Очевидно, что

$$t_{ij}^{(p)} = T_i^{(p)} + t_{ij}.$$

$t_{ij}^{(n)}$  – самый поздний допустимый срок окончания работы  $(P_i P_j)$ ,

$$t_{ij}^{(n)} = T_i^{(n)} - t_{ij}.$$

$S_{ij}^{(n)}$  – полный резерв времени у работы  $(P_i P_j)$ .

$S_{ij}^{(r)}$  – частный резерв времени у работы  $(P_i P_j)$ .

$S_{ij}^{(н)}$  – независимый резерв у работы  $(P_i P_j)$ .

$\Pi_j$  – потенциал события  $P_j$ .

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сыроежин, И.М. Азбука сетевых планов/ И.М. Сыроежин. – М.: Изд-во «Экономика», 1966. – 170 с.
2. Поспелов, Г.С. Автоматизация процессов разработки больших систем или сложных комплексов/ Г.С. Поспелов, А.И. Тейман. – М.: Известия АН СССР. Техническая кибернетика, 1963.
3. Зуховицкий, С.И. Математические методы сетевого планирования/ С.И. Зуховицкий, И.А. Радчик. – М.: Изд-во физико-математической литературы, 1965. – 296 с.
4. Форд, Л.Д. Потоки в сетях/ Л.Д. Форд, Д.Р. Фалкерсон. – М.: Изд-во «Мир», 1966. – 276 с.
5. Абрамов, С.А. Сетевые методы планирования и управления / С.А. Абрамов, М.И. Мариничев, П.Д. Поляков. – М.: Изд-во «Советское радио», 1965. – 135 с.
6. Рыбакова, Л.В. Разработка управленческих решений на основе сетевого моделирования на примере ОАО « Нерюнгринский городской водоканал»//Л.В. Рыбакова, А.В. Крендовская [«Электронныйресурс»]-Режим доступа [http://www.amursu.ru/attachments/article/10651/N63\\_30/pdf](http://www.amursu.ru/attachments/article/10651/N63_30/pdf)
7. Усов, А.Л. Методы решения задач линейного программирования: учебное пособие/ А.Л. Усов, Л.В. Усов. – Вологда: ВоГТУ, 2003.-110 с.
8. Усов, Л.В. Динамическое программирование: учебное пособие/ Л.В. Усов. - Вологда: ВоГТУ, 2003.-50с.
9. Усов, Л.В. Элементы сетевого планирования: учебное пособие / Л.В. Усов.- Куйбышев: Куйбышевский плановый институт, 1966.-78 с.
10. Усов, А.Л. Метод определения субоптимальной очередности возведения зданий по критерию минимума срока строительства / А.Л. Усов, Л.В. Усов // Сб. «Математические методы и интеллектуальные системы в экономике и образовании»: материалы Всероссийской заочной научно-практической конференции.- Ижевск: УдГУ, 2011.- С.44-47.
11. Усов, Л.В. Методика оценки затрат капитальных вложений в рост производительности труда на основе производственных функций комплексных переменных / Л.В. Усов, Д.Л. Усов, Р.А. Усова // Сб. «Леденцовские чтения. Бизнес. Наука. Образование». [Текст]: материалы международной науч.-практ. конференции. г. Вологда, 28-29 марта 2013г. : в 2 частях - ч. 2/под ред. д.э.н., проф. Ю.А. Дмитриева.- Вологда: Вологодский институт бизнеса, 2013.- 682 с.- С.344-351
12. Джонсон, С.М. Оптимальные двух- и трехоперационные календарные планы производства с учетом подготовительно-заключительного времени // Сб. «Календарное планирование» / сост. Дж. Ф. Мут и Дж.Л. Томпсон.- М.: Изд-во «Прогресс», 1966.- 466 с.- С. 33-41.
13. Питерсон, Дж. Теория сетей Петри и моделирование систем: пер. с англ. / Дж. Питерсон. – М., 1984. – 264 с.
14. Афанасьев, В.А. Определение рациональных очередностей освоения

частных фронтов работ применительно к различным методам поточной организации работ / В.А. Афанасьев, В.З. Величкин // Сб. «Математические методы решения экономических задач».- М.: Наука, 1980.- № 9.- С. 23-42.

15. Модели сетевого планирования и управления // Исследование операций в экономике / под ред. Н.Ш.Кремера . – 1998 . – С. 286-332

16. Бухалков, М.И. Стратегическое планирование. Сетевые методы планирования. Оперативное планирование производства. Экономическая оценка планов. Внутрифирменное планирование: учебник для вузов по эконом. спец. / М.И. Бухалков.– М., 1999. – С. 75-373.

17. Теличенко, В.И. Применение сетевого планирования при проектировании, подготовке и строительстве объектов городских программ жилищного строительства / В.И. Теличенко, Е.А. Король, Б.Ф. Ширшиков и др. // Строительные материалы, оборудование, технологии XXI века. – 2004. – № 12. – С. 40-41.

18. Полетаев, В.П. Модули интеграции АСУ ТООиР с системами календарно-сетевого планирования / В.П. Полетаев, А.Н. Ершов // Автоматизация машиностроительного производства, технология и надежность машин, приборов и оборудования: матер. второй науч.-техн. конф. 13-15 ноября 2006 г. – Вологда: ВоГТУ, 2006. – Т. 1 . – С. 146-149.

19. Алексеев, С.Р. Сетевое моделирование инновационных проектов развития транспортных предприятий / С.Р. Алексеев // Жилищное строительство. – 2007. – № 1. – С. 9-10.

20. Требуна, П. Применение методов сетевого анализа в моделировании / П. Требуна, М. Пекарчикова // Стратегия и тактика реализации социально-экономических реформ: региональный аспект: материалы VI Междунар. науч.-практ. конф. (г. Вологда, 6-8 окт. 2011 г.): в 3 ч. Ч. 3 / ИСЭРТ РАН. – Вологда, 2011. – С. 272-278. – Библиогр.: с. 278.

21. Сетевое планирование и управление: методические указания к практическому занятию: ЭЭФ: направление подготовки 140400.62 - Электроэнергетика и электротехника: [профиль подготовки "Электроснабжение"] /сост. Л. П. Летунова. – Вологда: ВоГТУ, 2012. – 7 с.

22. Булыга, Р. Добавленная стоимость как целевой критерий/Р. Булыга, П. Кохно //Экономист. – 2007. – №10. – С. 68-76.



Учебное издание

**Леонид Васильевич Усов**  
**Раиса Александровна Усова**

**СЕТЕВОЕ МОДЕЛИРОВАНИЕ СТРОИТЕЛЬНЫХ ПРОЦЕССОВ:**  
*прикладные модели и методы исследования*  
*строительных процессов*

Учебное пособие

Редактор Л.А. Перерукова

Подписано в печать 15.05.2014 г.  
Печать офсетная. Бумага офисная.  
Формат 60×90/16. Усл. печ. л. 7,1.  
Тираж 20 экз. Заказ

---

Отпечатано: РИО ВоГУ, г. Вологда, ул. Ленина, 15

