А.В. Полянский

2021

Автоматизированное решение задач проектного управления железнодорожным строительством

Учебное пособие для студентов специальности 23.05.06 «Строительство железных дорог, мостов и транспортных тоннелей»

Учебное пособие

МИР НАУКИ
ИЗДАТЕЛЬСТВО

УДК 625.1
ББК 39.2
П 545

Рецензенты: Ступникова Елена Анатольевна – кандидат экономических наук, доцент, доцент кафедры «Экономика транспортной инфраструктуры и управление строительным бизнесом» ФГАОУ ВО «Российский университет транспорта».

Польшин Максим Вячеславович – кандидат технических наук, доцент кафедры «Изыскания, проектирование и строительство железных дорог» ФГБОУ ВО «Ростовский государственный университет путей сообщения».

Полянский, Алексей Викторович
П 545 Автоматизированное решение задач проектного управления железнодорожным строительством. Учебное пособие. – М.: Мир науки, 2021. – Сетевое издание. Режим доступа: https://izd-mn.com/PDF/21MNNPU21.pdf – Загл. с экрана.

ISBN 978-5-6045772-0-2

В учебном пособии содержатся теоретические и практические материалы, необходимые для автоматизированного решения задач проектного управления железнодорожным строительством с применением системы Mathcad. Приведены основные приемы работы с системой Mathcad в объеме достаточном для решения рассматриваемых задач. Рассмотрены особенности решения задач управления очередностью реализации проектов реконструкции железных дорог при линейной и нелинейной зависимостях продолжительности от стоимости. Представлены методики принятия решений в условиях риска при управлении строительными проектами и расчет сетевого графика методами CPM и PERT в автоматизированном режиме.

Учебное пособие предназначено для студентов специальности «Строительство железных дорог, мостов и транспортных тоннелей», специализаций «Строительство магистральных железных дорог» и «Управление техническим состоянием пути». Пособие может быть использовано при изучении дисциплин: «Автоматизированные системы управления строительством», «Модели и методы инженерных расчетов», «Управление организационно-технологической надежностью строительства» а также может быть полезно аспирантам, научным работникам и специалистам проектных и строительных организаций.

Табл. 19. рис. 28, библиогр. 7 назв.

ISBN 978-5-6045772-0-2

Оглавление

Введение

С каждым годом компьютерные технологии все чаще становятся важным фактором, а также средством решения различных производственных задач. Бурное развитие кибернетики, теории управления и исследования операций позволило создать ряд формальных моделей и тем самым заложить систематическую научную основу проектного управления. Возрастающая сложность проектов, с одной стороны, и накопленный опыт управления, с другой, сделали необходимым и возможным создание идеологии и методологии проектного управления.

Понятие «проект» долгое время монопольно использовалось инженерами и было связано с представлением о комплекте технической и сметной документации, необходимой для строительства, в том числе железнодорожного. Теперь же понятие проект используется не только инженерами, но и финансистами, экономистами, предпринимателями, учеными - всеми теми людьми, которые задумывают и воплощают в жизнь намерения с заранее установленными целями и требованиями к срокам, стоимости, риску и качеству ожидаемых результатов. Это и есть проект.

Проектное управление (*Project Management*) – раздел теории управления социально-экономическими системами, изучающий методы, формы, средства и т.д. наиболее эффективного и рационального управления изменениями. В то же время – это искусство руководства и координации усилий людей и использования ресурсов с применением достижений современной науки и информационных технологий для успешного осуществления целей проекта по результатам, стоимости, времени и качеству и, что тоже немаловажно, удовлетворения участников проекта.

Задача учебного пособия - помочь в освоении основных технических приемов при принятии управленческих решений. Не нужно думать, что овладение этими приемами сразу же сделает обучающегося специалистом в области проектного управления железнодорожным строительством. Подготовка квалифицированных специалистов требует развития у них навыков обоснования и принятия управленческих решений. Эти навыки нужны не только будущим ученым, но и производственникам.

Изложение материала построено таким образом, что все лабораторные работы могут выполняться независимо друг от друга и, значит, изучаться и выполняться в любой последовательности, удобной преподавателю и студенту.

Опыт показывает, что материал, изложенный по структуре «теоретические положения – пример решения – проверка на ЭВМ» усваивается намного лучше. Поэтому структура и задания построены таким образом, чтобы студент выполнил соответствующий вариант вручную, а проверку расчетов выполнил на ЭВМ по специально разработанным автором программам.

Для автоматизации решения задач проектного управления выбрана система *Mathcad* фирмы *PTC*. Привлекательность этой системы для большинства пользователей кроется в том, что в ней описание решения управленческих задач дается с помощью привычных математических формул и знаков. Такой же вид имеют и результаты вычислений. Систему *Mathcad* отличает простой и удобный пользовательский интерфейс, позволяющий организовать интуитивно предсказуемый диалог с системой на общепринятом языке математических формул. Этим и многим другим объясняется заслуженная популярность системы *Mathcad* в научных и инженерных исследованиях.

Настоящее учебное пособие предназначено для студентов специальности 23.05.06 «Строительство железных дорог, мостов и транспортных тоннелей». При написании работы автор исходил из того, что студенты данной специальности при изучении на первом курсе дисциплины «Информатика» получают первичные знания о системе *Mathcad*, приобретают и закрепляют практические навыки работы с данным программным продуктом. Поэтому основное внимание в первой главе уделено описанию пользовательского интерфейса, входного языка системы, особенностям программирования в *Mathcad*.

Современные требования, предъявляемые к инженерам путей сообщения, предполагают знание современных информационных технологий, новейших программных средств, нацеленных на решение конкретных задач управления железнодорожным строительством, и умение применить эти знания в практической деятельности.

Глава 1. Теоретический курс

1.1. Сведения о системе компьютерной математики Mathcad

Mathcad – это популярная система компьютерной математики, предназначенная для автоматизации решения массовых математических задач в самых различных областях науки, техники, экономики, строительства, статистики, организации производства и управления... Название системы происходит от двух слов – *MATHematica* (математика) и *CAD* (*Computer Aided Design* – системы автоматизированного проектирования).

В настоящее время различные версии *Mathcad* являются математически ориентированными универсальными системами. Помимо собственно вычислений, как численных, так и аналитических, они позволяют решать сложные оформительские задачи, которые с трудом даются популярным текстовым редакторам или электронным таблицам. В то же время система *Mathcad* является превосходным средством для математического моделирования.

Оценивая функциональные возможности системы *Mathcad*, следует выделить следующие особенности:

- сравнительная простота;
- отсутствие высоких требований к пользователю как к программисту;
- возможность преобразования и представления данных в различных форматах;
- широкий набор «шаблонов» (предварительно запрограммированных на языке *C++* процедур) и встроенных функций для решения любых математических задач (решения систем алгебраических и дифференциальных уравнений, неравенств, циклических и рекуррентных процедур и пр.), визуализации результатов с помощью 2D и 3D-графиков;
- достаточно мощный арсенал операторов символьного преобразования математических выражений;
- возможность формирования пояснительных записок, отчетов, статей путем совместного использования текстовых редакторов и фрагментов документов *Mathcad*.

Таким образом, все вышеперечисленное позволяет заключить, что система *Mathcad* представляет собой мощный инструмент для инженерных расчетов и построения математических моделей в различных отраслях знаний.

1.2. Пользовательский интерфейс Mathcad

1.2.1. Особенности пользовательского интерфейса

Интерфейс системы *Mathcad* внешне весьма близок к интерфейсу известных текстовых редакторов *MS Word*. Поэтому в дальнейшем пользовательский интерфейс рассматриваемых систем описывается в сокращенном виде. В дальнейшем все то, что изображено на экране компьютера в конкретной системе *Mathcad* будет называться *документом Mathcad*.

После запуска системы или при создании нового документа можно видеть неименованную рабочую область с панелью инструментов, готовую к введению математических выражений и текстовых вставок, графиков, вычислениям или редактированию. Фрагмент окна *Mathcad* представлен на рисунке 1.1; в левой части этого рисунка виден крестик *визира*.

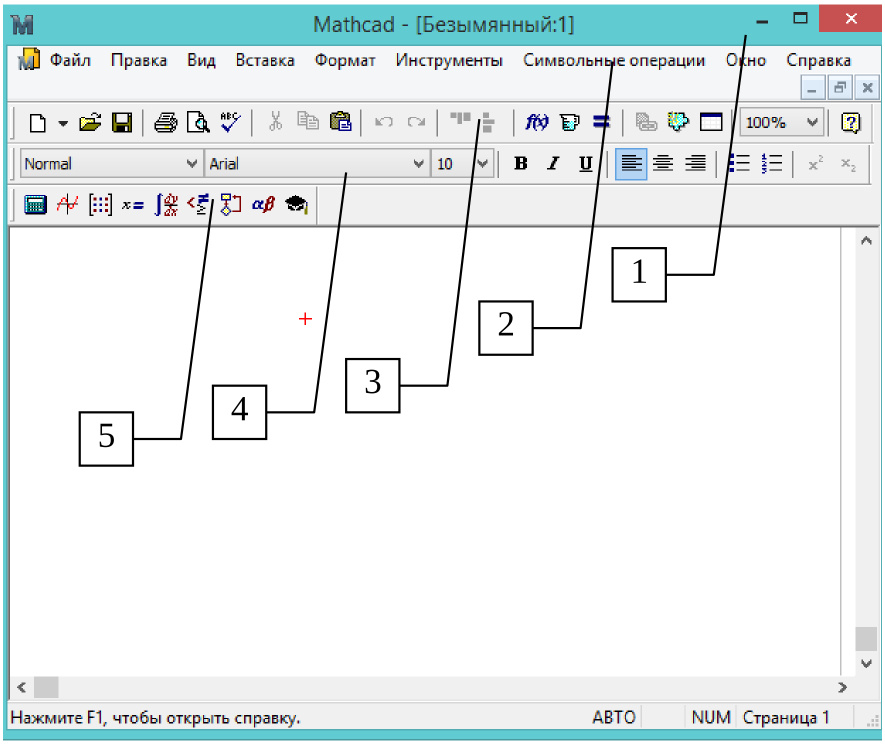


Рисунок 1.1 – Окно системы *Mathcad*

В верхней части экрана (рисунок 1.2) по умолчанию имеются четыре строки, содержащие соответственно:

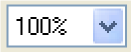
1. Название документа, кнопки управления окном.
2. Главное меню.
3. Панель выбора масштаба, копирования, вставки, печати, вызова справки и пр. – **Стандартная**.
4. Панель форматирования (выбор шрифтов, управления размещением фрагментов документа и пр.) – **Форматирование**.
5. Панель инструментов – **Математические**.

Две нижние строки могут быть скрыты или перемещены на другую позицию. Первая строка не отличается от аналогичной строки для редактора *MS Word*.

Главное меню систем, помимо стандартных для *Windows* позиций (**Файл**, **Правка**, **Вид**, **Вставка**, **Формат**, **Окно**, **Справка** имеет еще две позиции – **Инструменты** и **Символьные операции**. Содержание стандартных для *Windows* и дополнительных позиций достаточно подробно рассматривается в [1-4]. Далее будут приведены начальные сведения работы в системе *Mathcad*, позволяющие еще неопытному пользователю сделать свои первые шаги и достичь результата.

1.2.2. Панели инструментов

Панели инструментов позволяют выполнять наиболее часто используемые команды щелчком левой кнопки мыши по соответствующей кнопке-пиктограмме. Благодаря этому становится ненужным утомительный поиск в меню наиболее часто используемых команд. Панели можно расположить друг под другом сразу под строкой меню, как это показано на рисунке 1.2. Для того чтобы установить панель в нужном месте экрана, достаточно щелкнуть на ее названии и, удерживая левую кнопку мыши, переместить указатель, а вместе с ним и панель, в это место.

<u>Панель инструментов **Стандартная**</u>**.** Панель инструментов **Стандартная** содержит 20 кнопок и один раскрывающийся список 

для выбора масштаба изображения документа *Mathcad* на экране (рисунок 1.2).

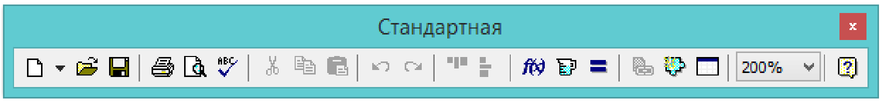


Рисунок 1.2 – Панель инструментов **Стандартная**

Далее приведены соответствия кнопок этой панели командам меню:

 - создание нового документа. Соответствует команде **Создать** меню **Файл** без вызова диалогового окна **Создать**;

 - загрузка ранее созданного документа в виде файла. Соответствует команде **Открыть** меню **Файл**;

 - запись текущего документа в файл под текущим именем. Соответствует команде **Сохранить** меню **Файл**;

 - печать документа на принтере. Соответствует команде **Печать** меню **Файл**;

 - предварительный просмотр документа перед печатью в том виде в котором он будет распечатан. Соответствует команде **Предварительный просмотр** меню **Файл**;

 - проверка орфографии (действует только для англоязычных документов). Соответствует команде **Проверка орфографии** меню **Правка**;

 - перенос выделенной части документа в буфер обмена с удалением этой части из документа. Соответствует команде **Вырезать** меню **Правка**;

 - копирование выделенной части документа в буфер обмена с сохранением этой части в документе. Соответствует команде **Копировать** меню **Правка**;

 - перенос содержимого буфера обмена в окно редактирования на место, в котором находится визир. Соответствует команде **Вставить** меню **Правка**;

 - отмена предшествующей операции редактирования. Соответствует команде **Отменить** меню **Правка**;

 - восстановление отмененной операции редактирования. Соответствует команде **Повторить** меню **Правка**;

 - выравнивание выделенных блоков по горизонтали. Соответствует команде **Выровнять регионы/По верхнему краю** меню **Формат**;

 - выравнивание выделенных блоков по вертикали. Соответствует команде **Выровнять регионы/По левому краю** меню **Формат**;

 - вставка функции из списка, появляющегося в диалоговом окне. Соответствует команде **Функция…** меню **Вставка**;

 - вставка единиц измерения. Соответствует команде **Единица измерения…** меню **Вставка**;

 - вычисление выделенного выражения. Соответствует команде **Вычислить** меню **Инструменты**;

 - вставка гиперссылок на файлы. Соответствует команде **Гиперссылка…** меню **Вставка**;

 - вставка компонентов других систем. Соответствует команде **Компонент…** меню **Вставка**;

 - вставка электронной таблицы. Соответствует команде **Данные/Таблица** меню **Вставка**;

 - запуск справочной базы данных.

<u>Панель инструментов **Форматирование**</u>. Панель инструментов **Форматирование** содержит 10 кнопок и 3 раскрывающихся списка (рисунок 1.3):

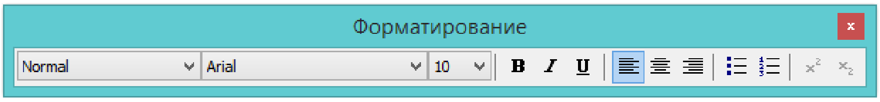
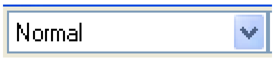


Рисунок 1.3 – Панель инструментов **Форматирование**

 - раскрывающийся список изменения текущего стиля документа, переменных или констант;

 - раскрывающийся список шрифтов, используемых при отображении констант, переменных или текста;

 - раскрывающийся список для отображения и изменения размеров символов в пунктах при отображении констант, переменных или текста;

 - полужирное начертание символов при отображении констант, переменных и текста;

 - наклонное начертание символов при отображении констант, переменных и текста;

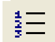
 - подчеркнутое начертание символов при отображении констант, переменных и текста;

 - выравнивание текста по левой границе текстового блока;

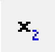
 - выравнивание текста по центру текстового блока;

 - выравнивание текста по правой границе текстового блока;

 - создание ненумерованного списка в текстовых блоках;

 - создание нумерованного списка в текстовых блоках;

 Вставка верхнего индекса в текстовой области

 Вставка нижнего индекса в текстовой области

<u>Панель инструментов **Математические**</u>. Панель инструментов **Математические** содержит 9 кнопок для вызова подпанелей (рисунок 1.4).

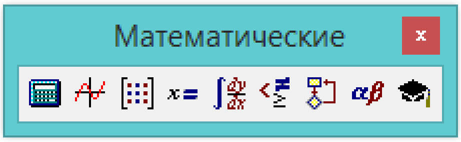
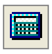


Рисунок 1.4 – Панель инструментов **Математические**

Кнопка  вызывает подпанель **Калькулятор** (рисунок 1.5) для ввода арифметических операций и некоторых наиболее часто используемых функций.

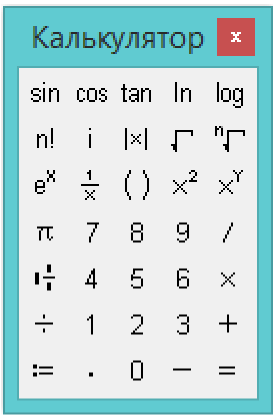


Рисунок 1.5 – Подпанель **Калькулятор**

Кнопка  вызывает подпанель **График** (рисунок 1.6) для построения двумерных и трехмерных графиков.

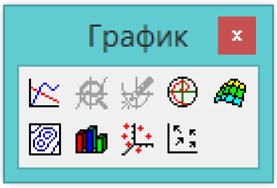


Рисунок 1.6 – Подпанель **График**

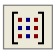
Кнопка  вызывает подпанель **Матрица** (рисунок 1.7) для ввода и обработки матриц и векторов.

Рисунок 1.7 – Подпанель **Матрица**

Кнопка  вызывает подпанель **Вычисление** (рисунок 1.8) для ввода знаков присваивания и равенства, а также для задания собственных операторов различных видов.

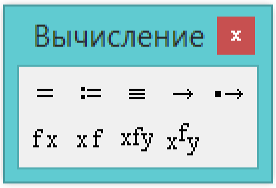
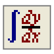


Рисунок 1.8 – Подпанель **Вычисление**

Кнопка  вызывает подпанель **Математический анализ** (рисунок 1.9) для вычисления производных, интегралов, сумм, произведений и пределов.

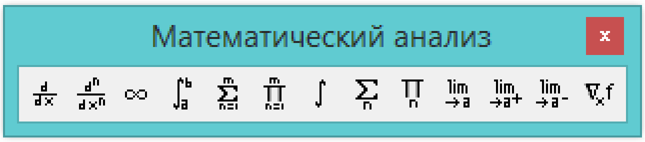
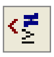


Рисунок 1.9 – Подпанель **Математический анализ**

Кнопка  вызывает подпанель **Булева алгебра** (рисунок 1.10) для ввода логических операторов булевой алгебры.

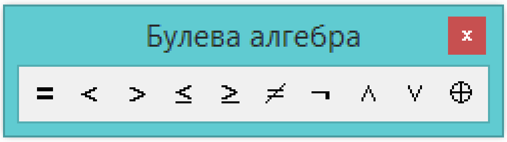
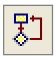


Рисунок 1.10 – Подпанель **Булева алгебра**

Кнопка  вызывает подпанель **Программирование** (рисунок 1.11) для ввода операторов программирования.

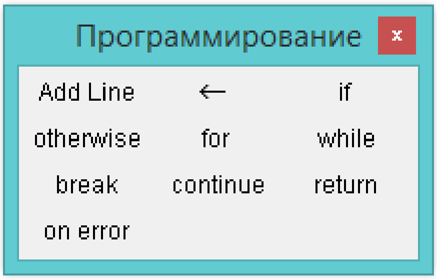


Рисунок 1.11 – Подпанель **Программирование**

Кнопка  вызывает подпанель **Греческий** (рисунок 1.12) для ввода греческих букв.

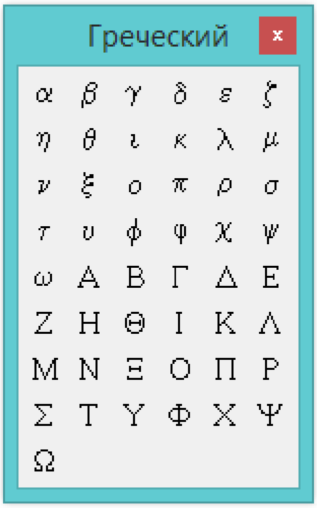


Рисунок 1.12 – Подпанель **Греческий**

Кнопка  вызывает подпанель **Символьные** (рисунок 1.13) для символьных вычислений.

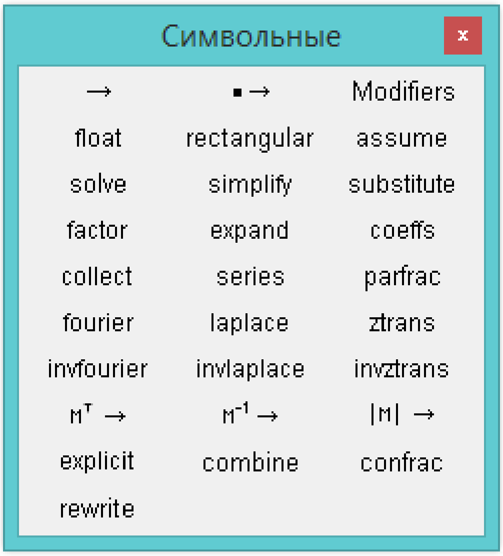


Рисунок. 1.13 – Подпанель **Символьные**

1.2.3. Окно редактирования

После вызова системы на чистом поле документа *Mathcad* можно увидеть обычный стрелочный курсор (курсор мыши), перемещение которого осуществляется мышью. Основа всех последующих вычислений закладывается на этапе ввода данных, формульных зависимостей и пр.

В распоряжении пользователя в системах *Mathcad* имеются:

- визир для выбора текущего места документа;
- уголковый курсор при вводе и редактировании математических выражений;
- курсор текстовых фрагментов.

Процедура ввода начинается в точке экрана (точке документа), которая выбирается из соображений удобства последующей работы. Определение места на экране, в которое будут вводится математические выражения или текстовые вставки, осуществляется **визиром**, имеющим вид крестика (см. рисунок 1.1). При вызове системы или создании нового документа визир по умолчанию располагается в левом верхнем углу экрана. Перемещение визира производится стрелками клавиатуры или щелчком мыши в произвольной точке экрана.

Перед началом операций ввода математических выражений, функций, матриц, шаблонов графиков, текстовых фрагментов и пр., в выбранной (с помощью обычного стрелочного курсора) точке экрана производится щелчок левой кнопкой мыши. В этом месте появляется визир (см. рисунок 1.1). Последующие нажатия на кнопки клавиатуры (латинские символы, цифры) будут восприниматься системой в качестве элементов математических выражений. Отнесение вводимых элементов к математическим выражениям объясняется тем, что по умолчанию система настроена на так называемую «математическую область» документа, в которой формируются фрагменты алгоритмов и осуществляются вычисления или символьные преобразования математических выражений. Для перехода к вводу текстовых фрагментов необходимо осуществить переключение режима работы системы. Для этого активизируется позиция **Вставка/Регион текста** главного меню. После завершения ввода текстового фрагмента щелчком мыши в выбранной точке документа производится обратное переключение режима (активизируется позиция **Вставка/Регион формул** главного меню); система готова к вводу математических выражений на месте появившегося визира.

При вводе на экране появляется последовательность вводимых символов, цифр, операторов и пр., которые заключены в черную рамку. Наличие рамки означает, что этот фрагмент документа в настоящее время является активным; рамка исчезает при щелчке мыши в другой (свободной от выражений) зоне документа.

Ввод первой цифры или первого символа математических выражений (внутри черной рамки) сопровождается появлением так называемого «синего **уголкового курсора»,** имеющего горизонтальную линию, расположенную слева или справа от вертикальной линии. В качестве примера рассмотрена последовательность действий при вводе простого выражения $(2a+3)c^3$. Особо следует отметить необходимость ввода операций умножения в процессе набора, отсутствие которых, возможное в текстовых описаниях, недопустимо в исполнимых математических выражениях.

Последовательность нажатий клавиш и результаты каждого нажатия приведены в таблице 1.1.

Следует обратить внимание на положение горизонтальной линии уголкового курсора. От этого положения зависит в какую сторону будет «развиваться» фрагмент, в котором осуществляется ввод - влево или вправо. Система «по умолчанию» предполагает естественное развитие выражения вправо, поэтому на различных этапах (см. таблицу 1.1) уголок курсора имеет преимущественное левостороннее состояние, при котором он охватывает фрагмент справа.

Уголковый курсор в системах *Mathcad* занимает исключительно важное место. С его помощью выполняется множество подготовительных операций при вводе и форматировании выражений, осуществляется копирование и удаление фрагментов и пр.

Таблица 1.1 - Последовательность нажатий клавиш и результаты нажатия

Клавиши	Получаемый результат	Примечание
(	$($	Появление маркера ввода и уголкового курсора (УК)
2	$(2$	УК устанавливается в позицию для развития выражения вправо
*****	$(2 \cdot$	Система предлагает ввести сомножитель
a	$(2 \cdot a$	
+	$(2 \cdot a +$	Система предлагает ввести слагаемое
3	$(2 \cdot a + 3$	
)	$(2 \cdot a + 3)$	Уголковый курсор «охватывает» всю скобку, рассматривая ее как единый элемент математического выражения
*****	$(2 \cdot a + 3) \cdot$	Система предлагает ввести сомножитель
b	$(2 \cdot a + 3) \cdot c$	
^	$(2 \cdot a + 3) \cdot c$	Система предлагает ввести степень
3	$(2 \cdot a + 3) \cdot c^{3}$	

Важную роль играет **положение горизонтальной линии** уголкового курсора - правое или левое. При левом положении, когда уголковый курсор имеет форму ⌋, набор выражения может быть осуществлен вправо от вертикальной линии курсора. Ниже приведены копии двух последовательных этапов набора математического выражения.

Этап 1: $(2 \cdot a + 3)$; этап 2: $(2 \cdot a + 3) \cdot c^{3}$.

На первом этапе уголковый курсор имеет левостороннее положение и охватывает всю скобку. Это позволяет умножить эту скобку на множитель справа.

В том случае, когда уголковый курсор имеет форму ⎣, дальнейшее развитие выражения возможно только влево от вертикальной линии курсора:

Этап 1: $(2 \cdot a + 3)$; этап 2: $c \cdot (2 \cdot a + 3)$

В этих простых примерах зафиксировано конечное положение уголкового курсора, соответствующее вводу последнего элемента выражения.

Изменение положения уголкового курсора возможно: стрелками клавиатуры, щелчком мыши в нужной точке выражения, клавишей «**Пробел**», сочетаниями перечисленных способов. Эти способы сильно отличаются удобством практического использования. Например, для перевода положения курсора из левостороннего, охватывающего все выражение (2*a*+3), в соответствующее правостороннее (также с полным охватом скобки), потребуется семь раз нажать на стрелку ←. Приведем положения курсора при таких последовательных нажатиях стрелки ←, начиная с первого, в копиях из документов *Mathcad*:

$(2 \cdot a + 3)$, $(2 \cdot a + 3)$, $(2 \cdot a + 3)$, $(2 \cdot a + 3)$, $(2 \cdot a + 3)$, $(2 \cdot a + 3)$, $(2 \cdot a + 3)$

Приведенный пример показывает, что действия одними стрелками крайне нерациональны для достижения требуемого положения уголкового курсора. В то же время в более сложных выражениях, включающих дроби, показатели степеней и пр., этот путь вообще может не привести к желаемому результату, поскольку последовательно перемещаемый стрелками клавиатуры курсор будет «обходить» все элементы выражения, образуя порою циклические переходы.

Значительно более рациональным, простым и удобным способом перемещения уголкового курсора служит комбинация действий стрелками, мышью и клавишей «**Пробел**». При этом мышью достигается требуемое (правое или левое) положение курсора с охватом любого элемента выражения, а нажатие клавиши «**Пробел**» увеличивает область охвата без изменения направления его горизонтальной линии. Основной смысл этой комбинации действий заключается в том, чтобы мышью добиться нужного положения курсора и расширить затем зону охвата клавишей «**Пробел**» до требуемого уровня. Например, перевод положения уголкового курсора в обратное может быть осуществлен этим способом в два этапа. На первом мышью выбирается точка слева от любого из элементов выражения и осуществляется щелчок. Этот элемент оказывается охваченным уголковым курсором слева. На втором этапе осуществляется последовательное нажатие клавиши «**Пробел**» до достижения нужной зоны охвата. Рассмотренный выше пример с переводом положения курсора в обратное может быть решен комбинированным способом так:

1. В исходном выражении (2a + 3) делается щелчок мышью слева от, например, тройки. Получаем следующий вариант:

$(2 \cdot a + 3)$.

2. Однократное нажатие клавиши «**Пробел**» в этом случае позволяет расширить зону охвата до полной и достичь требуемого положения уголкового курсора:

$(2 \cdot a + 3)$.

В ряде случаев не всегда удается сразу получить нужное положение уголкового курсора. Тогда, получая при щелчке, например, левостороннее положение курсора, изменяем его положение на обратное однократным нажатием стрелки ←, после чего можно расширить зону охвата элементов выражения клавишей «**Пробел**». Подобный вариант для нашего примера будет включать последовательность действий, приведенную в таблице 1.2:

Таблица 1.2 – Последовательность действий

Действие	Получаемый результат
Щелчок мыши справа от тройки	$(2\cdot a+3)$
Нажатие стрелки ←	$(2\cdot a+3)$
Нажатие клавиши «**Пробел**»	$(2\cdot a+3)$

Подчеркивая значение уголкового курсора в системах *Mathcad*, необходимо отметить, что от его состояния (положения и зоны охвата) зависят возможности копирования элементов выражений в буферную память и удаление элементов.

Например, в документе имеется некоторое математическое выражение, часть которого необходимо скопировать для последующего использования в других фрагментах. Для этого выбранная часть должна быть предварительно полностью охвачена уголковым курсором произвольного положения (с левым или правым расположением горизонтальной линии). Это значит, что выбранная часть выражения должна располагаться между вертикальной линией уголкового курсора и концом его горизонтальной линии. Таким образом, копировать можно только то, что охвачено уголковым курсором. Далее показан процесс выбора части математического выражения на примере рассмотренных ранее выражений.

Пусть в выражении $(2a+3)c^3$ требуется скопировать правый сомножитель.

Это значит, что уголковый курсор должен занимать только одно из двух следующих положений:

$(2\cdot a+3)\cdot c^3$ или $(2\cdot a+3)\cdot c^3$.

В положении уголкового курсора

$(2\cdot a+3)\cdot c^3$ или $(2\cdot a+3)\cdot c^3$.

в буферную память при копировании будет введен лишь параметр *c*.

При удалении части выражения необходимо также внимательно наблюдать за состоянием уголкового курсора. Удалить можно лишь ту часть, которая полностью охвачена уголком курсора. При этом для левостороннего его положения при удалении нужно пользоваться клавишей «**Backspace**», для правостороннего – клавишей «**Del**». Для удаления части выражения необходимо:

- нажать клавишу «**Backspace**» или «**Del**» в соответствии с положением уголкового курсора; при этом охваченная уголком курсора часть выражения закрашивается в черный цвет;
- дважды нажать клавишу «**Del**».

Пример подобных действий приведен в таблице 1.3:

Таблица 1.3 – Последовательность действий

Действие	Получаемый результат
Исходное положение курсора (удаляется правый сомножитель)	$(2\cdot a+3)\cdot c^3$
Нажатие клавиши «**Backspace**»	$(2\cdot a+3)\cdot c^3$
Первое нажатие клавиши «**Del**»	$(2\cdot a+3)\cdot \blacksquare$
Второе нажатие клавиши «**Del**»	$(2\cdot a+3)$

Аналогичные действия следует осуществить в нашем примере, если исходное состояние уголкового курсора – правостороннее. В этом случае потребуется трижды использовать клавишу «**Del**» – для выделения удаляемой части черным цветом и дважды – для окончательного удаления. Последовательность результатов при каждом нажатии «**Del**» будет иметь вид (первый фрагмент – исходное положение):

$$(2\cdot a+3)\cdot c^3,\quad (2\cdot a+3)\cdot c^3,\quad (2\cdot a+3)\cdot \blacksquare,\quad (2\cdot a+3).$$

Следует обратить внимание на промежуточную стадию, в которой появляется черная прямоугольная точка – так называемый маркер ввода. Этот маркер охвачен курсором. Появление маркера ввода означает, что система готова к вводу в этом месте новых элементов выражения, т.е. предлагается ввести на место маркера ввода какой-либо фрагмент математического выражения или удалить маркер.

Из представленных примеров следует, что процесс удаления клавишами сопровождается обязательным появлением на промежуточной стадии черной заливки удаляемой части. Эта заливка может быть вызвана и протяжкой мыши по этой части выражения, чем ускоряется процедура удаления.

При редактировании математических фрагментов документа щелчком мыши в выбранной точке выражения необходимо сделать активным весь фрагмент. Это отмечается системой появлением черной рамки вокруг всего фрагмента и появлением уголкового курсора в точке щелчка. Далее осуществляются операции с активным фрагментом, часть из которых (перемещение курсора, копирование, удаление, возможности коррекции элементов) рассмотрена выше.

Техника удаления всего фрагмента или группы фрагментов имеет незначительные особенности. Единичный активный фрагмент (отмеченный черной рамкой) может быть полностью удален двумя основными способами: набором клавиш «**Ctrl** + **D**», протягиванием мыши (при нажатой рабочей кнопке) по зоне фрагмента до появления черной заливки всего фрагмента с последующим нажатием клавиши «**Del**». Если после черной заливки фрагмента щелкнуть правой кнопкой мыши и выбрать режим *Copy*, отмеченный черной заливкой фрагмент может быть скопирован в буферную память для последующего использования. Выбирая при этом режим *Cut*, копируем фрагмент с удалением из документа. Все эти операции являются обычными для *MS Windows*, поэтому они здесь лишь упоминаются.

Группу фрагментов можно удалить (с помещением в буферную память или нет) и скопировать, предварительно протянув мышь по фрагментам, составляющим группу. При этом все фрагменты группы получают пунктирные рамки, появление которых свидетельствует о том, что вся группа рассматривается системой как единое целое и она может быть удалена (клавишей «**Del**»), скопирована и вырезана (*Copy* и *Cut*) или перемещена без изменения взаимного расположения фрагментов (протягиванием мыши при ее фиксации на одном из фрагментов). Перемещение происходит после появления изображения руки на одном из фрагментов, после чего протягиванием мыши (нажав рабочую кнопку мыши и не отпуская ее), производим перемещение группы в нужную зону документа.

Для перемещения одиночного фрагмента необходимо: активизировать его (появляется черная рамка); найти мышью позицию на границе рамки, при которой появится изображение руки; протягиванием мыши переместить фрагмент в нужную зону документа.

Совершенно аналогичные операции могут быть осуществлены при удалении или перемещении и других типов фрагментов – текстовых и графических.

Завершая общее описание работы с математическими фрагментами в системах *Mathcad*, отметим лишь некоторую особенность выбора типа и размера используемого шрифта. В системах *Mathcad* предусмотрена процедура раздельного выбора шрифта для символов и цифр математических выражений. При этом шрифт может быть установлен применительно к любому из математических фрагментов документа, но будет действовать для всех фрагментов в пределах рассматриваемого документа. Отдельная установка шрифта для каждого математического фрагмента не предусмотрена. В пределах выбранного типа и размера шрифта может быть отдельно установлен стиль (полужирный, курсив, подчеркивание) для символов и цифр.

При выборе типа и размера шрифта символов математических элементов необходимо:

- щелкнуть мышью на одном из символов любого математического фрагмента с тем, чтобы уголковый курсор охватил этот символ;
- выбрать тип и размер шрифта на панели форматирования.

Выбор типа и размера шрифта цифр в математических выражениях производится аналогично. Например, можно выбрать для цифр шрифт *Times New Roman*, размер 12 пт, стиль – обычный. Для символов применить шрифт *Arial*, размер 14 пт; стиль – полужирный, курсив, подчеркнутый. Результат такого выбора приведен ниже, после описания особенностей текстовых фрагментов.

Технология работы с текстовыми фрагментами в системах *Mathcad* практически не отличается от работы с текстовыми редакторами *MS Windows*. Текст вводится в месте фиксации визира после активизации позиции **Вставка/Регион текста** главного меню систем *Mathcad* или нажатием клавиши «"». При этом появляется черная рамка с расположенным внутри нее **текстовым** курсором, который имеет вид вертикальной полосы красного цвета. Выбирая тип и размер шрифта, английскую или русскую литерацию, можно приступать к набору текста. Наличие черного обрамления активного текстового фрагмента свидетельствует о возможности его перемещения, копирования, удаления, а также форматирования внутри рамки. Особого внимания заслуживает последняя операция, т.к. остальные, рассмотренные ранее, производятся аналогично.

Текстовый курсор может быть установлен щелчком мыши в любой точке текста для удаления прилегающих элементов клавишами «**Backspace**» или «**Del**», а также других действий. Далее приведена копия одного из текстовых фрагментов, на которой виден текстовый курсор, расположенный в первом слове.


Математические модели и методы

Для удаления слов и более длинных фрагментов текста используется протяжка мышью, после которой требуемая область текста получает черную заливку. Это означает, что выделенная таким образом область текста может быть удалена, скопирована в буфер или для нее может быть выбран свой вариант шрифта. Таким образом, в отличие от математических выражений, где выделение преимущественно осуществляется уголковым курсором (возможна и протяжка), в текстовых фрагментах для выделения с целью последующих операций используется исключительно протяжка мышью, которая сопровождается заливкой выделяемой зоны. Другим отличием текстовых фрагментов от математических служит возможность выбора различных шрифтов для каждого из текстовых фрагментов и, более того, выбора шрифтов для отдельных слов и элементов внутри фрагмента. Пример такого раздельного выбора приведен ниже:

(2 <u>**a**</u>+ 3) <u>**c**</u>^{3}	Пример раздельного выбора шрифтов для символов и цифр математического выражения, а также *для отдельных* слов <u>**текстового фрагмента**</u>

В документе *Mathcad* текстовые и математические фрагменты имеют равноценное значение; при активизации получают черную рамку, могут быть выделены в составе группы (пунктирные рамки), могут одинаково перемещаться пользователем в целях достижения рационального использования поля документа. Небольшие отличия касаются лишь отмеченных выше особенностей форматирования математических и текстовых фрагментов.

Более тонкие детали пользовательского интерфейса рассматриваются в процессе описания техники практического применения систем *Mathcad* при решении задач различных классов в [1–4].

1.2.4. Особенности входного языка системы

Входной язык *Mathcad* относится к интерпретирующим языкам, которые, при опознании какого-либо объекта системы, операции или функции, тут же исполняют предписанные инструкции. Реализация вычислений в системах осуществляется с использованием языка *С*++, однако его знание не является необходимым условием практического применения *Mathcad*. Пользователь должен знать лишь основные элементы входного языка систем, с помощью которых можно создать формульные, текстовые, графические и другие фрагменты документа. К таким элементам относятся идентификаторы, константы, переменные, массивы и другие типы данных, операторы, функции, элементы программных модулей и др.

При решении какой-либо задачи входной язык используется системами *Mathcad* для составления программы, которая размещается в ОЗУ. Программа хранится там до момента обращения к команде «сохранить» или «сохранить как», при выполнении которых текущий файл с расширением «*mcd»* сохраняется на жестком или гибком диске. Для пользователя входной язык выступает в качестве своеобразного макроязыка. Он позволяет решать возникающие задачи с помощью привычных математических символов и операторов (большая часть которых может быть введена мышью с использованием математической палитры и палитры символов), осуществлять как бы блочное программирование своей задачи. При этом от пользователя, естественно, требуется предварительно осуществить формализацию своей задачи в полном соответствии с канонами математики и разработать алгоритм получения решения.

Просмотр отдельных фрагментов текущего документа производится системами построчно слева направо и сверху вниз. При опознании очередного фрагмента, системы запускают внутренние подпрограммы для выполнения требуемых инструкций при условии, что к этому моменту определены все элементы используемых выражений, т.е. эти элементы и выражения введены надлежащим образом. Это определяет необходимость логической увязки фрагментов подготавливаемого документа «сверху вниз» и делает невозможным выполнение блоков до введения (описания, задания значений) участвующих в них переменных.

При нарушении логической согласованности фрагментов система выделяет красным цветом элементы первого из логически связанных фрагментов, из-за которого вычисления не могут быть продолжены (некорректно введенные элементы или невыполнимые фрагменты); подобной закраской выделяются и элементы последующих фрагментов. При щелчке мышки на таких фрагментах появляется выпадающая подсказка с краткой информацией о причинах нарушения вычислительного процесса. Ниже приведена горизонтальная цепочка элементарных вычислений, элементы которой введены правильно и которая завершается получением численного результата (он выделен фоновой заливкой).

$\boxed{x := 0.138} \quad y := x^3 \quad z := 3 - 4y \quad z = 2.989$

Если в том же алгоритме ввод значения исходного параметра передвинуть чуть ниже уровня последующего его использования, вычислительный процесс становится невозможным. При этом система выделяет красным цветом переменные *x*, *y*, *z* во втором, третьем и заключительном фрагментах соответственно, что служит сигналом о нарушении логической согласованности фрагментов документа. При щелчке мышки, например, на третьем фрагменте, он получает черную рамку и выпадающая подсказка говорит о том, что данная (выделенная красным в этом фрагменте) переменная не определена выше. Результат таких действий приведен ниже.

$$y := x^3 \qquad z := 3 - 4y \qquad z = \blacksquare$$

$$\boxed{x := 0.138}$$

Исключение из приведенного правила согласования фрагментов составляют лишь процедуры выполнения символьных вычислений, которые могут выполняться, несмотря на указанный «протест системы». На приведенном ниже простом примере система выполняет разложение разности кубов, несмотря на выделение красным цветом переменных *x* и *y* в первом и втором фрагментах соответственно.

$$y := x^3 \qquad z := y - 1 \qquad z \text{ factor}, 0 \ \rightarrow (x - 1) \cdot \left(x^2 + x + 1\right)$$

В двух последних примерах, представлен весь цикл выполнения вычислений от ввода до получения результата в числовой или в символьной формах. В них используются два оператора, которые не комментировались ранее – оператор присваивания и операторы вывода в числовом (знак равенства) и символьном (правая стрелка) виде. Кроме того, следует обратить внимание на формат чисел в системах *Mathcad*: числа имеют десятичную точку.

Оператор присваивания хорошо известен в программировании, имеет вид двоеточия с последующим знаком равенства. В системах *Mathcad* этот оператор вызывается нажатием на клавишу «**:**» (двоеточие; верхний регистр клавиатуры, т.е. предварительно нажать «**Shift**»**).** При вызове оператора на место визира получаем так называемый шаблон ввода

$$\boxed{\blacksquare := \blacksquare}$$

в состав которого входят два маркера ввода, разделенные оператором присваивания. Левый маркер предназначен для ввода имени выражения (переменной, функции, вектора, матрицы и проч.), правый — для ввода выражения. Состояние уголкового курсора показывает, что первым может вводиться выражение. Если предварительно набирается имя математического выражения, а затем вызывается оператор присваивания с последующим заполнением правого маркера ввода, то оператор присваивания будет иметь лишь правый маркер ввода. Например, для присваивания переменной *x* числового значения 0,138, можно выполнить один из следующих вариантов ввода, представленных в таблице 1.4.

Таблица 1.4 – Последовательность действий

№ операции	Вариант 1	Вариант 2	Вариант 3
1	■ := ■	0.138	x
2	■ := 0.138	0.138	x := ■
3	■ := 0.138	■ := 0.138	-
4	x := 0.138	x := 0.138	x := 0.138

Аналогично могут быть введены любые выражения, предназначенные для получения численного или символьного результата.

Необходимо дать краткие пояснения к таблице 1.4. Приведенные варианты отличаются стартовыми операциями. В первом сразу вводится шаблон оператора присваивания; во втором и третьем вариантах ввод начинается с набора числового значения и символа переменной соответственно. Вторая операция в зависимости от варианта состоит в вводе числа (вариант 1), изменении положения уголкового курсора на обратное (вариант 2) в обеспечение развития выражения влево, вводе оператора присваивания (вариант 3). На третьем этапе для варианта 1 производится щелчок мыши на левом маркере ввода для обеспечения возможности последующего ввода символа; для варианта 2 этот этап состоит в вызове оператора присваивания. Последняя строка таблицы 2.4 соответствует завершению ввода. В третьем варианте ввод выполняется за три этапа.

При щелчках мыши по маркерам ввода может появиться выпадающая подсказка с информацией о незавершенности выражения и необходимости ввода данных в этот маркер ввода. Подобная подсказка (*This expression is incomplete. You must fill in placeholders*) сопровождает, например, позицию 3 варианта 1 (см. таблицу 1.4).

<u>Для вывода результата в числовой форме</u> необходимо предварительно охватить математическое выражение или его имя уголком курсора и нажать на клавишу = (равно).

<u>Вывод результата в символьном виде</u> может быть осуществлен с помощью оператора символьного вывода, который вызывается сочетанием клавиш «**Ctrl** + **.**» и имеет вид правой стрелки.

Завершая описание начальных сведений о системах *Mathcad*, отметим, что в *Mathcad* предусмотрена избыточность средств достижения результатов, которая обеспечивает дополнительные удобства пользователям при решении конкретных задач. Так, вызов рассмотренных операторов и процедуры ввода могут быть реализованы с помощью соответствующих кнопок математической палитры, которая вызывается активизацией позиции **Вид/Панели инструментов (**главного меню системы *Mathcad).*

1.3. Программирование в Mathcad

Одну из наиболее значимых частей системы *Mathcad* составляют так называемые программные модули. С их помощью можно реализовать практически любую процедуру вычислений, дополняя тем самым обширные вычислительные возможности *Mathcad*.

Распространено мнение, что при обучении, например, численным методам с помощью пакетов прикладных программ страдает методическая сторона, поскольку ни один пакет не содержит неустойчивый метод, не слишком удачную аппроксимацию и т. п. Данная точка зрения, не совсем корректна. Во-первых, возможности любого пакета ограниченны. Это связано с тем, что решение задачи может быть не точным или вообще не найдено, хотя формально соответствующие опции в системе могут существовать. Однако при их применении наблюдаются типичные признаки неустойчивости применяемого в этой процедуре алгоритма. Во-вторых, практически все пакеты дают возможность написания программ для своих алгоритмов, причем компактность программ значительно облегчает их анализ.

1.3.1. Общая характеристика программных модулей

Программные модули в *Mathcad* предназначены для реализации различного рода программ, т.е. последовательностей операций над функциями и выражениями, позволяющих получить требуемый результат. Программные модули *Mathcad* представляют собой замкнутые вычислительные структуры, которые, при задании исходных данных, дают локальный результат. Замкнутость программного модуля означает ограничение реализованного в нем вычислительного процесса рамками модуля. При этом внутри программного модуля могут быть использованы обозначения, принятые и в последующих фрагментах документа *Mathcad*. Это совпадение обозначений не влияет на такие последующие фрагменты ввиду замкнутости программного модуля. Все внутренние операции присваивания имеют значение лишь для самого модуля, также не влияя на другие фрагменты документа. В то же время внешние обозначения и присваивания, расположенные в документе выше программного модуля, воспринимаются модулем как глобальные и влияют на процессы вычисления внутри него.

Основным требованием при формировании программных модулей является то, что операция присваивания внутри модуля должна осуществляться только с помощью символа локального присваивания ←. Введенные с помощью этого символа исходные данные внутри модуля имеют, как отмечалось, локальный статус, действующий только в пределах программного модуля. Использование обычных операторов присваивания (двоеточие со знаком равенства) внутри модулей недопустимо. Вместо него применяется оператор «эквивалентно», который вызывается клавишами «**Ctrl** + =».

Исходные данные для реализации вычислений можно вводить непосредственно внутри программного модуля. Эти данные могут быть введены или получены реализацией процесса вычислений и перед программным модулем, вне его рамок. Такие внешние исходные данные имеют для программного модуля статус глобальных, действующих в любой части модуля.

Характерным признаком программного модуля служит жирная вертикальная черта, которая вводится оператором **Add Line** (см. ниже). Программные модули могут быть простыми (с одной вертикальной чертой) и сложными (с несколькими вертикальными чертами), образованными по принципу вложения. Таким путем могут формироваться иерархические структуры программных модулей с соподчиненными модулями нижнего уровня и последовательностью модулей более высокого уровня. Именно такие структуры программных модулей применяются при реализации итеративных (циклических) процедур.

При формировании программных модулей и для реализации вычислительного процесса внутри модулей используется ряд специальных операторов, набор которых вызывается нажатием кнопки математической палитры с изображением блок-схемы программы (рисунок 1.14).

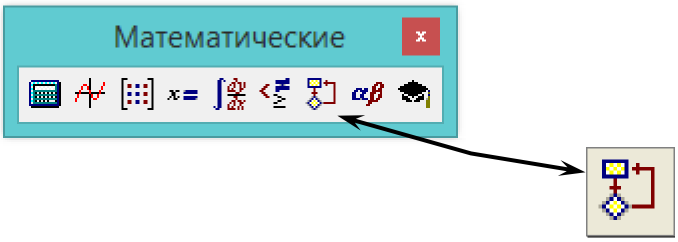


Рисунок 1.14 – Математическая палитра

Кроме специальных операторов, внутри программных модулей могут быть использованы любые функции, переменные и операторы, доступные другим приложениям систем *Mathcad*. В их число входят наиболее часто применяемые арифметические и булевы операторы, ранжированные переменные, векторные, матричные и тригонометрические встроенные функции и многие другие. Запрет при использовании внутри программных модулей, как отмечено выше, распространяется на обычные знаки присваивания.

Отмеченные особенности программных модулей позволяют рассматривать их в качестве своеобразных макрофункций, реализующих определенные процедуры и описанных с помощью специфических элементов. Такие функции-модули обладают всеми характерными признаками обычных функций. Им можно присваивать имя с перечнем аргументов или без них; они являются исполнимыми, т.е. позволяют получить результат с помощью обычных средств вывода. Этим функциям, как отмечалось, может быть присвоено имя, а, следовательно, после программного модуля можно поставить знак равенства (или стрелку символьного вывода) и получить результат.

Отдельные операторы программных модулей рассматриваются ниже. Следует отметить, что при сравнительно небольшом числе этих операторов возможностей каждого из них и их сочетаний вполне достаточно для программирования широкого класса прикладных задач в разных областях науки и техники.

1.3.2. Операторы программных модулей

Палитра операторов программных модулей, как было отмечено, вызываются нажатием кнопки математической палитры (см. рисунок 1.14) с изображением блок-схемы программы. Палитры операторов программных модулей для *Mathcad* показаны на рисунке 1.15.

Шаблоны двух структур программных модулей, приведенных на рисунке 1.15 справа, получены однократным и повторным нажатием клавиши **Add Line** соответственно. Во втором случае, после первого нажатия клавиши **Add Line**, уголок курсора должен находиться на одном из маркеров ввода.

Структура программного модуля имеет существенное значение при полнении циклических, итеративных и рекуррентных вычислений. Общим свойством любой структуры модуля служит то, что конечный результат ты модуля формируется его последней (завершающей) строкой. При формировании структуры пользователь может опираться на предварительный из алгоритма вычислений и заранее определить число и взаимное положение маркеров ввода в структуре программного модуля. Обычно, при решении достаточно сложных задач определить элементы структуры модуля точно не удается. В этих случаях, при формировании структуры возникает необходимость ее дополнения или изменения вида (добавление или изъятие вложенных циклов и проч.). Операции изменения некоторого варианта уже данной структуры любого вида осуществляются вводом необходимых операторов в тело программного модуля. При этом место ввода оператора и вариант получаемой структуры зависят от места размещения и положения (левостороннего, правостороннего) уголкового курсора. Последующие примеры иллюстрируют отмеченные возможности форматирования структур программных модулей.

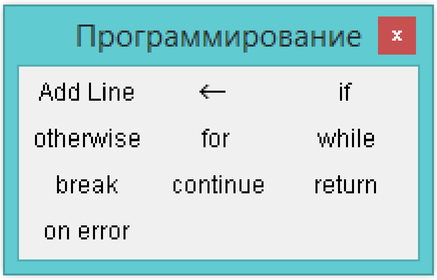


Рисунок 1.15 – Задание структур простых модулей

Палитра операторов программных модулей содержит 10 кнопок, нажатие которых вызывает шаблоны соответствующих операторов. В таблице 1.5 приведены шаблоны операторов, вызванные однократным нажатием кнопок палитры (рисунок 1.15). Шаблоны раскрывают формат соответствующих операторов.

Символы операторов в шаблонах (таблица 1.5) нельзя набирать с клавиатуры; при формировании программных модулей следует пользоваться только кнопками палитры программных операторов.

Таблица 1.5 - Символы операторов и кнопки палитры программных операторов

| ▪ | ▪	▪ ← ▪
▪ if ▪	▪ otherwise
for ▪ ∈ ▪ ▪	while ▪ ▪
break	continue
return ▪	▪ on error ▪

Перечислим операторы модулей и прокомментируем их назначение:

Add Line – используется для создания отличительного признака программного модуля — жирной вертикальной черты, которая, при однократном нажатии кнопки **Add Line**, имеет два маркера ввода справа от нее. При необходимости создания более сложной структуры программного модуля повторным нажатием кнопки **Add Line** вертикальная линия либо растягивается (при этом добавляются и маркеры ввода), либо, при нажатии; кнопки других операторов, справа от основной вертикальной линии появляются вертикальные линии вложенных программных модулей (более низкого уровня);

← – оператор локального присваивания внутри модуля. В теле программных модулей не допускается использование обычных знаков присваивания. Их заменяет оператор локального присваивания. В левый его маркер вводится имя переменной, которой присваивается значение, расположенное в правом маркере;

if – оператор условного перехода. Этот оператор имеет формат

выражение if условие

и обычно используется совместно с другими операторами, например, с оператором **otherwise**, в структурах типа

$$\left|\begin{array}{ll} 2 & \text{if } \ x > 1 \\ (-1) & \text{otherwise} \end{array}\right.$$

В условиях, как правило, используются булевы операторы, в том числе «больше», «меньше», «больше или равно», «меньше или равно», «эквивалентно» (жирный знак равенства, вызываемый клавишами «**Ctrl** + **=**»).

otherwise – оператор, используемый в качестве альтернативного оператора совместно с оператором **if**;

for – оператор организации циклических и итеративных вычислений с числом повторений в диапазоне от *nl* до *n2*. При нажатии кнопки этого оператора на программной палитре появляется шаблон, показанный на рисунке 1.16, *а*. В первый маркер после **for** необходимо ввести ранжированную переменную (например, *у*), а в правый маркер следует ввести (после нажатия клавиши **;**) диапазон ее изменения – значения целых чисел *n*1 и *n*2. В нижний маркер левой структуры вводится исполняемое выражение. Так поступают с каждым оператором **for.** Совмещая уголок курсора с нижним маркером (см. рисунок 1.16, *а*) и, нажимая кнопку **for** на программной панели, получаем структуру модуля, изображенную на рисунке 1.16, *б*.

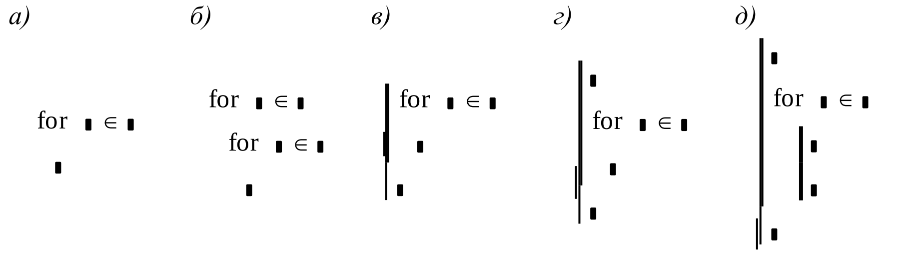


Рисунок 1.16. Структуры программных модулей с оператором **for**

Структуры программных модулей (см. рисунок 1.16, *в*, *г*) получены введением оператора **for** в левый и правый шаблоны модулей (см. рисунок 1.15). Структура г показывает возможности формирования более сложных программных блоков с вложениями. Структура *д* получена из *г* введением оператора **Add Line** для маркера, расположенного непосредственно под оператором **for**. Упрощение структур программных блоков производится в обратном порядке, путем исключения маркеров **(Del),** начиная с нижнего;

while – оператор для организации циклов и условных соотношений, действующих до тех пор, пока выполняется условие. Формат ввода этого оператора раскрыт в шаблоне (см. таблицу 1.5). В правый маркер вводится условие, а в нижний маркер – исполняемое выражение;

break – оператор, прерывающий программу в месте, где он встречается. Обычно используется совместно с операторами **if, while, for**;

continue – оператор, используемый для продолжения прерванной программы. Обычно используется совместно с операторами **while, for**;

return – оператор, возвращающий переменную (символ, число), стоящую непосредственно за ним, в качестве результата программы. Этот оператор, не прерывая цикла, останавливает процесс вычислений в точке своего расположения. Если оператор **return** используется условно, совместно с оператором **if**, то остановка вычислений осуществляется в момент выполнения условия. Шаблон оператора приведен в таблице 1.5;

on error – оператор, предназначенный для реагирования программы на возникающие ошибки; осуществляет переход вычислительного процесса на другой алгоритм вычислений (к другому выражению) в случае возникновения ошибок в выражении, стоящем непосредственно после оператора (справа). Форма оператора задается в виде **A2 *on error* A1.**

В случае возникновения ошибки в выражении A1 оператор возвращает (для выполнения) выражение A2. При отсутствии ошибок в выражении A1 оно исполняется по умолчанию. В качестве простых примеров работы этого оператора приведем результаты его использования:

$$3 \ \text{on error} \ \frac{1}{0} = 3 \qquad\qquad 5 \ \text{on error} \ (1+1) = 2$$

В первом примере деление на ноль воспринимается системой как ошибка, поэтому результата формируется по выражению слева. Во втором случае, в выражении справа ошибки нет, и результат формируется по этому выражению.

Программные модули широко используются при реализации циклических и рекуррентных процедур, применении разностных уравнений состояний систем.

Глава 2. Практический курс

2.1. Управление очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости

2.1.1. Теоретические основы

Необходимо определить оптимальную очередность реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости, используя критерий «упущенная выгода» [5,6].

Каждый проект (реконструируемый участок железной дороги) после его завершения дает строительной (подрядной) организации определенный доход. Задержка в сроках реализации проектов ведет к уменьшению дохода, то есть к упущенной выгоде.

Пусть *i*-й проект дает после завершения доход C_i в единицу времени. Тогда упущенная выгода при завершении *i*-го проекта в момент t_i составит $C_i t_i$,а суммарная упущенная выгода равна:

$$S = \sum_{i=1}^{n} C_i t_i \qquad (2.1.1)$$

Пусть мультипроект состоит из двух проектов (двух реконструируемых участков железной дороги), объемы которых W_1 и W_2, а w_i скорости выполнения проектов, которые линейно зависят от количества ресурсов a_i:

$$w_i = \begin{cases} a_i, a_i < N \\ N, a_i \geq N \end{cases}$$

Примем, что $a_1 + a_2 > N$, $a_1 \leq N$, $a_2 \leq N$, так что одновременно проекты нельзя выполнять с максимальными скоростями. Пусть первым завершается первый проект за минимальное время $\tau_1 = W_1 / a_1$. За время τ_1 будет выполнен объем работ $(N - a_1)\tau_1$ второго проекта. Оставшийся объем работ $W_2 - (N - a_1)\tau_1$ будет выполнен за время

$$\frac{W_2 - (N - a_1)\tau_1}{a_2}$$

Упущенная выгода составит:

$$S = C_1\tau_1 + C_2\left(\tau_1 + \frac{W_2 - (N - a_1)\tau_1}{a_2}\right) = C_2\left(\beta\tau_1 + \tau_2 + \frac{\delta\tau_1}{a_2}\right), \qquad (2.1.2)$$

где

$$\tau_1 = \frac{W_1}{a_1},\ \tau_2 = \frac{W_2}{a_2};\ \delta = a_1 + a_2 - N;\ \beta = \frac{C_1}{C_2}$$

Если первым завершается второй проект, то упущенная выгода составит:

$$S' = C_2\left(\beta\tau_1 + \tau_2 + \beta\frac{\delta\tau_2}{a_1}\right), \qquad (2.1.3)$$

Сравнивая (1.2) и (1.3), получим следующее решающее правило.

Если

$$S < S', \tag{2.1.4}$$

то первым завершается первый проект, в противном случае - второй. Следует учесть, что если $a_1 = a_2 = N$, то проекты выполняются последовательно за минимальные времена

$$\tau_i = \frac{W_i}{N}.$$

В этом случае получаем известную в теории расписаний задачу определения оптимальной очередности выполнения операций на одном рабочем месте. Решающее правило в этом случае совпадает с известным решающим правилом - упорядочение по убыванию отношения C_i / τ_i.

Алгоритм определения оптимальной очередности реализации проектов:

1. Рассматриваются 1 и 2 проекты. Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_2 > N$. Возникает конфликтная ситуация.
2. Пусть проект 1 выполняется на максимальном уровне. Определяется упущенная выгода по формуле (2.1.2).
3. Пусть проект 2 выполняется на максимальном уровне. Определяется упущенная выгода по формуле (2.1.3).
4. Аналогично рассматриваются все варианты реализации проектов.
5. Результаты расчетов сводятся в таблицу 2.1.

Таблица 2.1 – Форма для результатов расчета

Проект	τ_i мес.	δ	β	S_i, млн. р.	Очередность реализации
1					
2					
3					
4					
и т.д.					

6. Сравнивая критерии «упущенная выгода», определяется оптимальная очередность реализации проектов.

2.1.2. Пример решения

Исходные данные

Исходные данные помещены в таблице 2.2, необходимо определить оптимальную очередность реализации проектов.

Таблица 2.2 – Исходные данные

№ про-екта	Проект	W_i, млн. р.	a_i, млн. р./мес.	τ_i мес.	C_i, млн. р-
1	Участок 15 км	3500	310	11,3	280
2	Участок 22 км	6200	440	14,1	500
3	Участок 17 км	3900	480	8,1	320
4	Участок 48 км	12500	600	20,8	1000

Уровень финансирования мультипроекта N=500 млн.р./мес

Процедура вычислений

1. Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_2 = 310 + 440 = 750 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{280}{500} = 0{,}56\,;$$

$$\delta = 310 + 440 - 500 = 250 \text{ млн. р./мес.}$$

1.1. Проект 1 финансируется на максимальном уровне.

$$S(1;2) = 500\left(0{,}56 \cdot 11{,}3 + 14{,}1 + \frac{250 \cdot 11{,}3}{440}\right) = 13424{,}2 \text{ млн. р.;}$$

1.2. Проект 2 финансируется на максимальном уровне.

$$S(2;1) = 500\left(0{,}56 \cdot 11{,}3 + 14{,}1 + 0{,}56 \cdot \frac{250 \cdot 14{,}1}{310}\right) = 13397{,}9 \text{ млн. р.;}$$

2. Проекты 1 и 3 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_3 = 310 + 480 = 790 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{280}{320} = 0{,}875\,;$$

$$\delta = 310 + 480 - 500 = 290 \text{ млн. р./мес.}$$

2.1. Проект 1 финансируется на максимальном уровне.

$$S(1;3) = 320\left(0{,}875 \cdot 11{,}3 + 8{,}1 + \frac{290 \cdot 11{,}3}{480}\right) = 7940{,}7 \text{ млн. р.;}$$

2.2. Проект 3 финансируется на максимальном уровне.

$$S(3;1) = 320\left(0{,}875 \cdot 11{,}3 + 8{,}1 + 0{,}875 \cdot \frac{290 \cdot 8{,}1}{310}\right) = 7877{,}7 \text{ млн. р.;}$$

3. Проекты 1 и 4 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_4 = 310 + 600 = 910 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{280}{1000} = 0{,}28\,;$$

$$\delta = 310 + 600 - 500 = 410 \text{ млн. р./мес.}$$

3.1. Проект 1 финансируется на максимальном уровне.

$$S(1;4) = 1000\left(0{,}28 \cdot 11{,}3 + 20{,}8 + \frac{410 \cdot 11{,}3}{600}\right) = 31685{,}7 \text{ млн. р.;}$$

3.2. Проект 4 финансируется на максимальном уровне.

$$S(4;1) = 1000\left(0{,}28 \cdot 11{,}3 + 20{,}8 + 0{,}28 \cdot \frac{410 \cdot 20{,}8}{310}\right) = 31666{,}7 \text{ млн. р.;}$$

4. Проекты 2 и 3 не могут финансироваться на максимальном уровне, т.к. $a_2 + a_3 = 440 + 480 = 920 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{500}{320} = 1{,}563\,;$$

$$\delta = 440 + 480 - 500 = 420 \text{ млн. р./мес.}$$

4.1. Проект 2 финансируется на максимальном уровне.

$$S(2;3) = 320\left(1{,}56 \cdot 14{,}1 + 8{,}1 + \frac{420 \cdot 14{,}1}{480}\right) = 13590{,}0 \text{ млн. р.;}$$

4.2. Проект 3 финансируется на максимальном уровне.

$$S(3;2) = 320\left(1{,}56 \cdot 14{,}1 + 8{,}1 + 1{,}56 \cdot \frac{420 \cdot 8{,}1}{440}\right) = 13507{,}9 \text{ млн. р.;}$$

5. Проекты 2 и 4 не могут финансироваться на максимальном уровне, т.к. $a_2 + a_4 = 440 + 600 = 1040 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{500}{1000} = 0{,}5;$$

$$\delta = 440 + 600 - 500 = 540 \text{ млн. р./мес.}$$

5.1. Проект 2 финансируется на максимальном уровне.

$$S(2;4) = 1000\left(0{,}5 \cdot 14{,}1 + 20{,}8 + \frac{540 \cdot 14{,}1}{600}\right) = 40540 \text{ млн. р.;}$$

5.2. Проект 4 финансируется на максимальном уровне.

$$S(4;2) = 1000\left(0{,}5 \cdot 14{,}1 + 20{,}8 + 0{,}5 \cdot \frac{540 \cdot 20{,}8}{440}\right) = 40613{,}6 \text{ млн. р.;}$$

6. Проекты 3 и 4 не могут финансироваться на максимальном уровне, т.к. $a_3 + a_4 = 480 + 600 = 1080 > 500$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{320}{1000} = 0{,}32;$$

$$\delta = 480 + 600 - 500 = 580 \text{ млн. р./мес.}$$

6.1. Проект 3 финансируется на максимальном уровне.

$$S(3;4) = 1000\left(0{,}32 \cdot 8{,}1 + 20{,}8 + \frac{580 \cdot 8{,}1}{600}\right) = 31222 \text{ млн. р.;}$$

6.2. Проект 4 финансируется на максимальном уровне

$$S(4;3) = 1000\left(0{,}32 \cdot 8{,}1 + 20{,}8 + 0{,}32 \cdot \frac{580 \cdot 20{,}8}{480}\right) = 31434{,}6 \text{ млн. р.;}$$

7. Результаты расчетов сведены в таблице 2.3.

Таблица 2.3 – Результаты расчетов

Проект	τ_i мес.	δ	β	S_i, млн. р.	Очередность реализации
1	11,3	250	0,56	13424,2	2→1
2	14,1			13397,9	
1	11,3	290	0,875	7940,7	3→1
3	8,1			7877,7	
1	11,3	410	0,28	31685,7	4→1
4	20,8			31666,7	
2	14,1	420	1,563	13590,0	3→2
3	8,1			13507,9	
2	14,1	540	0,5	40540,0	2→4
4	20,8			40613,6	
3	8,1	580	0,32	31222,0	3→4
4	20,8			31434,6	

8. Оптимальная очередность реализации проектов 3 → 2 → 4 → 1.

На рисунке 2.1 представлен документ *Mathcad*, содержащий первый вариант решения задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости.

Решение задачи выполнено с применением средств программирования и встроенных функций *if* и *otherwise*, позволяющих реализовать режим проверки «если-то-иначе».

Результаты решения представляют собой значения «упущенной выгоды» для рассматриваемых проектов, а также значения промежуточных вычислений.

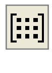
Алгоритм решения задачи с помощью *Mathcad* следующий:

1) Сначала следует заполнить значениями векторы данных (отмечены желтым фоном). Используя мышь, навести курсор на вектор *W* таким образом, чтобы он оказался справа от знака присваивания, после чего нажать на левую кнопку. Уголковый курсор должен принять правое положение ⌊ (если этого не произошло, то необходимо скорректировать его положение с помощью стрелки влево «**←**» на клавиатуре).

2) После появления уголкового курсора в правом положении необходимо нажать клавишу «**Пробел**» на клавиатуре. Уголковый курсор полностью охватит матрицу.

3) На клавиатуре следует дважды нажать на клавишу «***Del***». Матрица, существующая изначально в шаблоне *Mathcad* исчезнет.

4) Для ввода вектора *W* с данными, указанными в задании, необходимо предварительно вызвать панель **Матрица**.

5) На панели **Матрица** с помощью мыши следует нажать на значок матицы . В появившемся окне следует ввести количество строк и столбцов, после нажать клавишу «***Enter***» на клавиатуре. В данном случае строк – 4 и столбцов – 1.

6) В появившиеся ячейки вектора необходимо последовательно ввести исходные данные в соответствии с заданием. После ввода последнего значения нажать клавишу «***Enter***» на клавиатуре.

7) В отношении остальных векторов необходимо проделать операции аналогичные пунктам 2–7.

8) Также необходимо изменить значение для параметра *N*.

В результате определения оптимальной очередности реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости с использованием критерия «упущенная выгода» была установлена очередность реализации проектов: Участок 17 км → Участок 22 км → Участок 48 км → Участок 15 км.

Управление очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости

1) Исходные данные:

$$W := \begin{pmatrix} 3500 \\ 6200 \\ 3900 \\ 12500 \end{pmatrix} \text{ млн. р.} \qquad a := \begin{pmatrix} 310 \\ 440 \\ 480 \\ 600 \end{pmatrix} \text{ млн. р./мес.} \qquad \tau := \begin{pmatrix} 11.3 \\ 14.1 \\ 8.1 \\ 20.8 \end{pmatrix} \text{ мес.} \qquad C := \begin{pmatrix} 280 \\ 500 \\ 320 \\ 1000 \end{pmatrix} \text{ млн. р.}$$

$$N := 500 \text{ млн. р./мес.}$$

ORIGIN := 1

2) Блок решения

1) Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{vmatrix} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_2 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{vmatrix} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_1}{C_2} = 0.56 \qquad \delta := a_1 + a_2 - N = 250 \text{ млн. р./мес.}$$

1.1. Проект 1 финансируется на максимальном уровне.

$$S_{1\_2} := C_2 \cdot \left(\beta \cdot \tau_1 + \tau_2 + \frac{\delta \cdot \tau_1}{a_2} \right) = 13424.2 \text{ млн. р.}$$

1.2. Проект 2 финансируется на максимальном уровне.

$$S_{2\_1} := C_2 \cdot \left(\beta \cdot \tau_1 + \tau_2 + \beta \frac{\delta \cdot \tau_2}{a_1} \right) = 13397.9 \text{ млн. р.}$$

Рисунок 2.1 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* (*начало*)

2) Проекты 1 и 3 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_3 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_1}{C_3} = 0.875 \qquad \delta := a_1 + a_3 - N = 290 \text{ млн. р./мес.}$$

2.1. Проект 1 финансируется на максимальном уровне.

$$S_{1\_3} := C_3 \cdot \left(\beta \cdot \tau_1 + \tau_3 + \frac{\delta \cdot \tau_1}{a_3} \right) = 7940.7 \text{ млн. р.}$$

2.2. Проект 3 финансируется на максимальном уровне.

$$S_{3\_1} := C_3 \cdot \left(\beta \cdot \tau_1 + \tau_3 + \beta \frac{\delta \cdot \tau_3}{a_1} \right) = 7877.7 \text{ млн. р.}$$

3) Проекты 1 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_1}{C_4} = 0.28 \qquad \delta := a_1 + a_4 - N = 410 \text{ млн. р./мес.}$$

3.1. Проект 1 финансируется на максимальном уровне.

$$S_{1\_4} := C_4 \cdot \left(\beta \cdot \tau_1 + \tau_4 + \frac{\delta \cdot \tau_1}{a_4} \right) = 31685.7 \text{ млн. р.}$$

3.2. Проект 4 финансируется на максимальном уровне.

$$S_{4\_1} := C_4 \cdot \left(\beta \cdot \tau_1 + \tau_4 + \beta \frac{\delta \cdot \tau_4}{a_1} \right) = 31666.7 \text{ млн. р.}$$

Рисунок 2.1 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* (*продолжение*)

4) Проекты 2 и 3 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_2 + a_3 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_2}{C_3} = 1.563 \qquad \delta := a_2 + a_3 - N = 420 \text{ млн. р./мес.}$$

4.1. Проект 2 финансируется на максимальном уровне.

$$S_{2\_3} := C_3 \cdot \left(\beta \cdot \tau_2 + \tau_3 + \frac{\delta \cdot \tau_2}{a_3} \right) = 13590 \text{ млн. р.}$$

4.2. Проект 3 финансируется на максимальном уровне.

$$S_{3\_2} := C_3 \cdot \left(\beta \cdot \tau_2 + \tau_3 + \beta \frac{\delta \cdot \tau_3}{a_2} \right) = 13507.9 \text{ млн. р.}$$

5) Проекты 2 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_2 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_2}{C_4} = 0.5 \qquad \delta := a_2 + a_4 - N = 540 \text{ млн. р./мес.}$$

5.1. Проект 2 финансируется на максимальном уровне.

$$S_{2\_4} := C_4 \cdot \left(\beta \cdot \tau_2 + \tau_4 + \frac{\delta \cdot \tau_2}{a_4} \right) = 40540 \text{ млн. р.}$$

5.2. Проект 3 финансируется на максимальном уровне.

$$S_{4\_2} := C_4 \cdot \left(\beta \cdot \tau_2 + \tau_4 + \beta \frac{\delta \cdot \tau_4}{a_2} \right) = 40613.6 \text{ млн. р.}$$

Рисунок 2.1 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* (*продолжение*)

6) Проекты 3 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_3 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases}$$

$R =$ "Необходимо выполнить расчет"

На основе исходных данных необходимо определить:

$$\beta := \frac{C_3}{C_4} = 0.32 \qquad \delta := a_3 + a_4 - N = 580 \text{ млн. р./мес.}$$

6.1. Проект 2 финансируется на максимальном уровне.

$$S_{3\_4} := C_4 \cdot \left(\beta \cdot \tau_3 + \tau_4 + \frac{\delta \cdot \tau_3}{a_4} \right) = 31222 \text{ млн. р.}$$

6.2. Проект 3 финансируется на максимальном уровне.

$$S_{4\_3} := C_4 \cdot \left(\beta \cdot \tau_3 + \tau_4 + \beta \frac{\delta \cdot \tau_4}{a_3} \right) = 31434.7 \text{ млн. р.}$$

Рисунок 2.1 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* (*окончание*)

На рисунке 2.2 представлен документ *Mathcad*, содержащий второй вариант решения задачи.

Решение выполнено с применением средств программирования. Разработано три программных блока: блок расчета значений «упущенной выгоды» для рассматриваемых проектов (*R*), блок определения вариантов очередности реализации проектов (*f*), блок определения оптимальной очередности реализации проектов (*pos*).

Численные результаты, полученные с помощью программных блоков, представлены в матрицах *RR, H* и *KK*.

Управление очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости

1) Исходные данные:

$$Nm := \begin{pmatrix} \text{"Уч-к 15 км"} \\ \text{"Уч-к 22 км"} \\ \text{"Уч-к 17 км"} \\ \text{"Уч-к 48 км"} \end{pmatrix} \quad W := \begin{pmatrix} 3500 \\ 6200 \\ 3900 \\ 12500 \end{pmatrix} \text{ млн. р.} \quad a := \begin{pmatrix} 310 \\ 440 \\ 480 \\ 600 \end{pmatrix} \text{ млн. р./мес.} \quad \tau := \begin{pmatrix} 11.3 \\ 14.1 \\ 8.1 \\ 20.8 \end{pmatrix} \text{ мес.} \quad C := \begin{pmatrix} 280 \\ 500 \\ 320 \\ 1000 \end{pmatrix} \text{ млн. р.}$$

$N := 500$ млн. р./мес.

Рисунок 2.2 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*начало*)

2) Блок решения

$ORIGIN := 1$

$$
R(W, a, \tau, C, N) := \left|
\begin{array}{l}
q \leftarrow rows(W) \\
m \leftarrow 1 \\
n \leftarrow 1 \\
\text{for } i \in 1..q-1 \\
\quad \left| \begin{array}{l}
\text{for } j \in i+1..q \\
\quad \text{if } a_i + a_j > N \\
\qquad \left| \begin{array}{l}
Nn_m \leftarrow m \\
Ii_m \leftarrow i \\
Jj_m \leftarrow j \\
T1_m \leftarrow \tau_i \\
T2_m \leftarrow \tau_j \\
\beta_m \leftarrow \dfrac{C_i}{C_j} \\
\delta_m \leftarrow a_i + a_j - N \\
Sf_m \leftarrow C_j \cdot \left(\beta_m \cdot \tau_i + \tau_j + \dfrac{\delta_m \cdot \tau_i}{a_j} \right) \\
Ss_m \leftarrow C_j \cdot \left(\beta_m \cdot \tau_i + \tau_j + \beta_m \dfrac{\delta_m \cdot \tau_j}{a_i} \right) \\
\text{if } Sf_m < Ss_m \\
\quad \left| \begin{array}{l} Ik_m \leftarrow i \\ Jk_m \leftarrow j \end{array} \right. \\
\text{if } Sf_m > Ss_m \\
\quad \left| \begin{array}{l} Ik_m \leftarrow j \\ Jk_m \leftarrow i \end{array} \right. \\
m \leftarrow m + 1
\end{array} \right. \\
i \leftarrow i + 1
\end{array} \right. \\
augment(Nn, Ii, Jj, T1, T2, \delta, \beta, Sf, Ss, Ik, Jk)
\end{array}
\right.
$$

Рисунок 2.2 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*продолжение*)

$$RR := R(W, a, \tau, C, N)$$

$$RR = \begin{pmatrix} 1 & 1 & 2 & 11.3 & 14.1 & 250 & 0.56 & 13424.23 & 13397.87 & 2 & 1 \\ 2 & 1 & 3 & 11.3 & 8.1 & 290 & 0.88 & 7940.67 & 7877.68 & 3 & 1 \\ 3 & 1 & 4 & 11.3 & 20.8 & 410 & 0.28 & 31685.67 & 31666.71 & 4 & 1 \\ 4 & 2 & 3 & 14.1 & 8.1 & 420 & 1.56 & 13590 & 13507.91 & 3 & 2 \\ 5 & 2 & 4 & 14.1 & 20.8 & 540 & 0.5 & 40540 & 40613.64 & 2 & 4 \\ 6 & 3 & 4 & 8.1 & 20.8 & 580 & 0.32 & 31222 & 31434.67 & 3 & 4 \end{pmatrix}$$

```
f(RR, Nm) := | x ← rows(RR)
             | y ← rows(Nm)
             | mm ← 1
             | for ii ∈ 1..y
             |   | z ← 0
             |   | b ← 0
             |   | for jj ∈ 1..x
             |   |   | b ← b + 1  if  RR[jj,10] = ii
             |   |   | z ← z + 1  if  RR[jj,11] = ii
             |   | L1[ii] ← b
             |   | L2[ii] ← z
             |   | Pr[ii] ← ii
             |   | Nmm[ii] ← Nm[ii]
             | augment(Pr, Nmm, L1, L2)
```

$$H := f(RR, Nm)$$

H =

	1	2	3	4
1	1	"Уч-к 15 км"	0	3
2	2	"Уч-к 22 км"	2	1
3	3	"Уч-к 17 км"	3	0
4	4	"Уч-к 48 км"	1	2

Рисунок 2.2 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*продолжение*)

$$
\begin{aligned}
pos(H) := \;& H1 \leftarrow H \\
& \text{for } ik \in 1..rows(H1) - 1 \\
& \quad nmax \leftarrow ik \\
& \quad \text{for } jk \in ik + 1..rows(H) \\
& \quad\quad nmax \leftarrow jk \;\; \text{if } \left(H1^{\langle 3 \rangle}\right)_{jk} > \left(H1^{\langle 3 \rangle}\right)_{nmax} \\
& \quad \text{for } im \in 1..cols(H1) \qquad\qquad \text{if } nmax \neq ik \\
& \quad\quad cc_{im} \leftarrow \left(H1^{\langle im \rangle}\right)_{ik} \\
& \quad\quad \left(H1^{\langle im \rangle}\right)_{ik} \leftarrow \left(H1^{\langle im \rangle}\right)_{nmax} \\
& \quad\quad \left(H1^{\langle im \rangle}\right)_{nmax} \leftarrow cc_{im} \\
& augment\left(H1^{\langle 1 \rangle}, H1^{\langle 2 \rangle}\right)
\end{aligned}
$$

$$KK := pos(H)$$

KK =

	1	2
1	3	"Уч-к 17 км"
2	2	"Уч-к 22 км"
3	4	"Уч-к 48 км"
4	1	"Уч-к 15 км"

Рисунок 2.2 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*окончание*)

2.2. Управление очередностью реализации проектов реконструкции железных дорог при нелинейной зависимости продолжительности от стоимости

2.2.1. Теоретические основы

Необходимо определить оптимальную очередность реализации проектов реконструкции железных дорог при нелинейной зависимости продолжительности от стоимости, используя критерий «упущенная выгода» [5,6].

Пусть мультипроект состоит из двух проектов (двух реконструируемых участков железной дороги), объемы которых W_1 и W_2, а скорости - $w_1 = \sqrt{a_1}$ и $w_2 = \sqrt{a_2}$. Если первый проект завершается в момент t_1, $a_1 = (W_1 / t_1)^2$, $a_2 = (W_2 / t_2)^2$. За время t_1 будет выполнен объем работ

$$t_1 \sqrt{a_2} = t_1 \sqrt{N - \left(\frac{W_1}{t_1} \right)^2}$$

второго проекта, а в целом второй проект будет завершен в момент

$$t_2 = t_1 + \frac{W_2 - t_1 \sqrt{N - (W_1 / t_1)^2}}{\sqrt{N}} = t_1 + \tau_2 - \sqrt{t_1^2 - \tau_1^2} \,,$$

где $\tau_1 = \dfrac{W_1}{10\sqrt{N}}$, $\tau_2 = \dfrac{W_2}{10\sqrt{N}}$ - минимальные продолжительности проектов.

Упущенная выгода составит:

$$C_1 t_1 + C_2 t_2 = (C_1 + C_2) t_1 + C_2 \tau_2 - C_2 \sqrt{t_1^2 - \tau_1^2} \,. \tag{2.2.1}$$

Найдем t_1, минимизирующее (2.1):

$$t_1 = \frac{1 + \beta}{\sqrt{\beta (2 + \beta)}} \tau_1 \,, \tag{2.2.2}$$

где $\beta = \dfrac{C_1}{C_2}$.

При этой величине t_1 упущенная выгода равна:

$$S = C_2 \tau_2 \left(\gamma \sqrt{\beta (2 + \beta)} + 1 \right), \tag{2.2.3}$$

где $\gamma = \dfrac{\tau_1}{\tau_2}$.

Если первым завершается второй проект, то оптимальный момент его завершения:

$$t_2 = \frac{1 + \beta}{\sqrt{1 + 2\beta}} \tau_2 \,, \tag{2.2.4}$$

а упущенная выгода:

$$S' = C_2 \tau_2 \left(\sqrt{1 + 2\beta} + \gamma \beta \right). \tag{2.2.5}$$

Сравнивая (2.3) и (2.5), определяем:

$$\gamma' = \frac{\sqrt{2\beta + 1} - 1}{\sqrt{\beta (2 + \beta)} - \beta} \,. \tag{2.2.6}$$

Окончательно получаем, что при $S < S'$ и $\gamma < \gamma'$ в оптимальном решении сначала завершается первый проект, а затем второй, а в случае $S > S'$ и $\gamma > \gamma'$ наоборот, сначала завершится второй проект, а затем первый. Если проекты равноценны с точки зрения упущенной выгоды ($\beta = 1$), то $\gamma' = 1$, и первым завершится проект с меньшим объемом работ. Если проекты одинаковы по объему ($\gamma = 1$), то первым завершается более важный проект (то есть с большей величиной C_i).

Алгоритм определения оптимальной очередности реализации проектов:

1. Рассматриваются 1 и 2 проекты. Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_2 > N$. Возникает конфликтная ситуация.
2. Пусть первым завершается 1 проект. Определяется упущенная выгода по формуле (2.2.3).
3. Пусть первым завершается 2 проект. Определяется упущенная выгода по формуле (2.2.5).
4. Определяем γ' по формуле (2.2.6).
5. Аналогично рассматриваются все варианты реализации проектов.
6. Результаты расчетов сводятся в таблицу 2.4.

Таблица 2.4 - Форма для результатов расчета

Проекты	τ_i мес.	β	S_i, млн. р.	γ	γ'	Очередность реализации
1						
2						
1						
3						
и т.д.						

7. Сравнивая γ и γ' определяется оптимальная очередность реализации проектов.

2.2.2. Пример решения

Исходные данные

Исходные данные помещены в таблицу 2.5, необходимо определить оптимальную очередность реализации проектов.

Таблица 2.5 - Исходные данные

№ проекта	Проект	W_i, млн. р.	a_i, млн. р./мес.	τ_i мес.	C_i, млн. р-
1	Участок 15 км	5600	650	18,6	450
2	Участок 22 км	7500	500	25	600
3	Участок 8 км	2560	450	8,5	200
4	Участок 48 км	7820	550	26	620

Уровень финансирования мультипроекта N=900 млн. р./мес.

Процедура вычислений

1. Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_2 = 650 + 500 = 1150 > 900$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{450}{600} = 0,75;$$

$$\gamma = \frac{18,6}{25} = 0,744; \ \gamma' = \frac{\sqrt{2 \cdot 0,75 + 1} - 1}{\sqrt{0,75(2 + 0,75)} - 0,75} = 0,847.$$

1.1. Первым завершается 1 проект.

$$S(1;2) = 600 \cdot 25\left(0,74\sqrt{0,75(2 + 0,75)} + 1\right) = 31027,3 \text{ млн. р.}$$

1.2. Первым завершается 2 проект.

$$S(2;1) = 600 \cdot 25\left(\sqrt{1 + 2 \cdot 0,75} + 0,74 \cdot 0,75\right) = 32087,1 \text{ млн. р.}$$

2. Проекты 1 и 3 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_3 = 650 + 450 = 1100 > 900$ млн. р/мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{450}{200} = 2,25;$$

$$\gamma = \frac{18,6}{8,5} = 2,188; \ \gamma' = \frac{\sqrt{2 \cdot 2,25 + 1} - 1}{\sqrt{2,25(2 + 2,25)} - 2,25} = 1,597.$$

2.1. Первым завершается 1 проект.

$$S(1;3) = 200 \cdot 8,5\left(2,2\sqrt{2,25(2 + 2,25)} + 1\right) = 13203,5 \text{ млн. р.}$$

2.2. Первым завершается 3 проект.

$$S(3;1) = 200 \cdot 8,5\left(\sqrt{1 + 2 \cdot 2,25} + 2,2 \cdot 2,25\right) = 12356,9 \text{ млн. р.}$$

3. Проекты 1 и 4 не могут финансироваться на максимальном уровне, т.к. $a_1 + a_4 = 650 + 550 = 1200 > 900$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{450}{620} = 0,73;$$

$$\gamma = \frac{18,6}{26} = 0,726; \ \gamma' = \frac{\sqrt{2 \cdot 0,73 + 1} - 1}{\sqrt{0,73(2 + 0,73)} - 0,73} = 0,831.$$

3.1. Первым завершается 1 проект.

$$S(1;4) = 600 \cdot 26\left(0,72\sqrt{0,73(2 + 0,73)} + 1\right) = 32340,4 \text{ млн. р.}$$

3.2. Первым завершается 4 проект.

$$S(4;1) = 600 \cdot 26\left(\sqrt{1 + 2 \cdot 0,73} + 0,72 \cdot 0,73\right) = 33610,1 \text{ млн. р.}$$

4. Проекты 2 и 3 не могут финансироваться на максимальном уровне, т.к. $a_2 + a_3 = 500 + 450 = 950 > 900$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{600}{200} = 3;$$

$$\gamma = \frac{25}{8,5} = 2,941 \text{ ; } \gamma' = \frac{\sqrt{2 \cdot 3 + 1} - 1}{\sqrt{3(2+3)} - 3} = 1,885 \cdot$$

4.1. Первым завершается 2 проект.

$$S(2;3) = 200 \cdot 8,5\left(2,94\sqrt{3(2+3)} + 1\right) = 21064,9 \text{ млн. р.}$$

4.2. Первым завершается 3 проект.

$$S(3;2) = 200 \cdot 8,5\left(\sqrt{1 + 2 \cdot 3} + 2,94 \cdot 3\right) = 19497,8 \text{ млн. р.}$$

5. Проекты 2 и 4 не могут финансироваться на максимальном уровне, т.к. $a_2 + a_4 = 500 + 550 = 1050 > 900$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{600}{620} = 0,968 \text{ ;}$$

$$\gamma = \frac{25}{26} = 0,962 \text{ ; } \gamma' = \frac{\sqrt{2 \cdot 0,97 + 1} - 1}{\sqrt{0,97(2 + 0,97)} - 0,97} = 0,981 \cdot$$

5.1. Первым завершается 2 проект.

$$S(2;4) = 620 \cdot 26\left(0,96\sqrt{0,97(2 + 0,97)} + 1\right) = 42387,9 \text{ млн. р.}$$

5.2. Первым завершается 4 проект.

$$S(4;2) = 620 \cdot 26\left(\sqrt{1 + 2 \cdot 0,97} + 0,96 \cdot 0,97\right) = 42618,8 \text{ млн. р.}$$

6. Проекты 3 и 4 не могут финансироваться на максимальном уровне, т.к. $a_3 + a_4 = 450 + 550 = 1000 > 900$ млн. р./мес.

На основе исходных данных необходимо определить:

$$\beta = \frac{200}{620} = 0,323 \text{ ;}$$

$$\gamma = \frac{8,5}{26} = 0,327 \text{ ; } \gamma' = \frac{\sqrt{2 \cdot 0,32 + 1} - 1}{\sqrt{0,32(2 + 0,32)} - 0,32} = 0,521 \cdot$$

6.1. Первым завершается 3 проект.

$$S(3;4) = 620 \cdot 26\left(0,33\sqrt{0,32(2 + 0,32)} + 1\right) = 20681,6 \text{ млн. р..}$$

6.2. Первым завершается 4 проект.

$$S(4;3) = 620 \cdot 26\left(\sqrt{1 + 2 \cdot 0,32} + 0,33 \cdot 0,32\right) = 22376,1 \text{ млн. р.}$$

7. Результаты расчетов сведены в таблицу 2.6.

Таблица 2.6 - Результаты расчетов

Проекты	τ_i мес.	β	S_i, млн. р.	γ	γ'	Очередность реализации
1	18,6	0,75	31027,3	0,744	0,847	1→2
2	25		32087,1			
1	18,6	2,25	13203,5	2,188	1,597	3→1
3	8,5		12356,9			
1	18,6	0,726	32340,4	0,715	0,831	1→4
4	26		33610,1			
2	25	3	21064,9	2,941	1,885	3→2
3	8,5		19497,8			
2	25	0,968	42387,9	0,962	0,981	2→4
4	26		42618,8			
3	8,5	0,323	20681,6	0,327	0,521	3→4
4	26		22376,1			

8. Оптимальная очередность реализации проектов 3→1→2→4.

На рисунке 2.3 представлен документ *Mathcad*, содержащий решение задачи управления очередностью реализации проектов реконструкции железных дорог при линейной зависимости продолжительности от стоимости.

Решение задачи выполнено с применением средств программирования и встроенных функций *if* и *otherwise*, позволяющих реализовать режим проверки «если-то-иначе».

Результаты решения представляют собой значения «упущенной выгоды» для рассматриваемых проектов, а также значения промежуточных вычислений.

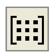
Алгоритм решения задачи с помощью *Mathcad* следующий:

1) Сначала следует заполнить значениями векторы данных (отмечены желтым фоном). Используя мышь, навести курсор на вектор *W* таким образом, чтобы он оказался справа от знака присваивания, после чего нажать на левую кнопку. Уголковый курсор должен принять правое положение ⌊ (если этого не произошло, то необходимо скорректировать его положение с помощью стрелки влево «**←**» на клавиатуре).

2) После появления уголкового курсора в правом положении необходимо нажать клавишу «**Пробел**» на клавиатуре. Уголковый курсор полностью охватит матрицу.

3) На клавиатуре следует дважды нажать на клавишу «***Del***». Матрица, существующая изначально в шаблоне *Mathcad* исчезнет.

4) Для ввода вектора *W* с данными, указанными в задании, необходимо предварительно вызвать панель **Матрица**.

5) На панели **Матрица** с помощью мыши следует нажать на значок матицы . В появившемся окне следует ввести количество строк и столбцов, после нажать клавишу «***Enter***» на клавиатуре. В данном случае строк – 4 и столбцов – 1.

6) В появившиеся ячейки вектора необходимо последовательно ввести исходные данные в соответствии с заданием. После ввода последнего значения нажать клавишу «***Enter***» на клавиатуре.

7) В отношении остальных векторов необходимо проделать операции аналогичные пунктам 2–7.

8) Также необходимо изменить значение для параметра *N*.

В результате определения оптимальной очередности реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости с использованием критерия «упущенная выгода» была установлена очередность реализации проектов: Участок 8 км → Участок 15 км → Участок 22 км → Участок 48 км.

Управление очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости

1) Исходные данные:

$W := \begin{pmatrix} 5600 \\ 7500 \\ 2560 \\ 7820 \end{pmatrix}$ млн. р. $\quad a := \begin{pmatrix} 650 \\ 500 \\ 450 \\ 550 \end{pmatrix}$ млн. р./мес. $\quad \tau := \begin{pmatrix} 18.6 \\ 25 \\ 8.5 \\ 26 \end{pmatrix}$ мес. $\quad C := \begin{pmatrix} 450 \\ 600 \\ 200 \\ 620 \end{pmatrix}$ млн. р.

$N := 900$ млн. р./мес.

ORIGIN := 1

2) Блок решения

1) Проекты 1 и 2 не могут финансироваться на максимальном уровне, т.к.

$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_2 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases}$ $\quad R = \text{"Необходимо выполнить расчет"}$

На основе исходных данных необходимо определить:

$\beta := \dfrac{C_1}{C_2} = 0.75 \quad \gamma := \dfrac{\tau_1}{\tau_2} = 0.744 \quad \gamma' := \dfrac{\sqrt{2\cdot\beta + 1} - 1}{\sqrt{\beta\cdot(2+\beta)} - \beta} = 0.847$

1.1. Проект 1 финансируется на максимальном уровне.

$S_{1\_2} := C_2\cdot\tau_2\cdot\left[\gamma\cdot\sqrt{\beta\cdot(2+\beta)} + 1\right] = 31027.3$ млн. р.

1.2. Проект 2 финансируется на максимальном уровне.

$S_{2\_1} := C_2\cdot\tau_2\cdot\left(\sqrt{1 + 2\cdot\beta} + \gamma\cdot\beta\right) = 32087.1$ млн. р.

Рисунок 2.3 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* (*начало*)

2) Проекты 1 и 3 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_3 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases}$$

R = "Необходимо выполнить расчет"

На основе исходных данных необходимо определить:

$$\beta := \frac{C_1}{C_3} = 2.25 \qquad \gamma := \frac{\tau_1}{\tau_3} = 2.188 \qquad \gamma' := \frac{\sqrt{2\cdot\beta + 1} - 1}{\sqrt{\beta\cdot(2+\beta)} - \beta} = 1.597$$

2.1. Проект 1 финансируется на максимальном уровне.

$$S_{1\_3} := C_3\cdot\tau_3\cdot\left[\gamma\cdot\sqrt{\beta\cdot(2+\beta)} + 1\right] = 13203.5 \text{ млн. р.}$$

2.2. Проект 3 финансируется на максимальном уровне.

$$S_{3\_1} := C_3\cdot\tau_3\cdot\left(\sqrt{1 + 2\cdot\beta} + \gamma\cdot\beta\right) = 12356.9 \text{ млн. р.}$$

3) Проекты 1 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_1 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases}$$

R = "Необходимо выполнить расчет"

На основе исходных данных необходимо определить:

$$\beta := \frac{C_1}{C_4} = 0.726 \qquad \gamma := \frac{\tau_1}{\tau_4} = 0.715 \qquad \gamma' := \frac{\sqrt{2\cdot\beta + 1} - 1}{\sqrt{\beta\cdot(2+\beta)} - \beta} = 0.831$$

3.1. Проект 1 финансируется на максимальном уровне.

$$S_{1\_4} := C_4\cdot\tau_4\cdot\left[\gamma\cdot\sqrt{\beta\cdot(2+\beta)} + 1\right] = 32340.4 \text{ млн. р.}$$

3.2. Проект 4 финансируется на максимальном уровне.

$$S_{4\_1} := C_4\cdot\tau_4\cdot\left(\sqrt{1 + 2\cdot\beta} + \gamma\cdot\beta\right) = 33610.1 \text{ млн. р.}$$

Рисунок 2.3 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* (*продолжение*)

4) Проекты 2 и 3 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_2 + a_3 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_2}{C_3} = 3 \qquad \gamma := \frac{\tau_2}{\tau_3} = 2.941 \qquad \gamma' := \frac{\sqrt{2 \cdot \beta + 1} - 1}{\sqrt{\beta \cdot (2 + \beta)} - \beta} = 1.885$$

4.1. Проект 2 финансируется на максимальном уровне.

$$S_{2\_3} := C_3 \cdot \tau_3 \cdot \left[\gamma \cdot \sqrt{\beta \cdot (2 + \beta)} + 1\right] = 21064.9 \text{ млн. р.}$$

4.2. Проект 3 финансируется на максимальном уровне.

$$S_{3\_2} := C_3 \cdot \tau_3 \cdot \left(\sqrt{1 + 2 \cdot \beta} + \gamma \cdot \beta\right) = 19497.8 \text{ млн. р.}$$

5) Проекты 2 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_2 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases} \qquad R = \text{"Необходимо выполнить расчет"}$$

На основе исходных данных необходимо определить:

$$\beta := \frac{C_2}{C_4} = 0.968 \qquad \gamma := \frac{\tau_2}{\tau_4} = 0.962 \qquad \gamma' := \frac{\sqrt{2 \cdot \beta + 1} - 1}{\sqrt{\beta \cdot (2 + \beta)} - \beta} = 0.981$$

5.1. Проект 2 финансируется на максимальном уровне.

$$S_{2\_4} := C_4 \cdot \tau_4 \cdot \left[\gamma \cdot \sqrt{\beta \cdot (2 + \beta)} + 1\right] = 42387.9 \text{ млн. р.}$$

5.2. Проект 4 финансируется на максимальном уровне.

$$S_{4\_2} := C_4 \cdot \tau_4 \cdot \left(\sqrt{1 + 2 \cdot \beta} + \gamma \cdot \beta\right) = 42618.8 \text{ млн. р.}$$

Рисунок 2.3 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* (*продолжение*)

6) Проекты 3 и 4 не могут финансироваться на максимальном уровне, т.к.

$$R := \begin{cases} \text{"Необходимо выполнить расчет"} & \text{if } a_3 + a_4 > N \\ \text{"Расчет не требуется"} & \text{otherwise} \end{cases}$$

$R = $ "Необходимо выполнить расчет"

На основе исходных данных необходимо определить:

$$\beta := \frac{C_3}{C_4} = 0.323 \qquad \gamma := \frac{\tau_3}{\tau_4} = 0.327 \qquad \gamma' := \frac{\sqrt{2\cdot\beta + 1} - 1}{\sqrt{\beta\cdot(2+\beta)} - \beta} = 0.521$$

6.1. Проект 3 финансируется на максимальном уровне.

$$S_{3\_4} := C_4\cdot\tau_4\cdot\left[\gamma\cdot\sqrt{\beta\cdot(2+\beta)} + 1\right] = 20681.6 \text{ млн. р.}$$

6.2. Проект 4 финансируется на максимальном уровне.

$$S_{4\_3} := C_4\cdot\tau_4\cdot\left(\sqrt{1 + 2\cdot\beta} + \gamma\cdot\beta\right) = 22376.1 \text{ млн. р.}$$

Рисунок 2.3 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* (*окончание*)

На рисунке 2.4 представлен документ *Mathcad*, содержащий второй вариант решения задачи.

Решение выполнено с применением средств программирования. Разработано три программных блока: блок расчета значений «упущенной выгоды» для рассматриваемых проектов (*R*), блок определения вариантов очередности реализации проектов (*f*), блок определения оптимальной очередности реализации проектов (*pos*).

Численные результаты, полученные с помощью программных блоков, представлены в матрицах *RR*, *H* и *KK*.

Управление очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости

1) Исходные данные:

$$Nm := \begin{pmatrix} \text{"Уч-к 15 км"} \\ \text{"Уч-к 22 км"} \\ \text{"Уч-к 8 км"} \\ \text{"Уч-к 48 км"} \end{pmatrix} \quad \overset{\text{млн. р.}}{W := \begin{pmatrix} 5600 \\ 7500 \\ 2560 \\ 7820 \end{pmatrix}} \quad \overset{\text{млн. р./мес.}}{a := \begin{pmatrix} 650 \\ 500 \\ 450 \\ 550 \end{pmatrix}} \quad \overset{\text{мес.}}{\tau := \begin{pmatrix} 18.6 \\ 25 \\ 8.5 \\ 26 \end{pmatrix}} \quad \overset{\text{млн. р.}}{C := \begin{pmatrix} 450 \\ 600 \\ 200 \\ 620 \end{pmatrix}}$$

$N := 900$ млн. р./мес.

Рисунок 2.4 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*начало*)

2) Блок решения

ORIGIN := 1

$$
R(W,a,\tau,C,N) := \left|
\begin{array}{l}
q \leftarrow \mathrm{rows}(W) \\
m \leftarrow 1 \\
n \leftarrow 1 \\
\text{for } i \in 1..q-1 \\
\quad \text{for } j \in i+1..q \\
\quad\quad \text{if } a_i + a_j > N \\
\quad\quad\quad Nn_m \leftarrow m \\
\quad\quad\quad Ii_m \leftarrow i \\
\quad\quad\quad Jj_m \leftarrow j \\
\quad\quad\quad T1_m \leftarrow \tau_i \\
\quad\quad\quad T2_m \leftarrow \tau_j \\
\quad\quad\quad \beta_m \leftarrow \dfrac{C_i}{C_j} \\
\quad\quad\quad \gamma_m \leftarrow \dfrac{\tau_i}{\tau_j} \\
\quad\quad\quad \gamma'_m \leftarrow \dfrac{\left(\sqrt{2\cdot\beta_m+1}-1\right)}{\sqrt{\beta_m\cdot\left(2+\beta_m\right)}-\beta_m} \\
\quad\quad\quad Sf_m \leftarrow C_j\cdot\tau_j\cdot\left[\gamma_m\cdot\sqrt{\beta_m\cdot\left(2+\beta_m\right)}+1\right] \\
\quad\quad\quad Ss_m \leftarrow C_j\cdot\tau_j\cdot\left(\sqrt{1+2\cdot\beta_m}+\gamma_m\cdot\beta_m\right) \\
\quad\quad\quad \text{if } \left(Sf_m < Ss_m\right) \wedge \left(\gamma_m < \gamma'_m\right) \\
\quad\quad\quad\quad Ik_m \leftarrow i \\
\quad\quad\quad\quad Jk_m \leftarrow j \\
\quad\quad\quad \text{if } Sf_m > Ss_m \\
\quad\quad\quad\quad Ik_m \leftarrow j \\
\quad\quad\quad\quad Jk_m \leftarrow i \\
\quad\quad\quad m \leftarrow m+1 \\
\quad i \leftarrow i+1 \\
\mathrm{augment}(Nn, Ii, Jj, T1, T2, \beta, Sf, Ss, \gamma, \gamma', Ik, Jk)
\end{array}\right.
$$

Рисунок 2.4 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*продолжение*)

$$RR := R(W, a, \tau, C, N)$$

$$RR = \begin{pmatrix} 1 & 1 & 2 & 18.6 & 25 & 0.75 & 31027.33 & 32087.08 & 0.74 & 0.85 & 1 & 2 \\ 2 & 1 & 3 & 18.6 & 8.5 & 2.25 & 13203.46 & 12356.85 & 2.19 & 1.6 & 3 & 1 \\ 3 & 1 & 4 & 18.6 & 26 & 0.73 & 32340.44 & 33610.1 & 0.72 & 0.83 & 1 & 4 \\ 4 & 2 & 3 & 25 & 8.5 & 3 & 21064.92 & 19497.78 & 2.94 & 1.89 & 3 & 2 \\ 5 & 2 & 4 & 25 & 26 & 0.97 & 42387.85 & 42618.81 & 0.96 & 0.98 & 2 & 4 \\ 6 & 3 & 4 & 8.5 & 26 & 0.32 & 20681.58 & 22376.13 & 0.33 & 0.52 & 3 & 4 \end{pmatrix}$$

```
f(RR,Nm) := | x ← rows(RR)
            | y ← rows(Nm)
            | mm ← 1
            | for  ii ∈ 1..y
            |   | z ← 0
            |   | b ← 0
            |   | for  jj ∈ 1..x
            |   |   | b ← b + 1  if  RR[jj,11] = ii
            |   |   | z ← z + 1  if  RR[jj,12] = ii
            |   | L1[ii] ← b
            |   | L2[ii] ← z
            |   | Pr[ii] ← ii
            |   | Nmm[ii] ← Nm[ii]
            | augment(Pr,Nmm,L1,L2)
```

f(RR,Nm) =

	1	2	3	4
1	1	"Уч-к 15 км"	2	1
2	2	"Уч-к 22 км"	1	2
3	3	"Уч-к 8 км"	3	0
4	4	"Уч-к 48 км"	0	3

Рисунок 2.4 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*продолжение*)

$$H := f(RR, Nm)$$

$$
pos(H) := \left|
\begin{array}{l}
H1 \leftarrow H \\
\text{for } ik \in 1..rows(H1) - 1 \\
\quad \left|
\begin{array}{l}
nmax \leftarrow ik \\
\text{for } jk \in ik + 1..rows(H) \\
\quad nmax \leftarrow jk \quad \text{if } \left(H1^{\langle 3 \rangle}\right)_{jk} > \left(H1^{\langle 3 \rangle}\right)_{nmax} \\
\text{for } im \in 1..cols(H1) \qquad\qquad \text{if } nmax \neq ik \\
\quad \left|
\begin{array}{l}
cc_{im} \leftarrow \left(H1^{\langle im \rangle}\right)_{ik} \\
\left(H1^{\langle im \rangle}\right)_{ik} \leftarrow \left(H1^{\langle im \rangle}\right)_{nmax} \\
\left(H1^{\langle im \rangle}\right)_{nmax} \leftarrow cc_{im}
\end{array}
\right.
\end{array}
\right. \\
augment\left(H1^{\langle 1 \rangle}, H1^{\langle 2 \rangle}\right)
\end{array}
\right.
$$

$$KK := pos(H)$$

KK =

	1	2
1	3	"Уч-к 8 км"
2	1	"Уч-к 15 км"
3	2	"Уч-к 22 км"
4	4	"Уч-к 48 км"

Рисунок 2.4 – Решение задачи управления очередностью реализации проектов реконструкции железных дорог при степенной зависимости продолжительности от стоимости в *Mathcad* с применением программных блоков (*окончание*)

2.3. Принятие решений в условиях риска при проектном управлении железнодорожным строительством

2.3.1. Теоретические основы

Процесс принятия решений заключается в выборе последовательности действий (альтернативы) для перевода объекта управления из состояния в текущий момент времени в желаемое состояние. Реализация той или иной альтернативы приводит к различным исходам (состояниям объекта). Задача принятия решений в условиях риска возникает в тех случаях, когда с каждым принимаемым решением связано множество исходов. Далее рассмотрены методы принятия решения, соответствующие наиболее простому - дискретному случаю, когда число принимаемых решений равно m: x_i, i=1,2,...,m; а число возможных вероятностных исходов равно n: s_j - возможные исходы с вероятностями $P(s_j \mid x_i)$, j=1,...,n, i=1,...,m. Интерес представляет случай, когда удается оценить полезность $u_{ij} = f(x_i, s_j)$ исхода s_j при принятии решения x_i [6].

Формально модель задачи принятия решения при риске представима в виде матрицы полезности $\|u_{ij}\|$ альтернатив x_i при различных вариантах внешних условий s_j (таблица 2.7):

Таблица 2.7 – Матрица полезности альтернатив

Альтернативы	Варианты внешних условий			
	S_1	S_2	…	S_n
X_1	U_{11}	U_{12}	…	U_{1n}
X_2	U_{21}	U_{22}	…	U_{2n}
…	…	…	…	…
X_m	U_{m1}	U_{m2}	…	U_{mn}

Далее приведены принципы поиска оптимального решения в сформулированных условиях.

Критерий «ожидаемое значение». Критерий ожидаемого значения используется, если необходимо максимизировать ожидаемую прибыль (или минимизировать ожидаемые затраты). Использование ожидаемых величин предполагает возможность многократного решения одной и той же задачи, пока не будут получены достаточно точные расчетные формулы. Математически это выглядит так: пусть X - это случайная величина с математическим ожиданием MX и дисперсией DX. Если x_1, x_2,.., x_n - значения случайной величины (CB)X, то среднее арифметическое их значений

$$\bar{x} = \frac{x_1 + x_2 + \ldots + x_n}{n} \qquad 2.3.1$$

имеет дисперсию

$$\frac{DX}{n}. \qquad 2.3.2$$

Таким образом, при $n \to \infty$,

$$\frac{DX}{n} \to 0 \text{ и } \bar{x} \to MX \qquad 2.3.3$$

Таким образом, при достаточно большом объеме выборки разница между средним арифметическим и математическим ожиданием стремится к нулю. Следовательно, использование критерия ожидаемого значения возможно только в случае, когда одно и то же решение приходится применять достаточно большое число раз. Для решений, которые приходится принимать небольшое число раз, критерий будет давать неверные результаты.

Перейдем к практическому применению критерия. Необходимо принять решение о том, когда нужно проводить профилактический ремонт парка автотехники (АТ) строительной (подрядной) организации (СПО), чтобы минимизировать потери из-за случайных поломок. Поскольку невозможно предсказать заранее, когда возникнет неисправность, необходимо найти вероятность того, что единица АТ выйдет из строя в период времени t. В этом и состоит элемент риска.

Математически это выглядит следующим образом: единица АТ ремонтируется индивидуально, если она остановилась из-за поломки. Через T интервалов времени выполняется профилактический ремонт всех n единиц АТ. Требуется определить оптимальное значение T, при котором минимизируются общие затраты на ремонт неисправных единиц АТ и проведение профилактического ремонта в расчете на один интервал времени.

Пусть p_t - вероятность выхода из строя одной единицы АТ в момент t, n_t - случайная величина, равная числу всех вышедших из строя единиц АТ в один и тот же момент. Пусть далее C_1 - затраты на ремонт неисправной единицы АТ и C_2 -затраты на профилактический ремонт одной единицы АТ.

Применение критерия ожидаемого значения в данном случае оправданно, если автотехника работает в течение большого периода времени. При этом ожидаемые затраты на один интервал составят

$$ОЗ = \frac{C_1 \sum_{t=1}^{T-1} M(n_t) + C_2 \cdot n}{T}, \qquad 2.3.4$$

где $M(n_t)$ - математическое ожидание числа вышедших из строя единиц АТ в момент t. Так как n_t имеет биномиальное распределение с параметрами (n, p_t), то $M(n_t) = np_t$. Таким образом

$$ОЗ = \frac{n\left(C_1 \sum_{t=1}^{T-1} p_t + C_2\right)}{T}. \qquad 2.3.5$$

Необходимые условия оптимальности T* имеют вид:

$$\begin{aligned} &ОЗ(T*-1) \geq ОЗ(T*), \\ &ОЗ(T*+1) \geq ОЗ(T*), \end{aligned} \qquad 2.3.6$$

Следовательно, начиная с малого значения T, вычисляют $ОЗ(T)$, пока не будут удовлетворены необходимые условия оптимальности; т.е. будет найден такой момент времени T*, что затраты и в предшествующий ему период, и в последующий будут превосходить ожидаемые затраты в период T*.

Критерий «ожидаемое значение – дисперсия». Критерий ожидаемого значения можно модифицировать так, что его можно будет применить и для редко повторяющихся ситуаций.

Если X-CB с дисперсией DX, то среднее арифметическое $\overline{X}$ имеет дисперсию $\frac{DX}{n}$, где n - число слагаемых в $\overline{X}$. Следовательно, если DX уменьшается, вероятность того, что $\overline{X}$ близко к MX, увеличивается. Следовательно, целесообразно ввести критерий, в котором максимизация ожидаемого значения прибыли сочетается с минимизацией ее дисперсии.

Для применения критерия «ожидаемое значение - дисперсия» найдем дисперсию затрат за один интервал времени, т.е. дисперсию величины

$$З_T = \frac{C_1 \sum_{t=1}^{T-1} n_t + C_2 \cdot n}{T}, \qquad 2.3.7$$

Так как n_t, t=1,...,T-1, - *СВ*, то $З_T$ также *СВ*. Случайная величина n_t имеет биномиальное распределение с математическим ожиданием $M(n_t) = np_t$, и дисперсией $D(n_t) = np_t(1 - p_t)$. Следовательно,

$$D(З_T) = D\left(\frac{C_1 \sum_{t=1}^{T-1} n_t + C_2 \cdot n}{T}\right) = \left(\frac{C_1}{T}\right)^2 D\left(\sum_{t=1}^{T-1} n_t\right) = n\left(\frac{C_1}{T}\right)^2 \left[\sum_{t=1}^{T-1} p_t - \sum_{t=1}^{T-1} p_t^{\,2}\right], \qquad 2.3.8$$

где $C_2 n = const$. Известно, что $M(З_T) = M(З(T))$.

Следовательно, искомым критерием будет минимум выражения $M(З(T)) + kD(З_T)$.

Константу k можно рассматривать как уровень «несклонности» к риску. Например, если нежелательны большие отрицательные отклонения прибыли вниз от $M(З(T))$, то нужно выбрать k много больше 1. Это придает больший вес дисперсии и приводит к решению, уменьшающему вероятность больших потерь прибыли.

При $k = 1$ получаем задачу со следующим критерием оптимизации

$$M(З(T)) + kD(З_T) = n\left[\left(\frac{C_1}{T} + \frac{C_1^2}{T^2}\right)\sum_{t=1}^{T-1} p_t - \left(\frac{C_1}{T}\right)^2 \sum_{t=1}^{T-1} p_t^2 + \frac{C_2}{T}\right]. \qquad 2.3.9$$

Оптимальное значение T* выбирается таким же образом, как и для предыдущего критерия.

2.3.2. Пример решения

<u>Исходные данные</u>

Необходимо принять решение о проведении профилактического ремонта парка из 50 машин, чтобы минимизировать потери от неисправностей при условии, что затраты на ремонт неисправной единицы C_1=100, затраты на профилактический ремонт одной единицы техники составляют C_2=10; а вероятности выхода из строя одной единицы p_t в момент времени t приведены во втором столбце таблице 2.8 [5,6].

Таблица 2.8 - Схема определения оптимального значения T* для критерия «ожидаемое значение»

T	P_t	$\sum_{t=1}^{T-1} P_t$	$ОЗ(T)$
1	0,05	0	$\frac{50(100 \cdot 0 + 10)}{1} = 500$
2	0,07	0,05	375
3	0,1	0,12	366,7
4	0,13	0,22	400
5	0,18	0,35	450

<u>Процедура вычислений</u>

В таблице 2.8 в столбцах 3 и 4 представлены результаты определения оптимального значения T^* для критерия «ожидаемое значение». В итоге установлено: $T^* \to 3$, $ОЗ(T^*) \to 366{,}7$.

Следовательно, профилактический ремонт необходимо делать через $T^*=3$ интервала времени.

Критерию «ожидаемое значение - дисперсия» соответствует таблица 2.9.

Таблица 2.9 - Схема определения оптимального значения T^* для критерия «ожидаемое значение – дисперсия»

T	P_t	P_t^2	$\sum_{t=1}^{T-1} P_t$	$\sum_{t=1}^{T-1} P_t^2$	$M(З(T))+kD(З_T)$
1	0,05	0,0025	0	0	500,00
2	0,07	0,0049	0,05	0,0025	6312,50
3	0,1	0,01	0,12	0,0074	6622,22
4	0,13	0,0169	0,22	0,0174	6731,25
5	0,18	0,0324	0,35	0,0343	6764,00

Из таблицы 2.9 видно, что профилактический ремонт надо проводить в течение каждого интервала $T^*=1$.

На рисунке 2.5 представлено решение задачи принятия решений в условиях риска при проектном управлении железнодорожным строительством в *Mathcad* с применением программных блоков.

Принятие решений в условиях риска при проектном управлении железнодорожным строительством

<u>1) Исходные данные:</u>

$n := 50$ - количество машин

$T := \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$ - время

$C_1 := 100$ - затраты на ремонт неисправной машины

$P_t := \begin{pmatrix} 0.05 \\ 0.07 \\ 0.10 \\ 0.13 \\ 0.18 \end{pmatrix}$ - вероятность выхода машины из строя в момент времни T

$C_2 := 10$ - затраты на профилактический ремонт машины

Рисунок 2.5 – Решение задачи принятия решений в условиях риска при проектном управлении железнодорожным строительством в *Mathcad* с применением программных блоков (*начало*)

2) Блок решения

$ORIGIN := 1$

Критерий «ожидаемое значение»

$$O3(n, C_1, C_2, T, P_t) := \left| \begin{array}{l} P_1 \leftarrow 0 \\ O_1 \leftarrow \dfrac{n \cdot (C_1 \cdot P_1 + C_2)}{T_1} \\ \text{for } i \in 2..\text{rows}(T) \\ \quad \left| \begin{array}{l} P_i \leftarrow P_{i-1} + P_{t_{i-1}} \\ O_i \leftarrow \dfrac{n \cdot (C_1 \cdot P_i + C_2)}{T_i} \end{array} \right. \\ \text{augment}(T, P_t, P, O) \end{array} \right.$$

$$O3_f := O3(n, C_1, C_2, T, P_t)$$

$$O3_f = \begin{pmatrix} 1 & 0.05 & 0 & 500 \\ 2 & 0.07 & 0.05 & 375 \\ 3 & 0.1 & 0.12 & 366.667 \\ 4 & 0.13 & 0.22 & 400 \\ 5 & 0.18 & 0.35 & 450 \end{pmatrix}$$

$$T_f(O3_f) := \left| \begin{array}{l} A \leftarrow \min\left(O3_f^{\langle 4 \rangle}\right) \\ \text{for } i \in 1..\text{rows}(O3_f) \\ \quad TT \leftarrow \left(O3_f^{\langle 1 \rangle}\right)_i \text{ if } A = \left(O3_f^{\langle 4 \rangle}\right)_i \\ \text{stack}(A, TT) \end{array} \right.$$

$$O3_{T'} := T_f(O3_f)_1 \qquad T' := T_f(O3_f)_2$$

$$O3_{T'} = 366.667 \qquad T' = 3$$

Рисунок 2.5 – Решение задачи принятия решений в условиях риска при проектном управлении железнодорожным строительством в *Mathcad* с применением программных блоков (*продолжение*)

Критерий «ожидаемое значение дисперсия»

$$O3D(n, C_1, C_2, T, P_t) := \left| \begin{array}{l} P_1 \leftarrow 0 \\ P2_1 \leftarrow 0 \\ M_1 \leftarrow n \cdot \left[\left[\frac{C_1}{T_1} + \frac{C_1^{\,2}}{(T_1)^2} \right] \cdot P_1 - \left(\frac{C_1}{T_1} \right)^2 \cdot P2_1 + \frac{C_2}{T_1} \right] \\ \text{for } i \in 2..\text{rows}(T) \\ \quad \left| \begin{array}{l} P_i \leftarrow P_{i-1} + P_{t_{i-1}} \\ P2_i \leftarrow P2_{i-1} + \left(P_{t_{i-1}}\right)^2 \\ M_i \leftarrow n \cdot \left[\left[\frac{C_1}{T_i} + \frac{C_1^{\,2}}{(T_i)^2} \right] \cdot P_i - \left(\frac{C_1}{T_i} \right)^2 \cdot P2_i + \frac{C_2}{T_i} \right] \end{array} \right. \\ \text{augment}\left(T, P_t, P_t^{\,2}, P, P2, M\right) \end{array} \right.$$

$$O3D_f := O3D(n, C_1, C_2, T, P_t)$$

$$O3D_f = \begin{pmatrix} 1 & 0.05 & 0.0025 & 0 & 0 & 500 \\ 2 & 0.07 & 0.0049 & 0.05 & 0.0025 & 6312.5 \\ 3 & 0.1 & 0.01 & 0.12 & 0.0074 & 6622.22222 \\ 4 & 0.13 & 0.0169 & 0.22 & 0.0174 & 6731.25 \\ 5 & 0.18 & 0.0324 & 0.35 & 0.0343 & 6764 \end{pmatrix}$$

$$T_f(O3D_f) := \left| \begin{array}{l} A \leftarrow \min\left(O3D_f^{\langle 6 \rangle}\right) \\ \text{for } i \in 1..\text{rows}(O3D_f) \\ \quad TT \leftarrow \left(O3D_f^{\langle 1 \rangle}\right)_i \ \text{if } A = \left(O3D_f^{\langle 6 \rangle}\right)_i \\ \text{stack}(A, TT) \end{array} \right.$$

$$O3D_{T'} := T_f(O3D_f)_1 \qquad T' := T_f(O3D_f)_2$$

$$O3D_{T'} = 500 \qquad T' = 1$$

Рисунок 2.5 – Решение задачи принятия решений в условиях риска при проектном управлении железнодорожным строительством в *Mathcad* с применением программных блоков (*окончание*)

2.4. Автоматизированный расчет числовых параметров календарных планов

2.4.1. Теоретические основы

Рассчитать сетевой график строительного процесса возведения опоры контактной сети (КС) с учетом длительности выполнения всех работ. Расчет можно выполнить с применением методов *CPM* и *PERT* [5,7].

Для понимания процедуры расчета сетевого графика указанными методами далее будут приведены необходимые теоретические сведения из области организации и управления строительным производством.

Строительный процесс — это комплекс работ, представляющий собой некоторую конечную совокупность отдельных взаимосвязанных работ, направленный на создание заранее определенной продукции с заданными параметрами при определенных ограничениях.

Работа — это процесс, происходящий во времени. Работа или несколько работ могут завершаться наступлением некоторого события, которое может быть завершающим, если оно не имеет последующих за ним работ, или исходным, если после него следует одна или несколько новых работ. Событие, не имеющее непосредственно предшествующих работ, называется исходным.

<u>Представление проекта в виде диаграммы Ганта.</u> Одним из наиболее распространенных способов наглядного представления производственного процесса или проекта во времени является линейный или ленточный календарный график — Диаграмма Ганта.

Диаграмма Ганта – это график, в котором строительный процесс представлен в двух видах. В *левой части* проект представлен в виде списка задач (работ, операций) проекта в табличном виде с указанием названия задачи и длительности ее выполнения, а часто и работ, предшествующих той или иной задаче. В *правой части* каждая задача проекта, а точнее длительность ее выполнения, отображается графически, обычно в виде отрезка определенной длины с учетом логики выполнения задач проекта.

<u>Представление проекта в виде сетевого графика.</u> Наиболее известным способом представления взаимосвязанного комплекса работ (задач) является сетевой график, который, в свою очередь, может отображать работы с детерминированным и вероятностным временем их выполнения.

Сетевой график – это график, представляющий собой совокупность взаимосвязанных работ и событий. Каждая работа на сетевом графике представляется в виде стрелки, а каждое событие в виде кружочка (в некомпьютерном варианте) или виде прямоугольника (в компьютерном варианте). При этом, сетевой график может быть представлен в виде временной шкалы.

Если сеть не слишком сложна, и содержит сравнительно небольшое число работ, то сетевой график можно изображать во временном масштабе с использованием оси времени. Для этого под сетевым графиком прочерчивается ось времени, а стрелки на сетевом графике изображаются так, чтобы длина их проекций на ось времени была бы равна величинам соответствующих работ (рисунок 2.6).

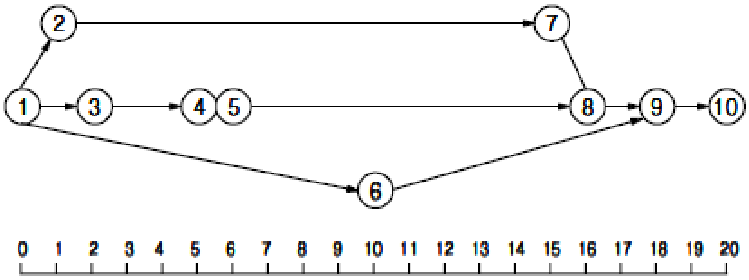


Рисунок 2.6 – Сетевой график строительного процесса

Головная часть стрелки показывает окончание работы, хвостовая – ее начало. Стрелки располагаются в строгом соответствии с ходом выполнения работ. Переход от одной или нескольких работ (стрелок) к одной или нескольким работам (стрелкам) образует событие (кружок).

Выделяют три вида событий: начальное событие, промежуточное событие и конечное событие.

Начальное событие – это событие, фиксирующее начало выполнения одной или нескольких работ процесса. В качестве начального события может выступать, например, выдача задания на выполнение того или иного процесса.

Конечное событие – это событие, фиксирующее завершение выполнения одной или нескольких работ процесса. Конечное событие – это тот момент времени, когда выданное ранее задание реализовано и выполнена последняя работа (задача).

Промежуточное событие – это состояние процесса, фиксирующее завершение выполнение одной или нескольких работ и начало выполнения одной или нескольких работ.

Все работы, ведущие к событию, должны быть закончены прежде, чем могут быть начаты работы, вытекающие из события. Событию, из которого выходит работа, приписывается индекс – i, а событию, в которое входит работа, индекс – j.

Тогда такая работа имеет индекс ij.

Для нумерации событий существует несколько правил:

- каждое событие имеет свой номер; если от события к событию идут несколько стрелок (работ), то с целью однозначного определения работ вводятся фиктивные события;
- для каждой работы номер события в конце работы должен быть больше, чем номер события в начале работы.

Как правило, нумерацию событий производят лишь после построения всего сетевого графика.

Выделяют две временные характеристики для каждого события, которые показаны на рисунке 2.7.

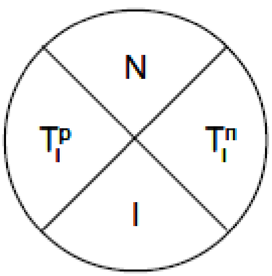


Рисунок 2.7 – Возможный способ изображения события на сетевом графике

T_i^p – **ранний срок свершения события** – минимальный из возможных моментов свершения данного события;

T_i^n – **поздний срок свершения события** – максимальный из допустимых моментов свершения данного события.

Выделяют три временные характеристики для каждой работы:

- **поздний срок начала работы** $t_{ij}^{пн}$ – максимальный из моментов времени, когда выполненный объем работы равен нулю;
- **ранний срок окончания работы** t_{ij}^{po} – это минимальный из моментов времени, когда выполненный объем работы равен ее полному объему;
- **продолжительность работы** t_{ij} – величина промежутка времени между моментами начала и окончания работы.

В процессе расчета сетевого графика для каждой работы, которой соответствует своя стрелка, ставят временные характеристики. В начале стрелки ставят поздний срок начала работы $t_{ij}^{пн}$, в середине стрелки продолжительность работы t_{ij}, а в конце – ранний срок окончания работы t_{ij}^{po}. Так, как показано на рисунке 2.8.

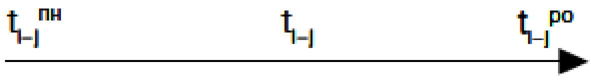


Рисунок 2.8. Изображение работы в сетевом графике

Расчет сетевого графика методом *CPM.* Для расчета сетевого графика методом критического пути *CPM* (*Critical Path Method*) требуются следующие входные данные:

- комплекс работ (операций, задач) процесса (проекта);
- связи между работами;
- оценки длительности выполнения каждой работы;
- календарь рабочего времени проекта;
- календарная дата начала проекта.

Первые три вида данных используются в ручном счете, а весь комплекс входных данным в машинном счете.

Алгоритм расчета сетевого графика методом МКП включает следующие основные этапы:

1. *Определение раннего срока свершения начального (исходного) события* T_1^p.

Для упрощения расчетов ранний срок свершения начального события T_1^p может быть принят равным нулю.

2. *Расчет ранних сроков окончания работ* t_{ij}^{po}. Расчет проводится для работ, которые выходят из событий с рассчитанными ранними сроками свершения событий, по формуле

$$t_{ij}^{po} = T_i^p + t_{ij}. \qquad 2.4.1$$

3. *Расчет раннего срока свершения события* T_j^p. Расчет проводится для событий, в которые входят работы с предварительно рассчитанными ранними сроками окончания работ, по формулам:

- если в событие входит только одна работа, то

$$T_j^p = t_{ij}^{po}; \qquad 2.4.2$$

- если в событие входит несколько работ, то

$$T_j^p = \max\{t_{ij}^{po}\}, i = \overline{1,k}. \qquad 2.4.3$$

Предварительно, если в событие входят несколько работ, то для них должны быть рассчитаны ранние сроки окончания t_{ij}^{po}.

Зная ранние сроки свершения других событий, можно переходить к расчету ранних сроков окончания работ $t_{ij}^{ро}$ для работ, выходящих из этих событий и так далее.

Этапы 2 и 3 выполняются до тех пор, пока не будут рассчитаны ранние сроки окончания каждой работы – $t_{ij}^{ро}$ и ранние сроки свершения каждого события – $T_j^р$.

4. *Расчет позднего срока свершения завершающего события* $T_k^п$. Величина $T_k^п$ определяется, исходя из следующих условий:

- если директивный срок завершения работ не задан, то

$$T_k^п = T_k^р, \qquad 2.4.4$$

- если директивный срок завершения работ задан, то

$$T_k^п = T_{дир}. \qquad 2.4.5$$

5. *Расчет поздних сроков начала работ* $t_{ij}^{пн}$. Расчет проводится для работ, которые входят в события с рассчитанными поздними сроками свершения событий, по формуле

$$t_{ij}^{пн} = T_j^п - t_{ij}. \qquad 2.4.6$$

Расчет проводится от конца сетевого графика до начала.

6. *Расчет позднего срока свершения события* $T_i^п$. Расчет проводится для событий, из которых выходят работы с рассчитанными поздними сроками начала работ, по формулам:

- если из события выходит только одна работа, то

$$T_i^п = t_{ij}^{пн}, \qquad 2.4.7$$

- если из события выходит несколько работ, то

$$T_i^п = \min\{t_{ij}^{пн}\},\ j = \overline{1,\text{p}}. \qquad 2.4.8$$

Если выходят из события несколько работ, то для них предварительно должны быть рассчитаны поздние сроки начала работ $t_{ij}^{пн}$.

Этапы 5 и 6 выполняются до тех пор, пока не будут рассчитаны поздние сроки начала каждой работы — $t_{ij}^{пн}$ и поздние сроки свершения каждого события — $T_i^п$.

7. *Определение резерва времени события* R_i. Вычисляется как разность между поздним и ранним сроками свершения события

$$R_i = T_i^п - T_i^р,\ i = \overline{1,k}. \qquad 2.4.9$$

8. *Определение полного резерва времени работы* $r_{ij}^п$.

$$r_{ij}^п = T_j^п - T_i^р - t_{ij}. \qquad 2.4.10$$

Полный резерв времени работы – это максимальное время, на которое можно отсрочить начало или увеличить продолжительность выполнения работы (*i*–*j*), не изменяя директивный или ранний срок наступления завершающего события.

Рассчитанную величину полного резерва времени выполнения работы размещают на сетевом графике под соответствующей стрелкой в середине.

9. *Определение свободного резерва времени работы* $r_{ij}^с$.

$$r_{ij}^с = T_j^р - T_i^п - t_{ij}. \qquad 2.4.11$$

Свободный резерв времени работы – это минимальное время, на которое можно отсрочить начало или увеличить продолжительность работы (*i*–*j*) при условии, что все события сети наступают в свои ранние сроки.

Рассчитанную величину свободного резерва времени выполнения работы размещают на сетевом графике под соответствующей стрелкой в середине, как правило, в таком виде r_{ij}^{n} / r_{ij}^{c}.

10. *Определение критических и подкритических путей.*

Критический путь – это путь, продолжительность которого равна критическому времени $T_{кр}$.

Критическое время $T_{кр}$ – это минимальное время, в течение которого может быть выполнен весь комплекс работ. Критический путь – это путь, на работах которого полные резервы времени работ принимают минимальные значения.

Если директивный срок не задан, то полные резервы времени работ, лежащих на критическом пути, равны 0.

Подкритические пути – это пути, у которых полный резерв времени отличается от минимального не более, чем на заданную величину. Множество всех критических и подкритических путей называют зоной комплекса работ.

Сетевой график может иметь любое количество критических путей. Отсутствие резервов времени на работы, расположенные на критическом пути, приводит к тому, что невыполнение срока окончания для любой из работ приведет к невыполнению в срок всего производственного процесса. Именно операции, лежащие на критическом пути, требуют бесперебойного обеспечения ресурсами, немедленного вмешательства руководства для ликвидации угрозы срыва выполнения всего производственного процесса в назначенный срок.

Изменение продолжительностей выполнения отдельных работ дает возможность проводить исследование влияния этих работ на продолжительность выполнения всего комплекса работ.

<u>Расчет сетевого графика методом *PERT.*</u> Для расчета сетевого графика методом *PERT* (*Program Evaluation Reserve Time*) требуются следующие входные данные:

- комплекс работ (операций, задач) процесса (проекта);
- связи между работами;
- две или три вероятностные оценки длительности выполнения каждой работы;
- календарь рабочего времени проекта;
- календарная дата начала проекта.

Первые три вида данных используются в ручном счете, а весь комплекс входных данным в машинном счете. Ограничимся пока ручным счетом.

Теперь рассмотрим расчет сетевого графика с вероятностным временем выполнения работ с использование метода *PERT*.

Вероятностный сетевой график – это график, в котором длительности работ (задач) имеют несколько оценок.

Каждая работа характеризуется несколькими временными оценками, которые дают специалисты (эксперты):

- оптимистическая оценка – $t_{\min ij}$;
- наиболее вероятная оценка – $t_{нв ij}$;
- пессимистическая оценка – $t_{\max ij}$.

Оптимистическая оценка – это минимально возможный период времени, в течение которого может быть выполнена данная работа при самых благоприятных условиях.

Наиболее вероятная оценка – это возможный период времени, в течение которого может быть выполнена работа, если бы требовалась только одна оценка.

Пессимистическая оценка – это максимально возможный период времени, в течение которого может быть выполнена данная работа при самых неблагоприятных условиях.

Допустим, что задан директивный срок $T_{дир}$ выполнения всего производственного процесса. Требуется определить вероятность выполнения производственного процесса в срок.

Для определения расчетной продолжительности выполнения работ наибольшее распространение нашли два способа их оценки.

Первый способ требует установления двух временных оценок: оптимистической оценки $t_{\min ij}$ и пессимистической оценки $t_{\max ij}$ для каждой работы. В расчетах же используется расчетная продолжительность выполнения работ, определяемая по формуле

$$t_{расч\, ij} = \frac{3t_{\min ij} + 2t_{\max ij}}{5}. \quad 2.4.12$$

Второй способ требует установления трех временных оценок: оптимистической $t_{\min ij}$, наиболее вероятной $t_{нв\, ij}$ и пессимистической оценки $t_{\max ij}$ для каждой работы, тогда $t_{расч\, ij}$ определяется по формуле

$$t_{расч\, ij} = \frac{t_{\min ij} + 4t_{нв\, ij} + t_{\max ij}}{6}. \quad 2.4.13$$

Предполагается, что существует лишь незначительная возможность того, что расчетное время выполнения операции $t_{расч\, ij}$ будет выходить за пределы значений $t_{\min ij}$ и $t_{\max ij}$.

Для определения вероятности выполнения производственного процесса в срок необходимо построить сетевой график производственного процесса, строго соблюдая логическую последовательность выполнения отдельных работ процесса.

Методика построения сетевого графика с вероятностным временем выполнения операций аналогична методике построения сетевого графика с детерминированным временем выполнения операций.

Алгоритм расчета сетевого графика с вероятностным временем выполнения работ включает следующие основные этапы:

1. *Определение ожидаемого времени выполнения работ $t_{ож\, ij}$ ($t_{расч\, ij}$).* Этот этап можно проводить с использованием как трех, так и двух оценок продолжительности выполнения работ. Используем для нашей задачи три вероятностные оценки.

Ожидаемое время определяется по формуле

$$t_{расч\, ij} = \frac{t_{\min ij} + 4t_{нв\, ij} + t_{\max ij}}{6}. \quad 2.4.14$$

2. *Расчет сетевого графика.* Этот этап использует методику расчета сетевого графика с детерминированным временем выполнения работ.

3. *Определение степени неопределенности выполнения работ, лежащих на критическом пути.* Для оценки степени неопределенности выполнения той или иной работы используется дисперсия.

В первом способе дисперсия рассчитывается по формуле

$$\sigma^2 = \frac{\left(t_{\max ij} - t_{\min ij}\right)^2}{5^2}, \quad 2.4.15$$

во втором способе — по формуле

$$\sigma^2 = \frac{\left(t_{\max ij} - t_{\min ij}\right)^2}{6^2}. \quad 2.4.16$$

В рассматриваемой задаче определение дисперсий производится по второй формуле.

4. *Определение вероятности завершения комплекса работ в заданный директивный срок.* Для этого вначале определяем аргумент функции нормального распределения x по формуле.

$$x = \frac{T_{дир} - T_x^p}{\sqrt{\sum \sigma_{кр\,ij}^2}}, \qquad 2.4.17$$

где T_x^p — ранний срок наступления конечного, завершающего события;

$\sum \sigma_{кр\,ij}^2$ — сумма дисперсий работ, лежащих на критическом пути.

Для определения вероятности завершения комплекса работ в заданный директивный срок используют значения функции распределения $P(x)$.

Расчет вероятности выполнения производственного процесса в заданный срок оказывается тем более необходимым, когда установленный руководством директивный срок $T_{дир}$ оказывается меньше наиболее раннего срока наступления конечного, завершающего события $T_{кр}$. В этом случае приходится принять $T_{кп}$, то есть наиболее поздний срок наступления конечного, завершающего события, равным директивному сроку. Резервы времени событий, лежащих на критическом пути, оказываются отрицательными, что означает, что события не могут свершаться в заданные сроки без сокращения продолжительности критических работ.

Основными параметрами сетевого графика, кроме критического пути, являются резервы времени свершения событий и различные разновидности резервов времени работ. Резервы времени как событий, так и работ должны внимательно анализироваться руководителями. Наличие резервов времени работ позволяет маневрировать сроками начала и окончания работы, ее продолжительностью [5,7].

2.4.2. Пример решения

<u>Исходные данные</u>

Рассмотрим на конкретном примере наиболее распространенные методы представления и расчета одного и того же процесса. В качестве примера возьмем простой строительный процесс – процесс установки опоры контактной сети [5,7].

Пусть необходимо установить опору контактной сети на фундамент. Известен комплекс работ, который необходимо выполнить, длительность выполнения каждой, а также последовательность их выполнения (таблица 2.10). Требуется построить календарный и сетевой графики процесса установки опоры КС, определить минимальное время установки опоры КС, время начала и окончания каждой работы, резервы времени, определить работы, лежащие на критическом пути, который характеризует длительность установки опоры КС на фундамент.

Таблица 2.10 – Исходные данные

№	Наименование работ строительного процесса	Время выполнения работ	Предшествующие работы
1.	Заказ фундаментного блока	1	-
2.	Изготовление блока	14	1
3.	Доставка блока на место монтажа	1	2
4.	Земляные работы	2	
5.	Устройство опалубки	3	4
6.	Бетонирование	1	5
7.	Твердение бетона	8	6
8.	Установка фундаментного блока	2	7, 3
9.	Изготовление опоры КС	10	-
10.	Доставка опоры КС на место	1	9
11.	Установка опоры КС	2	8, 10

В рассматриваемом примере в первый день может быть начато выполнение работы **Заказ фундаментного блока** продолжительностью 1 день. Затем, после ее выполнения может быть начата работа **Изготовление блока** продолжительностью 14 дней. И, наконец, после изготовления блока может быть осуществлена работа **Доставка блока на место монтажа** продолжительностью 1 день.

Одновременно с выполнением работ связанных с заказом, изготовлением и доставкой блока, могут выполняться работы связанные с подготовкой фундамента под блок и изготовлением и доставкой опоры КС. Так в первый день может быть начато выполнение работы **Земляные работы** продолжительностью 2 дня. Затем, после ее выполнения, может быть начата работа **Устройство опалубки** продолжительностью 3 дня. После монтажа опалубки может быть начата работа **Бетонирование** продолжительностью 1 день. И, наконец, после бетонирования может быть осуществлена работа **Твердение бетона** продолжительностью 8 дней.

Согласно логике выполнения работ в первый день может быть начато и выполнение работы **Изготовление опоры КС** продолжительностью 10 дней. Затем может быть начата работа **Доставка опоры на место** продолжительностью 1 день.

После доставки блока и подготовки фундамента может быть выполнена работа **Установка фундаментного блока** продолжительностью 2 дня. При этом работы, связанные с доставкой блока, занимают в сумме 1 + 14 + 1 = 16 дней, а работы связанные с подготовкой фундамента занимают в сумме 2 + 3 + 1 + 8 = 14 дней. Следовательно, работа, связанная с установкой блока, не может быть начата ранее, чем через 16 дней. После же установки блока и выполнения работ, связанных с доставкой опоры КС, может быть начата работа **Установка опоры КС**.

При этом работы, связанные с установкой блока, занимают в сумме 16 + 2 = 18 дней, а работы связанные с изготовлением и доставкой опоры КС занимают в сумме 10 + 1 = 15 дней. Следовательно, работа, связанная с установкой опоры КС, не может быть начата ранее, чем через 18 дней. Таким образом, весь процесс закончится не ранее, чем через 18 + 2 = 20 дней. Что и отражено на соответствующем графике Ганта, показанном на рисунке 2.9.

В верхней, правой части диаграммы Ганта располагается шкала времени. Длина отрезка и его расположение на шкале времени определяют время начала и окончания каждой задачи. Кроме того, взаимное расположение отрезков задач показывает, следуют ли задачи одна за другой или происходит их параллельное выполнение.

Рисунок 2.9 – График Ганта выполнения процесса «Установка опоры контактной сети»

На рисунке 2.10 представлен строительный процесс установки опоры контактной сети в виде сетевого графика.

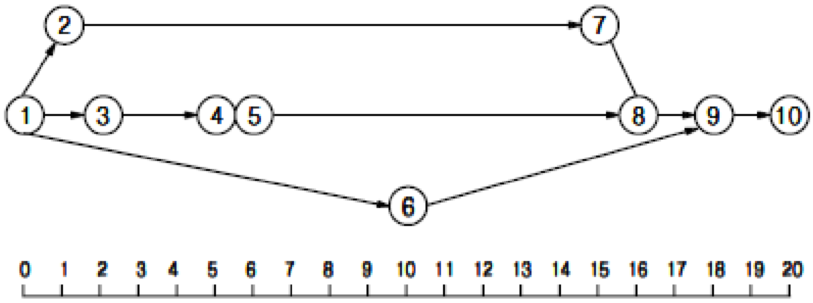


Рисунок 2.10 – Сетевой график процесса установки опоры контактной сети

Ниже, в таблице 2.11 дается расшифровка работ, представленного сетевого графика.

Таблица 2.11 - Расшифровка работ, представленного сетевого графика

Номер работы (i–j)	**Наименование работ производственного процесса**	**Длительность работ $t_{i–j}$**	**Предшествующие работы**
1–2	Заказ блока	1	
2–7	Изготовление блока	1	1–2
7–8	Доставка блока	1	2–7
1–3	Земляные работы	2	
3–4	Монтаж опалубки	3	1–3
4–5	Бетонирование	1	3–4
5–8	Твердение бетона	8	4–5
8–9	Установка блока	2	7–8, 5–8
1–6	Изготовление опоры КС	10	
6–9	Доставка опоры КС	1	1–6
9–10	Установка опоры КС	2	8–9, 6–9

<u>Процедура вычислений.</u> *Расчет сетевого графика методом CPM (МКП).* Алгоритм расчета сетевого графика методом МКП включает следующие основные этапы:

1. *Определение раннего срока свершения начального (исходного) события* T_1^p.

Примем для рассматриваемой задачи $T_1^p = 0$.

2. *Расчет ранних сроков окончания работ* t_{ij}^{po}.

Расчет проводится от начала сетевого графика до конца. В рассматриваемой задаче работы (1–2), (1–3) и (1–6) выходят из события 1, для которого известен ранний срок свершения события.

Следовательно, ранние сроки окончания этих работ составят

$$t_{12}^{po} = T_1^p + t_{12} = 0 + 1 = 1,$$

$$t_{13}^{po} = T_1^p + t_{13} = 0 + 2 = 2,$$

$$t_{16}^{po} = T_1^p + t_{16} = 0 + 10 = 10.$$

3. *Расчет раннего срока свершения события* T_j^p.

В рассматриваемой задаче в события 2, 3 и 6 входят по одной работе. Следовательно, ранние сроки свершения события составят

$$T_2^p = t_{12}^{po} = 1,$$

$$T_3^p = t_{13}^{po} = 2,$$

$$T_6^p = t_{16}^{po} = 10.$$

4. *Расчет позднего срока свершения завершающего события* T_k^n.

В рассматриваемой задаче директивный срок завершения комплекса работ не задан. Следовательно, поздний срок свершения завершающего события равен раннему сроку свершения завершающего (конечного) события

$$T_k^n = T_{10}^p = 20.$$

5. *Расчет поздних сроков начала работ* $t_{ij}^{пн}$.

В рассматриваемой задаче работа (9–10) входят в событие 10, для которого известен поздний срок свершения события. Следовательно, поздний срок начала этой работы составит

$$t_{910}^{пн} = T_{10}^n - t_{910} = 20 - 2 = 18.$$

6. *Расчет позднего срока свершения события* T_i^n.

В рассматриваемой задаче на первой итерации (цикле) из события 9 выходит одна работа (9–10) с рассчитанным поздним сроком начала работы. Следовательно, поздний срок свершения 9 события будет равен

$$T_9^n = t_{910}^{пн} = 18.$$

Далее выполняются расчеты (этапы 7-10) в соответствии с алгоритмом, изложенным в пункте 2.4.1.

Проведем расчет сетевого графика согласно вышеизложенному алгоритму и все результаты расчета зафиксируем на самом сетевом графике (рисунок 2.11).

Анализируя расчетные данные, представленные на самом сетевом графике, видим, что работы 1–2, 2–7, 7–8, 8–9 и 9–10 имеют нулевой полный резерв времени работы – r_{ij}^n. Следовательно, эти работы лежат на критическом пути и, именно этот путь, определяет суммарную длительность выполнения всего процесса (проекта). Любой из возможных сбоев при выполнении работ, лежащих на критическом пути, вызовет соответствующее увеличение срока выполнения всего процесса. Поскольку в нашем примере директивный срок выполнения процесса не задан, то полные резервы времени работ, лежащих на критическом пути, равны 0.

Расчет детерминированного сетевого графика при большом числе работ достаточно трудоемкая операция. Для облегчения таких расчетов можно использовать соответствующее программное обеспечение. Ниже приводится документ *Mathcad* с примером расчета сетевого графика методом *CPM* с применением программного блока (рисунок 2.12).

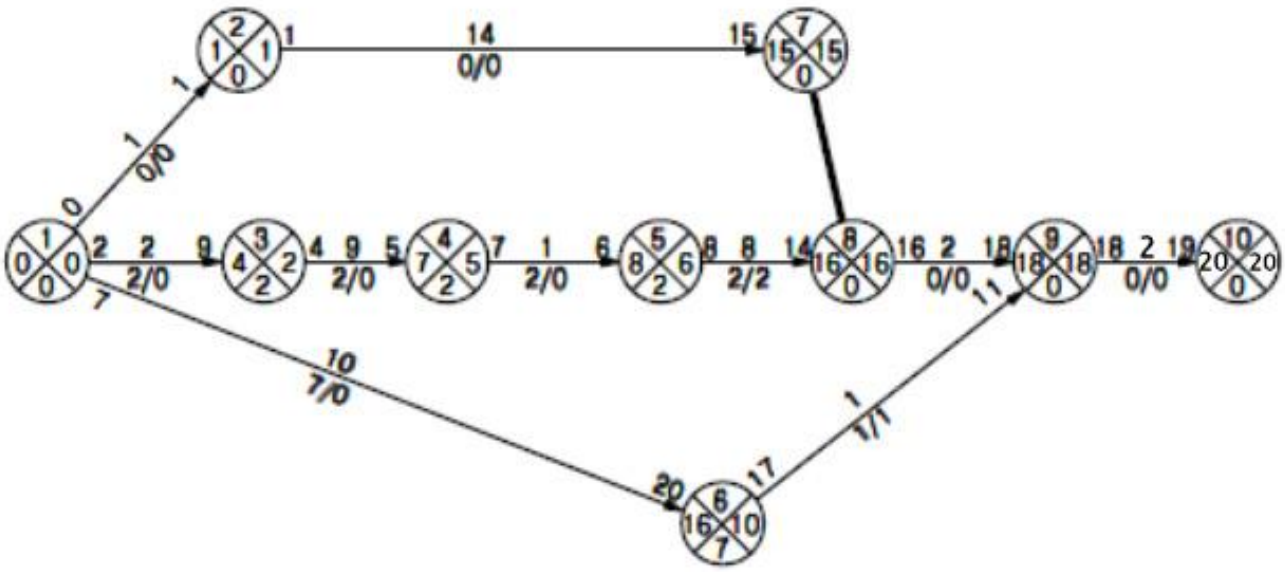


Рисунок 2.11 – Результаты расчета сетевого графика

Расчет сетевого графика методом CPM

1) Исходные данные:

$M := 10$ - число событий в сети $N := 11$ - число работ в сети

$$II := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix}$$ - номер события из которого выходит работа

$$JJ := \begin{pmatrix} 2 \\ 3 \\ 6 \\ 7 \\ 4 \\ 5 \\ 8 \\ 9 \\ 8 \\ 9 \\ 10 \end{pmatrix}$$ - номер события в который входит работа

$$T := \begin{pmatrix} 1 \\ 2 \\ 10 \\ 14 \\ 3 \\ 1 \\ 8 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix}$$ - продолжительность выполнения I-ой работы

Рисунок 2.12 – Расчет сетевого графика методом *CPM* в *Mathcad* с применением программного блока (*начало*)

2) Блок решения

```
ORIGIN := 1

SM(M,N,II,JJ,T) := | for I ∈ 1..N
                   |   L_I ← 0
                   | TP_1 ← 0
                   | for J ∈ 2..M
                   |   | TMAX ← 0
                   |   | for K ∈ 1..N
                   |   |   if JJ_K = J
                   |   |     | TP_J ← TP_(II_K) + T_K
                   |   |     | TMAX ← TP_J  if TP_J > TMAX
                   |   | TP_J ← TMAX
                   | TП_M ← TP_M
                   | for IP ∈ 2..M
                   |   | I ← M − IP + 1
                   |   | TMIN ← 100000
                   |   | for KF ∈ 1..N
                   |   |   | K ← N − KF + 1
                   |   |   | if II_K = I
                   |   |   |   | TП_I ← TП_(JJ_K) − T_K
                   |   |   |   | TMIN ← TП_I  if TП_I < TMIN
                   |   | TП_I ← TMIN
                   | for J ∈ 1..M
                   |   R_J ← TП_J − TP_J
                   | for K ∈ 1..N
                   |   | I ← II_K
                   |   | J ← JJ_K
                   |   | TPO_K ← TP_I + T_K
                   |   | TПH_K ← TП_J − T_K
                   |   | RП_K ← TП_J − TP_I − T_K
                   |   | RC_K ← TP_J − TP_I − T_K
                   |   | L_K ← 1  if |R_I − R_J| ≤ 0.0001 ∨ |TПH_K − TП_I| ≤ 0.0001  if |R_J − R_M| ≤ 0.0001  if |R_I − R_M| ≤ 0.0001
                   | KP_1 ← 1
                   | KK ← 1
                   | for K ∈ 1..N
                   |   | I ← II_K
                   |   | J ← JJ_K
                   |   | if I = KP_KK      if L_K ≠ 0
                   |   |   | KK ← KK + 1
                   |   |   | KP_KK ← J
                   | ( augment(II,JJ,T,RП,RC,TПH,TPO) )
                   | (      augment(TP,TП,R)          )
                   | (            KP^T                )
```

Рисунок 2.12 – Расчет сетевого графика методом *CPM* в *Mathcad* с применением программного блока (*продолжение*)

$G := SM(M, N, II, JJ, T)$

$G_1 =$

	i	j	$t_{i\text{-}j}$	$r_{i\text{-}j}^{п}$	$r_{i\text{-}j}^{с}$	$t_{i\text{-}j}^{пн}$	$t_{i\text{-}j}^{ро}$
	1	2	3	4	5	6	7
1	1	2	1	0	0	0	1
2	1	3	2	2	0	2	2
3	1	6	10	7	0	7	10
4	2	7	14	0	0	1	15
5	3	4	3	2	0	4	5
6	4	5	1	2	0	7	6
7	5	8	8	2	2	8	14
8	6	9	1	7	7	17	11
9	7	8	1	0	0	15	16
10	8	9	2	0	0	16	18
11	9	10	2	0	0	18	20

i - работа, выходящая из i-го события

j - работа, входящая в j-ое событие

$t_{i\text{-}j}$ - продолжительность выполнения (i-j)-ой работы

$r_{i\text{-}j}^{п}$ - полный резерв времени работы

$r_{i\text{-}j}^{с}$ - свободный резерв времени работы

$t_{i\text{-}j}^{пн}$ - поздний срок начала работы

$t_{i\text{-}j}^{ро}$ - ранний срок окончания работы

Рисунок 2.12 – Расчет сетевого графика методом *CPM* в *Mathcad* с применением программного блока (*продолжение*) событие $T_j^р$ $T_i^п$ R_j

$G_2 =$

	1	2	3
1	0	0	0
2	1	1	0
3	2	4	2
4	5	7	2
5	6	8	2
6	10	17	7
7	15	15	0
8	16	16	0
9	18	18	0
10	20	20	0

$T_j^р$ - ранний срок наступления j-го события

$T_i^п$ - поздний срок наступления j-го события

R_j - полный резерв времени j-го события

$\boxed{G_3 = (1\ \ 2\ \ 7\ \ 8\ \ 9\ \ 10)}$ - события, лежащие на критическом пути

Рисунок 2.12 – Расчет сетевого графика методом *CPM* в *Mathcad* с применением программного блока (*окончание*)

Расчет сетевого графика методом PERT. Допустим, что нам известны временные оценки продолжительности выполнения каждой работы (операции) (таблица 2.12).

Таблица 2.12 - Временные оценки продолжительности выполнения работ (операций)

Номер работы (i–j)	**Наименование работ производственного процесса**	t_{minij}	$t_{нвij}$	t_{maxij}
1–2	Заказ фундаментного блока	1	1	1
2–7	Изготовление блока	12	14	17
7–8	Доставка блока на место монтажа	1	1	1
1–3	Земляные работы	1	2	4
3–4	Устройство опалубки	2	3	5
4–5	Бетонирование	1	1	2
5–8	Твердение бетона	8	8	8
8–9	Установка фундаментного блока	1	2	4
1–6	Изготовление опоры КС	8	10	13
6–9	Доставка опоры КС на место	1	1	1
9–10	Установка опоры КС	1	2	4

Результаты расчета по формулам (2.4.12 – 2.4.14) представлены в таблице 2.13.

Таблица 2.13 - Результаты расчета временных параметров сетевого графика

Номер работы (i–j)	t_{minij}	$t_{нвij}$	t_{maxij}	$t_{расчij}$	σ^2
1–2	1	1	1	1	0
2–7	12	14	17	14,666	0,694
7–8	1	1	1	1	1
1–3	1	2	4	2,166	–
3–4	2	3	5	3,166	–
4–5	1	1	2	1,166	–
5–8	8	8	8	8	–
8–9	1	2	4	2,166	0,25
1–6	8	10	13	10,166	–
6–9	1	1	1	1	–
9–10	1	2	4	2,166	0,25

В рассматриваемой задаче определение дисперсий производится по формуле (2.4.16).

Вычисление дисперсий производится для работ 1–2, 2–7, 7–8, 8–9 и 9–10, которые лежат на критическом пути. Результаты расчета представлены в таблице 2.13.

Для определения вероятности завершения комплекса работ в заданный директивный срок необходимо определить аргумент функции нормального распределения x

$$x = \frac{T_{дир} - T_x^p}{\sqrt{\sum \sigma^2_{кр\,ij}}} = \frac{22,0 - 20,0}{\sqrt{\sum (0 + 0,694 + 1,0 + 0,25 + 0,25)}} = 1,35 \cdot$$

Для определения вероятности завершения комплекса работ в заданный директивный срок используют значения функции распределения $P(x)$. Результаты расчета представлены в таблице 2.14.

Таблица 2.14 - Результаты расчета вероятностей завершения комплекса работ в заданный директивный срок

x	**P(x)**	**x**	**P(x)**	**x**	**P(x)**	**x**	**P(x)**
0,0	0,500	1,6	0,945	–3,0	0,0013	–1,4	0,0808
0,2	0,579	1,8	0,964	–2,8	0,0026	–1,2	0,1151
0,4	0,656	2,0	0,977	–2,6	0,0047	–1,0	0,1587
0,6	0,726	2,2	0,986	–2,4	0,0082	–0,8	0,2119
0,8	0,788	2,4	0,992	–2,2	0,0139	–0,6	0,2743
1,0	0,841	2,6	0,995	–2,0	0,0228	–0,4	0,3446
1,2	0,885	2,8	0,997	–1,8	0,0352	–0,2	0,4207
1,4	0,919	3,0	0,999	–1,6	0,0548	–0,0	0,5000

Используя таблицу значений функции нормального распределения и метод интерполяции, определяем вероятность выполнения производственного процесса в заданный директивный срок

$$P(x) = 0,885 + (1,35 - 1,2) \cdot \frac{0,919 - 0,885}{1,4 - 1,2} = 0,885 + 0,15 \cdot 0,17 = 0,91 \cdot$$

Расчет сетевого графика при большом числе работ достаточно трудоемкая операция. Для облегчения таких расчетов можно использовать соответствующее программное обеспечение. Ниже приводится документ *Mathcad* с примером расчета сетевого графика методом *PERT* с применением программного блока (рисунок 2.13).

Расчет сетевого графика методом PERT

1) Исходные данные:

$M := 10$ - число событий в сети $N := 11$ - число работ в сети

$TD := 22$ - директивный срок

$$II := \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix} \quad JJ := \begin{pmatrix} 2 \\ 3 \\ 6 \\ 7 \\ 4 \\ 5 \\ 8 \\ 9 \\ 8 \\ 9 \\ 10 \end{pmatrix} \quad T1 := \begin{pmatrix} 1 \\ 1 \\ 8 \\ 12 \\ 2 \\ 1 \\ 8 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \quad T2 := \begin{pmatrix} 1 \\ 2 \\ 10 \\ 14 \\ 3 \\ 1 \\ 8 \\ 1 \\ 1 \\ 2 \\ 2 \end{pmatrix} \quad T3 := \begin{pmatrix} 1 \\ 4 \\ 13 \\ 17 \\ 5 \\ 1 \\ 8 \\ 1 \\ 1 \\ 4 \\ 4 \end{pmatrix}$$

Рисунок 2.13 – Расчет сетевого графика методом *PERT* в *Mathcad* с применением программного блока (*начало*)

2) Блок решения

```
ORIGIN := 1

PERT(II, JJ, T1, T2, T3) := | for I ∈ 1..N
                            |   T_I ← (T1_I + 4·T2_I + T3_I) / 6
                            |   D_I ← ((T3_I − T1_I) / 6)^2
                            |   L_I ← 0
                            | TP_1 ← 0
                            | S_1 ← 0
                            | for J ∈ 2..M
                            |   TMAX ← 0
                            |   SMAX ← 0
                            |   for K ∈ 1..N
                            |     if JJ_K = J
                            |       TP_J ← TP_(II_K) + T_K
                            |       S_J ← S_(II_K) + D_K
                            |       if TP_J > TMAX
                            |         TMAX ← TP_J
                            |         SMAX ← S_J
                            |   TP_J ← TMAX
                            |   S_J ← SMAX
                            | TII_M ← TP_M
                            | for IF ∈ 2..M
                            |   I ← M − IF + 1
                            |   TMIN ← 100000
                            |   for KF ∈ 1..N
                            |     K ← N − KF + 1
                            |     if II_K = I
                            |       TII_I ← TII_(JJ_K) − T_K
                            |       TMIN ← TII_I  if  TII_I < TMIN
                            |   TII_I ← TMIN
                            | for J ∈ 1..M
                            |   R_J ← TII_J − TP_J
                            | for K ∈ 1..N
                            |   I ← II_K
                            |   J ← JJ_K
                            |   TPO_K ← TP_I + T_K
                            |   TIIH_K ← TII_J − T_K
                            |   RII_K ← TII_J − TP_I − T_K
                            |   RC_K ← TP_J − TP_I − T_K
                            |   L_K ← 1  if  |R_I − R_J| ≤ 0.0001 ∨ |TIIH_K − TII_I| ≤ 0.0001  if  |R_J − R_M| ≤ 0.0001  if  |R_I − R_M| ≤ 0.0001
                            | KP_1 ← 1
                            | KK ← 1
                            | for K ∈ 1..N
                            |   I ← II_K
                            |   J ← JJ_K
                            |   if I = KP_KK      if L_K ≠ 0
                            |     KK ← KK + 1
                            |     KP_KK ← J
                            | X ← (TD − TP_M) / √(S_M)
                            | ( augment(II, JJ, T, RII, RC, TIIH, TPO) )
                            | ( augment(TII, TP, R)                   )
                            | ( KP^T                                  )
                            | ( X                                     )
                            | ( TD                                    )
                            | ( TP_M                                  )
```

Рисунок 2.13 – Расчет сетевого графика методом *PERT* в *Mathcad* с применением программного блока (*продолжение)*

$P := PERT(II, JJ, T1, T2, T3)$

$P_1 =$

	i	j	$t_{i\text{-}j}$	$r_{i\text{-}j}^{п}$	$r_{i\text{-}j}^{с}$	$t_{i\text{-}j}^{пн}$	$t_{i\text{-}j}^{ро}$
	1	2	3	4	5	6	7
1	1	2	1	0	0	0	1
2	1	3	2.2	1.8	0	1.8	2.2
3	1	6	10.2	7.2	0	7.2	10.2
4	2	7	14.2	0	0	1	15.2
5	3	4	3.2	1.8	0	4	5.3
6	4	5	1	1.8	0	7.2	6.3
7	5	8	8	1.8	1.8	8.2	14.3
8	6	9	1	7.2	7.2	17.3	11.2
9	7	8	1	0	0	15.2	16.2
10	8	9	2.2	0	0	16.2	18.3
11	9	10	2.2	0	0	18.3	20.5

i - работа, выходящая из i-го события

j - работа, входящая в j-ое событие

$t_{i\text{-}j}$ - продолжительность выполнения (i-j)-ой работы

$r_{i\text{-}j}^{п}$ - полный резерв времени работы

$r_{i\text{-}j}^{с}$ - свободный резерв времени работы

$t_{i\text{-}j}^{пн}$ - поздний срок начала работы

$t_{i\text{-}j}^{ро}$ - ранний срок окончания работы

Рисунок 2.13 – Расчет сетевого графика методом *PERT* в *Mathcad* с применением программного блока (*продолжение)*

$P_2 =$

событие	$T_j^р$	$T_i^п$	R_j
	1	2	3
1	0	0	0
2	1	1	0
3	4	2.2	1.8
4	7.2	5.3	1.8
5	8.2	6.3	1.8
6	17.3	10.2	7.2
7	15.2	15.2	0
8	16.2	16.2	0
9	18.3	18.3	0
10	20.5	20.5	0

$T_j^р$ - ранний срок наступления j-го события

$T_i^п$ - поздний срок наступления j-го события

R_j - полный резерв времени j-го события

$P_3 = (1\ \ 2\ \ 7\ \ 8\ \ 9\ \ 10)$ - события, лежащие на критическом пути

Рисунок 2.13 – Расчет сетевого графика методом *PERT* в *Mathcad* с применением программного блока (*продолжение)*

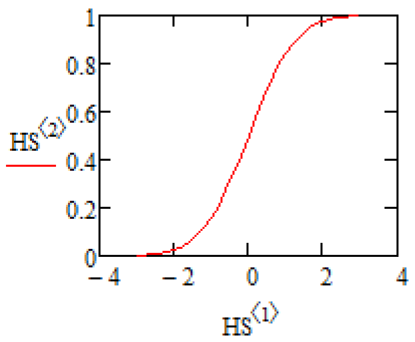
$P_5 = 22.00$ - директивный срок завершения работ

$P_6 = 20.50$ - ожидаемый срок завершения работ

$P_4 = 1.37$ - аргумент нормальной функции распределения

$HS := \mathrm{csort}(HH, 1)$

$LL(x) := \mathrm{linterp}\left(HS^{\langle 1 \rangle}, HS^{\langle 2 \rangle}, x\right)$



$LL(P_4) = 0.91$ - вероятность выполнения технологического процесса в заданный директивный срок

Рисунок 2.13 – Расчет сетевого графика методом *PERT* в *Mathcad* с применением программного блока (*окончание)*

Литература

1. Ивановский Р. И. Компьютерные технологии в науке и образовании. Практика применения систем Mathcad Pro: Учеб. пособие / Р. И. Ивановский. – М.: Высш. шк., 2003. – 431 с.: ил.

2. Кудрявцев Е. М. Mathcad 11: Полное руководство по русской версии. – М.: ДМК Пресс, 2005. – 592 с.: ил.

3. Охорзин В. А. Компьютерное моделирование в системе Mathcad: учеб. пособие. – М.: Финансы и статистика, 2006. – 144 с.: ил.

4. Семененко М. Г. Математическое моделирование в Mathcad. – М.: Альтекс-А, 2003. – 208 с.

5. Автоматизированное проектирование организации строительства железных дорог / Под ред. С.П. Першина. – М.: Транспорт, 1991. – 261 с.

6. Баркалов С. А., Бабкин В. Ф. Управление проектами в строительстве. Лабораторный практикум: Учеб. пособие. – М.: Издательство АСВ, 2003. – 288 с.: ил.

7. Кудрявцев Е. М. Microsoft Project. Методы сетевого планирования и управления проектом. – М.: ДМК Пресс, 2005. – 240 с.

Полянский Алексей Викторович

Автоматизированное решение задач проектного управления железнодорожным строительством

Учебное пособие
для студентов специальности 23.05.06 «Строительство железных дорог, мостов и транспортных тоннелей»

Учебное пособие издано в авторской редакции

Сетевое издание

Главный редактор – Кирсанов К.А.

Ответственный за выпуск – Алимова Н.К.

Учебное издание

Системные требования:
операционная система Windows XP или новее, macOS 10.12 или новее, Linux.
Программное обеспечение для чтения файлов PDF.

Объем данных 2 Мб

Принято к публикации «23» апреля 2021 года

Режим доступа: https://izd-mn.com/PDF/21MNNPU21.pdf свободный. – Загл. с экрана. – Яз. рус., англ.

ООО «Издательство «Мир науки»

«Publishing company «World of science», LLC

Адрес:

Юридический адрес — 127055, г. Москва, пер. Порядковый, д. 21, офис 401.

Почтовый адрес — 127055, г. Москва, пер. Порядковый, д. 21, офис 401.

https://izd-mn.com/

ДАННОЕ ИЗДАНИЕ ПРЕДНАЗНАЧЕНО ИСКЛЮЧИТЕЛЬНО ДЛЯ ПУБЛИКАЦИИ НА ЭЛЕКТРОННЫХ НОСИТЕЛЯХ