Реферат

Курсовой проект 43 с., 5 рис., 6 блок-схем, 1 таблица, 1 источник.

СЕТЕВОЙ ГРАФИК, АНАЛИЗ ОПТЕМАЛЬНОСТИ СЕТЕВЫХ ГРАФИКОВ, РАЦИОНАЛЬНЫЕ МЕТОДИКИ ПОИСКА ОСОБЫХ ПУТЕЙ СЕТЕВЫХ ГРАФИКОВ, АВТОМАТИЗАЦИЯ АНАЛИЗА СЕТЕВЫХ ГРАФИКОВ НА ЭВМ.

Направление работы – изучение математических и алгоритмических аспек­тов анализа оптимальности сетевых графиков.

Основная цель работы – найти и доказать рациональные методики поиска особых путей сетевых графиков, легко поддающиеся автоматизации на ЭВМ и со­кращающие затраты на сетевое планирование, за счёт уменьшения сроков разра­ботки оптимальных сетевых графиков.

Используемый в работе метод исследований – аппарат формальной логики, позволяющий осуществлять математические доказательства с минимальным при­влечением, для этого, формул.

В ходе работы получены блок-схемы алгоритмов расчёта параметров сете­вых графиков и поиска их особых путей, которые предполагается использовать при создании конкретной программы анализа оптимальности сетевых графиков на любом из известных языках программирования.

Новизна работы состоит в том, что разработанные методы позволяют найти критический и наикратчайший пути сетевого графика без перебора всех возмож­ных вариантов, что даёт: во-первых – высокую скорость разработки оптимальных сетевых графиков, а во-вторых – возможность точного ответа на вопрос об оптимальности уже готового сетевого графика и высокую степень оптимизации сетевых графиков по длительности в случае их неоптимальности.

Содержание

Введение 4

1 Постановка задачи 6

2 Теоретические основы сетевого планирования 9

3 Обоснование рациональных методик поиска особых путей сете­вых графиков 15

4 Автоматизация анализа оптимальности сетевых графиков на ЭВМ 22

4.1 Представление сетевого графика в машинной форме 22

4.2 Автоматизация расчёта параметров сетевого графика 27

4.3 Автоматизация процесса поиска особых путей сетевого гра­фика 40

Заключение 42

Список использованных источников 43

Введение

Одним из основных экономических показателей, определяющих себестои­мость проведения проектных, научно-исследовательских, опытно-конструктор­ских и других, поддающихся экономическому анализу, работ, связанных с раз­ра­боткой и внедрением на предприятие новой техники или с организацией и управ­лением деятельности всего предприятия, является общая продолжительность их выполнения. Естественно, что в рамках некоторого рассматриваемого проекта, эта продолжительность существенно зависит от структуры упорядочивания отдель­ных, входящих в него работ. Поэтому, построение оптимальной структуры упоря­дочивания проектных работ является основной задачей сетевого планирования.

В основе решения указанной задачи лежит анализ смыслового содержания работ и ус­тановление взаимо­связей между ними, что позволяет выявить возмож­ность их параллельного выполнения. Последнее, является основным фактором со­кращения дли­тельности всего проекта.

Распространены два метода оптимального планирования или упорядочива­ния проектных работ. Один из методов, основан на построении ленточного гра­фика, где каждой работе присваи­ваются такие характеристики как время начала её выполнения, её длительность, которые затем, в виде параллельных от­резков, на­но­сятся на шкалу времени. Другой из ме­тодов, ос­нован на построении сетевого графика, где структура упорядочивания работ изо­бражается графически в виде сигнального графа.

Выбор того или иного метода планирования зависит от числа работ, входя­щих в состав проекта. Принято, что если число работ превышает 25, то наиболее наглядный и удобный метод опти­мального планирования – есть метод, основан­ный на построении сетевого графика. На практике этот метод более употребите­лен, в силу того, что число работ, входящих в некоторый рассматриваемый проект, как правило, достигает не­скольких сотен.

Для сетевого графика, существует два понятия оптимальности: оп­тималь­ность по структуре и оптимальность по длительности. Оптимальность по струк­туре характеризуется степенью параллельности исполнения отдельных ра­бот. Оп­тимальность по длительности характеризуется рациональным распре­деле­нием тру­довых ресурсов между параллельными видами работами, которое обеспечивает при­мерно равную их продолжительность.

На сегодняшний день нет, и не предвидится появление, строгих методов и алгоритмов построения оптимального сетевого графика, поддающихся автомати­зации на ЭВМ. Это связано с тем, что процесс построения оптимального сетевого графика требует от экономиста-проектировщика опыта и интуитивных свойств мышления, реализовать которые на ЭВМ практически не возможно.

По другому обстоит дело с задачей анализа оптимальности уже готового се­тевого графика. Надо сказать, что с этой задачей экономист-проектировщик стал­кивается систематически при оптимизации сетевого графика по длительности, ко­гда каждое очередное принятое решение о перераспределении трудовых ресурсов требует проверки на достижение оптимального варианта. Очевидно, что если ав­томатизи­ровать процесс решения рассматриваемой задачи, то это существенно снизит про­должитель­ность разработки сетевого графика, а значит и затраты на се­тевое пла­нирование в целом. Так вот, задача анализа оптимальности сетевого гра­фика математиче­ски формализуема и, с некоторыми трудностями, решаема на ЭВМ. В данном курсовом проекте, как раз и будут предложены и обоснованы ра­циональные методики решения задачи анализа оптимальности сетевых графиков, легко автоматизируемые на ЭВМ.

# Постановка задачи

Как правило, экономисту-проектировщику не представляется сложным, с первого раза, построить оптимальный по структуре сетевой график, когда будет обеспечена максимальная параллельность исполнения отдельных работ. Всё зави­сит от понимания им сущности и содержания каждой работы, входящей в состав сетевого графика.

Труднее обстоит дело с распределением трудовых ресурсов по отдельным видам работ, от которого зависит оптимальность сетевого графика по длительно­сти. Проблема в том, что практически невозможно предугадать, как от­разится на длительности всего проекта и соотношении длительностей различных путей его сетевого графика, перенос трудовых ресурсов с одних работ на другие, в резуль­тате которого, при неизменной трудоемкости работ, происходит увеличение дли­тельности первых и уменьшение длительности вторых. В таких условиях, ос­та­ётся только один способ оптимизации сетевого графика по длительности. Этот способ основан на методе проб и ошибок, когда, первостепенную важность играет задача проверки и анализа оптимальности уже готового, полностью рассчитанного сете­вого графика, с целью выявления ошибок в распределении трудовых ресурсов. Рассмотрим эту задачу и связанные с ней трудности подробнее.

Для сетевого графика существуют понятия пути и его продолжительности. Под путем понимается любая цепочка непрерывно следующих, друг за другом, последовательных во времени работ, от начала проекта до его завершения. Под длительностью пути понимается суммарная длительность всех, входящих в него, последовательных работ. Более понятными, данные определения станут при рас­смотрении следующего раздела. Сейчас же, важно другое, что каждый сетевой график имеет в своём составе два особых пути: критиче­ский и наикратчайший. Критическим путём является путь, имеющий наибольшую продолжительность среди других возможных путей сетевого графика. Наикрат­чайшим путём является путь, который, в отличие от критического пути, имеет наименьшую продолжи­тельность во всём сетевом графике. На понятиях этих двух путей основан наибо­лее простой и распространенный критерий оптимальности сетевого графика, фор­мализуемый следующим образом:

, (1.1)

1.  – коэффициент напряжённости наикратчайшего пути;
2.  – длительность наикратчайшего пути, ;
3.  – длительность критического пути, .

Из критерия (1.1) следует, что некоторый рассматриваемый сетевой график принимается оптимальным, если отношение длительности его наикратчайшего пути к длительности его критического пути не менее 0.7, или, что тоже самое, если длительность наикратчайшего пути отличается от длительности критиче­ского пути не более чем на 30%.

Забегая вперёд, можно сказать, что длительность критического пути, легко найти путём расчёта некоторых, принятых параметров сетевого графика, которые будут подробно рассмотрены в следующем разделе. Длитель­ность же наикрат­чайшего пути, в общем случае неизвестна, и для её нахождения требуется сумми­ровать длительности всех, входящих в него работ.

Теперь встаёт проблема, – а как найти работы, принадлежащие наикратчай­шему пути, чтобы иметь возможность просуммировать их длительности? Решить данную проблему, для человека, интуитивно или простым перебором вариантов, очень проблематично, особенно при большой, сильно разветвленной структуре се­тевого графика. Зачастую и ЭВМ справиться с этой задачей не может, в силу того, что её быстродействие ограничено, а число всех возможных вариантов путей сете­вого графика, уже при стах событиях, может достигать миллионов или даже сотен миллионов.

Так вот, оказывается, эта проблема решаема, причём без перебора вариантов и срав­нительно быстро даже для человека, не говоря уже об ЭВМ. Основной це­лью дан­ной курсового проекта, как раз и является цель показать, а точнее доказать рациональные ме­тодики поиска особых путей сетевого графика, которые не только дают возмож­ность проверки его оптимальности, но и позволяют рационально вы­полнить его оптимизацию по длительности. Последнее заключается в том, что если экономист-проектировщик будет знать, как проходят особые пути сетевого графика, то он сможет, в целях оптимизации, правильно перераспределить трудо­вые ресурсы, а именно – перенести ресурсы с работ, принадлежащих наикратчай­шему пути, на работы, принадлежащие критическому пути, и тем самым уровнять длительности этих путей, для обеспечения выполнения критерия оптимальности (1.1).

# Теоретические основы сетевого планирования

Прежде, чем преступать к обоснованию рациональных методик поиска осо­бых путей сетевого графика, необходимо напомнить, что вообще собой представ­ляет сетевой график, и какими основными параметрами он характеризу­ется.

Итак, сетевой график – есть математическая модель упорядочивания про­ектных работ типа “Сигнальный граф” (см. пример на рис.2.1 ). Любой сигналь­ный граф состоит только из двух элементов: дуг и вершин. В контексте сетевого пла­нирования, дугами являются отдельные работы, изображаемые на сетевом графике в виде стрелок так, что начала стрелок, соответствует началам выполне­ния работ, концы стрелок – их завершению. Вершинами сигнального графа явля­ются так на­зывае­мые события, которые изображаются на сетевом графике в виде кружков, с поряд­ковыми номерами в нижних квадрантах. Как раз события сете­вого графика и служат для целей упорядочивания проектных работ, которое за­ключается в том, что исходящая из неко­торого события работа не может начаться, пока не завер­шаться все входящие в него работы.

Существует масса правил, узаконенных стандартом, придерживаться кото­рых необходимо при построении сетевых графиков. Наиболее важные из них:

* Любой сетевой график должен иметь начальное событие, ра­боты из ко­то­рого только исходят, и конечное событие, в которое они только входят;
* Любой путь сетевого графика должен быть полным. То есть, любая це­почка, непрерывно следующих друг за другом, последовательных во времени ра­бот, должна начинаться в исходном событии сетевого графика, а заканчиваться в конечном;
* Сетевой график не должен иметь замкнутых петель. То есть, недопус­тимо, чтобы конец некоторой работы являлся бы началом другой работы, предше­ствующей первой по времени.

Имея только структуру сетевого графика, невозможно разрешить вопрос о его оптимальности. Требуется проводить расчеты еще целого ряда, принятых па­раметров. К этим параметрам относятся:

* ранние и поздние сроки наступления событий;

1. 2.1 – Пример сетевого графика

0

2

1

4

3

5

6

7



0

3

4

12

5

9

12

15

15

12

4

12

9

5

3

0

0

0

0

0

0

0

0

0



* ранние и поздние сроки начала и окончания работ;
* резервы времени работ и событий.

Ранний срок наступления события – это минимально возможный срок, необ­ходимый для выполнения всех работ, предшествующих данному событию. Расчёт ранних сроков наступления событий ведут в порядке – от начального собы­тия проекта (с номером 0) до завершающего. При расчёте принимают, что ран­ний срок наступления начального события равен 0. Для определения ран­него срока наступ­ления -го события пользуются правилом, математически записывае­мым так:

 , (2.1)

1.  – ранний срок наступления рассматриваемого события, ;
2.  – номер рассматриваемого события;
3.  – номера предшествующих событий, соединенных с рассматривае­мым работами;
4.  – ранний срок наступления -го предшествующего события, ;
5.  – длительность работы, соединяющей -е предшествующее собы­тие с рассматриваемым, .

Таким образом, ранний срок наступления -го события – есть максимально воз­можная сумма из сумм ранних сроков наступления предшествующих событий и длитель­ностей работ соединяющих предшествующие события с рассматривае­мым. Забегая вперёд, надо сказать, что эти суммы равны ранним срокам окончания соответствующих работ. Тогда, ранний срок свершения события – есть макси­мальный из ранних сроков окончания, входящих в него работ.

Поздний срок наступления события – это максимально допустимый срок на­ступления рассматриваемого события, определяемый из условия, что после насту­пления этого события в свой поздний срок остаётся достаточно времени, чтобы выполнить следующие за ним работы. Расчёт поздних сроков наступлений собы­тий ведут в обратном порядке – от завершающего события проекта до на­чального (с номером 0). При расчёте принимают, что поздний срок на­сту­пления завершаю­щего события совпадает с его ранним сроком наступле­ния. Для расчёта позднего срока наступления -го события пользуются правилом, матема­тически записывае­мым так:

, (2.2)

1.  – поздний срок наступления рассматриваемого события, ;
2.  – номер рассматриваемого события;
3.  – номера последующих событий, соединённых с рассматриваемым работами;
4.  – поздний срок наступления -го последующего события, ;
5.  – длительность работы, соединяющей -е последующее событие с рассматриваемым, .

Таким образом, поздний срок наступления -го события – есть минимально воз­можная разность из разностей поздних сроков наступления последующих событий и дли­тельностей работ, соединяющих последующие события с рассматриваемым. Забегая вперёд, необходимо сказать, что эти разности равны позд­ним срокам на­чала соответствующих работ. Тогда, поздний срок свершения события – есть ми­нимальный среди поздних сроков начала, исходящих из него работ.

Зная ранний и поздний сроки наступления события, можно определить ре­зерв времени события:

, (2.3)

1.  – резерв времени рассматриваемого события, .

Резерв времени события показывает насколько можно отсрочить наступление со­бытия по сравнению с его ранним сроком наступления без изменения об­щей про­должительности всего проекта.

Ранний срок начала работы совпадает с ранним сроком наступления её на­чального события, а ранний срок окончания работы превышает его на величину продолжительности этой работы:

; (2.4)

, (2.5)

1.  – ранний срок начала работы, исходящей из -го события и входящей в -е событие, ;
2.  – ранний срок окончания данной работы, ;
3.  – длительность этой работы, ;
4.  – раннее начало события, из которого исходит рассматриваемая работа, ;

Поздний срок окончания работы совпадает с поздним сроком наступ­ления её конечного события, а поздний срок начала работы меньше на величину продолжи­тельности этой работы:

; (2.6)

, (2.7)

1.  – поздний срок окончания работы, исходящей из -го события и входящей в -е событие, ;
2.  – поздний срок начала данной работы, ;
3.  – длительность этой работы, ;
4.  – позднее окончание события, в которое входит рассматриваемая работа, .

Полный резерв времени некоторой работы – это максимальное время, на ко­торое можно отсрочить её начало или увеличить продолжительность, не из­меняя директивного срока наступления завершающего события сетевого графика:

, (2.8)

1.  – полный резерв времени работы, исходящей из -го события и входящей в -е событие, .

Свободный резерв времени некоторой работы – максимальное время, на ко­торое можно отсрочить её начало или увеличить её продолжительность при усло­вии, что все события наступают в свои ранние сроки:

, (2.9)

1.  – свободный резерв времени работы, исходящей из -го собы­тия и входящей в -е событие, .

В качестве примера, который потребуется и в дальнейшем, основные рас­смотренные параметры сетевого графика рассчитаны для случая, представленного на рисунке 2.1 . Здесь, длительности работ, являющиеся исходными данными для расчёта, выбраны произвольным образом. Параметры работ обозначены соответ­ствующими символами возле стрелок. Параметры событий отражены в трёх квад­рантах соответствующих кружков. В левых квадрантах отражены значения ранних сроков свершения событий. В правых – значения поздних сроков свершения собы­тий. В верхних – значения резервов времени событий.

Как говорилось в предыдущем разделе, длительность критического пути легко найти из расчёта параметров сетевого графика. Теперь можно сказать, чему она равна, – она равна сроку свершения завершающего события сетевого графика и, соответственно, определяет длительность выполнения всех проектных работ. По­следнее заключается в том, что проектные работы не могут завершиться в срок, меньший, чем длительность критического пути, и в тоже время, если все проект­ные работы выполняются вовремя, то срок их завершения равен длительности критического пути.

# Обоснование рациональных методик поиска особых путей сетевых графиков

Обоснование рациональных методик поиска особых путей сетевого графика основано на смысле полного резерва времени работы, который показывает, на сколько можно отсрочить начало или увеличить продолжительность работы без изменения продолжительности всего проекта. Надо сказать, что этот смысл выте­кает из правил расчёта сетевого графика и давно известен, поэтому сейчас он не требуется в специальном рассмотрении. Важно другое – из смысла полного ре­зерва времени работы следует истинность следующего утверждения, на котором основаны некоторые, приводимые ниже доказательства, – полный резерв времени работы может появиться только за счёт существования другого более длительного пути, нежели путь, в состав которого входит рассматриваемая работа. Это утвер­ждение становится очевидным, если подумать – за счёт чего, у некоторой работы, может появиться возможность отсрочить начало её выполнения или увеличить её продолжительность без изменения срока свершения завершающего события сете­вого графика? Естественно, только за счёт того, что этот срок свершения опреде­ляется другим, более продолжительным путём.

Начнём с доказательства методики поиска критического пути сетевого гра­фика. Для этого рассмотрим ряд вспомогательных теорем.

**Теорема 3.1** – Для того, чтобы некоторый путь сетевого графика был бы кри­тическим, необходимо и достаточно, чтобы полные резервы времени всех вхо­дя­щих в него работ были бы равны нулю.

**Необходимость** – Если некоторый путь является критическим, то полные резервы времени всех входящих в него работ равны нулю.

Докажем это утверждение методом от противного.

Пусть известно, что некоторый рассматриваемый путь заведомо критиче­ский. Теперь предположим противное – на нём лежит хотя бы одна работа с нену­левым резервом времени. Это означает, что есть другой путь, с большей продол­жительностью, чем рассматриваемый, за счёт чего и получается данный резерв времени. Но, раз имеется более продолжительный путь, то рассматриваемый путь уже не может быть критическим. Полученное противоречие доказывает невоз­можность существования на критическом пути работы с ненулевым полным ре­зервом времени, так как в противном случае, он уже не будет являться критиче­ским. Тогда, для любой работы критического пути остаётся другая возможная си­туация – её полный резерв времени равен нулю. Утверждение доказано.

Поскольку любой сетевой график имеет критический путь, то есть путь с наибольшей продолжительностью, то, на основании только что доказанного, в лю­бом сетевом графике можно найти путь, работы которого имеют только нулевые полные резервы времени.

**Достаточность** – Если все работы некоторого пути имеют нулевые полные резервы времени, то этот путь обязательно является критическим.

Если некоторый путь имеет работы только с нулевыми полными резервами времени, то это означает, что ни одну работу, указанного пути, нельзя увеличить по длительности без изменения срока свершения завершающего события сетевого графика. Это возможно, только когда сумма длительностей работ, рассматривае­мого пути равна сроку свершения завершающего события, то есть длительности критического пути. Тогда, рассматриваемый путь и является критическим, в силу того, что он равен критическому пути по длительности. Утверждение доказано.

**Теорема** **3.2** – Если в некоторое событие сетевого графика входит работа с ну­левым полным резервом времени, то среди всех исходящих из данного события работ, обязательно найдётся хотя бы одна, имеющая также нулевой резерв вре­мени. То есть, работы с нулевыми резервами времени следуют друг за другом не­прерывно.

Для доказательства данной теоремы рассмотрим обобщенный пример на ри­сунке 3.1 , где, в целях удобства, событиям присвоены условные номера.

Докажем теорему методом от противного.

Пусть для работы, входящеё в событие 2, полный резерв времени . Предположим противное – среди всех работ, исходящих из события 2, нет ни од­ной работы с нулевым полным резервом времени.



1. 3.1 – Поясняющий рисунок к Теореме 3.2

Для начала найдём, чему равен поздний срок свершения события 2. Он, в соответствии с формулой (2.2), определяется как минимальное время позднего на­чала работы среди всех работ, исходящих из рассматриваемого события. Пусть поздний срок свершения события 2 равен позднему началу работы, входящей, на­пример, в событие 4:

,

или, в соответствии с выражением (2.8) для полного резерва времени,

. (3.1)

Теперь рассмотрим, какое может иметь значение полный резерв времени ра­боты, исходящей из события 1 и входящей в событие 2. В соответствии с форму­лой (2.8):

. (3.2)

Из формулы (3.2) видно, что минимально возможное значение полного ре­зерва времени работы, исходящей из события 1 и входящей в событие 2, достига­ется тогда, когда величина  достигает своего максимального значения. Из правила определения раннего срока свершения события, задаваемого формулой (2.1), следует, что максимальное значение этой величины может быть равно только раннему сроку свершения события 2, когда ранний срок окончания рассматривае­мой работы самый большой из всех ранних сроков окончания работ, входящих в событие 2. Тогда, минимально возможное значение полного резерва времени ра­боты, исходящей из события 1 и входящей в событие 2 равно:

,

или, исходя из формулы (3.1):

. (3.3)

Поскольку мы предположили от противного, что среди всех исходящих из события 2 работ нет работ с нулевым полным резервом времени, то отсюда сразу вытекает, что и работа, исходящая из события 1 и входящая в событие 2, также не может иметь нулевой полный резерв времени, уж если его минимальное значение заведомо неравно нулю, в соответствии с полученным равенством (3.3). Последнее противоречит условию теоремы. Из этого противоречия следует то, что невоз­можна ситуация, когда при нулевом резерве времени работы, входящей в событие 2, все исходящие из этого события работы имели бы ненулевые резервы времени. Если бы это имело место, то в соответствии с приведённым доказательством, ра­бота, входящая в событие 2 также бы имела ненулевой полный резерв времени. Но ведь это не так по условию теоремы. Тогда для работ, исходящих из события 2 ос­таётся другая возможная ситуация – хотя бы одна из них имеет также нулевой полный резерв времени. Теорема доказана.

Из доказанных выше теорем, непосредственно, следует методика поиска критического пути, приводимая ниже.

**Рациональная методика поиска критического пути сетевого графика:**

1. Просмотр сетевого графика ведётся от его начального события к конеч­ному;
2. При рассмотрении начального события сетевого графика, в качестве ра­боты, лежащей на критическом пути, выбирается та, которая имеет нулевой пол­ный резерв времени. В соответствии с теоремой 3.1 (утверждение-необходимость), такая работа обязательно будет существовать;
3. При рассмотрении работ, исходящих из события, к которому привила ра­бота с нулевым полным резервом времени, выбирается работа, также имеющая нулевой полный резерв времени. В соответствии с теоремой 3.2, такая работа су­ще­ствует;
4. Если, среди исходящих из некоторого события работ, есть несколько ра­бот с нулевыми полными резервами времени, то выбирается любая. При этом, со­гласно теореме 3.2, процесс построения критического пути в тупик зайти не мо­жет, и рано или поздно дойдет до завершающего события сетевого графика.

Реализация указанных правил даёт путь, состоящий только из работ с нуле­выми полными резервами времени. Тогда, на основании теоремы 3.1 (утвержде­ние-достаточность), этот путь и будет являться критическим.

В целях проверки, доказанная методика применена для сетевого графика, представленного на рисунке 2.1 . Здесь, найденные критические пути, выделены жирными стрелками. Как видно, таких путей два, благодаря тому, что среди работ, исходящих из события 0, есть две работы с нулевыми полными резервами вре­мени. Проверить то, что найденные пути являются критическими легко, просум­мировав длительности принадлежащих им работ. Суммы окажутся: во-первых, равными между собой, а во-вторых, наибольшими среди аналогичных сумм дру­гих возможных путей.

Теперь рассмотрим вопрос поиска наикратчайшего пути сетевого графика. Оказывается, для его поиска можно применять, методику поиска критического пути, если использовать идею, высказываемую в следующей теореме.

**Теорема 3.3** – Если произвести расчёт параметров заданного сетевого гра­фика по установленным правилам, но заменяя известные длительности работ на те же значения с отрицательным знаком (длительности всех работ будут меньше нуля), то наикратчайший путь сетевого графика станет подчиняться всем свойст­вам кри­тического пути.

Эту теорему легко доказать, используя правило сравнения отрицательных чисел. Данное правило заключается в том, что одно отрицательное число счита­ется большим другого, если абсолютное значение первого меньше абсолютного значения второго. Поскольку длительность наикратчайшего пути, по абсолютному значению наименьшая, среди длительностей всех других путей сетевого графика, то, на основании указанного правила, отрицательная длительность наикратчай­шего пути будет наибольшей среди отрицательных длительностей остальных пу­тей. Тогда, наикратчайший путь, состоящий из работ с отрицательными длитель­ностями, будет критическим, при условии, что все остальные пути, также состоят из работ с отрицательными длительностями. Теорема доказана.

Для проверки доказанной теоремы, параметры сетевого графика на рисунке 2.1 пересчитаны заново, при отрицательных значениях длительностей работ, и пред­ставлены на рисунке 3.2 . Как видно, сетевой график на рисунке 3.2 содержит путь, работы которого имеют только нулевые полные резервы времени. Данный путь выделен жирными стрелками. Этот путь, являясь критическим для сетевого гра­фика на рисунке 3.2 , в тоже время является наикратчайшим путем для сетевого гра­фика на рисунке 2.1 . Последнее можно проверить простым суммированием дли­тельностей его работ. Полученная сумма должна быть наименьшей по абсо­лют­ному значению, среди аналогичных сумм других путей сетевого графика на ри­сунке 2.1 .

Вообще говоря, для нахождения продолжительности наикратчайшего пути, необходимой при анализе оптимальности сетевого графика по критерию (1.1), не обязательно суммировать длительности всех, принадлежащих ему работ. Она уже известна из рассчитанных, при отрицательных длительностях работ, парамет­ров сетевого графика, и равна, как и для любого критического пути, сроку свер­шения завершающего события. Естественно, что данный срок свершения имеет отрица­тельное значение, и поэтому, для нахождения фактической длительности наикрат­чайшего пути, требуется менять это значение на противоположное.

Необходимо сказать, что можно поставить и решить общую задачу поиска пути заданной продолжительности. Но данная задача принципиаль­ной важности, при анализе сетевого графика, не несёт. Для анализа оптимально­сти сетевого гра­фика и осуществления его оптимизации, достаточно знать лишь, как проходят особые пути, и какова их продолжительность. Ответы на эти вопросы и дают ра­циональные методики поиска особых путей, доказанные в этом разделе.

1. 3.2 – Пример сетевого графика при отрицательных длительностях работ

0

2

1

4

3

5

6

7



0

-3

-1

-6

-5

-7

-7

-9

-9

-6

2

-6

-6

-4

-3

0

0

1

1

1

0

0

3

0



# Автоматизация анализа оптимальности сетевых графиков на ЭВМ

## Представление сетевого графика в машинной форме

Любая ЭВМ нуждается в преобразовании различных абстрактных понятий, ясных для человека, в удобную для неё форму. Сетевой график, как графическое изображение упорядоченных кружков и стрелок само по себе для ЭВМ нечего не значить. Для того, чтобы ЭВМ могла понимать структуру сетевого графика и, главное, обрабатывать её, необходимо представить эту структуру в эквивалентной машинной форме.

Наиболее удобный способ представления структуры сетевого графика в ма­шинной форме, основан на понятии матрицы смежностей . Пример данной матрицы для структуры сетевого графика на рисунке 2.1 представлен на рисунке 4.1 .

Матрица смежностей квадратная и имеет размерность , где  – число событий сетевого графика. Номера строк матрицы задаются номерами событий , из которых работы сетевого графика исходят, номера столбцов матрицы зада­ются номерами событий , в которые работы сетевого графика входят. На пере­сечении строки и столбца , в матрице смежностей, может быть только одно из двух значений: 0 или 1. Если , то это означает, что на сетевом гра­фике существует работа, исходящая из события с номером  и входящая в со­бытие с номером . Если , то такой работы на сетевом графике нет.

Матрица смежностей будет верно отражать структуру сетевого графика, если сетевой график построен по всем, узаконенным стандартом правилам. Здесь, наиболее важны следующие:

* Событиям присваиваются номера с таким расчётом, что старший номер соответствует более позднему по времени событию. То есть, если рассмотреть не­которое событие и все входящие в него работы, то номер этого события должен быть больше номеров всех событий, из которых эти работы исходят. В этом случае первая строка и первый столбец матрицы смежностей соответствует начальному событию сетевого графика , а последние строка и столбец – завершающему со­бытию сетевого графика , где  – число всех событий в сетевом графике.

С0

С1

С2

С3

С4

С5

С6

С7

С0

С1

С2

С3

С4

С5

С6

С7

1

1

1

1

1

1

1

1

1

1

1

1

1

1

0

0

0

0

0

0

0

0

0

0

0

0

0

0

1. 4.1 – Матрица смежностей сетевого графика

Ci

Cj

* Два события сетевого графика может соединять только одна работа. Если все же имеет место факт соединения двух событий несколькими работами, то, для выполнения указанного правила, необходимо ввести дополнительные события, разрывающие лишние работы и дополняющие их фиктивными работами с нулевой длительностью (см. пример на рис. 4.2 ). Дополнительные события также должны иметь свои уникальные, в сетевом графике, номера, присвоенные им в соответст­вии с первым правилом.

Верно построенная матрица смежностей обладает радом полезных свойств:

* Если задаться некоторым номером события , то единицы в соответст­вующей строке укажут на номера событий , с которыми событие  соединено, исходящими из него работами. Это свойство следует из правила построения мат­рицы смежностей.

Верно

Неверно

1

1

3

2

2

1. 4.2 – Пример разрыва параллельных работ

Фиктивная работа

* Если задаться некоторым номером события , то единицы в соответст­вующем столбце укажут на номера событий , с которыми событие  соеди­нено, входящими в него работами. Это свойство, также, следует из правила по­строения матрицы смежностей.
* Если некоторое событие  указывает единицами в соответствующей строке матрицы смежностей на соединённые с ним события , то номера этих событий  могут быть только больше номера , что ясно из правила присвоения номеров событиям сетевого графика. Из этого свойства следует, что матрица смежностей носит диагональный характер, то есть, единицы в матрице смежно­стей могут присутствовать только в верхней диагональной части матрицы (см. рис. 4.1 ).

Любопытно заметить, что если последнее из перечисленных свойств не вы­полняется, то в сетевом графике есть петли, то есть, работы, концы которых явля­ются началами других работ, предшествующих первым по времени, при условии, что все события занумерованы, верно. Из этого следует возможность легкой авто­матизации на ЭВМ процесса проверки правильности построения сетевого гра­фика. Данный процесс проверки, алгоритмически, представляется в виде блок-схемы 4.1 .

Суть алгоритма проверки заключается в определении содержимого элемен­тов нижней диагональной части матрицы смежностей. Если там встретится хотя бы одна единица, то это будет означать, что сетевой график построен неправильно – либо в нем есть петли, либо события занумерованы не верно.

Блок-схема 4.1 – Алгоритм тестирования матрицы смежностей

Начало

Матрица смежностей уже задана

i:=0

j:=0

Mi j<> 0

Ошибка в структуре графика

Err:=1

Err=1 – Ошибка в

структуре;

Err=0 –Ошибок нет.

Да

Нет

j:=j+1

j > i

Да

Нет

j:=0

i:=i+1

i < p

Нет

Err:=0

p – Число событий на графике

В структуре ошибок нет

Конец

Да

## Автоматизация расчёта параметров сетевого графика

Анализ оптимальности сетевого графика возможно провести, только после расчёта всех, присущих ему параметров. Исходными данными для расчёта явля­ются длительности всех, входящих в сетевой график работ. Результатами расчёта являются значения, описанных в раз­деле 2, параметров. И первое и второе, можно объединить в одной таблице исход­ных данных и результатов 4.1 .

Данная таблица – есть двумерная матрица с пронумерованными строками и столбцами. Номера строк изменяются от 0 до  (см. таб. 4.1 ), где  – число ра­бот в сетевом графике, которое можно найти, подсчитав все единицы в матрице смежностей. Номера столбцов изменяются от 0 до 13, где каждый номер соответ­ствует своему параметру сетевого графика. Нумерация строк и столбцов необхо­дима для представления таблицы исходных данных и результатов в машинной форме.

Столбцы под номерами 0,1 и 2 определяют часть таблицы 4.1 , отведённую под хранение исходных данных, к которым относятся коды работ и длительности работ. Как видно, коды работ задаются ячейками двух столбцов под номерами 0 и 1. Здесь индекс  (столбец 0) определяет номер события, из которого работа исхо­дит, а индекс  (столбец 1) определяет номер события, в которое она входит. Найти все возможные коды работ сетевого графика легко по матрице смежностей , если, просматривая её строки, номера которых соответствуют индексу , выбирать в качестве индекса  номера тех столбцов, для которых будут отыски­ваться единицы.

Алгоритм заполнения таблицы 4.1 исходными данными представлен в виде блок-схемы 4.2 , где ячейки самой таблицы обозначены символом . Для дан­ного обозначения:  – номер строки таблицы исходных данных и результатов,  – номер столбца той же таблицы. Алгоритм предполагает, что таблица исходных данных и результатов  уже зарезервирована и имеет размерность ,  – число работ в сетевом графике.

Таблица 4.1 – Таблица исходных данных и результатов

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | Код ра­боты | | | Параметры работ и событий | | | | | | | | | | | | |
|  | | *i* | *j* | |  |  |  |  |  |  |  |  |  |  |  |  | |
|  | | 0 | 1 | | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | |
| 0 | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | |
| 1 | |  |  | |  |  |  |  |  |  |  |  |  |  |  |  | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| k-1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

N

строки

N Столбца

Блок-схема 4.2 – Алгоритм заполнения исходными данными таблицы исходных данных и результатов

После заполнения таблицы 4.1 исходными данными для расчёта, идёт сле­дующая стадия, – непосредственно, сам расчёт параметров сетевого графика. Данная стадия выполняется в три этапа. На первом этапе осуществляется расчёт сетевого графика в порядке – от начального события до завершающего, и опреде­ляются ранние сроки свершения событий, ранние начала и окончания работ. На втором этапе осуществляется расчёт сетевого графика в обратном порядке – от за­вершающего события до начального, и определяются поздние сроки свершения событий, поздние начала и окончания работ. На третьем этапе осуществляется расчёт резервов времени событий и работ, в произвольном порядке.

j:= j+1

j ≥ p

Нет

Да

i:= i+1

j:=i+1

i < p-1

Нет

Да

Конец

p – Число событий на графике

Начало

i := 0

j := 1

s := 0

Mi j = 1

Нет

Да

As 0 := i

As 1 := j

As 2 := ti j

ti j

Ввод оператором

длительности

работы с кодом i-j

s := s+1

Рассмотрим расчёт параметров сетевого графика на первом этапе.

Ясно, что в общем случае, при попытке определить ранний срок свершения некоторого события, как максимальный из ранних окончаний всех работ, входя­щих в это событие, может быть неудача, так как к этому моменту не все ранние сроки окончаний работ могут быть известны. Тогда, встает задача найти такой по­рядок расчёта сетевого графика, при котором, переходя от события к событию, всегда удаётся находить их ранние сроки свершения. Оказывается, для всех сете­вых графиков, с правильно занумерованными событиями этот порядок один и тот же, и основывается на следующей теореме.

**Теорема 4.1** – Если события сетевого графика занумерованы так, что любая его работа исходит из события с меньшим номером и входит в событие с большим номером, то расчёт ранних сроков свершения событий в порядке: 0-е событие, 1-е, 2-е, и так далее, до завершающего события, в тупик зайти не может, при условии, что рассчитывая ранний срок свершения очередного события, сразу же определя­ются ранние окончания всех, исходящих из него работ.

Докажем эту теорему методом математической индукции.

Зададимся нулевым сроком свершения 0-го события, и рассчитаем ранние окончания всех, исходящих из него работ. Далее. Рассмотрим 1-е событие. В него могут входить только работы, исходящие из событий с меньшими номерами – в данном случае только из 0-го события, при этом ранние окончания этих работ уже известны. Тогда можно рассчитать ранний срок свершения 1-го события. Рассчи­тав ранний срок свершения 1-го события, сразу же рассчитаем ранние окончания всех, исходящих из него работ. Далее. Рассмотрим 2-е событие. В него могут вхо­дить работы, только из 0-го и 1-го события, и ранние окончания которых уже из­вестны. Тогда можем рассчитать ранний срок свершения 2-го события. Рассчитав ранний срок свершения 2-го события, сразу же рассчитаем ранние окончания всех, исходящих из него работ. Далее. Рассмотрим 3-е событие. В него могут входить работы, только из 0-го, 1-го и 2-го события, и ранние окончания которых уже из­вестны. Тогда можем рассчитать ранний срок свершения 3-го события….

Продолжая данные рассуждения, по индукции, рано или поздно дойдём до завершающего события сетевого графика, ранний срок которого окажется возмож­ным рассчитать, так как к этому времени, уже будут известны ранние окончания всех работ сетевого графика. Теорема доказана.

Из данной теоремы, непосредственно, вырисовывается алгоритм расчёта па­раметров сетевого графика на первом этапе. Данный алгоритм представлен в виде блок-схемы 4.3 , и основан на том, что после выполнения алгоритма 4.2 , в таблице исходных данных и результатов  уже находятся коды работ сетевого графика и их длительности.

Блок-схема 4.3 – Алгоритм расчета ранних сроков свершения событий сетевого графика

Начало

max:= 0

i:= 0

max – Хранит ранний срок свершения текущего события

s:=0

As 0 = i

Нет

Да

Цикл расчёта ранних окончаний работ, исходящих из i-го события

As 3:= max

As 7:= max

As 9:= max+As 2

s := s+1

s < k

Нет

Да

k – Число строк в таблице исходных данных и результатов

i:= i+1

i < p

Нет

Да

p –Число событий

max:= -∞

c:= 2

-∞ – Минимально возможное машинное число

c – Счётчик двух циклов для заполнения ячеек A s 5 табл. исходных данных и результатов

As 1= i

Нет

Да

Цикл поиска максимального из ранних окончаний работ, входящих в i-е событие

max < As 9

Нет

Да

max:= As 9

s:= 0

1

2

3

4

5

Конец

1

As 5:= max

2

s := s+1

s < k

Нет

Да

3

c:= c-1

c > 0

Нет

Да

4

5

Рассмотрим расчёт параметров сетевого графика на втором этапе.

В общем случае, при попытке определить поздний срок свершения некото­рого события, как минимальный из поздних начал всех работ, исходящих из этого события, может быть неудача, так как к этому моменту не все поздние сроки начал работ могут быть известны. Тогда, встает задача найти такой порядок расчёта се­тевого графика, при котором, переходя от события к событию, всегда удаётся на­ходить их поздние сроки свершения. Оказывается, для всех сетевых графиков, с правильно занумерованными событиями этот порядок, опять, один и тот же, и ос­новывается на следующей теореме.

**Теорема 4.2** – Если события сетевого графика занумерованы так, что любая его работа исходит из события с меньшим номером и входит в событие с большим номером, то расчёт поздних сроков свершения событий в порядке: последнее со­бытие, предпоследние событие, предшествующее предпоследнему событию, и так далее, до начального (0-го) события, в тупик зайти не может, при условии, что рас­считывая поздний срок свершения очередного события, сразу же определяются поздние начала всех, входящих в него работ.

Докажем эту теорему методом математической индукции.

Зададимся поздним сроком свершения последнего события, равным его ран­нему сроку свершения, и рассчитаем поздние начала всех, входящих в него работ. Далее. Рассмотрим предпоследнее событие. Из него могут исходит только работы, входящие в события с большими номерами – в данном случае только в последнее событие, при этом поздние начала этих работ уже известны. Тогда можно рассчи­тать поздний срок свершения предпоследнего события. Рассчитав поздний срок свершения предпоследнего события, сразу же рассчитаем поздние начала всех, входящих в него работ. Далее. Рассмотрим событие, предшествующее предпо­следнему. Из него могут исходить работы, только в предпоследнее и в последнее событие, и поздние начала которых уже известны. Тогда можем рассчитать позд­ний срок свершения события, предшествующего предпоследнему….

Продолжая данные рассуждения, по индукции, рано или поздно дойдём до начального события сетевого графика, поздний срок которого окажется возмож­ным рассчитать, так как к этому времени, уже будут известны поздние начала всех работ сетевого графика. Теорема доказана.

Из данной теоремы, непосредственно, следует алгоритм расчёта параметров сетевого графика на втором этапе. Данный алгоритм представлен в виде блок-схемы 4.4 , и основан на том, что после выполнения алгоритма 4.3 , в таблице ис­ходных данных и результатов  уже рассчитаны все ранние сроки свершения событий.

Блок-схема 4.4 – Алгоритм расчёта поздних сроков свершения событий сетевого графика

Начало

min:= Ak-1, 5

j:= p-1

min – Хранит поздний срок свершения текущего события

s:=k

Цикл расчёта поздних начал работ, входящих в j-е событие

k – Число строк в таблице исходных данных и результатов

s:= s -1

As 1 = j

Нет

Да

As 6:= min

As 10:= min

As 8:= min-As 2

s > 0

Нет

Да

j:= j-1

j < 0

Нет

Да

Конец

min:= +∞

c:= 2

+∞ – Максимально возможное машинное число

c – Счётчик двух циклов для заполнения ячеек As 4 табл. исходных данных и результатов

s:= k

s:= s-1

As 0 = j

Нет

Да

min > As 8

Нет

Да

min:= As 8

2

1

3

4

5

Цикл поиска минимального из поздних начал работ, исходящих из j-го события

Рассмотрим расчёт параметров сетевого графика на третьем этапе.

1

2

3

4

5

As 4:= min

s > 0

Нет

Да

c:= c-1

c > 0

Нет

Да

Если, сначала выполнить алгоритм расчёта ранних сроков свершения собы­тий 4.3 , а затем алгоритм расчёта поздних сроков свершения 4.4 , то в таблице ис­ходных данных и результатов останутся не заполненными только три последних столбца, с номерами: 11, 12 и 13. Данные столбцы, как видно из таблицы 4.1 , отве­дены под расчёт резервов времени сетевого графика. Расчёт резервов времени се­тевого графика можно осуществить в любом порядке строк таблицы исходных данных и результатов, например, подряд – с 0-й строки по последнюю. Такой по­рядок расчёта представлен ниже, в виде блок-схемы 4.5 . Данный алгоритм явля­ется завершающим для процесса расчёта параметров сетевого графика, после вы­полнения которого, все ячейки таблицы исходных данных и результатов 4.1 , будут заполнены значениями соответствующих параметров.

Блок-схема 4.5 – Алгоритм расчёта резервов времени сетевого графика

Начало

s:=0

As 11:= As 10 – As 9

As 12:= As 5 – As 9

As 13:= As 4 – As 3

s:= s +1

s < k

Нет

Да

Конец

## Автоматизация процесса поиска особых путей сетевого графика

Как уже известно, найти особые пути сетевого графика представляется воз­можным только, если будут рассчитаны полные резервы времени всех, входящих в него работ. Тогда, перед поиском особых путей, необходимо выполнять, описан­ные в предыдущем подразделе алгоритмы по расчёту параметров сетевого гра­фика.

Из раздела 3 ясно, что для поиска, и критического пути и наикратчайшего, возможно использовать одну и туже методику. Данная методика заключается в по­следовательном выборе, от 0-го события до завершающего, тех работ, которые имеют нулевые полные резервы времени. В случае, если параметры сетевого гра­фика рассчитывались для положительных длительностей, входящих в него работ, то указанная методика даёт критический путь сетевого графика. Если же пара­метры рассчитывались при отрицательных длительностях работ, то методика даст наикратчайший путь сетевого графика.

Алгоритм, реализующий методику поиска особого пути сетевого графика, представлен в виде блок-схемы 4.6 , и основан на том, что таблица исходных дан­ных и результатов  уже полностью рассчитана, либо при положительных, либо при отрицательных длительностях работ.

Имея в арсенале, все рассмотренные в данном разделе алгоритмы, любому программисту не составит труда объединить их в одну, общую программу анализа оптимальности сетевого графика по критерию оптимальности, подробно описан­ному в разделе 1. Проверка данного критерия, с целью выявления оптимальности сетевого графика, на столько проста в алгоритмической реализации, что специ­ального рассмотрения не требует.

Блок-схема 4.6 – Алгоритм поиска особого пути сетевого графика

Начало

i:= 0

s:= 0

i < p – 1

Нет

Да

As 0 = i

Нет

Да

As 11 = 0

Нет

Да

Вывод кода работы особого пути:

< i, As 1 >

i:= As 1

s:= s+1

Конец

Заключение

В данном курсовом проекте были предложены и обоснованы рациональные методики поиска особых путей сетевых графиков. Рациональность данных мето­дик заключается в том, что они позволяют найти критический и наикротчайший пути сетевого графика без перебора всех возможных вариантов. Последнее, позво­ляет в короткие сроки осуществить решение двух основных задач сетевого плани­рования: задачу анализа оптимальности уже готового сетевого графика и задачу его оптимизации по длительности, в случае, если сетевой график оказывается не оптимальным.

Кроме того, в курсовом проекте были рассмотрены вопросы автоматизации на ЭВМ рациональных методик поиска особых путей сетевого графика. В резуль­тате – разработаны блок схемы алгоритмов расчёта параметров сетевых графиков и поиска их особых путей, которые предполагается использовать при создании конкретной программы анализа оптимальности сетевых графиков на любом из из­вестных языках программирования.

Значимость проделанной работы заключается в том, что применение пред­ложенных методик, во-первых – позволяет точно судить об оптимальности сете­вых графиков любой сложности, а во-вторых – сокращает затраты на сетевое пла­нирование в целом, прежде всего, за счёт сокращения длительности разработки оптимальных сетевых графиков.

Список использованных источников

Технико-экономическое обоснование дипломных проектов проектов: Учеб. Пособие для втузов / Л. А. Астреина, В. В. Балдесов, В. К. Беклешов и др.; Под ред. В. К. Беклешова. – М.: Высш. Шк., 1991. – 176 c.: ил.