Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»

Н.М. Ефромеев, Е.В. Ефромеева

Методы исследования операций в машиностроении

Рекомендовано ФГБОУ ВО «МГТУ «СТАНКИН» кафедрой Информационных технологий и вычислительных систем в качестве учебного пособия по направлению подготовки 09.03.01 «Информатика и вычислительная техника». Протокол заседания кафедры № 3 от 15.11.2017 г.

Москва
2018

УДК 519.852(075)
ББК 22.18
Е 92

Рецензенты: д-р. пед. наук, проф. В.К. Жаров (заведующий кафедрой фундаментальной и прикладной математики ФГБОУ ВО «Российский государственный гуманитарный университет» (РГГУ));

канд. соц. наук, доц. И.И. Грунтовский (заведующий кафедрой общегуманитарных и научных дисциплин Международного юридического института).

Ефромеев Н.М., Ефромеева Е.В.

Е 92 Методы исследования операций в машиностроении: учебное пособие. – 2-е издание, перераб. и доп. / Н.М. Ефромеев, Е.В. Ефромеева. – М.: ФГБОУ ВО «МГТУ «Станкин», 2018. – 155 с.
ISBN 978-5-7028-0556-6

В пособии изложены основные возможности практического применения методов исследования операций, когда целевая функция и все ограничения являются линейными функциями.

Учебное пособие предназначено для бакалавров высших учебных заведений, обучающихся по направлению 09.03.01 «Информатика и вычислительная техника».

Пособие может быть полезно студентам технических вузов, а также студентам экономических специальностей и практическим работникам, занимающимся анализом текущего финансово-экономического состояния и будущего развития фирм, предприятий, отраслей и регионов.

УДК 519.852(075)
ББК 22.18

ISBN 978-5-7028-0556-6

СОДЕРЖАНИЕ

Введение

Исследование операций направлено на решение практических задач, которые можно описать с помощью математических моделей. Сегодня теория исследования операций является основным и неотъемлемым инструментом при принятии решений в самых разнообразных областях машиностроительного производства. А от уровня развития машиностроения зависят материалоемкость, энергоемкость валового внутреннего продукта, производительность труда, промышленная безопасность и обороноспособность государства.

Машиностроительный комплекс включает более двадцати подотраслей металлообрабатывающей промышленности, производящих средства производства, транспорта, оборонную продукцию, а также предметы потребления. Многие интересные и важные виды деятельности в машиностроении можно трактовать как процессы исследования операций, когда целевая функция и все ограничения являются линейными функциями.

В пособии представлены наиболее известные и эффективные методы линейного математического программирования, часто использующиеся в задачах машиностроительного профиля.

Для машиностроения это задачи размещения производства, транспортные задачи, задачи назначения, маршрутизации, распределительные задачи.

1. Интегрированная информационная технология общего назначения: электронный офис

1.1. Знакомство с Microsoft Excel 2016

1.1.1. Общие сведения

Microsoft Office Excel 2016 представляет собой мощный табличный процессор, который широко используется как рядовыми пользователями, так и специалистами узкого профиля для работы с электронными таблицами.

Электронная таблица – это информационная технология для профессиональной работы с данными, представляющая собой аналог обычной таблицы и позволяющая проводить разнообразные вычисления с числовыми данными. Электронные таблицы позволяют автоматизировать выполнение однотипных вычислений и пересчета с изменяющимися исходными данными, а также обрабатывать числовую информацию в массиве баз данных, анализировать финансы, доходы, налоги и так далее. В ячейки электронной таблицы можно вносить текст, числа, формулы. Электронные таблицы часто используются в качестве простых баз данных или как приложение для построения графиков и диаграмм. Для управления электронной таблицей созданы специальные программные продукты – табличные процессоры.

Табличный процессор – комплекс программных средств для математической, статистической и графической обработки текстовых и числовых данных в табличном виде.

Основные возможности применения *Excel 2016*:

- ✓ решение числовых задач, требующих больших вычислений (создание отчетов, анализ результатов);
- ✓ создание диаграмм;
- ✓ организация списков (создание и использование сложно структурированных таблиц);
- ✓ доступ к данным других типов (возможность импортирования данных из множества различных источников);
- ✓ создание рисунков и схем (использование фигур и объектов SmartArt);
- ✓ автоматизация сложных задач (с использованием макросов).

Приложение *Microsoft Excel 2016* открывает совершенно новые возможности анализа данных, а также управления и обмена ими, что позволяет принимать более правильные и обоснованные решения. Новые средства анализа и визуализации данных позволяют отслеживать и выделять важные тенденции. Файлы можно легко отправлять в Интернет для работы с ними вместе с другими пользователями. С данными можно легко работать в пути практически через любой браузер или смартфон. Приложение *Excel 2016* позволяет более эффективно и гибко выполнять поставленные задачи: составление финансовых отчетов или ведение личной бухгалтерии.

Преимущества *Microsoft Excel 2016:*

- ✓ Быстрое и эффективное сравнение данных
  Новые средства и возможности *Excel 2016* позволяют выявлять модели или тенденции, облегчающие принятие решений и анализ крупных наборов данных.
  - ➢ Обобщение данных с помощью спарклайнов – небольших диаграмм, которые помещаются в ячейку вместе с текстом.
  - ➢ Для быстрой фильтрации больших объемов данных можно использовать новые функции срезов, которые расширяют возможности визуального анализа сводных таблиц и диаграмм.
- ✓ Мощные возможности анализа
  Благодаря повышению производительности и улучшениям приложения *Excel 2016* работать в нем стало проще и быстрее.
  - ➢ Новый фильтр поиска позволяет быстро сузить область просмотра в таблицах, сводных таблицах и сводных диаграммах. Нужные данные среди миллионов элементов можно найти за считаные секунды.
  - ➢ Бесплатная надстройка *PowerPivot* для *Excel 2016* позволяет быстро выполнять операции с крупными наборами данных (включающими миллионы строк) и упрощает их интеграцию. Кроме того, результатами анализа можно без труда поделиться с другими пользователями с помощью сервера *SharePoint Server 2016*.
- ✓ Экономия времени, упрощение работы и повышение производительности

  Создавать книги и управлять ими гораздо проще, если можно работать так, как удобно.
  - ➢ В *Excel 2016* можно восстанавливать несохраненные версии случайно закрытых файлов (рис. 1.1). И это всего лишь одна из множества функций, доступных в представлении *Microsoft Office Backstage*™, которое предлагает все средства управления книгами.
  - ➢ Улучшенная лента теперь легко настраивается, что упрощает доступ к часто используемым командам. Можно создавать пользовательские вкладки и настраивать встроенные – приложение *Excel 2016* позволяет сделать интерфейс таким, как вам удобно.

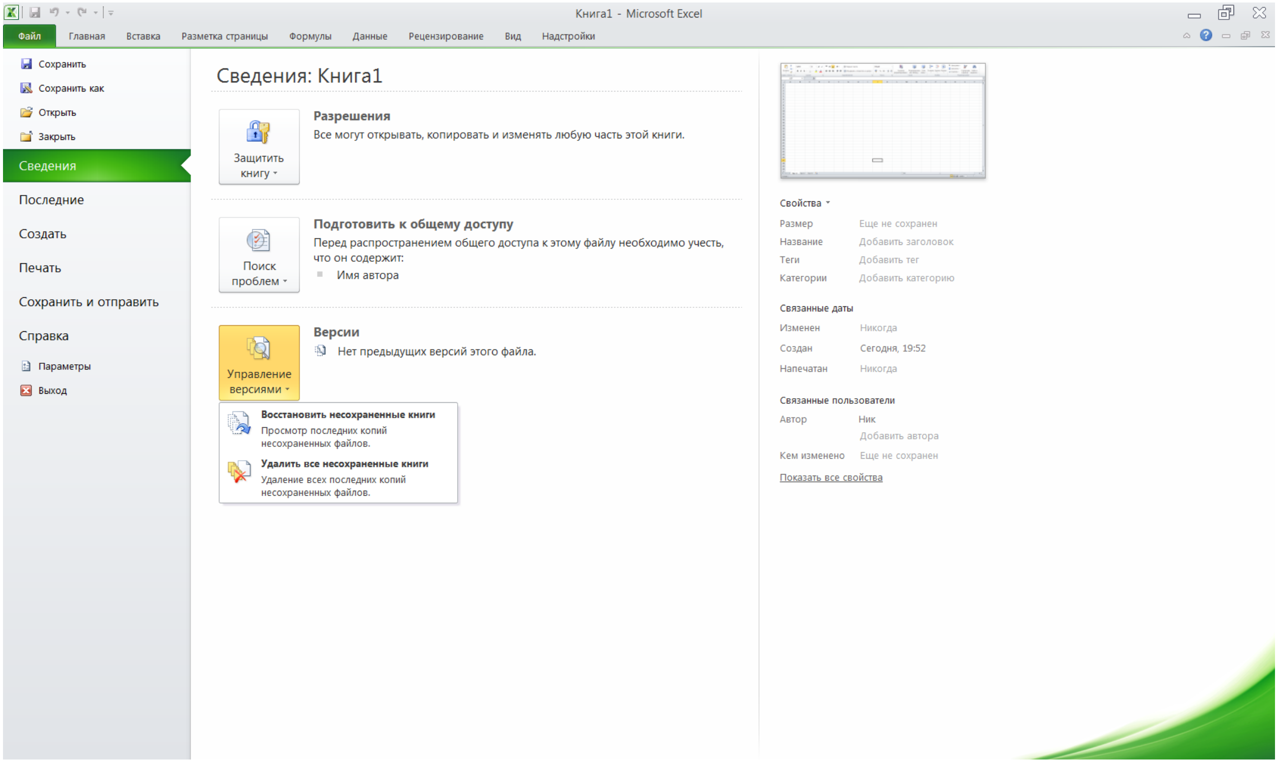


Рис. 1.1. Вкладка «Файл»

1.1.2. Интерфейс табличного процессора Microsoft Excel

В окне *Microsoft Excel 2016*, как и в других программах *MS Office 2016*, используется ленточный интерфейс (рис. 1.2).

В верхней части окна расположена лента со вкладками инструментов, панель быстрого доступа, строка заголовка. Под лентой меню расположена строка, в которой отображается название активной ячейки, а также строка ввода формул или содержимого выделенной ячейки. В нижней части окна расположена строка состояния, которая содержит вспомогательную информацию по работе с программой.

Вкладка «Файл»

Вкладка «Файл» (рис. 1.3) предназначена для вызова наиболее часто использующихся команд по работе с файлами книг: *Сведения, Создать, Открыть, Сохранить, Сохранить как, Печать, Общий доступ, Экспорт, Опубликовать, Закрыть, Учетная запись, Параметры*.

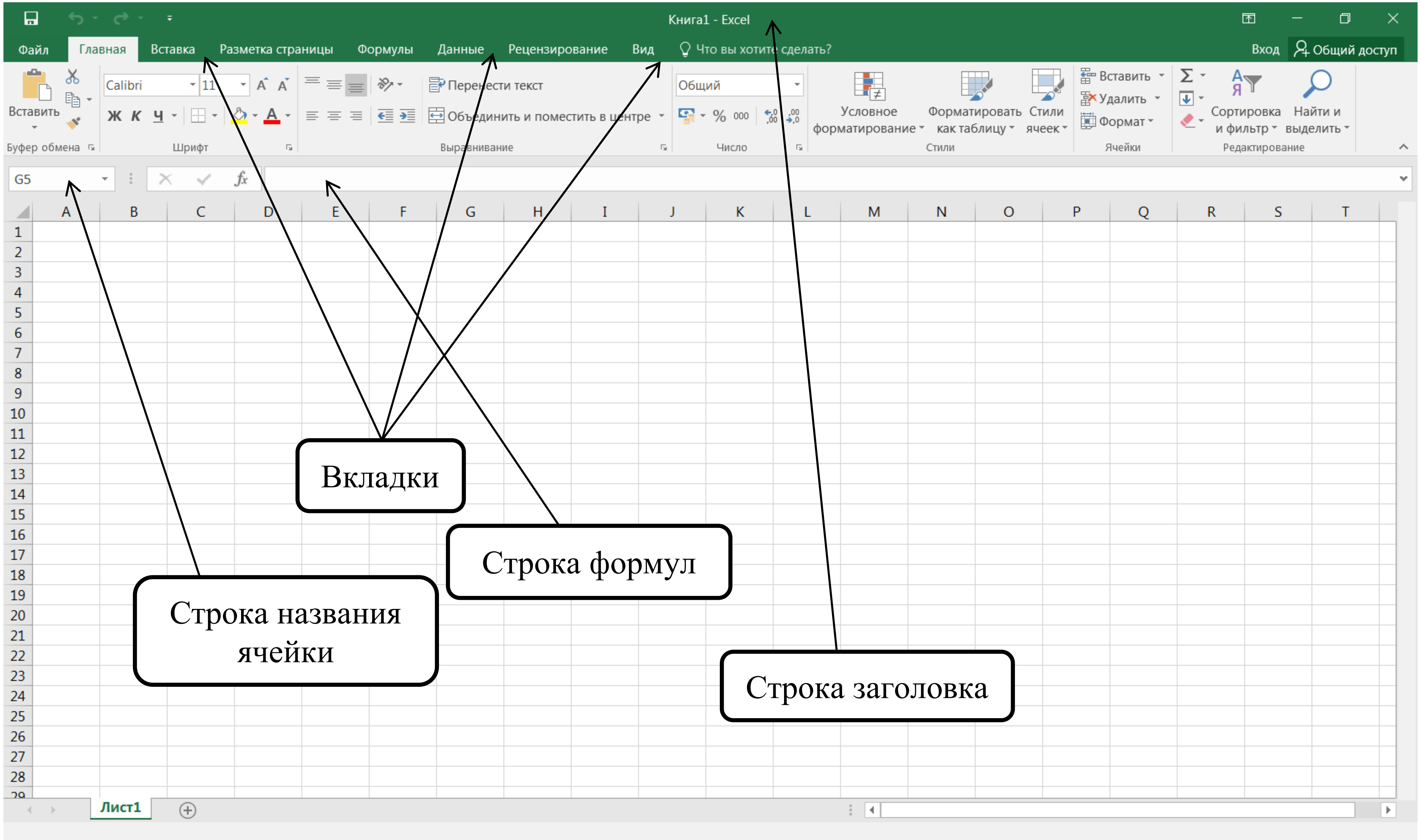


Рис.1.2. Основные элементы окна MS Excel 2016

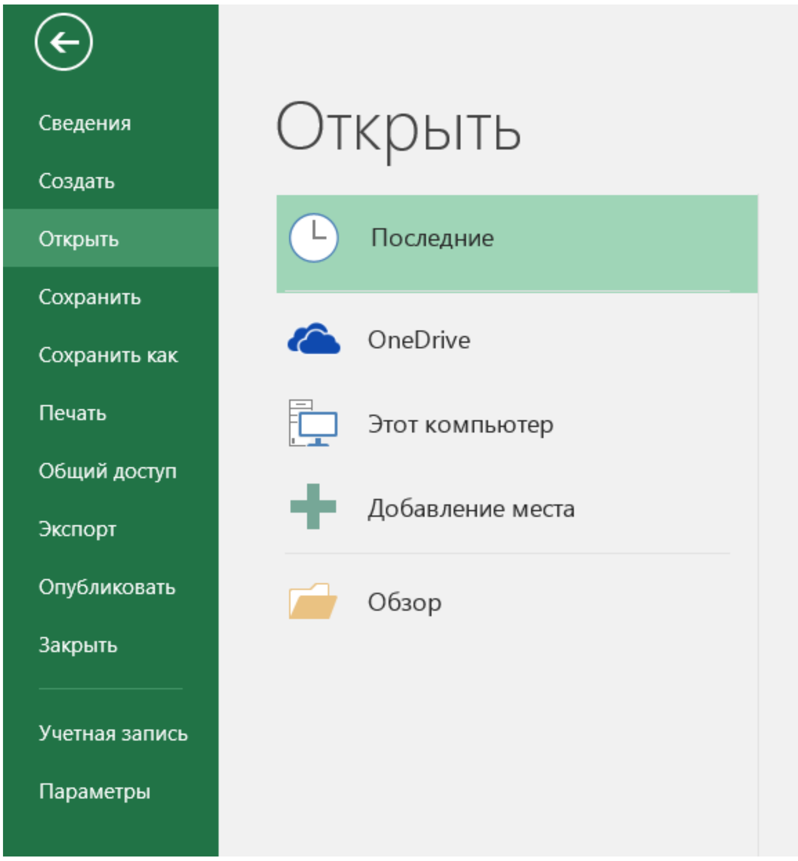


Рис. 1.3. Вкладка «Файл»

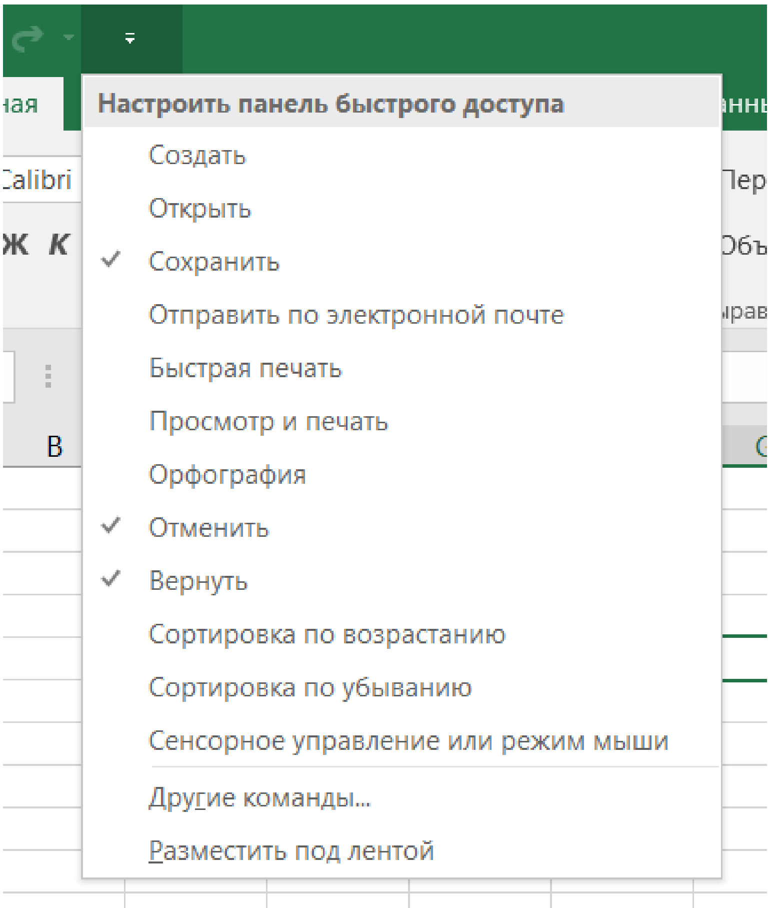


Рис. 1.4. Панель быстрого доступа

Панель быстрого доступа

На панели (рис. 1.4) размещены кнопки наиболее часто выполняемых операций. По умолчанию это: *Сохранить*, *Отменить ввод*, *Повторить ввод*. На панель могут быть добавлены кнопки из раскрывающегося по кнопке списка настройки, либо любые другие команды при использовании пункта *Другие Команды – Настройка*. Для добавления и удаления определенных команд, необходимо их выделить в левой части окна и добавить на правую, а также указать, будет ли панель иметь заданный вид при открытии всех других документов, выбрав из списка пункт *Для всех документов*, или только *для определенного документа*.

Лента главного меню

Лента главного меню включает в себя вкладки: *Главная, Вставка, Разметка страницы, Формулы, Данные, Рецензирование, Вид*. Каждая вкладка содержит группы инструментов, предназначенных для выполнения определенного класса задач. Также существуют специализированные вкладки, которые появляются в ленте меню на время работы с определенными объектами (рис. 1.5).

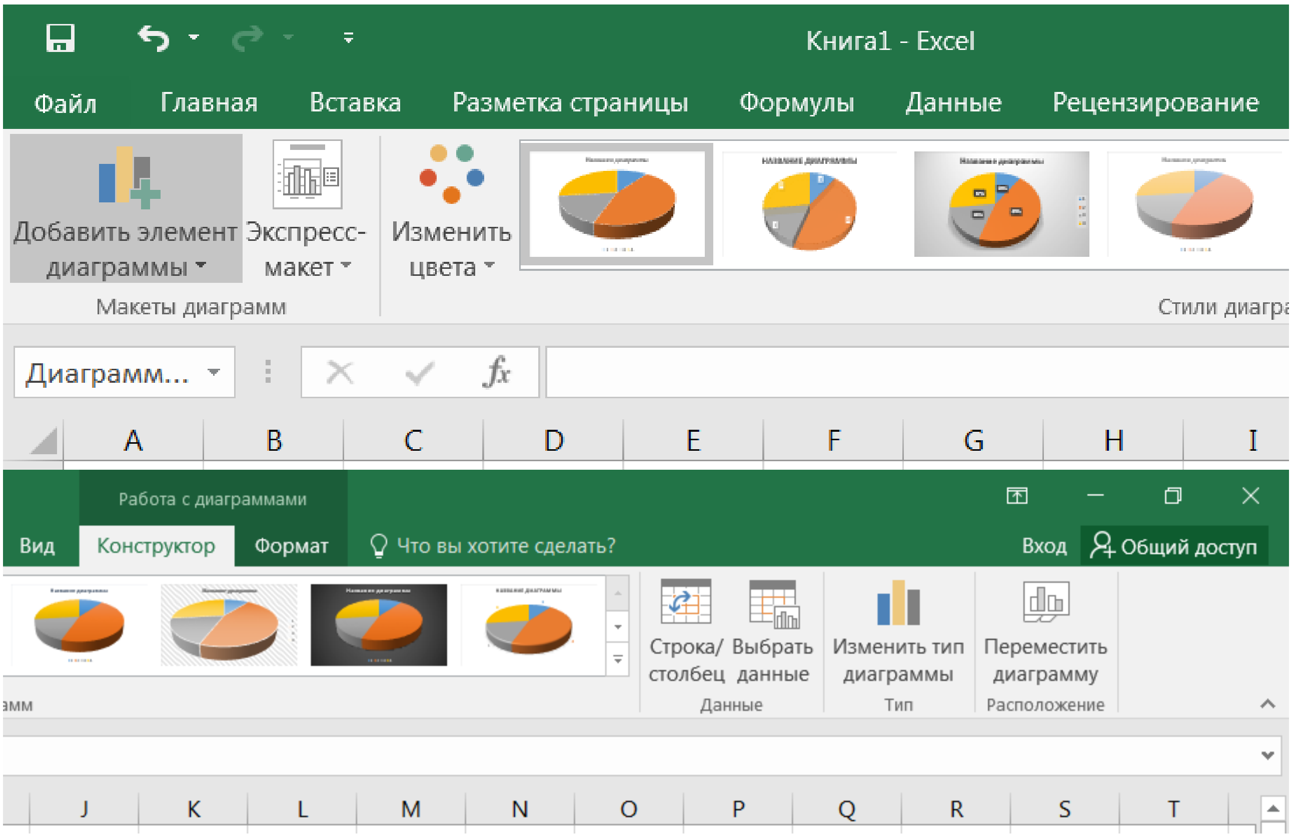


Рис. 1.5. Лента главного меню

На панелях инструментов вкладок вынесены наиболее часто используемые кнопки. Другие нужные команды группы можно вызвать, нажав на небольшую стрелку в правом нижнем углу определенной группы. Это позволит вызвать диалоговое окно (рис. 1.6), содержащее все команды данной группы. При наведении курсора на кнопки инструментов появляется всплывающая подсказка, которая информирует об их предназначении.

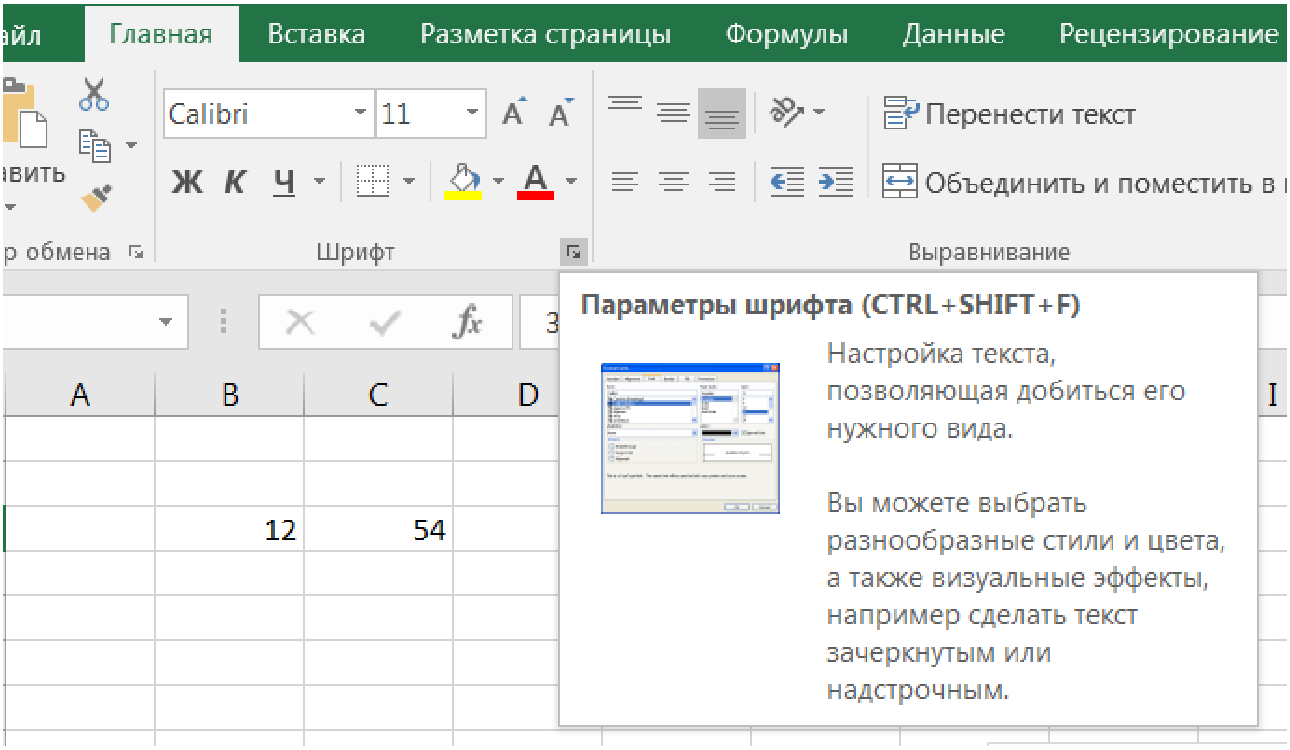


Рис. 1.6. Кнопка открытия диалогового окна Шрифт

Рассмотрим основные вкладки главного меню.

<u>Вкладка Главная</u> (рис. 1.7) состоит из следующих групп инструментов, позволяющих осуществлять базовые операции по редактированию и оформлению текста в ячейках, форматированию самих ячеек и работе с ними:

- *буфер обмена* (позволяет осуществлять копирование, вставку, специальную вставку, удаление, формат по образцу);
- *шрифт* (позволяет задавать разнообразные параметры шрифта, заливки и границ ячеек);
- *выравнивание* (позволяет устанавливать выравнивание текста в ячейках по горизонтали и вертикали, направление и перенос текста, объединение/разъединение ячеек);
- *число* (используется для задания формата отображения значений ячейки, регулирования разрядности числовых значений);
- *стили* (позволяет задавать разнообразные параметры стилей оформления ячеек, условное форматирование);

- *ячейки* (позволяет выполнять операции вставки, удаления, формата ячеек, строк, столбцов, листов, а также выставлять параметры защиты различных объектов);
- *редактирование* (предназначена для вставки функций в формулы, задания прогрессии, сортировки и фильтрации, очистки содержимого ячеек, поиска и выбора различных объектов листа).

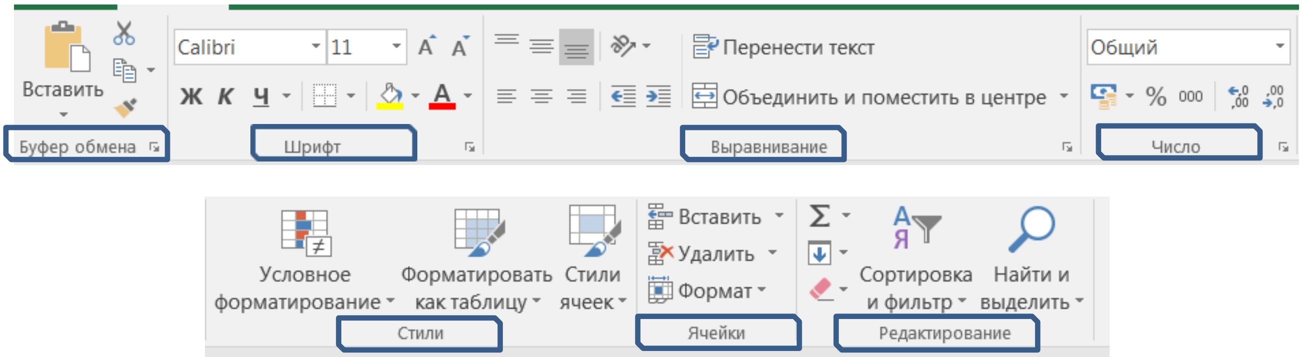


Рис. 1.7. Состав групп вкладки Главная

<u>Вкладка Вставка</u> (рис. 1.8) состоит из следующих групп, позволяющих осуществлять вставку в электронную таблицу различных элементов:

- *таблицы* (позволяет создать на листе новый объект – таблицу, вставить сводную таблицу);
- *иллюстрации* (позволяет вставлять рисунки, картинки, фигуры, объекты SmartArt, снимки);
- *надстройки;*
- *диаграммы* (позволяет вставить на листы диаграммы);
- *обзоры;*
- *спарклайны* (мини-диаграммы, помещенные в отдельные ячейки);
- *фильтры*;
- *ссылки*;
- *текст* (позволяет вставить предварительно отформатированные надписи, колонтитулы, объекты WordArt и другие объекты);
- *символы* (позволяет вставить формулы и специальные символы).

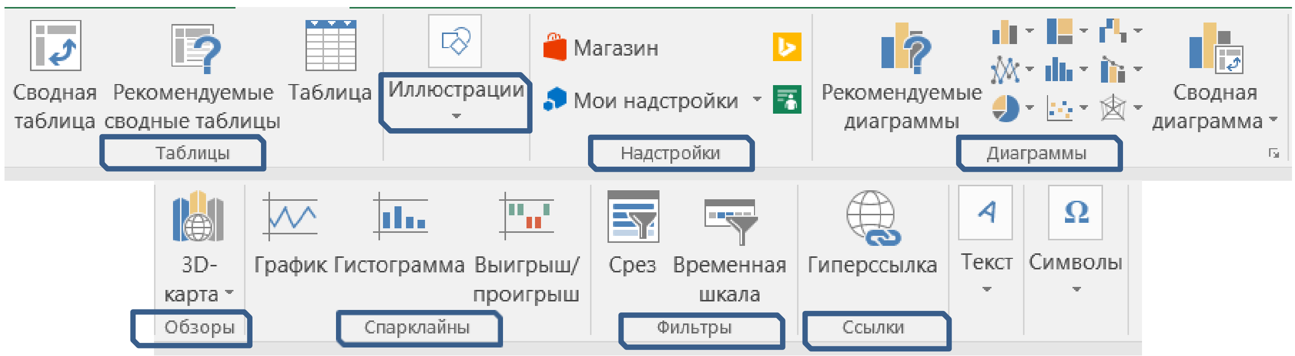


Рис. 1.8. Состав групп вкладки Вставка

Вкладка Разметка страницы (рис. 1.9) состоит из следующих групп инструментов, ориентированных на установку и настройку различных параметров разметки страницы:

- *темы* (изменение вида оформления электронной таблицы, в том числе цветов, шрифтов, эффектов);
- *параметры страницы* (выбор размеров полей, ориентации и размера бумаги, добавление в документ разрывов страниц, включение режима печати заголовков таблицы, задание подложки листов);
- *вписать* (изменение масштаба документов, задание распределения таблицы на определенное количество листов при печати);
- *параметры листа* (задание вида отображения данных на листе);
- *упорядочение* (корректировка параметров размещения выделенного объекта на листе, привязка к сетке).

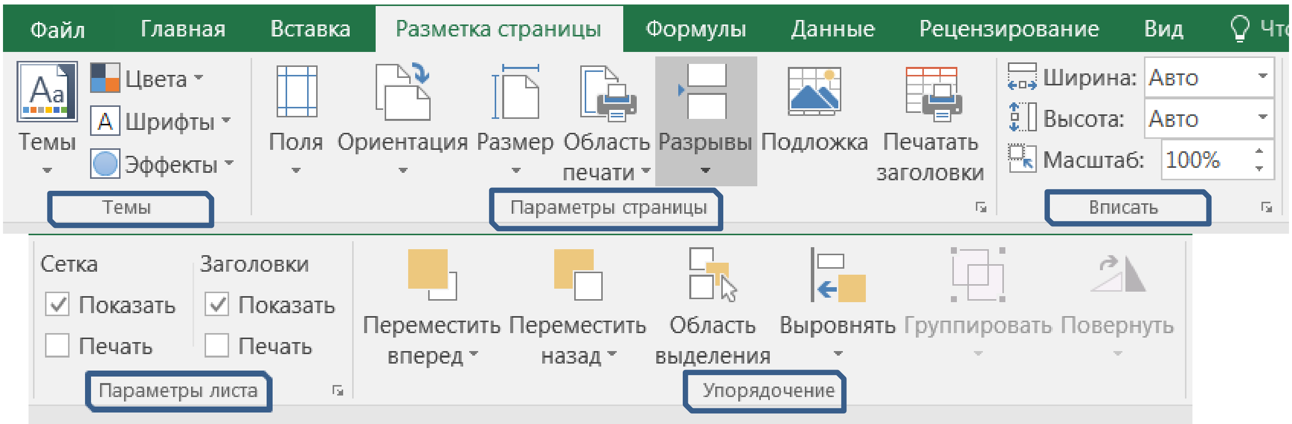


Рис. 1.9. Состав групп вкладки Разметка страницы

Вкладка Формулы (рис. 1.10) состоит из групп инструментов по созданию и использованию формул на листах таблицы:

- *библиотека функций* (используется для вставки функций различных типов);
- *определенные имена* (задание и использование имен для более удобной работы ячейками в формулах);
- *зависимости формул* (нахождение зависимостей и проверка формул);
- *вычисление* (задание пересчета формул).

Вкладка Данные (рис. 1.11) состоит из групп инструментов для различных операций с данными:

- *получение внешних данных* и *скачать и преобразовать* (импорт данных из различных приложений);
- *подключения* (интерактивный сбор данных);
- *сортировка и фильтр* (расширенные возможности сортировки и отбора данных);
- *работа с данными* (проверка, объединение данных, подбор заданных параметров, удаление повторяющихся значений);

- *прогноз* (проверка и прогнозирование данных);
- *структура* (группировка и подведение итогов).

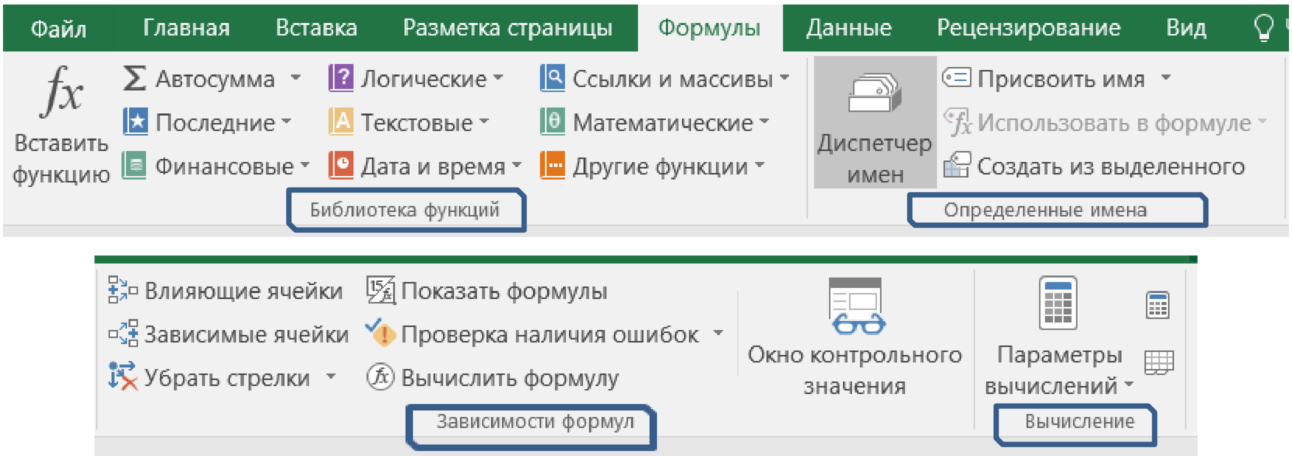


Рис. 1.10. Состав групп вкладки Формулы

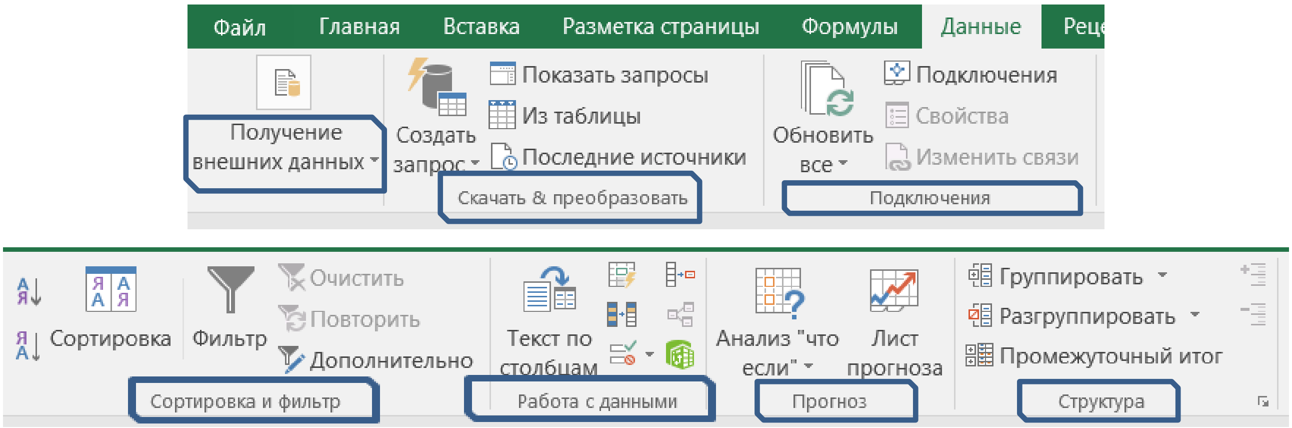


Рис. 1.11. Состав групп вкладки Данные

<u>Вкладка Рецензирование</u> (рис. 1.12) состоит из следующих групп:

- *правописание (проверка орфографии, грамматики, использование справочников);*
- *подробные сведения* (определения и другая информация из различных источников в Интернете);
- *язык (перевод выделенного текста на другой язык);*
- *примечания* (создание, редактирование и удаление примечаний к ячейкам);
- *изменения* (настройка параметров защиты листов и книг).

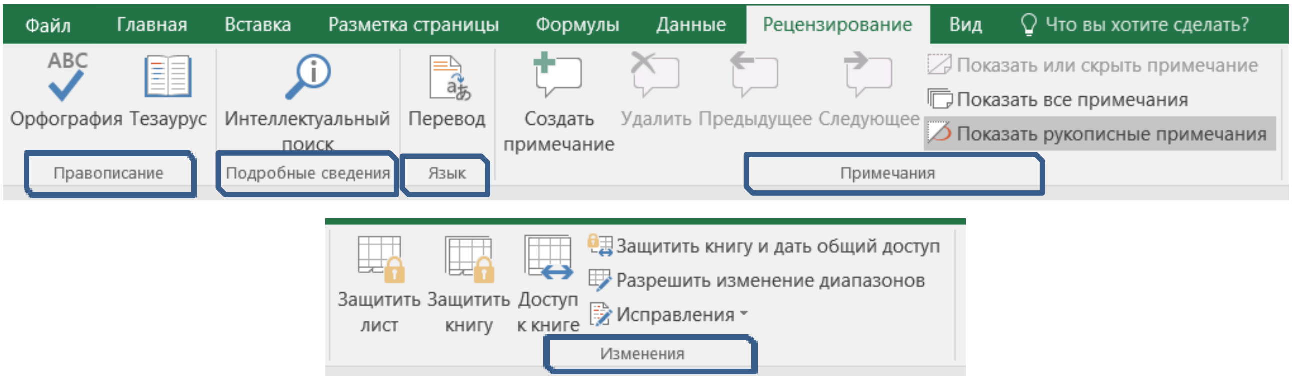


Рис. 1.12. Состав групп вкладки Рецензирование

<u>Вкладка Вид</u> (рис. 1.13) состоит из следующих групп инструментов, предназначенных для настройки режимов просмотра документов:

- *режимы просмотра книги* (просмотр документов в различных видах);
- *показать или скрыть* (дополнительные элементы настройки отображения элементов окна);
- *масштаб* (изменение масштаба книги и ее частей);
- *окно (открытие нового окна, упорядочивание* и управление открытыми окнами, разделение текущего окна на два окна для одновременного просмотра разных частей документа);
- *макросы* (работа с макросами в таблицах).

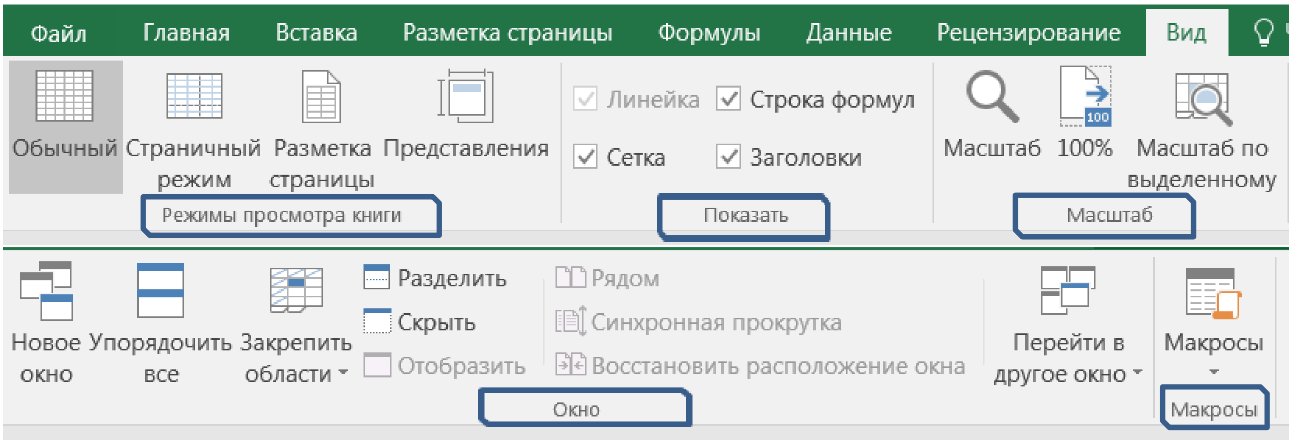


Рис. 1.13. Состав групп вкладки Вид

<u>Строка состояния</u>

В нижней части окна программы находится строка состояния (рис. 1.14). По умолчанию в правой части строки отображаются ярлыки режимов просмотра книги, масштаб. Чтобы изменить набор отображаемых элементов, необходимо щелкнуть правой кнопкой мыши на строке состояния. Снимая или устанавливая флажки соответствующих пунктов меню, можно настроить вид строки состояния по своему желанию.

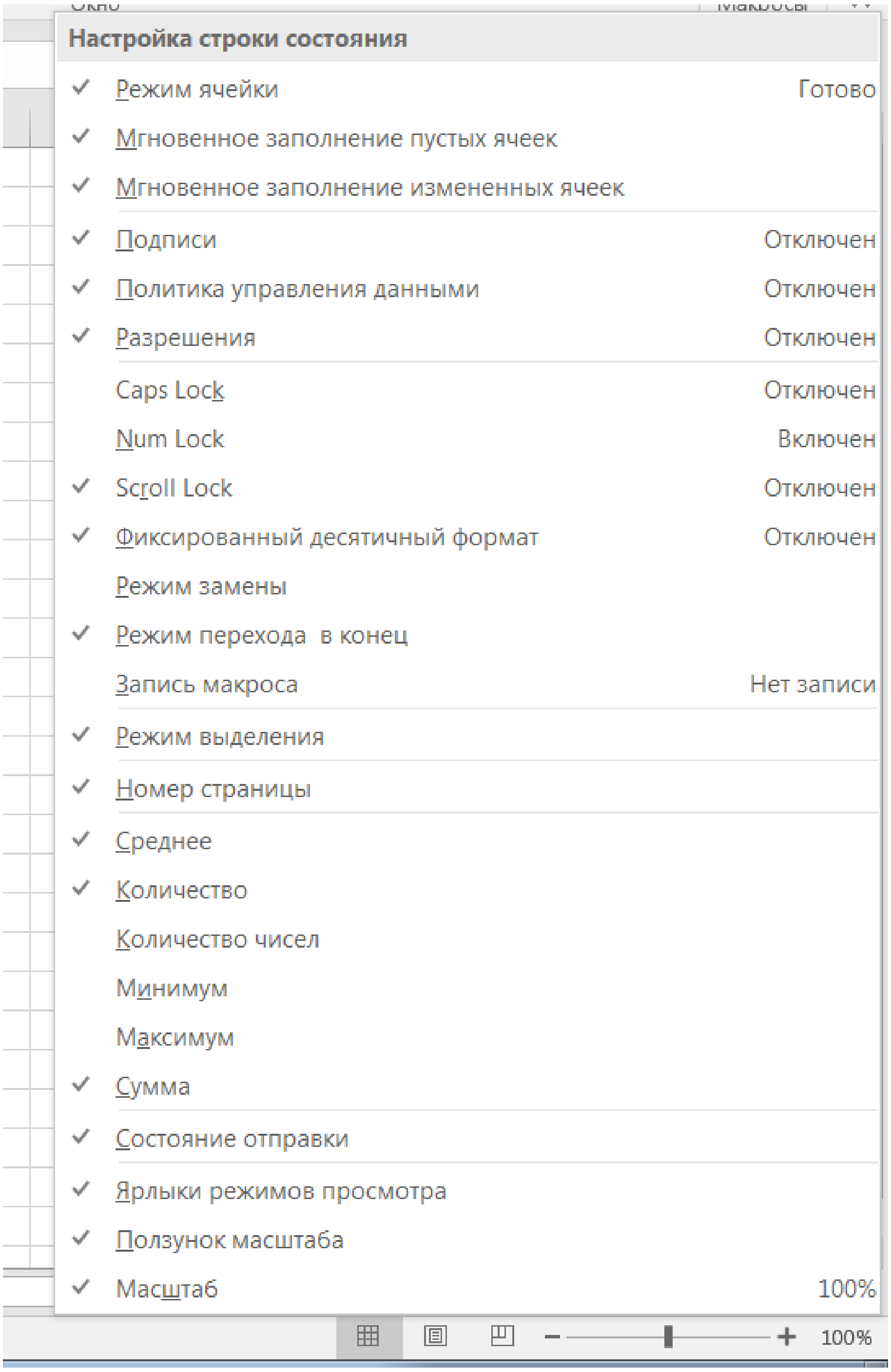


Рис. 1.14. Настройка строки состояния

Контрольные вопросы

1. Что такое электронная таблица?
2. Для чего могут использоваться табличные процессоры?
3. Как дополнить панель быстрого доступа новыми кнопками?
4. Для чего нужен буфер обмена?
5. С помощью инструментов какой вкладки можно настроить параметры страницы?
6. Что отображается в строке состояния?

1.2. Работа с книгами в MS Excel 2016

1.2.1. Создание новой книги

Все создаваемые в *Excel* файлы называются книгами, они состоят из множества рабочих листов и имеют расширение *.xlsx*[1] (если книга не содержит макросы) и *.xlsm* (если книга содержит макросы), также книга может иметь расширение *.xltx* (книга-шаблон) и *.xltm* (книга-шаблон с макросами). При запуске программы автоматически создается новая пустая книга. Для создания новой книги также предназначен пункт *Создать* вкладки *Файл*. При его выборе появляется окно *Доступные шаблоны* (рис. 1.15). В нем необходимо указать категорию шаблонов, на основе которых будет создана книга. По умолчанию используется вариант *Пустая книга*, но может быть выбран необходимый шаблон из имеющихся или есть возможность поиска необходимого шаблона в Интернете.

Для завершения создания нового файла необходимо кликнуть на нужный шаблон.

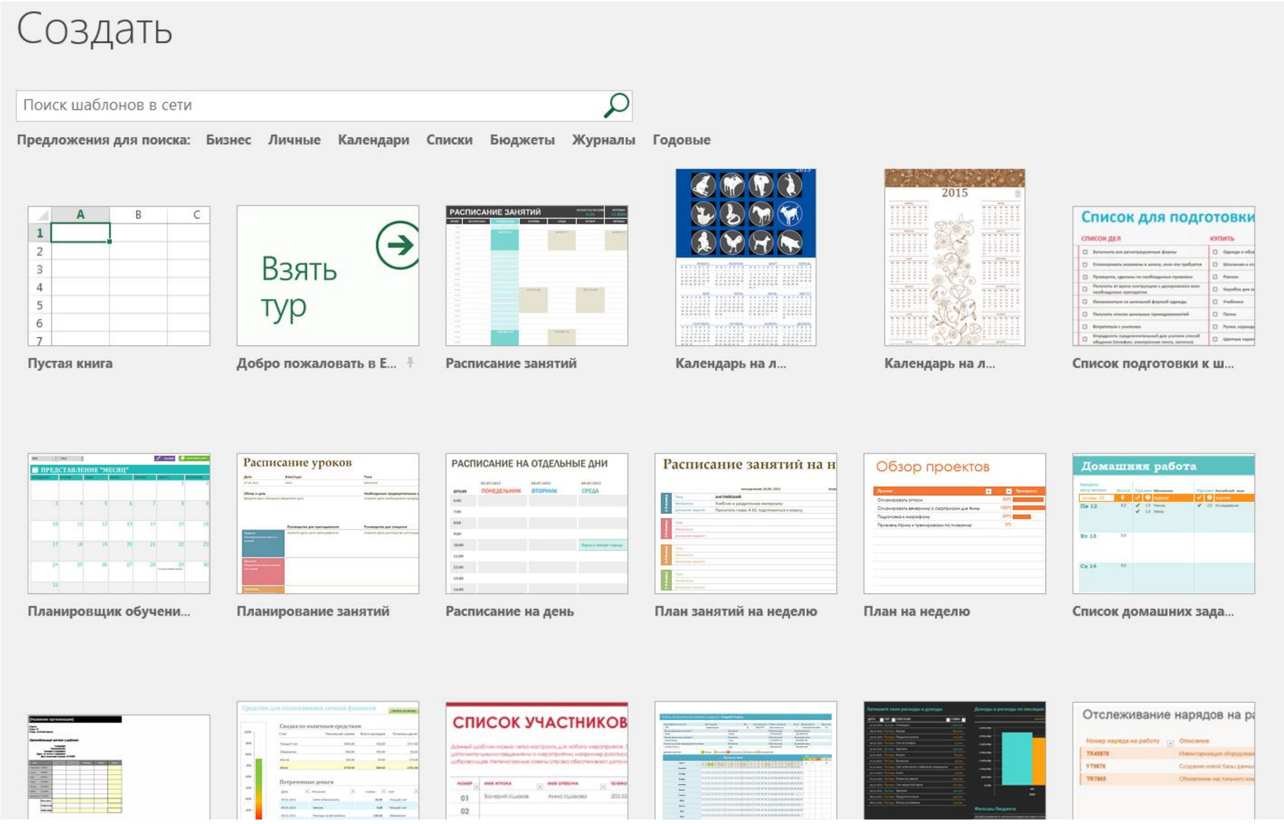


Рис. 1.15. Диалоговое окно Создать

[1] *.xls* в *Ms Excel 97-2003*

Сохранение книги

Excel 2016 по умолчанию сохраняет файлы в формате *.xlsx*. Этот формат не поддерживается старыми версиями программы[2]. Поэтому, чтобы файл можно было прочитать не только в *Excel 2007, Excel 2010, Excel 2013 и Excel 2016*, необходимо сохранять его в формате *Книга Excel 97-2003* (вкладка *Файл – Сохранить как…*). Также при помощи этой команды можно сохранить книгу в других доступных форматах.

Открытие книги

При открытии книги, созданной в *Excel* более ранних версий, она будет открыта в режиме совместимости. Чтобы для книги были доступны все функции *Excel 2016*, необходимо преобразовать файл с помощью команды *Преобразовать* вкладки *Файл – Сведения*.

Excel позволяет работать с несколькими книгами одновременно, открытыми в отдельных окнах. Кнопки панели *Окно* (рис. 1.16) вкладки *Вид* помогают упростить работу пользователя при этом:

- *Новое окно* – создает новое окно для рабочей книги;
- *Упорядочить все* – позволяет по-разному разместить окна всех открытых книг;
- *Перейти в другое окно* – переключение между окнами открытых книг.

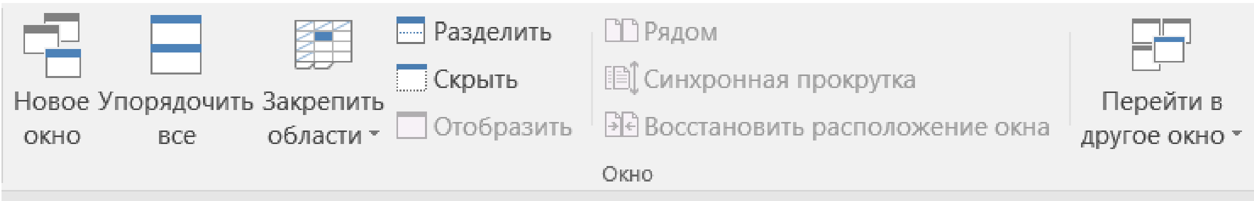


Рис. 1.16. Панель Окно

Для более удобной работы с частями одной книги, используются следующие команды:

Закрепить области – позволяет оставить на месте во время прокрутки определенные строки и столбцы;

Разделить  – разделение одного окна на несколько частей;

Рядом  – располагает окна открытых книг или листов рядом для сравнения их содержимого;

Синхронная прокрутка  – становится активной, при нажатой кнопке *Рядом* и позволяет синхронно прокручивать книги.

[2] До *Ms Office 2003* включительно

1.2.2. Защита книг и совместное использование

Для случаев, когда книгу нужно защитить от копирования и модификаций, в *Excel 2016* существуют способы защиты файлов книг (рис. 1.17).

Защита может быть задана:

- паролем для открытия книги;
- запретом манипулирования с ее листами;
- запретом изменения положения и размера окна рабочей книги.

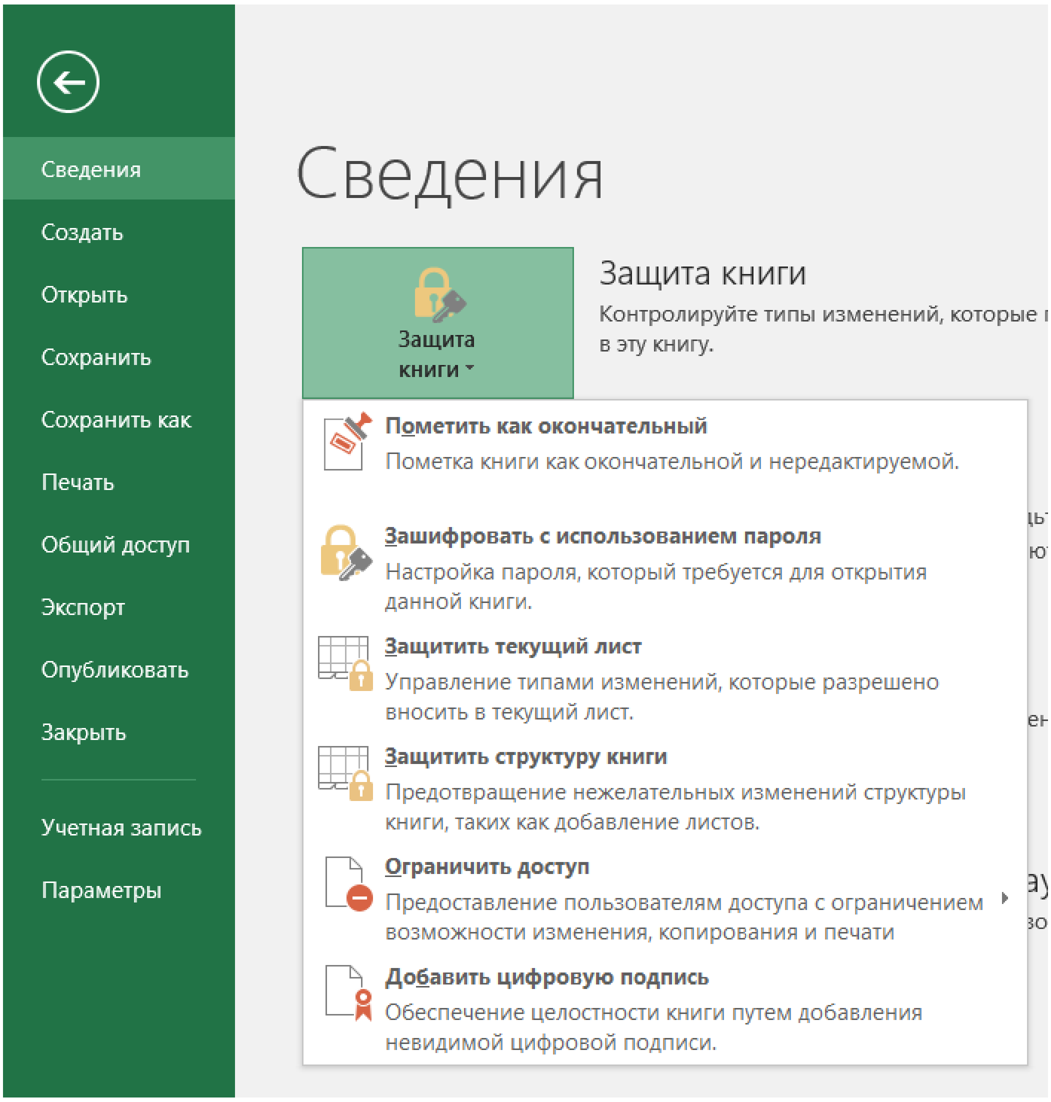


Рис. 1.17. Защита книги

Пароль на открытие книги можно задать командой *Файл – Сведения – Защитить книгу – Зашифровать с использованием пароля.*

Также для защиты от изменений книге может быть присвоен статус «окончательная» рабочая книга (*Файл – Сведения – Защитить книгу – Пометить как окончательный*), тогда книга сможет быть открыта только для чтения без возможности внесения изменений и сохранения под новым именем.

Контрольные вопросы

1. Какое расширение имеет файл книги, содержащей макросы?
2. Как создать книгу, используя шаблон?
3. Какие есть способы защиты файлов *Excel*?

1.3. Работа с листами и ячейками в MS Excel 2016

1.3.1. Основные операции с листами

Каждая рабочая книга в *Excel* состоит из рабочих листов. Лист состоит из ячеек, образующих строки и столбцы. На одном листе может содержаться 1048576 строк и 16384 столбцов.

Для *активизации* листа нужно нажать на его название в группе ярлыков листов либо на кнопки со стрелками для перехода к нужному листу.

Листы можно *добавлять, удалять, перемещать, копировать, переименовывать, выделять, задавать цвет для ярлыка*. Для этого можно воспользоваться соответствующей командой из контекстного меню по нажатию правой кнопки мыши на ярлыке листа.

1.3.2. Основные операции с ячейками, строками и столбцами

<u>Выделение</u>

Перед выполнением каких-либо действий (копирование, перемещение и удаление данных, размещение ссылок на ячейки в формулах и т.д.) с ячейками, строками, столбцами необходимо их выделить.

Чтобы *выделить* ячейки, установите курсор в нужную позицию и выделите смежные ячейки при помощи левой кнопки мыши. Чтобы *выделить диапазон* ячеек, установите курсор в левый верхний угол диапазона, удерживая клавишу *Shift*, установите курсор в правый нижний угол диапазона. Блок ячеек между этими позициями будет выделен. Чтобы *выделить несмежные* ячейки, проводите выделение нужных ячеек при нажатой клавише *Ctrl*.

Чтобы *выделить* строку/столбец, установите курсор на заголовок строки/столбца. Чтобы *выделить несколько смежных* строк/столбцов, установите курсор на заголовок строки/столбца начала диапазона, удерживая клавишу *Shift*, переместите курсор на заголовок строки/столбца конца диапазона. Чтобы выделить *несмежные* строки/столбцы, начинайте выделение нужных строк/столбцов при нажатой клавише *Ctrl*.

Чтобы *выделить все ячейки* листа, примените комбинацию клавиш *Ctrl+A* либо нажмите на кнопку на пересечении заголовков строк и столбцов.

Для *выделения текста внутри* ячейки нужно войти в режим редактирования (по клавише F2 или дважды щелкнуть мышью по ячейке) и выделить нужный текст.

<u>Копирование и перемещение</u>

Чтобы *скопировать* данные из ячейки/строки/столбца, нужно выделить необходимый элемент и по контекстному меню по нажатию правой кнопки мыши выбрать пункт *Копировать*, затем *Вставить*, переместив курсор и выделив нужное для вставки место. Также можно воспользоваться сочетаниями клавиш *Ctrl+Insert или Ctrl+C* (для копирования) и *Shift+Insert или Ctrl+V* (для вставки), либо с помощью левой кнопки мыши с нажатой одновременно клавишей *Ctrl* «перетащить» элемент в нужное место для получения там его копии, либо воспользоваться соответствующими кнопками на панели *Буфер обмена* вкладки *Главная*.

Чтобы *переместить* данные из ячейки/строки/столбца, нужно выделить необходимый элемент и по контекстному меню по нажатию правой кнопки мыши выбрать пункт *Вырезать*, затем *Вставить*, переместив курсор и выделив нужное для вставки место. Также можно воспользоваться сочетаниями клавиш *Shift+Delete* или *Ctrl+X* (для вырезания) и *Shift+Insert* или *Ctrl+V* (для вставки), либо просто перетащить на новое место элемент левой кнопкой мыши, либо воспользоваться соответствующими кнопками на панели *Буфер обмена* вкладки *Главная*.

Данные, которые были скопированы, могут быть вставлены в новое место на листе с определенными параметрами. Для этого при вставке скопированных данных используйте команду *Специальная вставка* контекстного меню (рис. 1.18). В открывшемся диалоговом окне можно выбрать нужные параметры вставки (рис. 1.19). В этом же окне можно задать транспонирование ячеек, т.е. изменение расположения данных в них с горизонтального на вертикальное и наоборот.

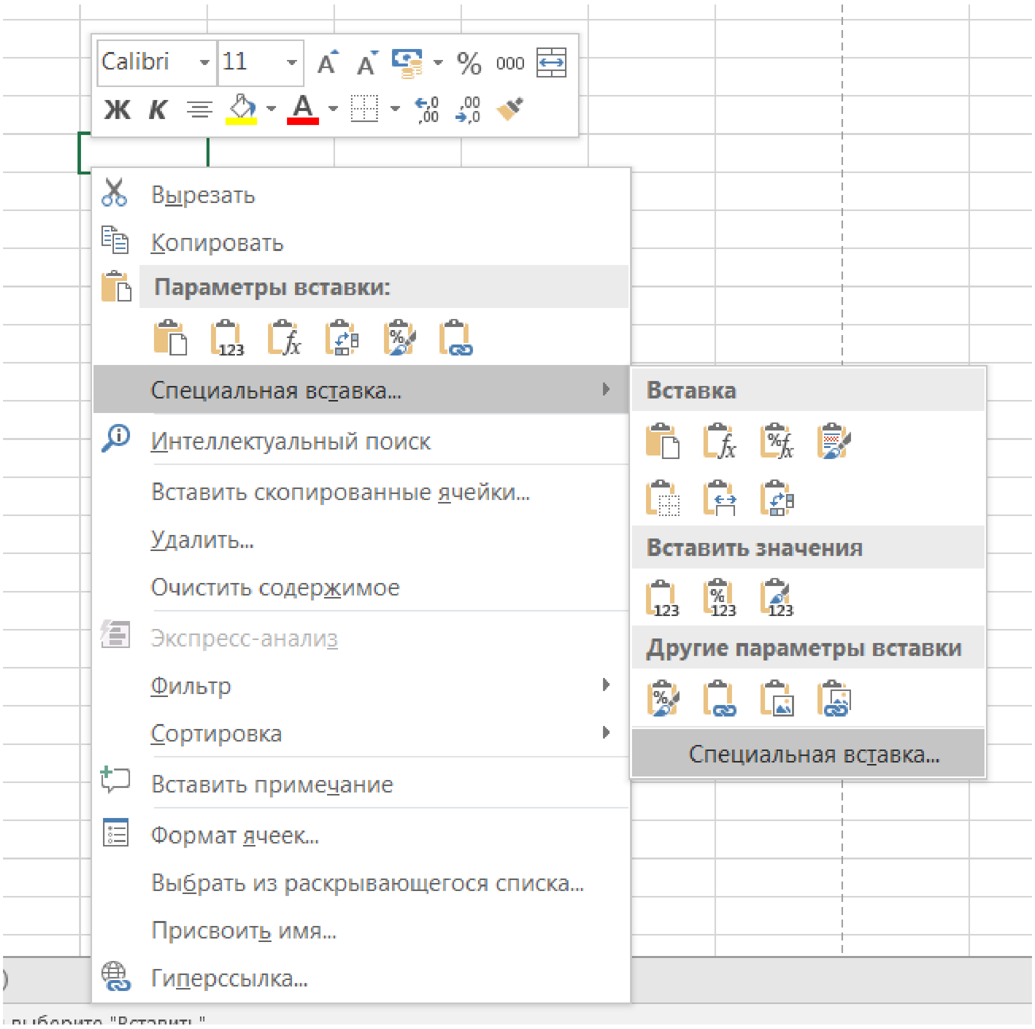


Рис. 1.18. Контекстное меню

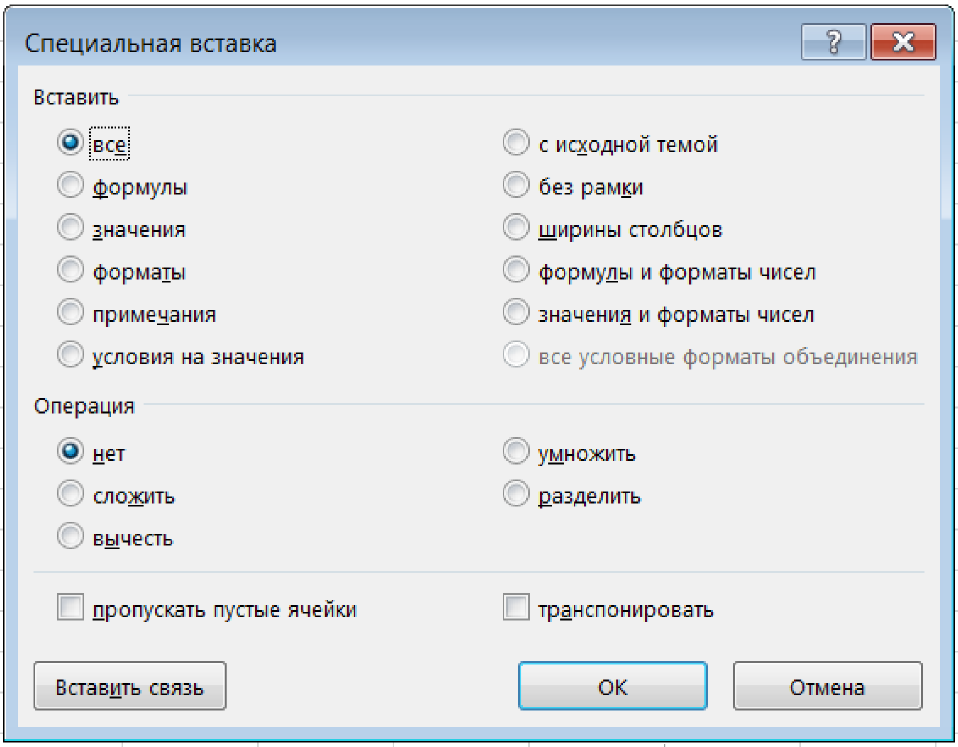


Рис. 1.19. Окно специальной вставки

Добавление и удаление

Чтобы *добавить новую ячейку* на лист, нужно выделить место вставки новой ячейки, по контекстному меню выбрать команду *Вставить…* и в появившемся окне *Добавление ячеек* выбрать нужный вариант.

Чтобы *добавить новую строку/столбец*, нужно выделить строку/столбец, перед которой будет вставлена новая/новый, и по контекстному меню командой *Вставить* осуществить вставку элемента, либо использовать команду *Главная – Ячейки – Вставить*.

Чтобы *удалить строку/столбец*, нужно выделить данный элемент, и по контекстному меню командой *Удалить*, выполнить удаление, либо применить команду *Главная – Ячейки – Удалить*. При удалении строки произойдет сдвиг вверх, при удалении столбца – сдвиг влево.

Для *удаления ячеек* со сдвигом выберите из контекстного меню команду *Удалить..* и укажите способ удаления.

Для *удаления данных из ячеек* воспользуйтесь командой контекстного меню *Очистить содержимое*. Также на панели *Редактирование* вкладки *Главная* существует кнопка *Очистить* , позволяющая выбрать, что именно вы хотите очистить в ячейке (все, формат, содержимое, примечания).

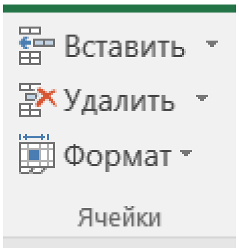


Рис. 1.20. Панель Ячейки

Контрольные вопросы

1. Что можно «делать» с листом в книге?
2. Как переименовать лист в книге?
3. С помощью чего можно выделить все ячейки листа?
4. Как разделить лист на несколько частей для просмотра?
5. Для чего используется «специальная вставка»?
6. Как удалить одну ячейку на листе?

1.4. Форматирование таблиц в MS Excel 2016

Форматирование придает таблицам, созданным в табличном процессоре, законченный вид и позволяет акцентировать внимание пользователей на нужных деталях.

1.4.1. Форматирование ячеек

Изменение основных параметров формата

Основные средства для форматирования ячеек расположены на панелях *Шрифт, Выравнивание, Стили, Число, Ячейки* вкладки *Главная*, в диалоговом окне *Формат ячеек (*открываемой с панелей инструментов, либо из контекстного меню*)* и на мини-панели (рис. 1.21), вызываемой при нажатии на ячейке правой кнопки мыши.

Рис. 1.21. Мини-панель форматирования

Выделив ячейку и вызвав диалоговое окно форматирования (рис. 1.22), можно применить различные способы оформления ячеек.

Для задания параметров шрифта (вид, начертание, цвет, размер, видоизменение) можно использовать инструменты с мини-панели, с панели Шрифт вкладки Главная либо со вкладки Шрифт окна Формат ячеек.

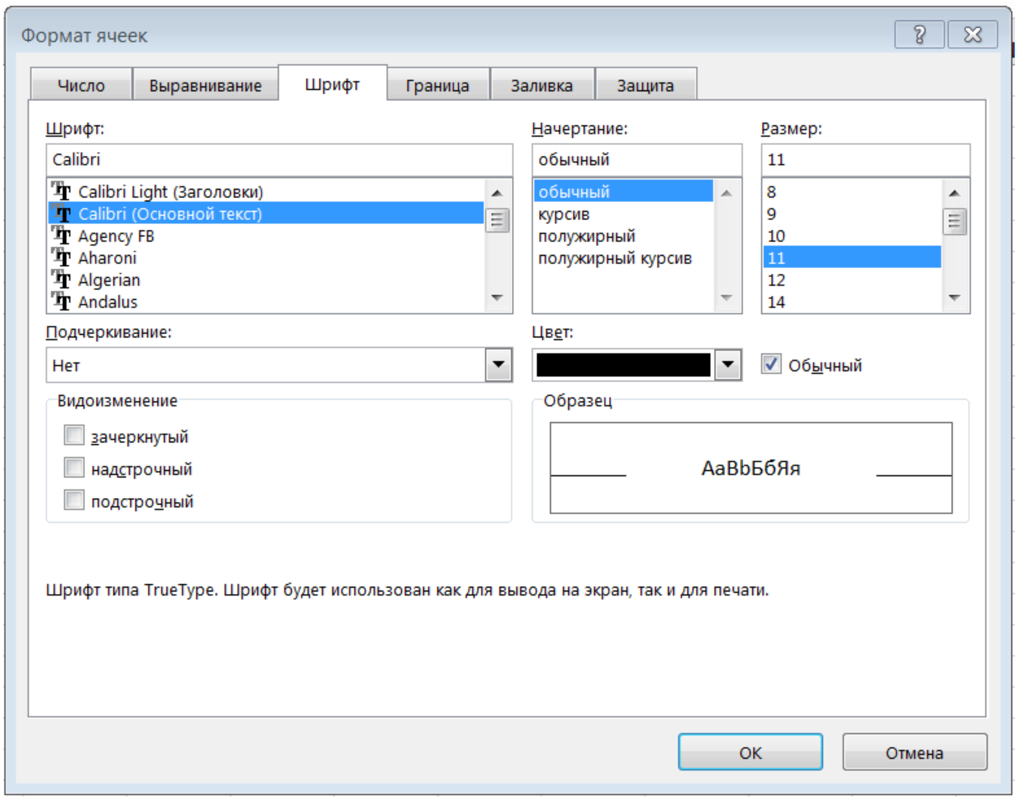


Рис. 1.22. Диалоговое окно Формат ячеек

1.4.1. Формат по образцу

Полезной при форматировании является опция «Формат по образцу», вызываемая кнопкой  с панели *Буфер обмена* вкладки *Главная*. Она переносит параметры форматирования выделенной ячейки на новый фрагмент таблицы. Чтобы перенести все заданные параметры форматирования на новый элемент необходимо:

- установить курсор в ячейке, параметры форматирования которой мы хотим использовать;
- нажать кнопку *Формат по образцу* на вкладке *Главная – Буфер обмена* (рис. 1.23) (если необходимо форматировать за один раз несколько разных фрагментов, следует сделать двойной щелчок на кнопке);
- выделить ячейку, на которую надо перенести форматирование (если был сделан двойной щелчок на кнопке *Формат по образцу*, то можно выделять последовательно несколько ячеек; по завершении всей операции форматирования надо один раз щелкнуть на кнопке *Формат по образцу*, чтобы «отжать» ее).

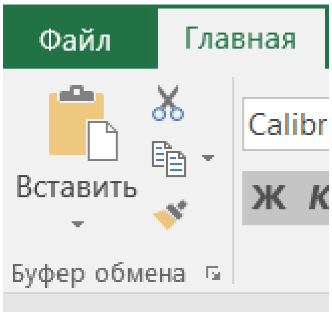


Рис. 1.23. Панель *Буфер обмена*

1.4.2. Форматирование с помощью стилей

Форматирование ячеек может быть осуществлено с использованием стилей – заготовок, включающих в себя определенный набор параметров форматирования ячеек. Удобство стилей заключается в том, что все ячейки, отформатированные одним стилем, будут изменять свой вид при редактировании параметров стиля.

Чтобы использовать готовый стиль, нужно выделить ячейки и по команде *Главная – Стили – Стили ячеек* применить нужный стиль (рис. 1.24). После применения стиля к ячейкам можно дополнительно использовать любые другие методы форматирования.

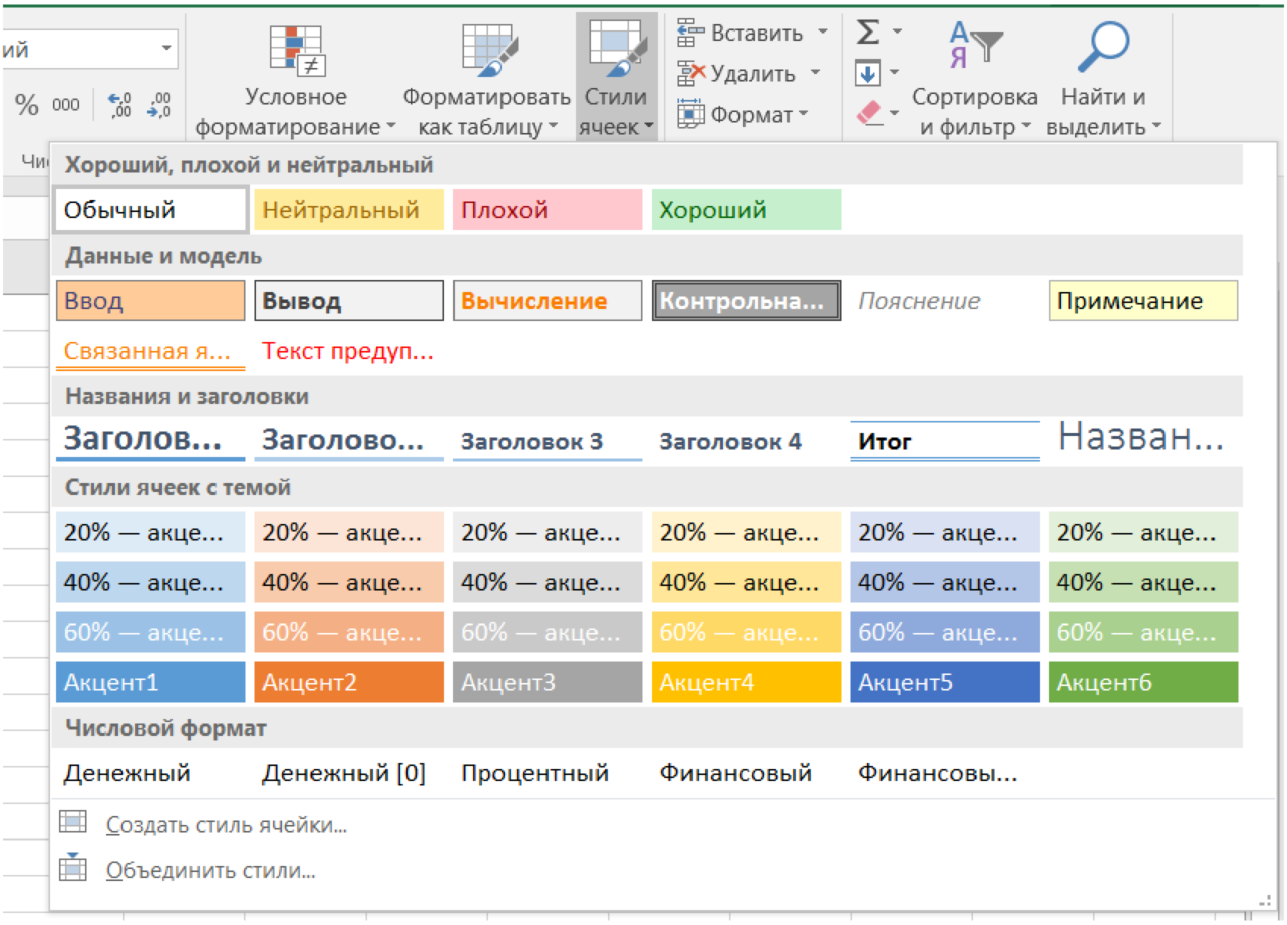


Рис. 1.24. Окно выбора стилей ячеек

1.5. Ввод данных в MS Excel 2016

В ячейках электронной таблицы могут находиться данные трех типов: числовые значения (включая время и дату), текст, формулы. На рабочем листе, но в «графическом слое» поверх листа, могут также находиться рисунки, диаграммы, изображения, кнопки и другие объекты.

Ввод чисел

Числа вводятся с помощью верхнего ряда клавиатуры или числовой клавиатуры. В качестве десятичного разделителя применяется запятая или точка, можно вводить знаки денежных единиц. Если перед числом ввести «минус» или скобки, то оно считается отрицательным. Нули, набранные перед числом, игнорируются программой (рис. 1.25). Если необходимо получить значение с нулями впереди, его необходимо интерпретировать как текстовое.

Числовые значения автоматически выравниваются по правой границе ячейки.

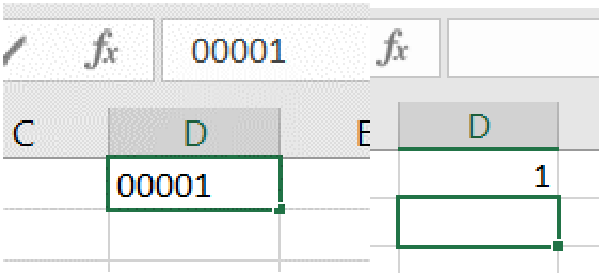


Рис. 1.25. Игнорирование нулей

<u>Ввод значений дат и времени</u>

Excel для представления дат использует внутреннюю систему порядковой нумерации дат. (Так, самая ранняя дата, которую может распознать программа, 1 января 1900 года, этой дате присвоен порядковый номер 1, следующей дате – порядковый номер 2 и т. д.). Даты вводятся в привычном для пользователя формате и распознаются автоматически (рис. 1.26). Временные значения также вводятся в одном из распознаваемом форматов времени. Представление даты и времени непосредственно на листе регулируется заданием формата отображения ячейки.

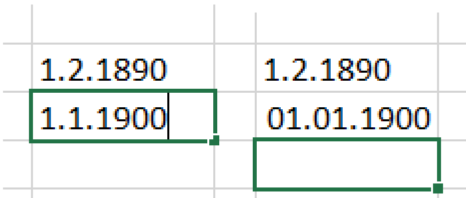


Рис. 1.26. Распознавание дат

<u>Ввод текста</u>

Как текстовые значения воспринимаются все введенные данные, не распознаваемые как числа или формулы. Текстовые значения выравниваются по левой границе таблицы. Если текст не умещается в одной ячейке, то он располагается поверх соседних ячеек, если они свободны. Параметры расположения текста в ячейке задаются через формат ячеек.

<u>Ввод формулы</u>

Формулой считается любое математическое выражение. Формула всегда начинается со знака «=», может включать в себя, кроме операторов и ссылок на ячейки, встроенные функции *Excel* (рис. 1.27).

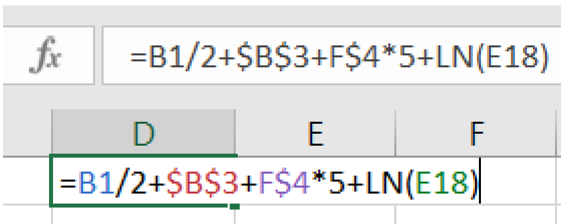


Рис. 1.27. Ввод формулы

1.6. Обработка и анализ данных

Набор строк в *Excel*, содержащий взаимосвязанные данные и определенную структуру, называется *списком.* Такой диапазон можно сортировать, группировать, фильтровать, проводить в нем поиск и выполнять вычисления. Данные на рис. 1.28. структурированы и удовлетворяют следующим критериям:

- Ячейки каждого столбца (т.е. поля) содержат однотипную информацию, имеют одну размерность, поэтому, например, дата и время представлены в одном формате (ДД.ММ.ГГ и ЧЧ.ММ).
- Каждый столбец имеет уникальный заголовок, расположенный прямо над данными.
- Все ячейки в каждой строке образуют одну запись и занимают не более одной строки.
- Абсолютно незаполненные строки не могут считаться данными – создав запись, следует заполнить хотя бы одно поле.

ФИО	Телефон	Дата рождения	Должность	№ кабинета
Смирнов Ф.И.	648-85-96	20.12.1968	Генеральный директор	21
Федорова И.А.	648-85-65	05.02.1972	Зам. директора	15
Емельянов В.П.	648-85-75	09.09.1975	Глав. бух	24
Портнов М.В.	648-85-84	24.04.1987	Юрист	2
Васильева Н.В.	648-85-92	30.06.1973	Юрист	6
Иванова П.Р.	648-85-97	05.11.1962	Главный юрист	26

Рис. 1.28. Пример списка данных

1.6.1. Сортировка данных

Для осуществления сортировки необходимо сначала выделить диапазон ячеек, содержащих данные для сортировки, либо ячейку столбца, по которому будет проведена сортировка. После выделения можно применить команды (для сортировки по возрастанию значений) и (для сортировки по убыванию значений). Команду настраиваемой сортировки можно вызвать через *Главная – Редактирование – Сортировка и фильтр* (рис. 1.29) либо через *Данные – Сортировка и фильтр – Сортировка* (рис. 1.30). В появившемся окне *Сортировка* (рис. 1.31) нужно указать столбец, порядок и особенности сортировки списка данных.

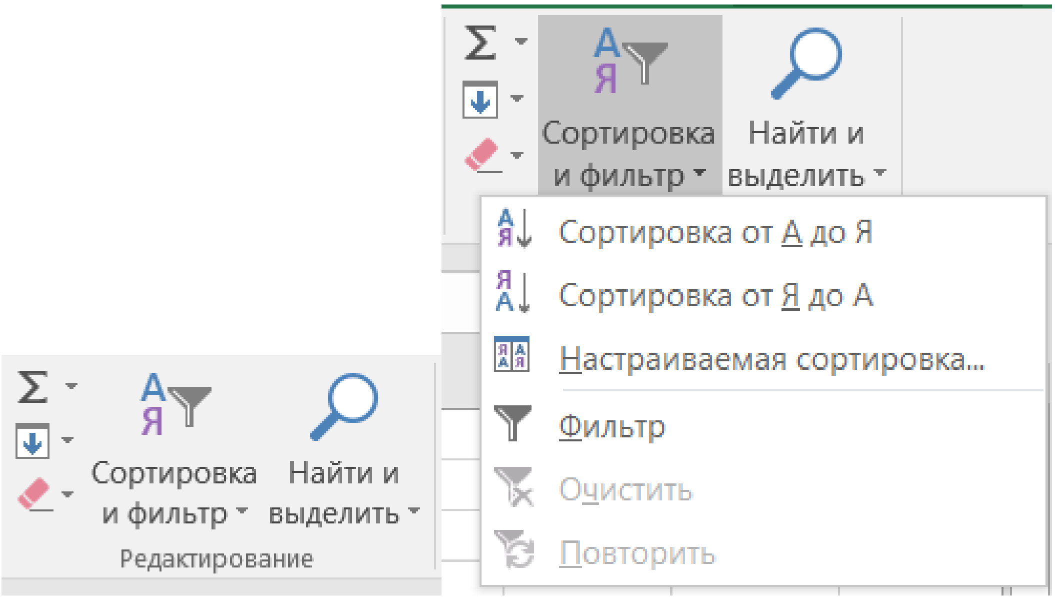


Рис. 1.29. Панель *Сортировка и фильтр*

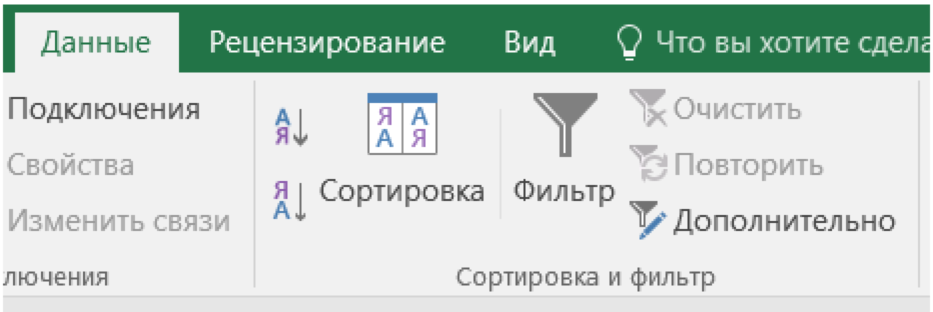


Рис. 1.30. Панель *Сортировка*

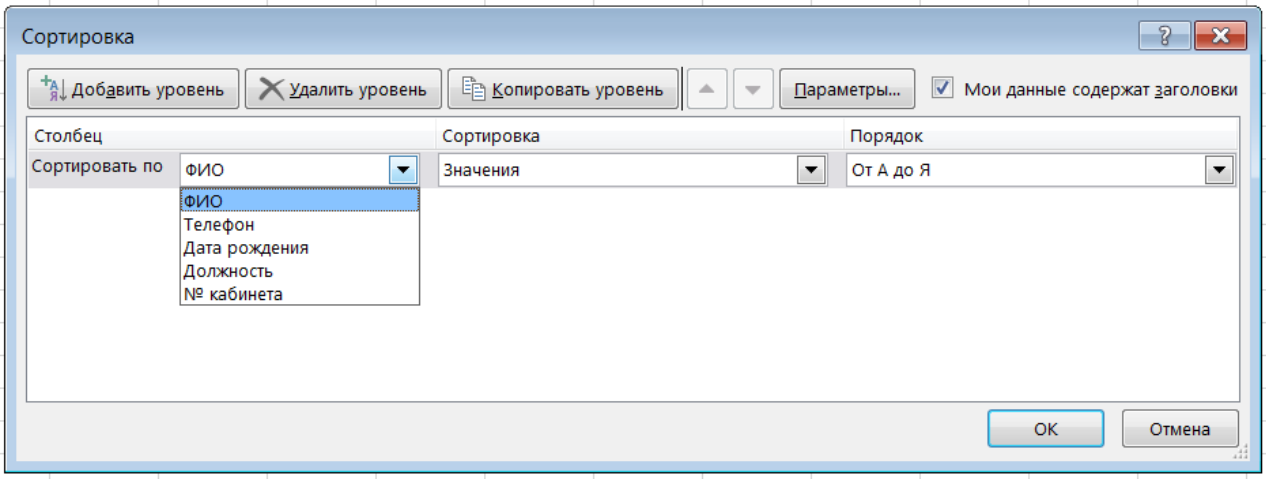


Рис. 1.31. Диалоговое окно *Сортировка*

2. Система компьютерной алгебры. Mathcad

2.1. Знакомство с Mathcad

Математический пакет *Mathcad* – система компьютерной математики, предназначенная для решения математических задач в различных областях науки, техники и образования. Название системы образовано от двух слов: *MATHematics* (математика) и *CAD* (*Computer Aided Design* – системы автоматического проектирования – САПР). Пакет *Mathcad* создан фирмой *MathSoft Inc*.

Mathcad – система компьютерной алгебры из класса систем автоматизированного проектирования, ориентированная на подготовку интерактивных документов с вычислениями и визуальным сопровождением, отличается легкостью использования и применения для коллективной работы.

Mathcad был задуман и первоначально написан Алленом Раздовым из Массачусетского технологического института (MIT), соучредителем компании Mathsoft, которая с 2006 года является частью корпорации PTC (Parametric Technology Corporation).

Mathcad имеет простой и интуитивный для использования интерфейс пользователя. Для ввода формул и данных можно использовать как клавиатуру, так и специальные панели инструментов.

Некоторые из математических возможностей *Mathcad* (версии до 13.1 включительно) основаны на подмножестве системы компьютерной алгебры *Maple* (MKM, Maple Kernel Mathsoft). Начиная с 14 версии – использует символьное ядро MuPAD.

Работа осуществляется в пределах рабочего листа, на котором уравнения и выражения отображаются графически, в противовес текстовой записи в языках программирования. При создании документов-приложений используется принцип WYSIWYG (What You See Is What You Get – "что видишь, то и получаешь").

Несмотря на то, что эта программа в основном ориентирована на пользователей-непрограммистов, *Mathcad* также используется в сложных проектах, чтобы визуализировать результаты математического моделирования путем использования распределённых вычислений и традиционных языков программирования. Также *Mathcad* часто используется в крупных инженерных проектах, где большое значение имеет трассируемость и соответствие стандартам.

Mathcad достаточно удобно использовать для обучения, вычислений и инженерных расчетов. Открытая архитектура приложения в сочетании с поддержкой технологий .NET и XML позволяют легко интегрировать *Mathcad* практически в любые ИТ-структуры и инженерные приложения. Есть возможность создания электронных книг (e-Book).

Mathcad относится к системам компьютерной алгебры, то есть средств автоматизации математических расчетов. В этом классе программного обеспечения существует много аналогов различной направленности и принципа построения. Наиболее часто *Mathcad* сравнивают с такими программными комплексами, как *Maple, Mathematica, MATLAB*, а также с их аналогами *MuPAD, Scilab, Maxima* и др. Впрочем, объективное сравнение осложняется в связи с разным назначением программ и идеологией их использования.

Система *Maple*, например, предназначена главным образом для выполнения аналитических (символьных) вычислений и имеет для этого один из самых мощных в своем классе арсенал специализированных процедур и функций (более 3000). Такая комплектация для большинства пользователей, которые сталкиваются с необходимостью выполнения математических расчетов среднего уровня сложности, является избыточной. Возможности Maple ориентированы на пользователей – профессиональных математиков; решения задач в среде *Maple* требуют не только умения оперировать какой-либо функцией, но и знания методов решения, в неё заложенных: во многих встроенных функциях Maple фигурирует аргумент, задающий метод решения.

То же самое можно сказать и о *Mathematica*. Это одна из самых мощных систем, имеет чрезвычайно большую функциональную наполненность (есть даже синтезирование звука). *Mathematica* обладает высокой скоростью вычислений, но требует изучения довольно необычного языка программирования.

Разработчики *Mathcad* сделали ставку на расширение системы в соответствии с потребностями пользователя. Для этого назначены дополнительные библиотеки и пакеты расширения, которые можно приобрести отдельно и которые имеют дополнительные функции, встраиваемые в систему при установке, а также электронные книги с описанием методов решения специфических задач, с примерами действующих алгоритмов и документов, которые можно использовать непосредственно в собственных расчетах. Кроме того, в случае необходимости и при условии наличия навыков программирования в С, есть возможность создания собственных функций и их прикрепления к ядру системы через механизм DLL.

Mathcad, в отличие от *Maple*, изначально создавался для численного решения математических задач, он ориентирован на решение задач именно прикладной, а не теоретической математики, когда нужно получить результат без углубления в математическую суть задачи. Впрочем, для тех, кому нужны символьные вычисления и предназначено интегрированное ядро *Maple* (с версии 14 – MuPAD). Особенно это полезно, когда речь идет о создании документов образовательного назначения, когда необходимо продемонстрировать построение математической модели, исходя из физической картины процесса или явления. Символьное ядро *Mathcad*, в отличие от оригинального *Maple* (MuPAD), искусственно ограничено (доступно около 300 функций), но этого в большинстве случаев вполне достаточно для решения задач инженерного характера.

Mathcad представляет собой совокупность следующих основных компонентов:

- редактор текстов (с возможностью вставки математических выражений, шаблонов графиков и текстовых комментариев);
- графический редактор (вставка графических областей двумерных и трехмерных графиков с использованием шаблонов);
- редактор формул (вставка математических выражений);
- справочная система (система для получения справочных данных по тематическому и индексному каталогу, а также для поиска нужных данных по ключевой фразе или слову).

Любое изменение содержимого рабочего документа *Mathcad* вызывает обновление всех зависимых результатов и областей с графиками.

Система реализует типовые возможности Windows последних версий, включая использование шрифтов, работу со всеми типами принтеров, одновременное выполнение разнохарактерных задач. В режиме редактирования возможна одновременная работа с рядом документов и перенос объектов из одного окна в другое.

Предусмотрен импорт графических изображений различной сложности. Есть возможность вращения трехмерных графиков, а также возможность анимации объектов.

Возможно дополнение *Mathcad* новыми возможностями с помощью специализированных пакетов расширений и библиотек, которые пополняют систему дополнительными функциями и константами для решения специализированных задач:

- Пакет для анализа данных (англ. Data Analysis Extension Pack) – обеспечивает *Mathcad* необходимыми инструментами для анализа данных.
- Пакет для обработки сигналов (англ. Signal Processing Extension Pack) – содержит более 70 встроенных функций для аналоговой и цифровой обработки сигналов, анализа и представления результатов в графическом виде.
- Пакет для обработки изображений (англ. Image Processing Extension Pack) – обеспечивает *Mathcad* необходимыми инструментами для обработки изображений, анализа и визуализации.
- Пакет для работы с функциями волнового преобразования (англ. Wavelets Extension Pack) – содержит большой набор дополнительных вейвлет-функций, которые можно добавить в библиотеку встроенных функций базового модуля *Mathcad Professional*. Пакет предоставляет возможность применить новый подход к анализу сигналов и изображений, статистической оценки сигналов, анализа сжатия данных, а также специальных численных методов. Функциональность включает одно- и двухмерные вейвлеты, дискретные вейвлет-преобразования, мультианализ разрешения и многое другое. Пакет объединяет более 60 функций ключевых вейвлетов. Включены

ортогональные и биортогональные семейства вейвлетов, среди прочего – вейвлет Хаара, вейвлет Добеши, симлет, койфлет и B-сплайны. Пакет также содержит обширную диалоговую документацию по основным принципам вейвлетов, приложения, примеры и таблицы ссылок.

- Библиотека строительства (англ. Civil Engineering Library) – включает справочник англ. Roark's Formulas for Stress and Strain (Формулы Роарка для расчета напряжений и деформаций), настраиваемые шаблоны для строительного проектирования и примеры тепловых расчётов.
- Электротехническая библиотека (англ. Electrical Engineering Library) – содержит стандартные вычислительные процедуры, формулы и справочные таблицы, используемые в электротехнике. Текстовые пояснения и примеры облегчают работу с библиотекой – каждый заголовок имеет гиперссылку на оглавление и указатель, и его можно найти в системе поиска.
- Библиотека машиностроения (англ. Mechanical Engineering Library) – включает справочник англ. Roark's Formulas for Stress and Strain (Формулы Роарка для расчета напряжений и деформаций), содержащий более пяти тысяч формул, вычислительные процедуры из справочника McGraw-Hill и метод конечных элементов. Текстовые пояснения, поисковая система и примеры облегчают работу. В состав библиотеки включена электронная книга Дэвида Пинтура “Введение в метод конечных элементов”.

2.2. Основные возможности пакета Mathcad

К основным элементам математических выражений MathCAD относятся *типы данных, операторы, функции* и *управляющие структуры.*

Операторы

Операторы – элементы MathCAD, с помощью которых можно создавать математические выражения. К ним, например, относятся символы арифметических операций, знаки вычисления сумм, произведений, производной и интеграла и т.д.

Оператор определяет:

1) действие, которое должно выполняться при наличии тех или иных значений операндов;
2) сколько, где и какие операнды должны быть введены в оператор.

Операнд – число или выражение, на которое действует оператор. Например, в выражении **5! + 3** число **3** и выражение **5!** – операнды оператора **+** (плюс), а число **5** операнд оператора факториал (!). После указания *операндов* операторы становятся исполняемыми по документу блоками.

Типы данных

К *типам данных* относятся числовые константы, обычные и системные переменные, массивы (векторы и матрицы) и данные файлового типа.

Константами называют поименованные объекты, хранящие некоторые значения, которые не могут быть изменены. *Переменные* являются поименованными объектами, имеющими некоторое значение, которое может изменяться по ходу выполнения программы. Тип переменной определяется ее значением; переменные могут быть числовыми, строковыми, символьными и т. д. Имена констант, переменных и иных объектов называют *идентификаторами*. Идентификаторы в MathCAD представляют собой набор латинских или греческих букв и цифр.

В MathCAD содержится небольшая группа особых объектов, которые нельзя отнести ни к классу констант, ни к классу переменных, значения которых определены сразу после запуска программы. Их правильнее считать *системными переменными*, имеющими предопределенные системой начальные значения.

Обычные переменные отличаются от системных тем, что они должны быть предварительно *определены* пользователем, т. е. им необходимо хотя бы однажды *присвоить значение*. В качестве *оператора присваивания* используется знак **:=**, тогда как знак = отведен для *вывода значения* константы или переменной.

Если переменной присваивается начальное значение с помощью оператора ***:=***, вызывается нажатием клавиши ***:*** (двоеточие) на клавиатуре, такое присваивание называется *локальным*. До этого присваивания переменная не определена и ее нельзя использовать. Однако с помощью знака ≡ (клавиша **~** на клавиатуре) можно обеспечить *глобальное* присваивание.

MathCAD прочитывает весь документ дважды слева направо и сверху вниз. При первом проходе выполняются все действия, предписанные локальным оператором присваивания (≡), а при втором – производятся действия, предписанные локальным оператором присваивания (**:=**), и отображаются все необходимые результаты вычислений (=).

Существуют также жирный знак равенства **=** (комбинация клавиш ***Ctrl* + =**), который используется, например, как оператор приближенного равенства при решении систем уравнений, и символьный знак равенства → (комбинация клавиш ***Ctrl* + .**).

Дискретные аргументы – особый класс переменных, который в пакете MathCAD зачастую заменяет *управляющие структуры*, называемые циклами (однако полноценной такая замена не является). Эти переменные имеют ряд фиксированных значений, либо целочисленных (1 способ), либо в виде чисел с определенным шагом, меняющихся от начального значения до конечного (2 способ).

1. *Name* := *Nbegin* .. *Nend,*

   где *Name* – имя переменной, *Nbegin* – ее начальное значение, *Nend* – конечное значение, **..** – символ, указывающий на изменение переменной в заданных пределах (вводится клавишей **;**). Если *Nbegin* < *Nend*, то шаг переменной будет равен +1, иначе –1.

2. *Name* := *Nbegin*, (*Nbegin* + *Step*) .. *Nend*

   здесь *Step* – заданный шаг изменения переменной (он должен быть положительным, если *Nbegin* < *Nend*, или отрицательным в обратном случае).

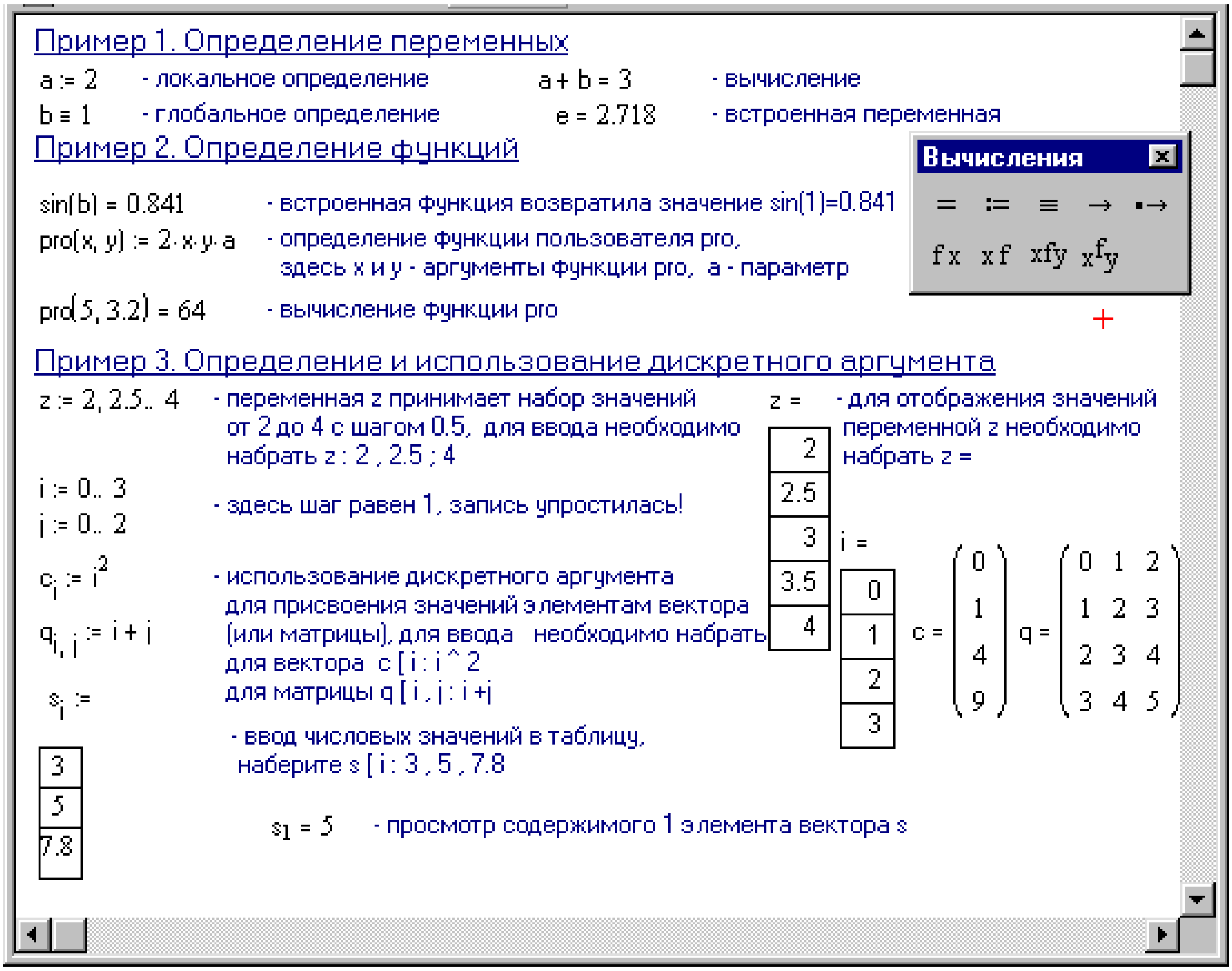


Рис. 2.1. Пример определения переменных, функций и аргументов

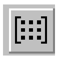
Дискретные аргументы значительно расширяют возможности MathCAD, позволяя выполнять многократные вычисления или циклы с повторяющимися вычислениями, формировать векторы и матрицы.

Массив – имеющая уникальное имя совокупность конечного числа числовых или символьных элементов, упорядоченных некоторым образом и имеющих определенные адреса. В пакете MathCAD используются массивы двух наиболее распространенных типов:

- одномерные (векторы);
- двумерные (матрицы).

Порядковый номер элемента, который является его адресом, называется *индексом*. Индексы могут иметь только целочисленные значения. Они могут

начинаться с нуля или единицы, в соответствии со значением системной переменной *ORIGIN*.
Векторы и матрицы можно задавать различными способами:

- с помощью команды *Вставка => Матрица*, или комбинации клавиш ***Ctrl + M***, или щелчком на кнопке  панели *Матрица*, заполнив массив пустых полей для не слишком больших массивов;
- с использованием дискретного аргумента, когда имеется некоторая явная зависимость для вычисления элементов через их индексы.

Функции

Функция – выражение, согласно которому проводятся некоторые вычисления с *аргументами* и определяется его числовое значение.
Следует особо отметить разницу между *аргументами* и *параметрами* функции. Переменные, указанные в скобках после имени функции, являются ее *аргументами* и заменяются при вычислении функции значениями из скобок. Переменные в правой части определения функции, не указанные скобках в левой части, являются *параметрами* и должны задаваться *до* определения функции.

Главным признаком функции является *возврат значения*, т.е. функция в ответ на обращение к ней по имени с указанием ее аргументов должна возвратить свое значение.

Функции в пакете MathCAD могут быть *встроенные*, т. е. заблаговременно введенные разработчиками, и *определенные пользователем*.
Способы вставки встроенной функции:

1. Выбрать пункт меню *Вставка => Функция*.
2. Нажать комбинацию клавиш ***Ctrl + E***.
3. Щелкнуть на кнопке .

Текстовые фрагменты

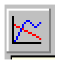
Текстовые фрагменты представляют собой куски текста, которые пользователь хотел бы видеть в своем документе. Существуют два вида текстовых фрагментов:

- *текстовая область* предназначена для небольших кусков текста – подписей, комментариев и т. п. Вставляется с помощью команды *Вставка => Регион текста* или комбинации клавиш ***Shift + "*** (двойная кавычка);
- *текстовый абзац* применяется в том случае, если необходимо работать с абзацами или страницами. Вставляется с помощью комбинации клавиш ***Shift + Enter***.

Графические области

Графические области делятся на три основных типа – двумерные графики, трехмерные графики и импортированные графические образы. Двумерные и трехмерные графики строятся самим MathCAD на основании обработанных данных.

Для создания *декартового графика* необходимо:

1. Установить визир в пустом месте рабочего документа.
2. Выбрать команду *Вставка => График => X-У график*, или нажать комбинацию клавиш ***Shift* + @,** или щелкнуть кнопку  панели *Графики*. Появится шаблон декартового графика.
3. Ввести в средней метке под осью *X* первую независимую переменную, через запятую – вторую и так до 10, например *x*1, *x*2, …
4. Ввести в средней метке слева от вертикальной оси *Y* первую независимую переменную, через запятую – вторую и т. д., например *y*1(*x*1), *y*2(*x*2), …, или соответствующие выражения.
5. Щелкнуть за пределами области графика. После щелчка вне графика Mathcad вычисляет промежуточные значения и строит точки графика (рис. 2.2).

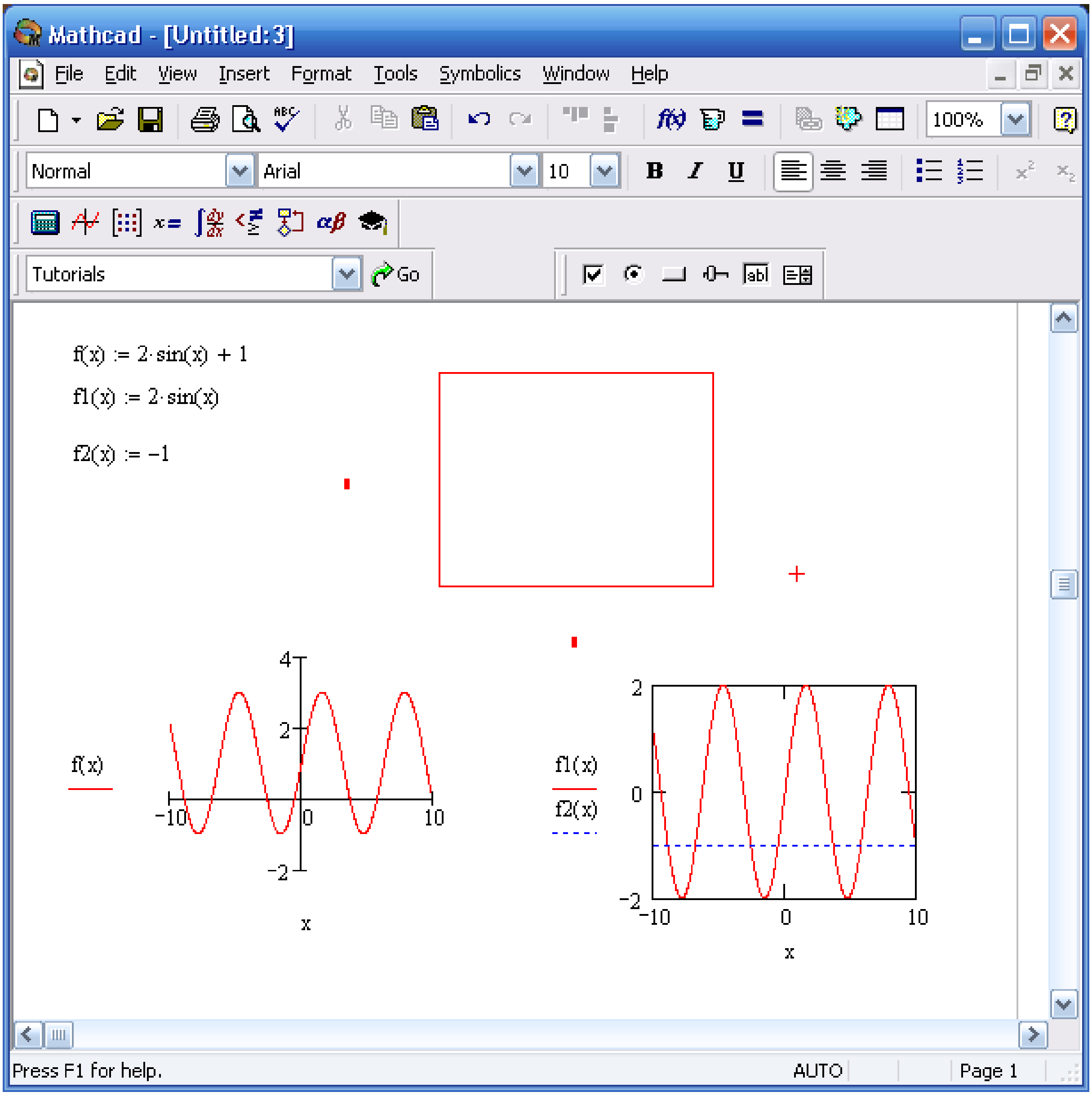
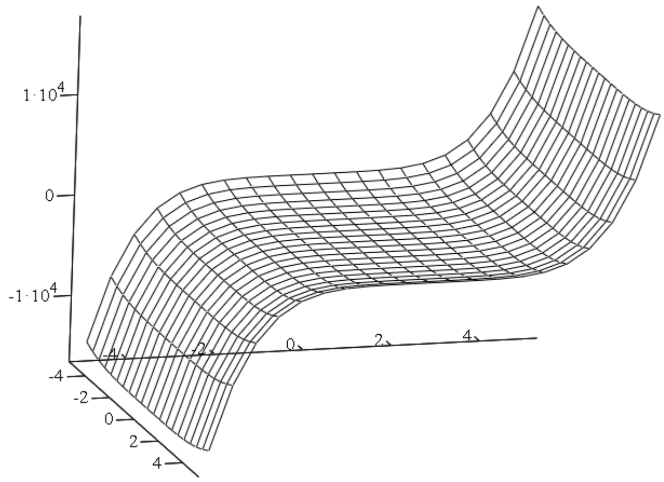


Рис. 2.2. Незаполненный шаблон двухмерного графика и варианты результатов построения графиков одной и двух функций в графической области *X-У Plot*

Для построения графиков нескольких функций необходимо ввести их названия через запятую. По умолчанию *Mathcad* берет интервал по оси x, где $x \in [-10; 10]$. Если необходим другой интервал, то его следует ввести перед областью графика.

Для построения графиков поверхностей используется подпанель *Graph* шаблон *Surface Plot* (рис. 2.3).

$$f(x, y) := 2x^4 + 5y^5$$



f

Рис. 2.3. График поверхности

График поверхности можно вращать в пространстве. Вращение поверхности эквивалентно ее просмотру с разных сторон. Для визуализации этих действий надо поместить указатель мыши в область графика, нажать левую кнопку мыши и, удерживая ее, начать перемещать мышь в некотором направлении. Сразу будет видно, что фигура вместе с осями координат и призмой, в которой она находится, начнет вращаться в ту ли иную сторону. Если двигать скроллинг мыши вперед или назад, то можно удалять или приближать объект.

Трехмерные, или *3D-графики*, отображают функции двух переменных вида $Z(X,\ Y)$. При построении трехмерных графиков в ранних версиях MathCAD поверхность нужно было определить математически (2 способ). Теперь применяют функцию MathCAD *CreateMesh.*

1 способ

$$f(x,y) := \sin\left(x^2 + y^2\right)$$

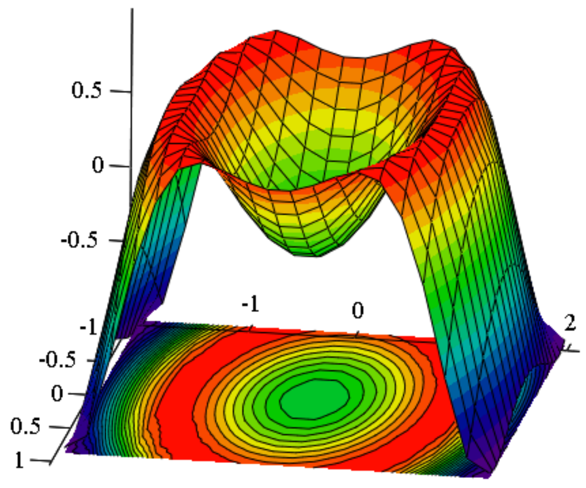
$$G1 := \mathrm{CreateMesh}(f, -1, 1, -2, 2, 20, 20)$$

2 способ

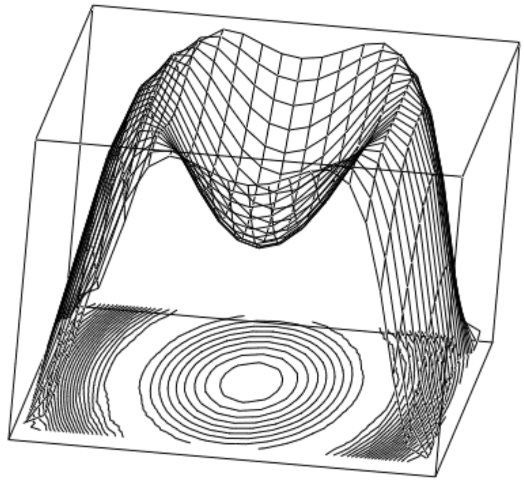
$$i := 0..20 \qquad x_i := -1 + i \cdot 0.1$$

$$j := 0..20 \qquad y_j := -2 + j \cdot 0.2$$

$$G2_{i,j} := f\left(x_i, y_j\right)$$



G1 ,G1



G2 ,G2

Рис. 2.4. Пример трехмерного графика

CreateMesh(*F* (или *G*, или *f*1, *f*2, *f*3), *x*0, *x*1, *y*0, *y*1, *xgrid*, *ygrid*, *fmap*)
Создает сетку на поверхности, определенной функцией *F*. *x*0, *x*1, *y*0, *y*1 – диапазон изменения переменных, *xgrid*, *ygrid* – размеры сетки переменных, *fmap* – функция отображения. Все параметры, за исключением *F*, – факультативные. Функция *CreateMesh* по умолчанию создает сетку на поверхности с диапазоном изменения переменных от -5 до 5 и с сеткой 20*20 точек.
Нередко поверхности и пространственные кривые представляют в виде точек, кружочков или иных фигур. Такой график создается операцией *Вставка => График => 3D-график разброса*, причем поверхность задается параметрически – с помощью трех матриц (*X*, *Y*, *Z*).

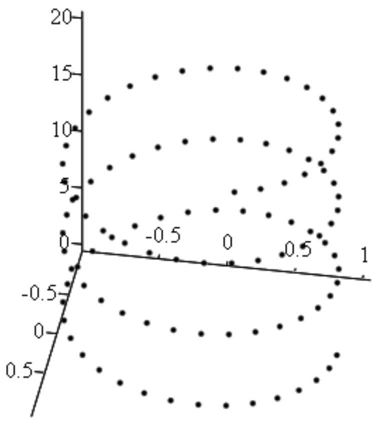
Для определения исходных данных для такого вида графиков используется функция *CreateSpace* .

CreateSpace (*F*, *t*0, *t*1, *tgrid*, *fmap*) – возвращает вложенный массив трех векторов, представляющих *x*-, *y*-, и *z*-координаты пространственной кривой, определенной функцией *F*. *t*0 и *t*1 – диапазон изменения переменной, *tgrid* – размер сетки переменной, *fmap* – функция отображения. Все параметры, за исключением *F*, – факультативные.

1 способ

$$f(t) := \begin{pmatrix} \sin(t) \\ \cos(t) \\ t \end{pmatrix}$$

$$P := \mathrm{CreateSpace}(f, 0, 20, 100)$$



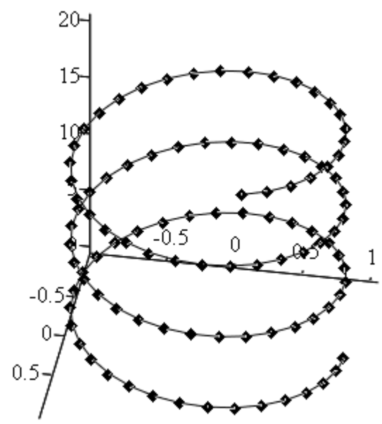
P

2 способ

$$t := 0 .. 100$$

$$x_t := \sin\left(\frac{t}{5}\right) \qquad y_t := \cos\left(\frac{t}{5}\right)$$

$$z_t := \frac{t}{5}$$



(x, y, z)

Рис. 2.5. Пример трехмерного графика разброса

Построение *пересекающихся фигур*

Особый интерес представляет собой возможность построения на одном графике ряда разных фигур или поверхностей с автоматическим учетом их взаимного пересечения. Для этого надо раздельно задать матрицы соответствующих поверхностей и после вывода шаблона 3D-графика перечислить эти матрицы под ним с использованием в качестве разделителя запятой (рис. 2.6).

$$f1(x,y) := \cos\left(x^2 + y^2\right) \qquad f2(x,y) := x^2 + y^2 - 5 \qquad x := 0..20$$

$$y := 0..20$$

$$G1_{x,y} := f1\left(\frac{x-7}{5}, \frac{y-7}{5}\right) \qquad G2_{x,y} := f2\left(\frac{x-10}{5}, \frac{y-10}{5}\right)$$

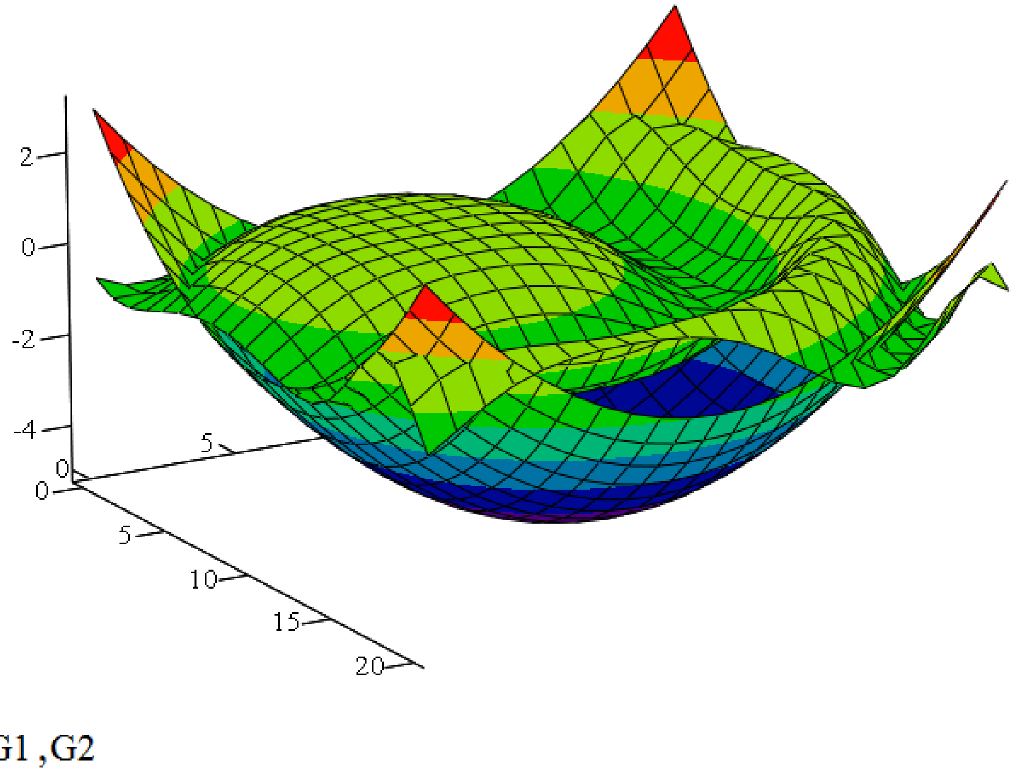


Рис. 2.6. Пример пересечения поверхностей

Создание анимационного клипа

MathCAD имеет встроенную переменную FRAME, чье единственное назначение – управление *анимациями*:

- Создайте объект, чей вид зависит от FRAME.
- Убедитесь, что установлен режим автоматического расчета (*Инструменты => Вычислить => Автоматическое Вычисление*).
- Выберите *Инструменты => Анимация* для вызова одноименного диалогового окна.
- Заключите в выделяющий пунктирный прямоугольник часть рабочего документа, которую нужно анимировать.
- Установите нижние и верхние границы FRAME (поля **От:** и **До:**).
- В поле **Скорость** введите значение скорости воспроизведения (кадров/сек).
- Выберите **Анимация**. Сейчас анимация только создается.
- Сохраните анимацию как AVI файл (**Сохранить как**).
- Воспроизведите сохраненную анимацию *Вид => Воспроизведение*.

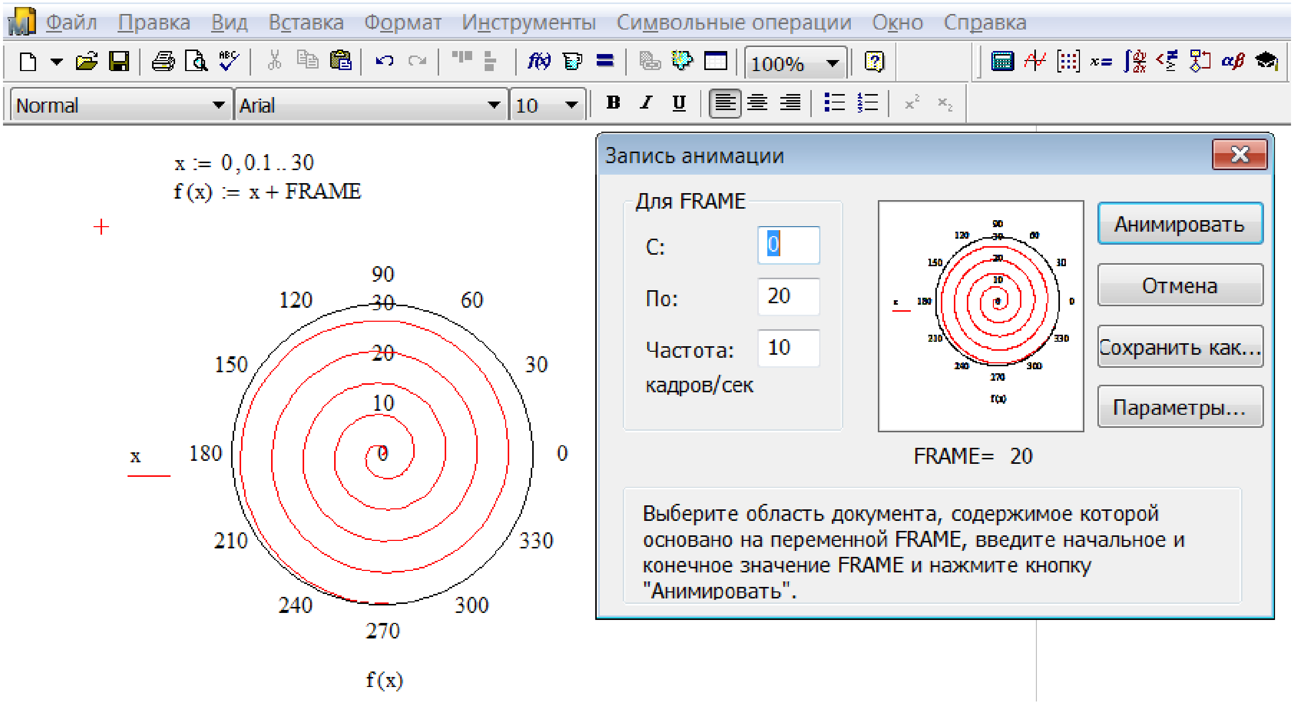


Рис. 2.7. Пример анимации

2.3. *Редактор формул*

Для запуска редактора достаточно установить указатель мыши в любом свободном месте окна редактирования и щелкнуть левой кнопкой. Курсор указывает место, с которого можно начать набор формулы – вычислительного блока. В области ввода формул курсор превращается в синий уголок, указывающий направление и место ввода. Для расширения охваченной уголком области можно использовать клавиши управления курсором и клавишу *<Пробел>*.

Особенности работы *Mathcad* при выполнении простых вычислений:

- *Mathcad* отображает формулы в общепринятом виде. *Mathcad* самостоятельно подбирает размеры дробных черт, скобок и других математических символов пропорционально написанной фразе, части формулы и пр.
- *Mathcad* понимает порядок действий.
- Некоторые комбинированные операторы (например, :=) вводятся одним символом (нажатием на клавишу <: >).
- *Mathcad* вставляет пробелы до и после арифметических операторов.

- Оператор умножения вводится как звездочка, но представляется точкой в середине строки.
- Оператор деления вводится как косая черта, но заменяется горизонтальной линией.
- Целая часть числа от дробной в числах десятичной записи отделяется точкой.
- По умолчанию десятичные числа имеют представление с тремя цифрами после разделительной точки. Для задания большего количества цифр после запятой необходимо изменить формат вывода числа в пункте *Результат…* меню *Формат* (рис. 2.8).
- Числа могут быть представлены в виде обыкновенной дроби – параметр *Дробь* (рис. 2.9).

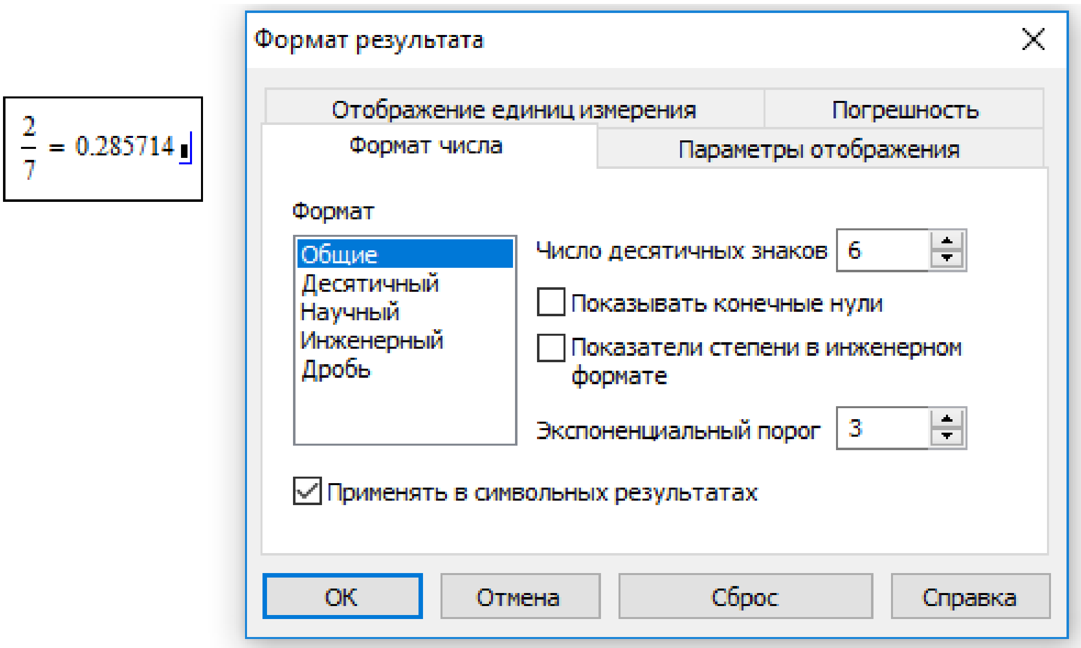


Рис. 2.8. Пример работы с окном Формат результата

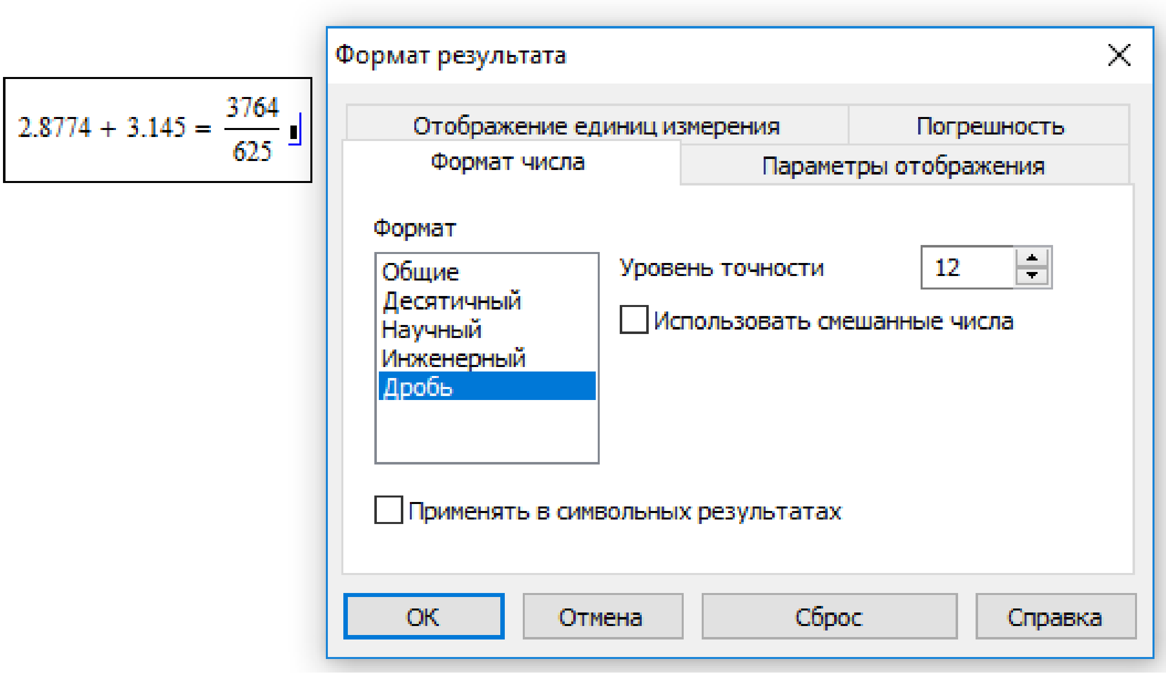


Рис. 2.9. Пример представления числа в виде обыкновенной дроби

- *Mathcad* понимает наиболее распространенные константы (например, число $\pi = 3{,}1415926$, ускорение свободного падения $g = 9{,}8$ м/с и т. д.).
- Математические выражения можно редактировать внутри формального блока, используя для этого курсор ввода и типовые приемы редактирования.
- Выражение на экране можно редактировать, устанавливая курсор в нужном месте и вводя новые символы, формулы, начальные значения переменных.
- После нажатия на клавишу <=> *Mathcad* показывает результат. Как говорилось выше, по умолчанию *Mathcad* отображает результат с тремя цифрами после запятой.

Рабочая область окна представляет собой калькулятор, т. е. формулы можно писать в любой рабочей области. Вычисления в *Mathcad* можно осуществлять с любой точки экрана, надо для этого лишь щелкнуть мышью в любом месте рабочей области.
В *Mathcad* большинство команд собраны на панели инструментов *Математика* (рис. 2.10).

Панель инструментов позволяет быстро вызывать команды главного меню. Каждая кнопка панели инструментов (рис. 2.13) активизируется щелчком мыши по ней. Палитры инструментов можно перемещать по рабочей области в любом направлении. Размеры панелей можно изменять.

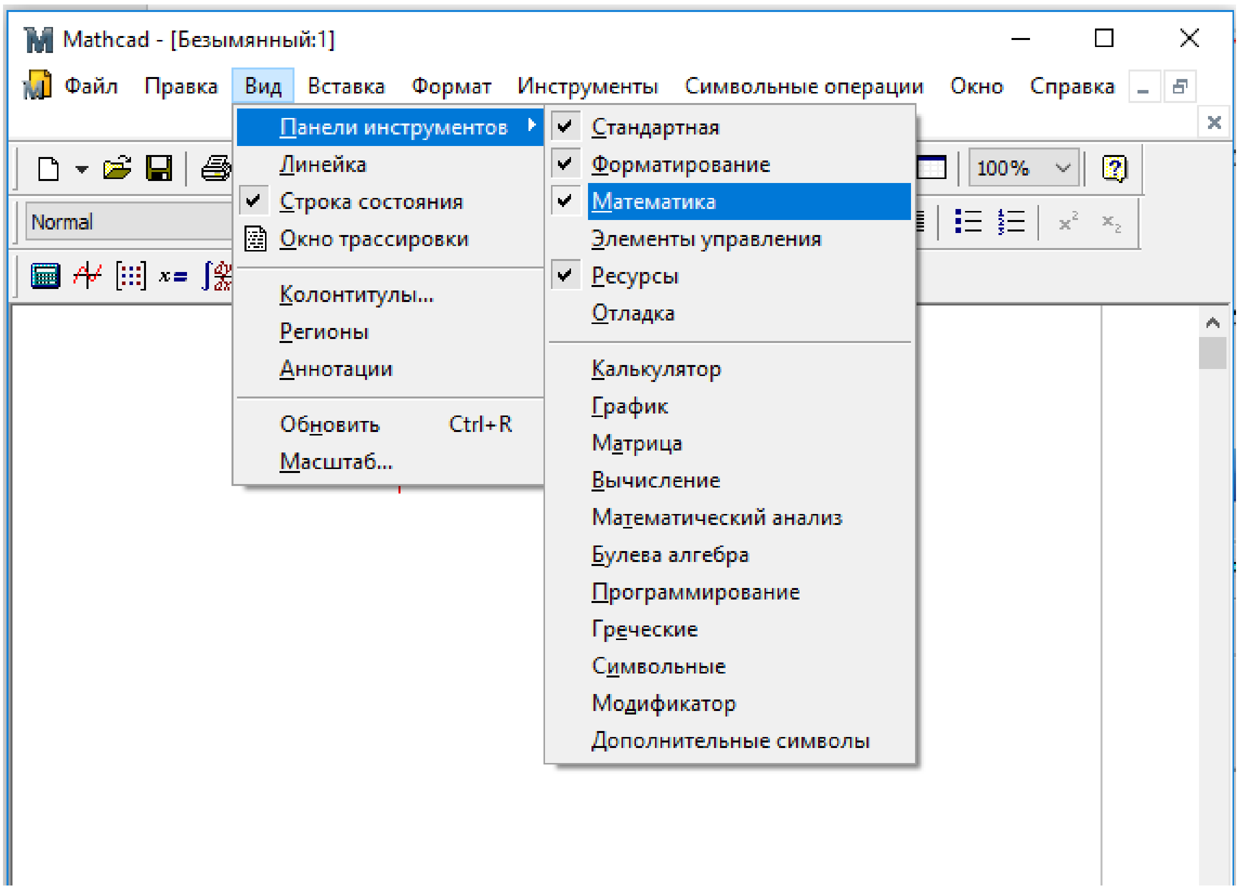


Рис.2.10. Вызов панели инструментов

После нажатия на клавишу <=> получаем результат вычисления интеграла:

$$f(x) := 8\cdot\sqrt{\left(25 - x^2\right)\cdot\left(5 - x^2\right)}$$

$$b := 2$$

$$a := 0$$

$$\int_{\blacksquare}^{\blacksquare} \blacksquare \, d\blacksquare$$

$$\int_{b}^{a} f(x)\, dx = -147.269$$

Рис. 2.11. Шаблон ввода определенного интеграла

В *Mathcad* есть специальные типы переменных, называемые ранжированными. Они имеют множественные значения. Например (рис. 2.12), переменная i представляет собой целые числа от 1 до 4 с шагом 0,5, переменная n принимает значения от 1 до 7 с шагом 1:

$n := 1..7$ $\qquad\qquad$ $i := 1, 1.5..4$

n =

1
2
3
4
5
6
7

i =

1
1.5
2
2.5
3
3.5
4

Рис. 2.12. Пример задания ранжированной переменной

Примечание.

Второй член в задании ранжированной переменной – это сумма ее начального значения и шага. На рис. 2.12 шаг переменной i равен $1{,}5 - 1 = 0{,}5$. Если второй член не указан, шаг по умолчанию принимается равным единице.

Ранжированная переменная может быть использована для той или иной функции или представления значений вектора. Она задает пределы изменения независимой переменной и ее шаг значений при построении графиков.

<u>Формулы с константами</u>

Константы в *Mathcad* представляют собой действительные и комплексные числа. Действительные константы могут иметь знак “ $+$ ” или “ $-$ “. Дробная часть константы отделяется от целой части точкой. Комплексная константа представляется в виде суммы (разности) действительной и мнимой частей. При этом за мнимой частью константы без какого-либо знака операции ставится символ i.

В *Mathcad* имеется несколько зарезервированных имен констант. Это число π, которое вводится с подпанели *Калькулятор* (*Calculator*); основание натурального логарифма, которое вводится клавишей $< e >$; значение 0,01 для вычисления процентов, которое вводится клавишей $<\%>$; значение компьютерной бесконечности (это число 10^{307}), которое вводится кнопкой $\boxed{\infty}$ подпанели *Исчисление* (*Calculus*).

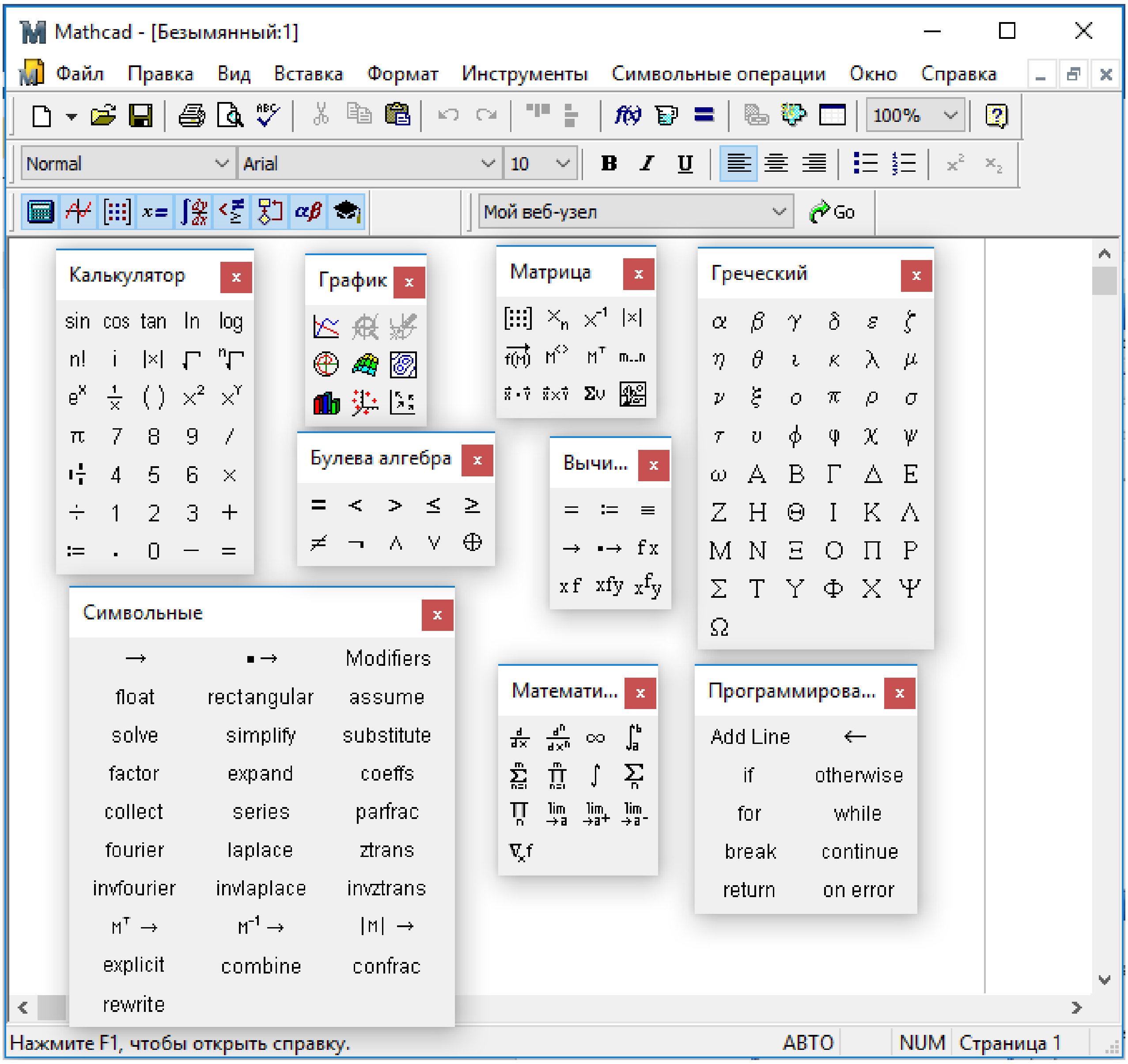


Рис.2.13. Математические палитры

Переменные и формулы с переменными

Имена переменных и функций называются идентификаторами. Идентификаторы в *Mathcad* могут иметь практически любую длину. Они состоят из символов, каждый из которых может быть буквой (в том числе и греческой), а также цифрой. При этом первым символом должна быть буква.

Для получения численного результата необходимо всем переменным из формулы присвоить числовые значения. Присваивания бывают двух видов: локальные и глобальные. Локальное присваивание осуществляется кнопкой *:=* подпанели *Калькулятор* (*Calculator*) или клавишей <:>. Присвоенное значение в документе начинает действовать с момента его записи (слева-направо и сверху-вниз), поэтому необходимо следить, чтобы очередная формула была в строке правее или ниже предыдущей.

Глобальное присваивание действует в пределах всего документа, независимо от места его определения. Глобальное присваивание задается кнопкой ≡ подпанели *Оценка* (*Evaluation*).

Пример цепочки формул с использованием локального (для x) и глобального (для a) присваивания:

$$x := 1 \qquad y := x + 2 \qquad z := y \cdot x + 3 \qquad z = 6$$

$$x := 2 \qquad \mu := \frac{z \cdot x}{a} \qquad \mu := 4$$

$$a \equiv 3$$

Если для одной и той же переменной в документе задано локальное и глобальное присваивание, то локальное присваивание отменяет глобальное присваивание, превратив последнее в локальное.

Если для одной и той же переменной в документе глобальное присваивание задается несколько раз, то действует всегда самое последнее из них.

В *Mathcad* наиболее часто встречающиеся функции вынесены на подпанели *Калькулятор* (*Calculator*) и *Исчисление* (*Calculus*). К большинству же функций можно обратиться с помощью кнопки $\boxed{f(x)}$ стандартной панели инструментов. При этом ввод функции осуществляется с использованием диалогового окна *Вставить Функцию* (*Insert Function*). Диалоговое окно (рис. 2.14) содержит два списка. С помощью левого списка *Категория функции* (*Function Category*) осуществляется выбор раздела, к которому относится извлекаемая функция. Если есть затруднения в выборе раздела, то можно воспользоваться элементом списка *All*, по которому извлекаются все стандартные функции *Mathcad* в алфавитном порядке. В правом списке *Имя функции* (*Function Name*) содержится в алфавитном порядке перечень функций выбранного раздела:

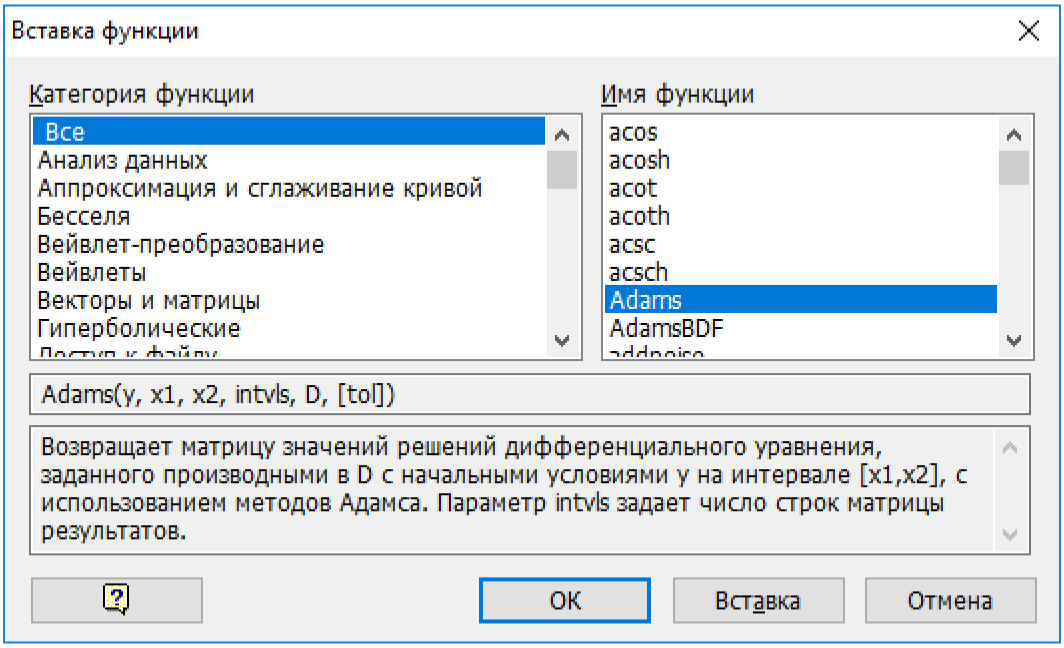


Рис.2.14. Диалоговое окно ввода стандартных функций

Стандартные функции можно вставлять в *Mathcad* – документ и без использования диалогового окна *Вставить Функцию* (*Insert Function*). Для этого в нужном месте документа набирают имя функции со списком аргументов в круглых скобках.

Собственные функции пользователя

Помимо широкого набора стандартных функций, в *Mathcad* возможно определение собственных функций пользователя. В простейшем случае функция может быть определена формулой пользователя. Функция определяется следующим образом:

имя_функции(аргументы) := формула,

где *имя_функции* – любой уникальный для данного документа идентификатор; *аргументы* – список аргументов функции через запятую; *формула* – любая формула с использованием констант, стандартных функций и функций пользователя. В *Mathcad* имя функции должно быть уникальным идентификатором среди всех других идентификаторов.

Формулы с векторами и матрицами

Работа с векторами и матрицами осуществляется по аналогии с обычными константами и переменными, т. е. векторы и матрицы можно использовать в операциях присваивания и в формулах. Вектор и матрица, как и переменная, задаются своим идентификатором.

Для ввода числовых значений матрицы используют кнопку $\begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix}$ подпанели *Матрица* (*Matrix*). При этом вызывается диалоговое окно *Вставить Матрицу* (*Insert Matrix*), в котором необходимо задать размер матрицы: число ее строк и число столбцов.

На месте меток шаблона необходимо ввести константы или формулы, как это показано в примере:

$a := 2 \qquad b := 4$

$$M := \begin{pmatrix} a+b & 1 & 2 \\ 3 & a*b & 4 \\ 5 & 6 & a^2+b^2 \end{pmatrix} \qquad M = \begin{pmatrix} 6 & 1 & 2 \\ 3 & 8 & 4 \\ 5 & 6 & 20 \end{pmatrix}$$

При работе с векторами и матрицами необходимо помнить, что нумерация столбцов и строк в них (по умолчанию) начинается с нуля.

Для задания нумерации с единицы необходимо предварительно локальным или глобальным присваиванием изменить значение системной переменной *origin* на 1:

$$m := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad \mathrm{ORIGIN} := 1 \quad m^{\langle 1 \rangle} = \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix}$$

Помимо операций над матрицами в *Mathcad* возможны операции над элементами матриц по аналогии с переменными. Элементы матрицы определяются с помощью индексированных переменных. Индексами здесь могут быть целые константы и неотрицательные переменные. При использовании в качестве индекса идентификатора i необходимо помнить, что этот идентификатор используется также для указания мнимой единицы комплексного числа, поэтому в выражениях типа $2i$ не надо забывать указывать в явном виде операцию умножения (в противном случае $2i$ будет восприниматься как мнимая часть комплексного числа).

Нестандартные операции над элементами матриц выполняются с использованием всех операций и функций непосредственно над элементами матриц. При этом часто используют ранжированные переменные, которые определяются следующим образом: вводится имя переменной и знак присваивания. Затем кнопкой $m..n$ подпанели *Матрица* вводится шаблон и на месте появившихся меток вводится начальное и конечное значение переменной, при этом шаг изменения переменной будет равен 1.

При определении ранжированной переменной можно задавать любой шаг ее изменения, отличный от единицы. Для этого кнопкой $m..n$ подпанели *Матрица* (*Matrix*) вводится шаблон $i := \blacksquare..\blacksquare$, в котором после первой метки вводится запятая: $i := \blacksquare, \blacksquare..\blacksquare$.

На месте первой метки нового шаблона вводится первое значение ранжированной переменной, на месте второй метки – второе значение ранжированной переменной, а на месте третьей метки – последнее значение ранжированной переменной.

Операция векторизации позволяет поэлементно оперировать векторами и матрицами одинакового размера. Эта операция проводится с помощью клавиши $\boxed{\overrightarrow{f(M)}}$ подпанели *Матрица* (*Matrix*). Пусть, например, даны векторы $\vec{a} := (2, 4, 6)$, $\vec{b} := (2, 8, 3)$, $\vec{c} := (3, 4, 5)$ и требуется определить вектор $\vec{d}$, i-я координата $\overrightarrow{d_i}$ которого будет равна $\frac{\overrightarrow{a_i}}{\overrightarrow{b_i}} \overrightarrow{c_i}$, где a_i, b_i, c_i – соответственно i-е координаты векторов $\vec{a}, \vec{b}, \vec{c}$.

Для этого в нужном месте рабочего листа введите выражение $d := \frac{a}{b} \cdot c$ и синим курсором ввода выделите выражение, стоящее справа от знака присваивания: $\boxed{d := \frac{a}{b} \cdot c}$. После щелчка по кнопке $\boxed{\overrightarrow{f(M)}}$ произойдет векторизация $d := \overrightarrow{\left(\frac{a}{b}\cdot c\right)}$, в результате которой будет получен искомый вектор $\vec{d} = (3, 2, 10)$.

Рассмотрим другой пример. Пусть дана матрица:

$$M := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \qquad M^2 = \begin{pmatrix} 30 & 36 & 42 \\ 66 & 81 & 96 \\ 102 & 126 & 150 \end{pmatrix}$$

$$\overrightarrow{M^2} = \begin{pmatrix} 1 & 4 & 9 \\ 16 & 25 & 36 \\ 49 & 64 & 81 \end{pmatrix}$$

Отсюда видно, что невекторизованный квадрат воспринимается *Mathcad* как матричное умножение, а операция $\overrightarrow{M^2}$ определяет поэлементное умножение.

В *Mathcad* допускается использование так называемых "гнездовых" матриц, т.е. матриц, элементами которых являются также матрицы.

Например:

$$a := \begin{pmatrix} 1 & 2 & 3 \end{pmatrix} \quad b := \begin{pmatrix} 1 \\ 2 \end{pmatrix} \quad c := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

$$m := \begin{pmatrix} a & b & c \end{pmatrix} \quad m = \begin{pmatrix} \{1,3\} & \{2,1\} & \{2,2\} \end{pmatrix}$$

Это означает, что элемент m_0 является, в свою очередь, матрицей размера 1×3, элемент m_1 – матрицей размера 2×1, а элемент m_2 – матрицей размера 2×2. Отобразить все подэлементы элемента m можно так:

$$m_2 = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

Гнездовые матрицы не допускают многих матричных операций, например, операцию обращения:

$$v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad s := \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \quad t := \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

$$M := \begin{pmatrix} v & s & t \end{pmatrix}$$

$$M = \begin{pmatrix} \{3,1\} & \{3,1\} & \{3,1\} \end{pmatrix}$$

$$M^{-1} = \blacksquare$$

В этом примере обратная матрица не вычисляется, о чем свидетельствует отсутствие решения и выделенный на экране красным цветом идентификатор матрицы M. Однако операции сложения и вычитания для гнездовых матриц одинакового размера допускаются.

Обработка формул в символьном виде

Mathcad имеет набор команд для преобразования и упрощения символьных выражений. Для их отображения на панели *Математика* откройте подпанель *Символьные* (*Symbolic*) (рис. 2.15).

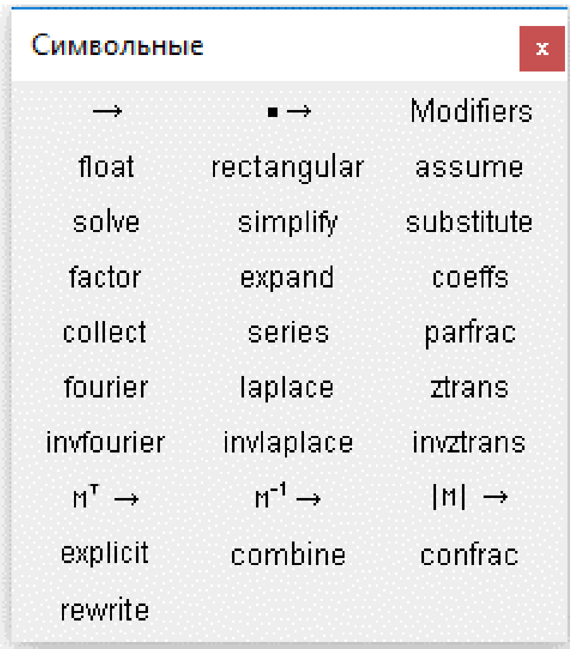


Рис. 2.15. Подпанель Символьные для выполнения символьных операций

Команды подпанели *Символьные*:

1. Простейшим примером символьных вычислений является вычисление неопределенных интегралов. Для этого с помощью кнопки [∫] подпанели *Исчисление* (*Calculus*) введите требуемый неопределенный интеграл. Затем кнопкой [→] подпанели *Символьные* введите знак символьного вывода.

   Пример вычисления в символьном виде неопределенного интеграла:

$$\int x\,dx \to \frac{1}{2}x^2.$$

2. Команда *simplify* используется для упрощения выражений. В нужном месте рабочего листа введите формулу, выделите всю формулу и щелкните $\boxed{\text{simplify}}$ кнопкой подпанели *Символьные* (*Symbolic*); справа от знака символьного вывода появится выражение в упрощенном виде:

$$\frac{a^3-b^3}{a-b}-ab+\left(\frac{b}{3}\right)^2 \ \text{simplify} \ \to a^2+\frac{10}{9}b^2.$$

3. Команда *expand* предназначена для представления формулы в развернутом виде (то есть в некотором смысле она противоположна команде *simplify)* относительно выражения, записанного на месте метки:

$$\sin 5x \ \text{expand}, x \to 16\sin x \cos x^4 - 12\sin x\cos x^2 + \sin x$$
$$\sin 5x \ \text{expand}, 5x \to \sin 5x.$$

4. Команда *factor* предназначена для разложения целых чисел или выражений на множители:

$$a^3-6a^2+9a \quad \text{factor} \quad \to \mathrm{a}(\mathrm{a}-3)^2$$
$$a^2-b^2 \quad \text{factor} \quad \to (\mathrm{a}-\mathrm{b})(\mathrm{a}+\mathrm{b}).$$

5. Команда *collect* разлагает формулу по степеням переменной, указанной в этой команде на месте метки:

$$(a+b)^5 \quad \text{collect}, a$$
$$\to a^5+5ba^4+10b^2a^3+10b^3a^2+5b^4a+b^5$$
$$(x-a)(x-b)(x-c) \ \text{collect}, x$$
$$\to x^3+(-a-b-c)x^2+[ab-(-a-b)c]x$$
$$-abc.$$

6. Команда *coeffs* используется для вычисления коэффициентов полинома относительно переменной, указанной в команде на месте метки:

$$(4x^3 + 3x^2)(x + 1) \quad \text{coeffs, x} \rightarrow \begin{pmatrix} 0 \\ 0 \\ 3 \\ 7 \\ 4 \end{pmatrix}$$

$$(x - b)(x + b) \;\; \text{coeffs, x} \rightarrow \begin{pmatrix} -b^2 \\ 0 \\ 1 \end{pmatrix}.$$

7. Команда *solve* позволяет решить уравнение или неравенство с нулевой правой частью относительно переменной, указанной в этой команде на месте метки:

$$x^2 + ax + b \;\; \textit{solve, x} \rightarrow \begin{bmatrix} -\frac{1}{2}a + \frac{1}{2}(a^2 - 4b)^{\frac{1}{2}} \\ -\frac{1}{2}a - \frac{1}{2}(a^2 - 4b)^{\frac{1}{2}} \end{bmatrix} x^2 + ax + b \quad \text{solve, x} \rightarrow$$

$$\begin{bmatrix} -\frac{1}{2}a + \frac{1}{2}(a^2 - 4b)^{\frac{1}{2}} \\ -\frac{1}{2}a - \frac{1}{2}(a^2 - 4b)^{\frac{1}{2}} \end{bmatrix}$$

$$e^x - a \quad \text{solve, x} \rightarrow \ln a.$$

Команда *solve* позволяет также решать системы линейных и нелинейных уравнений с нулевой правой частью:

$$\begin{pmatrix} x^2 + y^2 - 7 \\ 2x - y \end{pmatrix} \quad \text{solve, x, y} \rightarrow \begin{pmatrix} \frac{1}{2}35^{\frac{1}{2}} & \frac{2}{5}35^{\frac{1}{2}} \\ -\frac{1}{2}35^{\frac{1}{2}} & -\frac{2}{5}35^{\frac{1}{2}} \end{pmatrix}.$$

8. Команда *substitute* используется для подстановки значений переменных в формулу с последующим вычислением этой формулы:

$$ax^2 + bx + \mathrm{c} \quad \text{substitute, x} = 5 \;\rightarrow\; 25\mathrm{a} + 5\mathrm{b} + \mathrm{c}.$$

Задание функций и формул с помощью программных модулей

Формирование программных модулей осуществляется в *Mathcad* с помощью подпанели *Программирование* (*Programming*) (рис. 2.16).

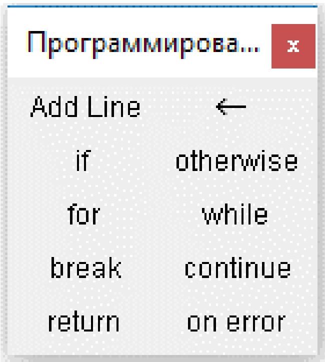


Рис.2.16. Подпанель Программирование

Программный модуль обозначается в *Mathcad* вертикальной чертой, справа от которой последовательно записываются операторы языка программирования. Чтобы создать программный модуль, необходимо:

- ✓ ввести часть выражения, которая будет находиться слева от знака присваивания, и сам знак присваивания;
- ✓ нажать на подпанели *Программирование* кнопку *Add Line* (*Добавить линию*);
- ✓ если приблизительно известно, сколько строк кода будет содержать программа, можно создать нужное количество линий повторным нажатием кнопки *Add Line* соответствующее число раз;
- ✓ ввести желаемый программный код, используя программные операторы.

Пример:

$$a := 3$$

$$\left| \begin{array}{l} x \leftarrow a + 4 \\ x^2 \end{array} \right.$$

В этой цепочке формул вычисляется выражение $(a + 4)^2$. Как видно из примера, программный модуль ограничивается слева вертикальной линией. Внутри программного модуля могут присутствовать внешние (a) и внутренние (x) переменные. В программном модуле значения внешних переменных определяются в соответствии с общими правилами операций локального и глобального присваивания (значение 3 для переменной a). Внутренняя переменная программного модуля определяется с момента присваивания ей числового значения операцией внутреннего присваивания (кнопка ← подпанели *Программирование*). Если идентификаторы внутренней и внешней переменной совпадают, то в пределах программного модуля действует внутренняя переменная. Результатом вычисления программного модуля считается последняя выполняемая в модуле формула (в данном примере это x^2).

В отличие от правил записи формул на рабочем листе *Mathcad* – документа, внутри программного модуля в одной строке можно записать только один оператор или формулу.

Очень часто программные модули используются для определения функций пользователя. В конце программного модуля должна быть указана формула, являющаяся результатом вычисления функции.

Пример:

$$f(x) := \left| \begin{array}{l} a \leftarrow (x+1)\cdot 2 \\ a^2 + x \end{array} \right.$$

$$f(0.5) = 9.5$$

Операторы программирования

Условный оператор *if* предназначен для выполнения вычислений в зависимости от условия. При вызове оператора *if* появляется шаблон с двумя метками. На месте правой метки вводится логическое выражение. На месте левой метки вводится или формула, или операция внутреннего присваивания для указанной переменной.

Оператор *otherwise* используется совместно с одним или несколькими условными операторами *if* и указывает на выражение, которое будет выполняться, если ни одно из условий не оказалось истинным.

Пример:

$$f(x) := \left| \begin{array}{l} \text{if } \ x < 5 \\ \quad \left| \begin{array}{l} y \leftarrow 2 \\ x \leftarrow x^2 \end{array} \right. \\ \text{otherwise} \\ \quad \left| \begin{array}{l} y \leftarrow 4 \\ x \leftarrow x^3 \end{array} \right. \\ x\cdot y \end{array} \right.$$

$$f(3) = 18$$

$$f(8) = 2.048\times 10^3$$

В *Mathcad* имеются два оператора цикла: *for* и *while*. Первый из них дает возможность организовать цикл по некоторой переменной, заставляя ее пробегать некоторый диапазон значений. Второй создает цикл с выходом из него по некоторому логическому условию.

Для ввода оператора *for* необходимо выполнить следующие действия:

- выбрать кнопку *for* подпанели *Программирование*. На экране появятся поля ввода, изображенные на рис. 2.17.

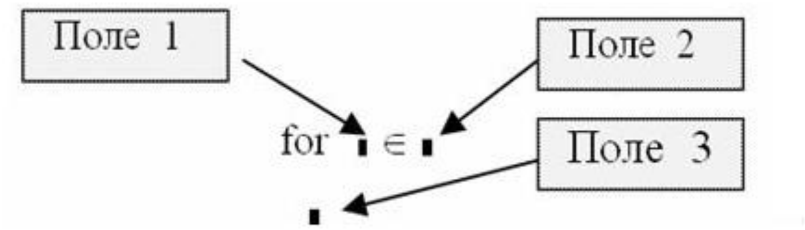


Рис.2.17. Поля оператора цикла for

- в поле ввода 1 ввести имя переменной, являющейся параметром цикла;
- в поле 2 – закон изменения параметра цикла, используя для этого описание дискретной переменной или описание массива;
- в поле 3 – операторы, составляющие тело цикла. Если одной строки недостаточно, то дополнительные поля ввода (дополнительные строки) создаются щелчком по кнопке *Add line* подпанели *Программирование*, и тогда слева от тела цикла появляется вертикальная черта, охватывающая тело цикла.

Пример реализации:

$$\mathrm{form}(n) := \left| \begin{array}{l} \text{for } i \in 1..n \\ \quad z_i \leftarrow \dfrac{1}{i^2+1} \\ z \end{array} \right.$$

$$\mathrm{form}(5) = \begin{pmatrix} 0 \\ 0.5 \\ 0.2 \\ 0.1 \\ 0.059 \\ 0.038 \end{pmatrix}$$

При программировании итерационных циклов используется оператор цикла *while*. Для этого необходимо:

- выбрать кнопку *while* подпанели *Программирование*. На экране появляются элементы, показанные на рис. 2.18:

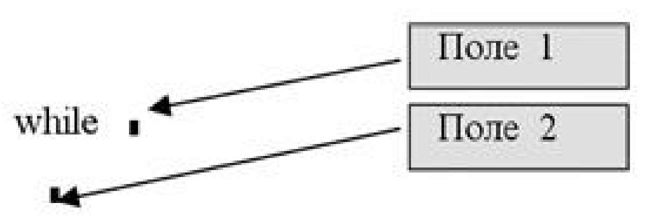


Рис. 2.18. Структура оператора цикла while

- в поле 1 ввести условие выполнения цикла;
- в поле 2 ввести операторы тела цикла. В теле цикла должны присутствовать операторы, которые могут изменить значение условия цикла, иначе цикл будет продолжаться бесконечно.

Оператор цикла *while* выполняется следующим образом: обнаружив оператор *while*, *Mathcad* проверяет указанное в операторе условие. Если оно равно 1 (т.е. выполняется), то выполняется тело цикла, и снова проверяется условие. Если условие принимает значение 0, то цикл заканчивается.

Пример реализации:

$$sh(a,d) := \left| \begin{array}{l} x \leftarrow a \\ \text{while } \left|x^2 - a\right| > d \\ \quad \left| \begin{array}{l} y \leftarrow x \\ x \leftarrow \dfrac{y + \dfrac{a}{y}}{2} \end{array} \right. \\ x \end{array} \right.$$

$$sh(9, 0.1) = 3$$

Организация итерационного цикла с помощью оператора *while* без дополнительных средств контроля может привести к зацикливанию. Поэтому в *Mathcad* имеется специальный оператор *break*, который позволяет выйти из цикла или приостановить исполнение программы при выполнении заданного в операторе *break* условия.

Оператор *break* используется в левом поле ввода условного оператор *if*, а в правом размещается условие, при выполнении которого происходит прекращение работы цикла или программы. Поэтому первоначально вводится оператор *if*, а затем заполняются поля этого оператора.

Оператор *continue* обычно используется для продолжения выполнения цикла возвратом в начало тела цикла.

Оператор *return* прерывает выполнение модуля и возвращает значение операнда, стоящего в поле 1 (рис. 2.19).

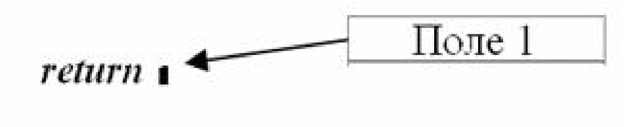


Рис. 2.19. Структура оператора return

Оператор *on error* является обработчиком возникающих при выполнении тех или иных вычислений ошибок и записывается в виде:

< *конструкция 1* > on error < *конструкция 2* >

Если при выполнении <*конструкция 2*> возникает ошибка, то выполняется <*конструкция 1*>. Если ошибка не возникает, то выполняется <*конструкция 2*>.

Справочная система в Mathcad (меню Help)

В пункте *Help* (*Справка*) главного меню *Math* есть команды, обеспечивающие доступ к справочной информации (рис. 2.20).

Основные разделы пункта:

- *Mathcad Help* (*Справка по Mathcad*) – система справок по всем вопросам работы в *Mathcad*.

  Помощь в *Mathcad* является контекстной. Ее можно вызвать, нажав в любой момент работы с программой клавишу <*Fl*>. Откроется окно справочной системы *Mathcad*. Содержание окна будет зависеть от того, где вы находились в момент вызова помощи. Например, при вызове справки по функции *solve* появится окно *Mathcad Help*, открытое на функции *solve* (рис. 2.21).

  Окно справочной системы состоит из двух частей: слева отображается список статей, справа – их содержание.

  Справочная система насыщена гиперссылками, которые находятся в тексте статей и обеспечивают переход от одной статьи к другой. Для вывода большинства статей иногда необходимо несколько переходов по гиперссылкам.

  В конце почти каждой статьи есть кнопка *Related Topics*. Щелчок по этой кнопке открывает перечень статей, близких по тематике, связанных гиперссылками с текущей статьей.

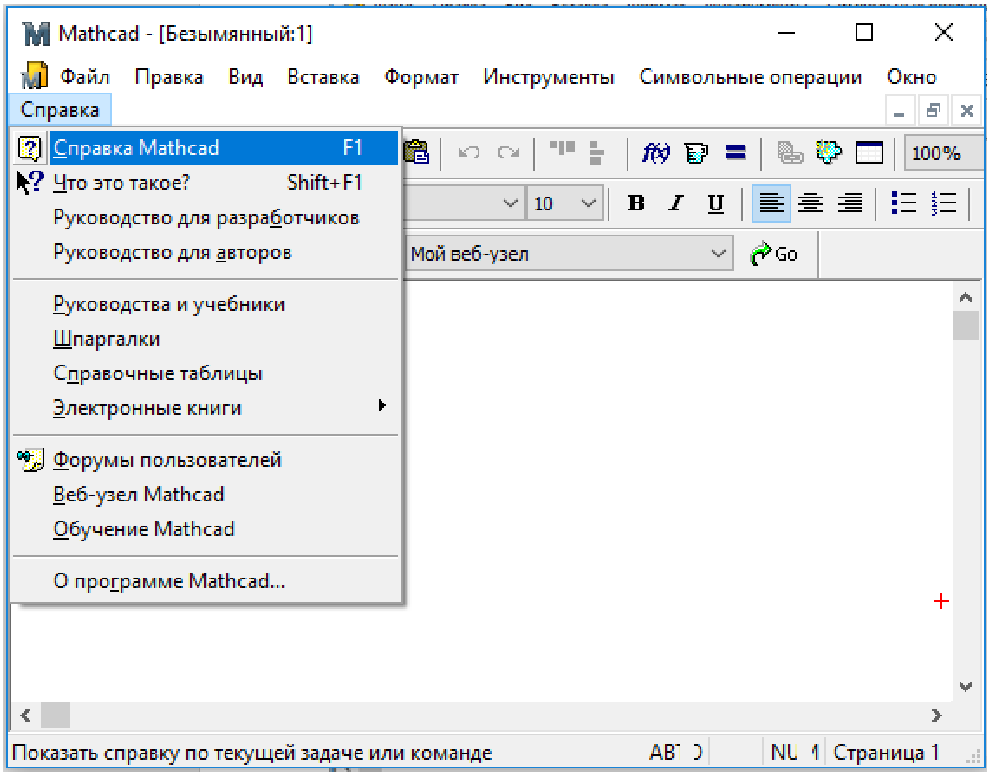


Рис. 2.20. Вызов справки в Mathcad

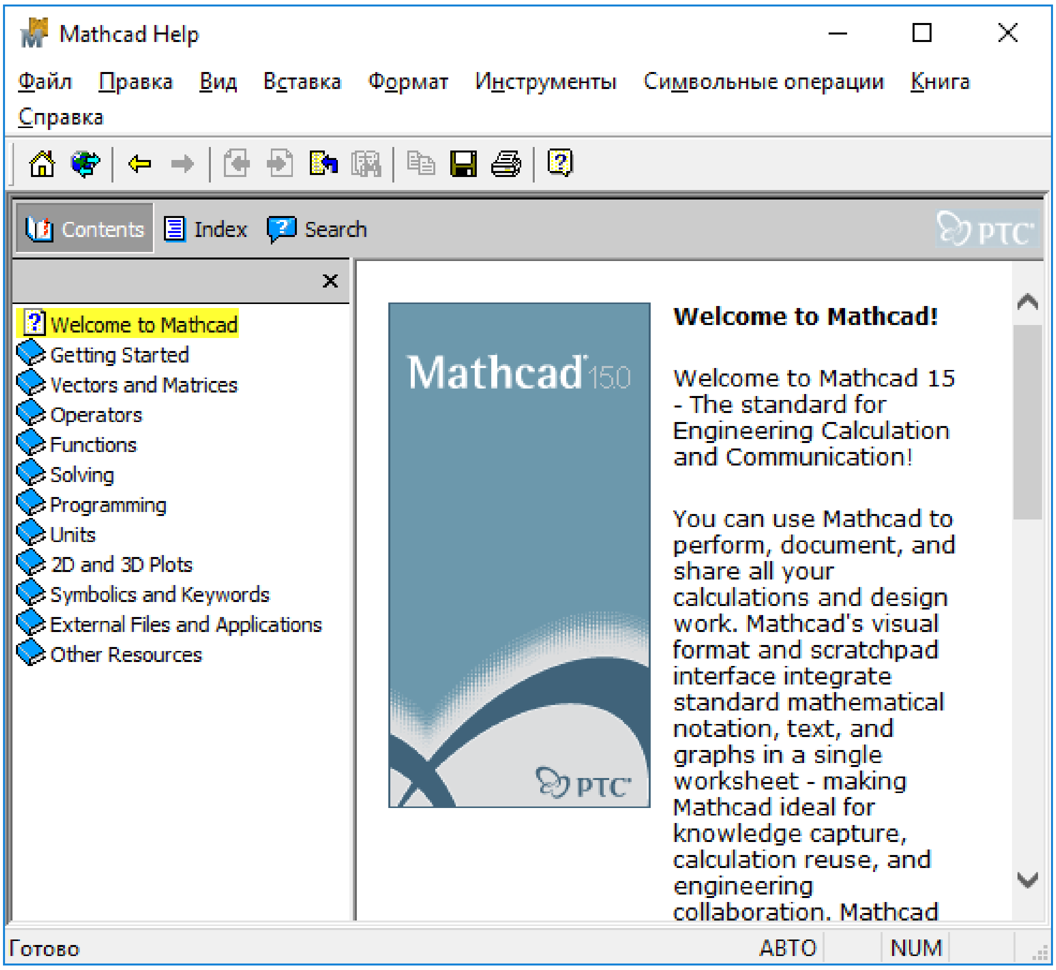


Рис. 2.21. Окно справочной системы Mathcad

- *Developer's Reference* (*Справка для разработчиков*) – дополнительные справки для разработчиков приложений *Mathcad*;
- *Author's Reference* (*Справка для авторов*) – дополнительные главы справки для авторов электронных книг;
- *Tutorials* (*Учебники*) – информация о возможностях системы и основные приемы работы с *Mathcad*.

  В этом разделе пользователь может найти основные сведения по текущей версии *Mathcad*.

  В кратком обзоре рассказывается о новинках текущей версии и перечисляются основные возможности *Mathcad*, приводится большое количество примеров.
- *QuickSheets* и *Reference Tables* (*Шпаргалки и справочные таблицы*) – сборник примеров решения различных задач в *Mathcad*.

  В этих разделах сосредоточено большое количество примеров из математики и физики с использованием встроенных функций, графических возможностей и других средств *Mathcad*.

3. Задачи линейного программирования

3.1. Постановка задачи линейного программирования

Линейное программирование – область математики, разрабатывающая теорию и численные методы решения задач нахождения экстремума (максимума или минимума) линейной функции многих переменных при наличии линейных ограничений, т.е. равенств или неравенств, связывающих эти переменные.

Методы линейного программирования применяют к практическим задачам, в которых:

1) необходимо выбрать наилучшее решение (оптимальный план) из множества возможных;
2) решение можно выразить как набор значений некоторых переменных величин;
3) ограничения, накладываемые на допустимые решения специфическими условиями задачи, формулируются в виде линейных уравнений или неравенств;
4) цель выражается в форме линейной функции основных переменных. Значения целевой функции, позволяя сопоставлять различные решения, служат критерием качества решения.

Для практического решения экономической задачи математическими методами прежде всего ее следует записать с помощью математических выражений: уравнений, неравенств и т.п., т.е. составить экономико-математическую модель.

Исходя из отмеченных выше особенностей задач линейного программирования, можно наметить следующую общую схему формирования модели:

1) выбор некоторого числа переменных величин, заданием числовых значений которых однозначно определяется одно из возможных состояний исследуемого явления;
2) выражение взаимосвязей, присущих исследуемому явлению, в виде математических соотношений (уравнений, неравенств). Эти соотношения образуют систему ограничений задачи;
3) количественное выражение выбранного критерия оптимальности в форме целевой функции;
4) математическая формулировка задачи как задачи отыскания экстремума целевой функции при условии выполнения ограничений, накладываемых на переменные.

Временем рождения линейного программирования принято считать 1939г., когда была напечатана брошюра Л.В. Канторовича «Математические методы организации и планирования производства». Поскольку методы, изложенные Л.В. Канторовичем, были мало пригодны для ручного счета, а быстродействующих вычислительных машин тогда не существовало, то эта работа осталась почти не замеченной.

Свое второе рождение линейное программирование получило в начале пятидесятых годов с появлением ЭВМ. Тогда началось всеобщее увлечение линейным программированием, вызвавшее в свою очередь развитие других разделов математического программирования. В 1975 году академик Л.В. Канторович и американец профессор Т. Купманс получили Нобелевскую премию по экономическим наукам за «вклад в разработку теории и оптимального использования ресурсов в экономике».

Американский экономист Т. Купманс в течение многих лет привлекал внимание математиков к ряду задач, связанных с военной тематикой. Он активно способствовал тому, чтобы был организован математический коллектив для разработки этих проблем. В итоге было осознано, что надо научиться решать задачи о нахождении экстремумов линейных функций на многогранниках, задаваемых линейными неравенствами.

По предложению Т. Купманса этот раздел математики получил название линейного программирования.

Позже (в 1947 году) американский математик А. Данциг разработал весьма эффективный конкретный метод численного решения задач линейного программирования (он получил название симплекс-метода). Идеи линейного программирования в течение пяти – шести лет получили грандиозное распространение в мире, и имена Т. Купманса и А. Данцига стали повсюду широко известны.

Линейное программирование послужило основой для разработки других математических методов исследования операций, например целочисленного, стохастического и нелинейного программирования. Важно и то, что после трех десятилетий глубоких разработок, практической реализации и критического анализа результатов применение методов линейного программирования привело к значительным успехам в решении широкого круга задач, относящихся к таким сферам, как промышленное производство, военное дело, сельское хозяйство, экономические исследования, транспорт, здравоохранение и даже психология и социальные науки.

Линейное программирование – один из первых и наиболее подробно изученных разделов математического программирования.

Линейное программирование представляет собой теоретический аппарат модельного исследования, направленного на отыскание наилучшего способа распределения ограниченных ресурсов по нескольким взаимосвязанным по цели и использованию ресурсов, по видам производственной деятельности

Задачами линейного программирования называются задачи, в которых линейны как целевая функция, так и ограничения в виде равенств и неравенств, а сами переменные неотрицательны.

В свою очередь в линейном программировании существуют классы задач, структура которых позволяет создать специальные методы их решения, выгодно отличающиеся от методов решения задач общего характера. Так, в линейном программировании появился раздел транспортных задач, блочного программирования и др.

3.2. Основная задача линейного программирования. Построение математических моделей в виде задач линейного программирования

Задачей линейного программирования (ЗЛП) называется задача отыскания экстремума (максимума или минимума) линейной функции от нескольких переменных при линейных ограничениях на эти переменные.

В общем виде задача линейного программирования состоит в максимизации (или минимизации) линейной функции

$$F = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

при условии, что переменные x_i имеют линейные ограничения вида

$$\begin{gathered} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ \leq \ (=, \geq)\ b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ \leq \ (=, \geq)\ b_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ \leq \ (=, \geq)\ b_m, \\ x_i \ \geq 0, i = 1, \ldots n. \end{gathered}$$

Значения b_i, c_j, a_{ij} предполагаются известными.

Пример. Найти максимальное значение функции

$$f(x_1, x_2) = 10x_1 + 12x_2$$

при следующих ограничениях на переменные x_1 и x_2:

$$\begin{cases} 9x_1 + 7x_2 \leq 18, \\ 4x_1 + 5x_2 \leq 11, \\ x_1 \geq 0, x_2 \geq 0. \end{cases}$$

Приведенная задача есть ЗЛП максимизации от двух переменных с ограничениями-неравенствами (могут быть и ограничения-равенства). ***Линейная функция f называется*** функцией цели, или ***целевой функцией.*** Ограничения

$$x_1 \geq 0, x_2 \geq 0$$

называются ***ограничениями неотрицательности*** (или условиями неотрицательности), а система линейных неравенств и (или) уравнений называется ***системой ограничений ЗЛП.*** Запись ЗЛП с ограничениями-неравенствами выглядит следующим образом:

Целевая функция

$$f(x_1, x_2) = 10x_1 + 12x_2 \rightarrow max$$

Система ограничений

$$\begin{cases} 9\ _1 + 7x_2 \quad 18 \\ 4\ _1 + 5x_2 \quad 11 \end{cases}$$

$$x_1 \geq 0,\ x\ \ \geq 0$$

ЗЛП является удобной ***математической моделью*** для большого числа экономических задач (планирование производства, расходование ресурсов, раскрой материалов, транспортные перевозки и т.д.). Рассмотрим на примерах процесс построения математической модели (в виде ЗЛП на максимум или минимум) для некоторых задач.

Основа построения математических моделей в виде ЗЛП - это, прежде всего, правильный выбор параметров экономической задачи (или некоторого процесса), через которые требуемая цель выражалась бы в виде линейной целевой функции, а ограничения на процесс записывались бы в виде системы линейных уравнений или неравенств.

3.3. Пример решения задачи линейного программирования в Microsoft Excel с использованием модуля Поиск решения

MS Excel содержит модуль *Поиск решения*, позволяющий осуществлять поиск оптимальных решений, в том числе *решение* задач линейного, целочисленного, нелинейного и стохастического программирования.

По умолчанию надстройка *Поиск решения* отключена. Если у вас *Excel 2003*: выберите команду *Сервис-Надстройки* и активизируйте надстройку *Поиск решения*. Чтобы активизировать ее в *Excel 2007* или *Excel 2010*, щелкните значок *Кнопка Microsoft Office* (для *MS 2007*) или выберите вкладку *Файл* (для *MS 2010*), щелкните *Параметры* (рис. 3.1), а затем выберите категорию *Надстройки*. В поле *Управление* выберите значение *Надстройки Excel* и нажмите кнопку *Перейти* (рис. 3.2). В поле *Доступные надстройки* установите флажок рядом с пунктом *Поиск решения* и нажмите кнопку *ОК* (рис. 3.3).

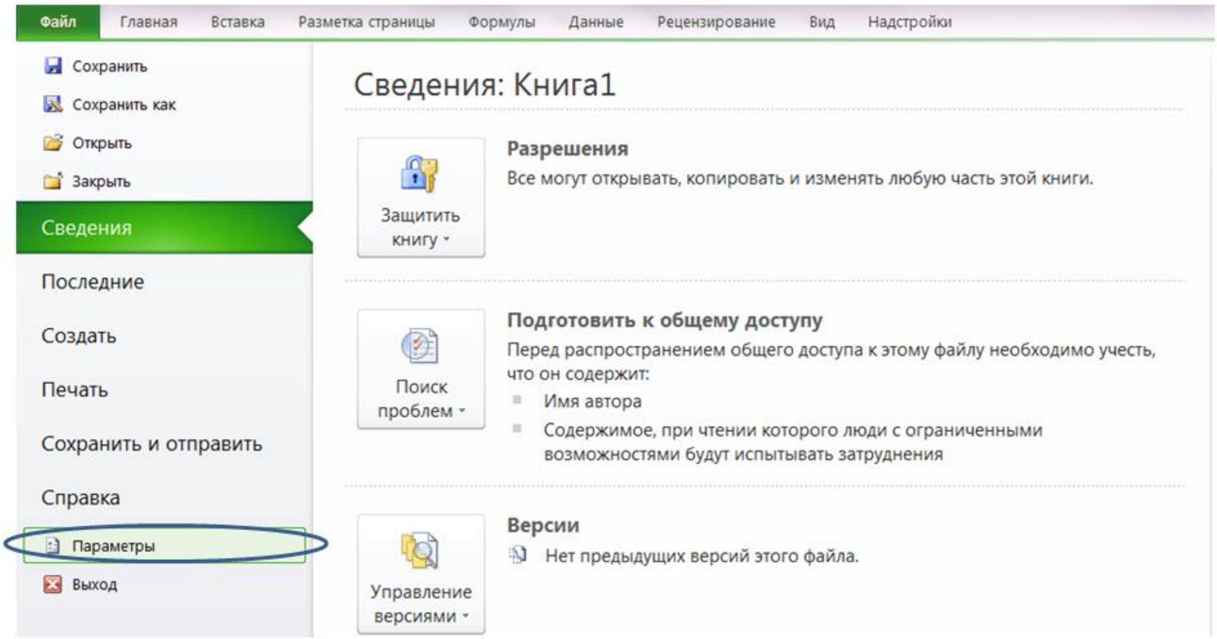


Рис. 3.1. Вкладка *Файл MS Excel 2010*

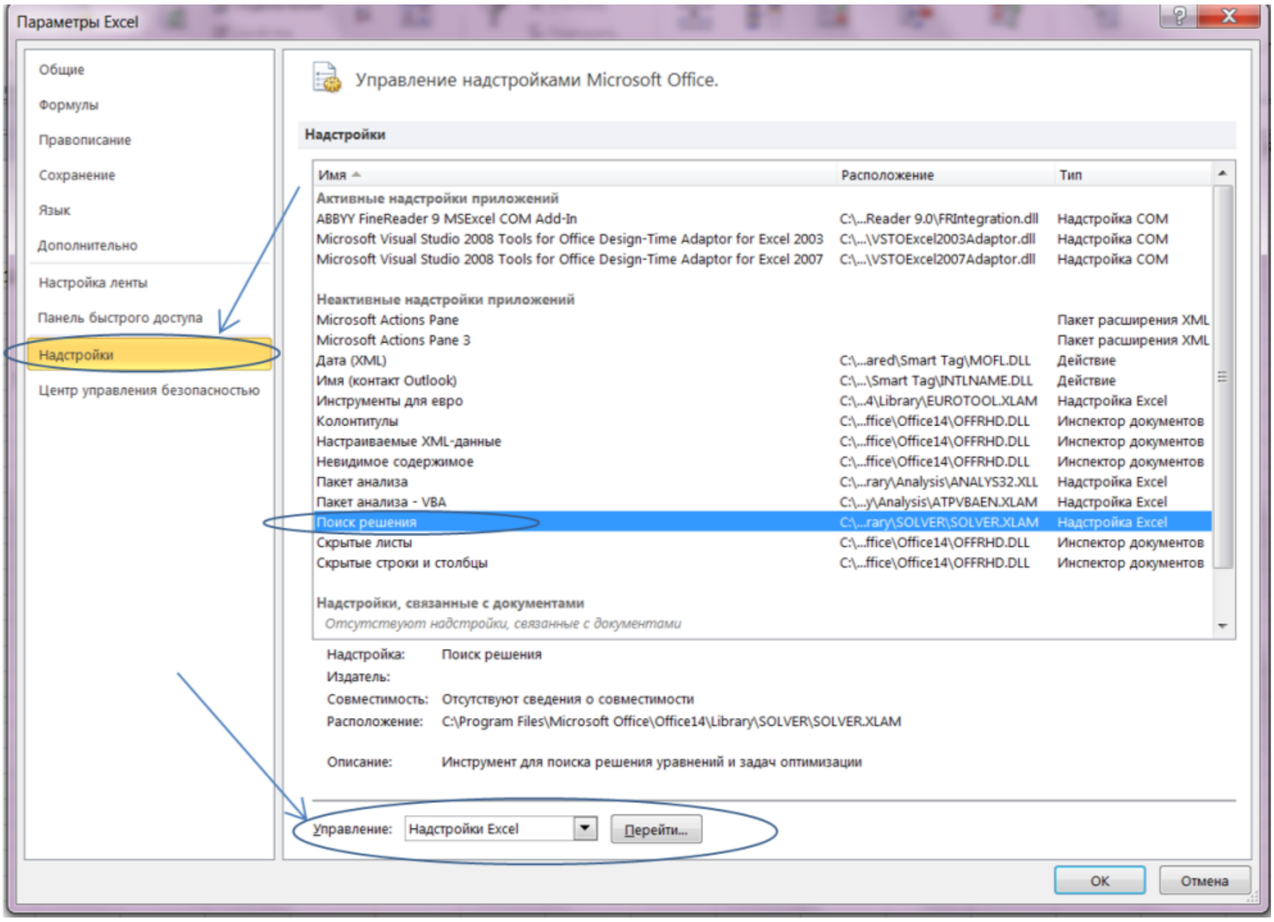


Рис. 3.2. Окно *Параметры Excel*

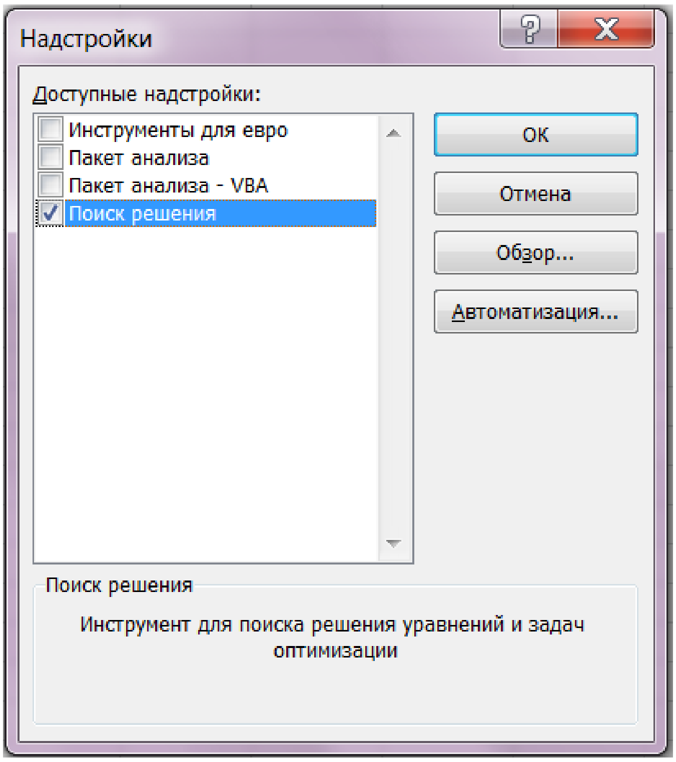


Рис. 3.3. Окно *Надстройки*

Постановка задачи осуществляется посредством задания ячеек для переменных и записи формул с использованием этих ячеек для целевой функции и системы ограничений.

Например, на рис. 3.4 приведена постановка задачи линейного программирования с целочисленным решением.

Математическая модель этой задачи, записанная в обычной математической форме:

- целевая функция

$$S = 60x_1+70x_2+120x_3+130x_4 \rightarrow \max,$$

- система ограничений

$$\begin{gathered} x_1+x_2+x_3+x_4 \le 16, \\ 6x_1+5x_2+4x_3+3x_4 \le 110, \\ 4x_1+6x_2+10x_3+13x_4 \le 150, \\ x_1,\ x_2,\ x_3,\ x_4 \ge 0 \text{ – целые.} \end{gathered}$$

В *Excel* математическая модель может быть представлена, в лучшем случае, в виде таблицы чисел. Размещение данных в *Excel* оформляют в свободном порядке, формы ввода не предусмотрены.

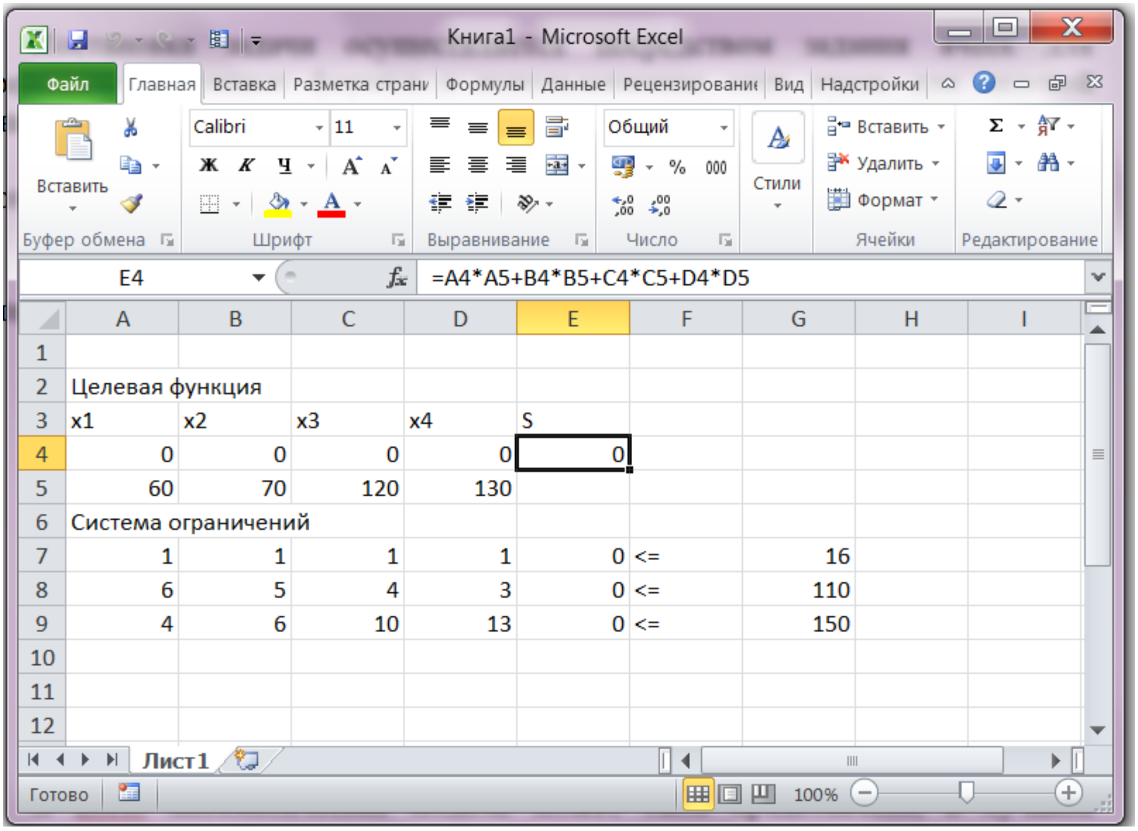


Рис. 3.4. Постановка задачи

Далее, переходим к решению. Выбираем в меню *Сервис* | *Поиск решения* (для *MS Excel 2003*) или *Данные* | *Анализ* | *Поиск решения* (для *MS Excel 2007* и *2010*). Открывается диалоговое окно *Параметры поиска решения* (рис. 3.5). Здесь указывают ячейки целевой функции, переменных и устанавливают ограничения исходя из заданной системы ограничений. Можно начать решение или установить *Параметры* решения.

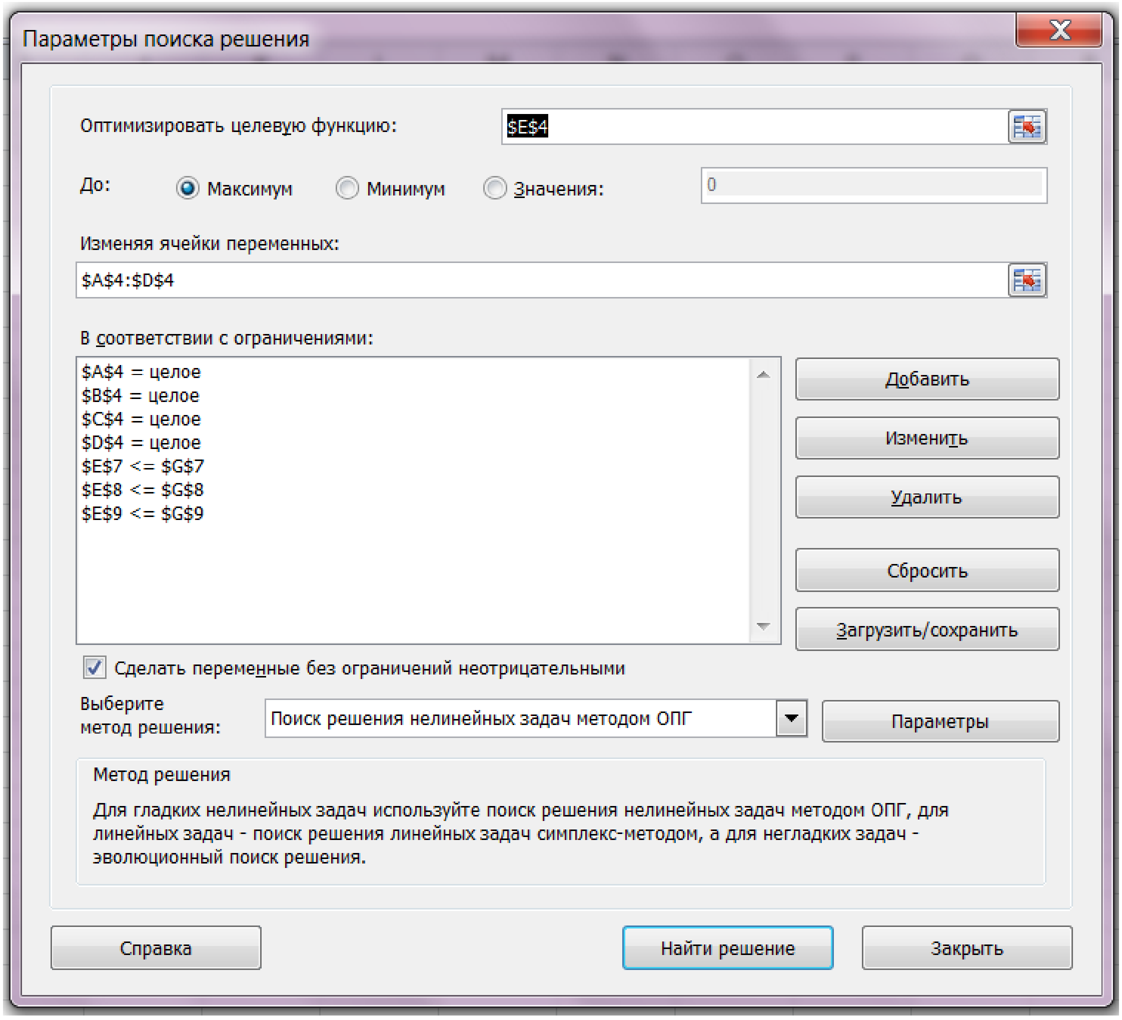


Рис. 3.5. Модуль *Поиск решения* программы *MS Excel*

Дальнейший процесс решения отличается в зависимости от версии *MS Office*:

→ *MS Office 2003*

- Нажать кнопку *Параметры* (рис. 3.6). Выбираем:
    - *Линейная модель*.
    - *Неотрицательные значения*.
    - *Показывать результаты итераций*.

  Линейная модель служит для ускорения поиска решения линейной задачи оптимизации. Функция *Показывать результаты итераций* служит для приостановки поиска решения для просмотра результатов отдельных итераций.

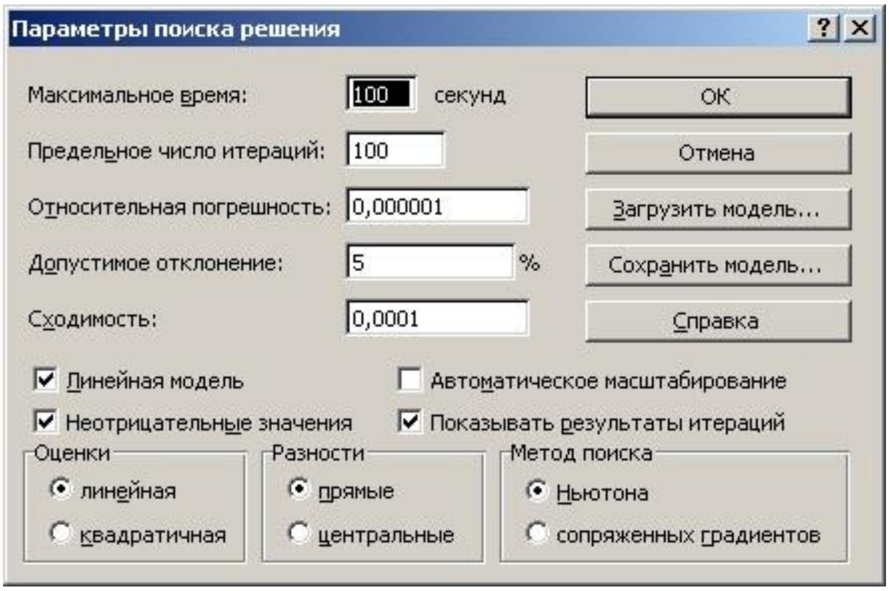


Рис. 3.6. Окно *Параметры поиска* решения модуля *Поиск решения*

- По умолчанию выбран метод поиска *Ньютона*, есть возможность выбрать метод *сопряженных градиентов*.
    - *Метод Ньютона* – реализация квазиньютоновского метода, в котором запрашивается больше памяти, но выполняется меньше итераций, чем в методе сопряженных градиентов.
    - *Метод сопряженных градиентов* – реализация метода сопряженных градиентов. Данный метод следует использовать, если задача достаточно велика и необходимо экономить память, а также если итерации дают слишком малое отличие в последовательных приближениях.

Причем здесь нет возможности выбрать ни графический симплекс-метод, ни метод симплекс-таблиц. Применение метода, позволяющего найти целочисленное решение, определяется лишь добавлением условия на каждую переменную – «целое».

→ *MS Office 2010*

- В диалоговом окне *Параметры поиска решения* можно выбрать любой из указанных ниже алгоритмов или методов поиска решения:
    - *Нелинейный метод обобщенного понижающего градиента (ОПГ).* Используется для гладких нелинейных задач.
    - *Симплекс-метод*. Используется для линейных задач.
    - *Эволюционный метод*. Используется для негладких нелинейных задач.
- Так как у нас линейная задача, то выбор метода однозначен.

После нажатия кнопки *Найти решение* через несколько секунд появится диалоговое окно *Результаты поиска решения* (рис. 3.7). Есть возможность создать отчет по результатам. Отчет создается на новом листе (рис. 3.8).

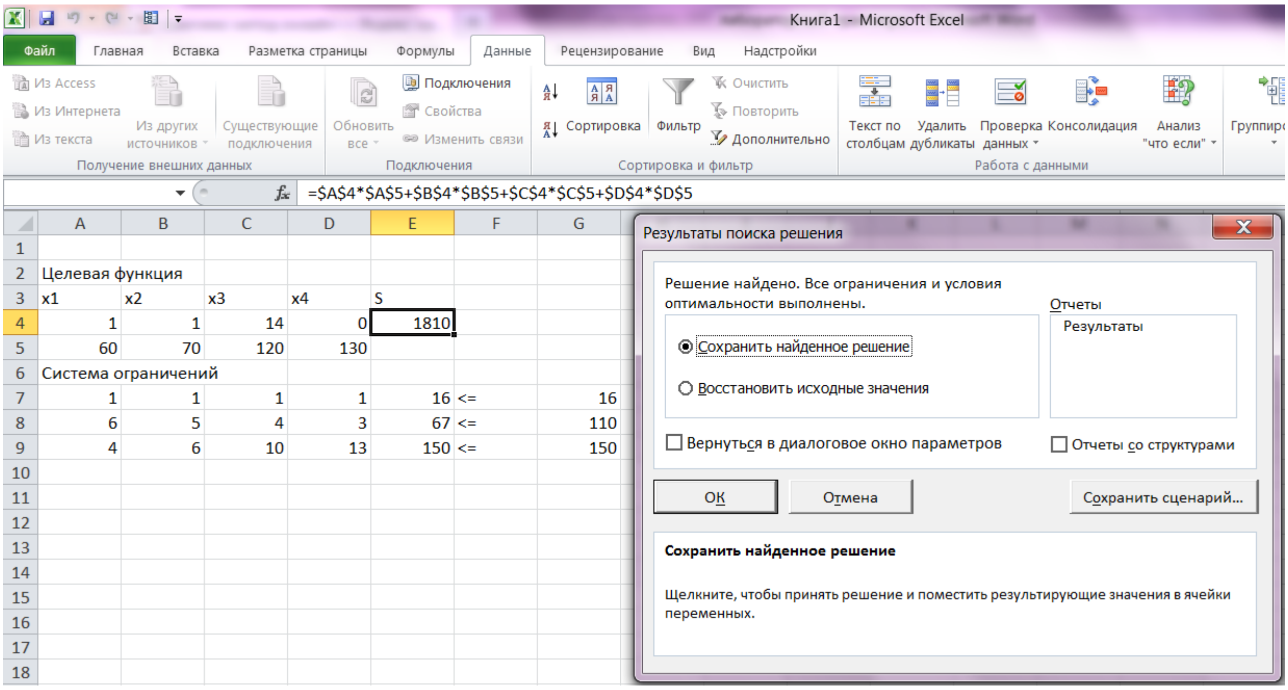


Рис. 3.7. Результат работы модуля *Поиск решения* при решении целочисленной задачи

Если в рассматриваемой задаче убрать ограничения на целочисленность, то получится другой ответ. В диалоговом окне *Результаты поиска решений* будет возможность создать 3 отчета (рис. 3.9.):

- ✓ Результаты.
- ✓ Устойчивость.
- ✓ Пределы.

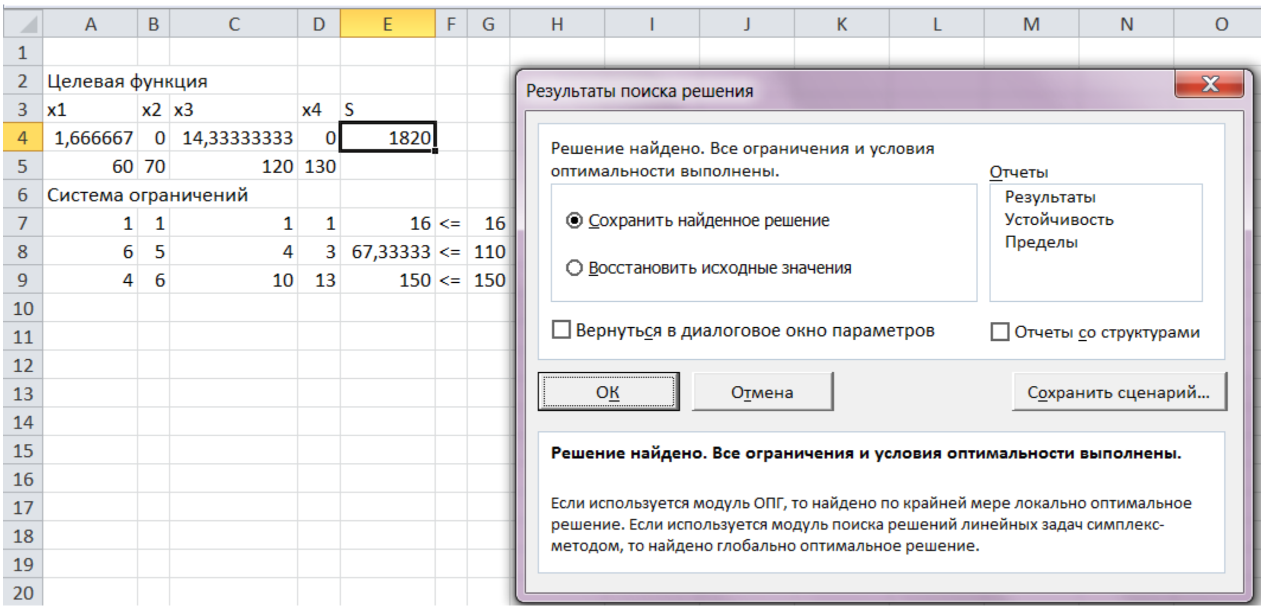


Рис. 3.9. Результат работы модуля *Поиск решения*

Microsoft Excel 14.0 Отчет о результатах
Лист: [Книга1]Лист1
Отчет создан: дата время
Результат: Решение найдено. Все ограничения и условия оптимальности выполнены.
Модуль поиска решения
Модуль: Поиск решения линейных задач симплекс-методом
Время решения: 12,89 секунд.
Число итераций: 2 Число подзадач: 4
Параметры поиска решения
Максимальное время 100 с, Число итераций Без пределов, Precision 0,0001, Показывать результаты итераций
Максимальное число подзадач Без пределов, Максимальное число целочисленных решений Без пределов
Целочисленное отклонение 5%, Считать неотрицательными

Ячейка целевой функции (Максимум)

Ячейка	Имя	Исходное значение	Окончательное значение
$E$4	S	0	1810

Ячейки переменных

Ячейка	Имя	Исходное значение	Окончательное значение	Целочисленное
$A$4	x1	0	1	Целочисленное
$B$4	x2	0	1	Целочисленное
$C$4	x3	0	14	Целочисленное
$D$4	x4	0	0	Целочисленное

Ограничения

Ячейка	Имя	Значение ячейки	Формула	Состояние	Допуск
$E$7	S	16	$E$7<=$G$7	Привязка	0
$E$8	S	67	$E$8<=$G$8	Без привязки	43
$E$9	S	150	$E$9<=$G$9	Привязка	0
$A$4=Целочисленное					
$B$4=Целочисленное					
$C$4=Целочисленное					
$D$4=Целочисленное					

Рис. 3.8. Пример отчета

Такой подход к решению задач широко используют для проведения практических расчетов, когда важен лишь результат решения, а не сам процесс получения оптимального решения.

Если важен сам процесс решения задачи с применением какого-либо метода решения, модуль *Поиск решения* может быть использован лишь для сравнения результатов решения задачи в качестве проверки правильности применения методов. Тут следует отметить, что при решении некоторых задач получается несколько вариантов оптимального решения. Так, например, транспортная задача (частный случай задачи ЛП) может иметь два (и более) оптимальных плана перевозок с одинаковой стоимостью.

На рис. 3.10 показан ввод данных другой задачи:

18	15	12	360
6	4	8	192
5	3	3	180
9	10	16	

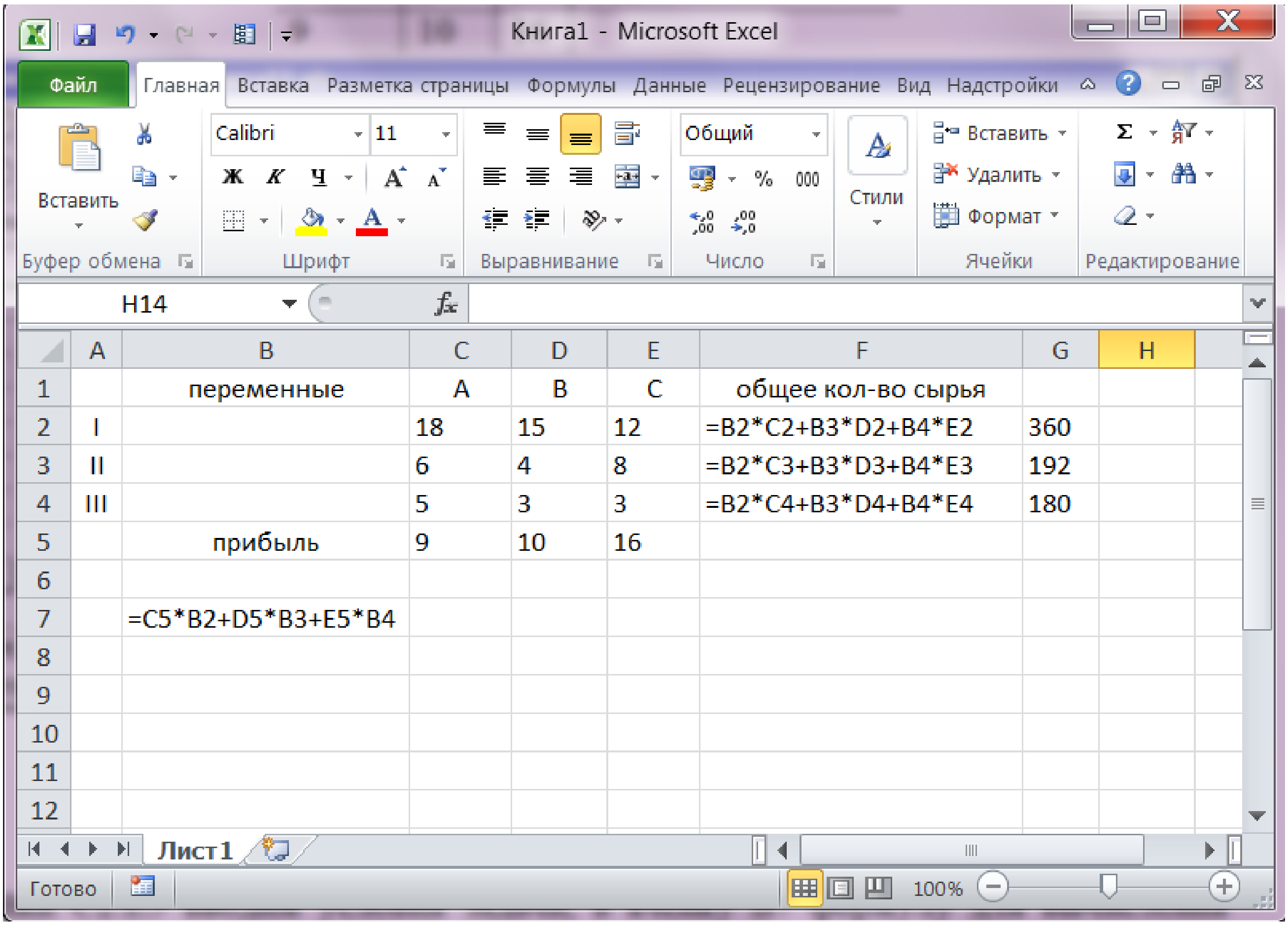


Рис. 3.10. Данные задачи

Ячейки B2:B4 предназначены для значений изменяемых переменных. В ячейки *C2:E5* вводим условия задачи, в ячейку *B7* формулу для вычисления целевой функции.

Для решения задачи воспользуемся надстройкой *MS Excel Поиск решения*.

Сначала нужно заполнить поле *Оптимизировать целевую функцию – $B$7*. Затем установить переключатель *До* на значение *Максимум*. Наконец, определить данные поля *Изменяя ячейки переменных*, выделив ячейки *$B$2:$B$4*. После этого определяют ограничения: нужно активизировать кнопку *Добавить*.

Необходимо ввести ограничения на количество используемого сырья:

$F$2<=$G$2, $F$3<=$G$3, $F$4<=$G$4.

Чтобы не вводить дополнительные ограничения на неотрицательность переменных, выбираем *Сделать переменные без ограничений неотрицательными* (рис. 3.11).

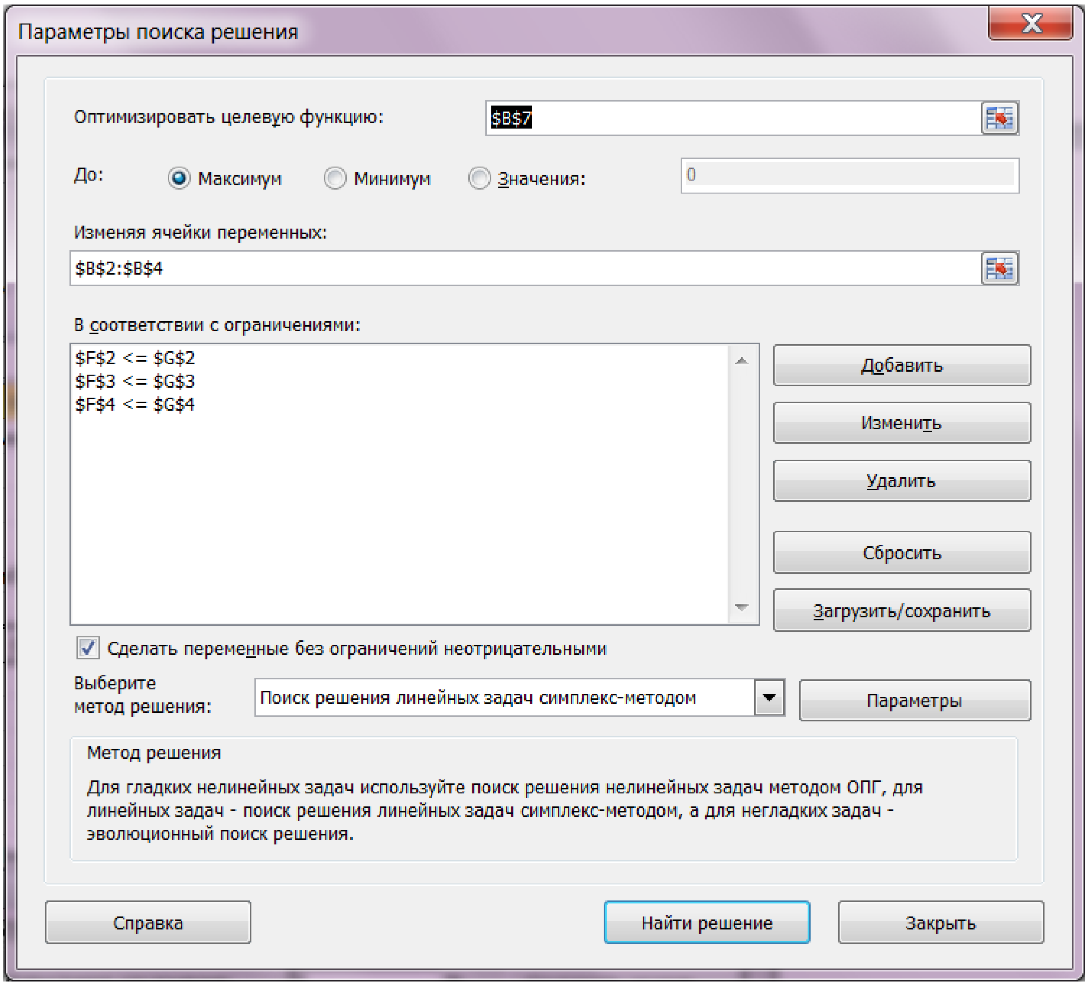


Рис. 3.11. Диалоговое окно *Параметры поиска решения*

Теперь для процедуры *Поиска решения* готовы все исходные данные. Для того чтобы начать процесс решения задачи, нужно выбрать кнопку *Найти решение*. В строке состояния будет отражаться ход решения задачи.

Через некоторое время на экране появится диалоговое окно *Результаты поиска решения* (рис. 3.12). Теперь можно выбрать одну из следующих возможностей:

- сохранить найденное решение;
- восстановить исходные значения в изменяемых ячейках.

В итоге получается требуемое решение – необходимое количество каждого вида ресурсов и величина максимальной прибыли. На рис. 3.13 в ячейках *B2:B4* найдены значения переменных $x_1 = 0, x_2 = 8, x_3 = 20$, а в ячейке *B7* значение целевой функции $F = 400$.

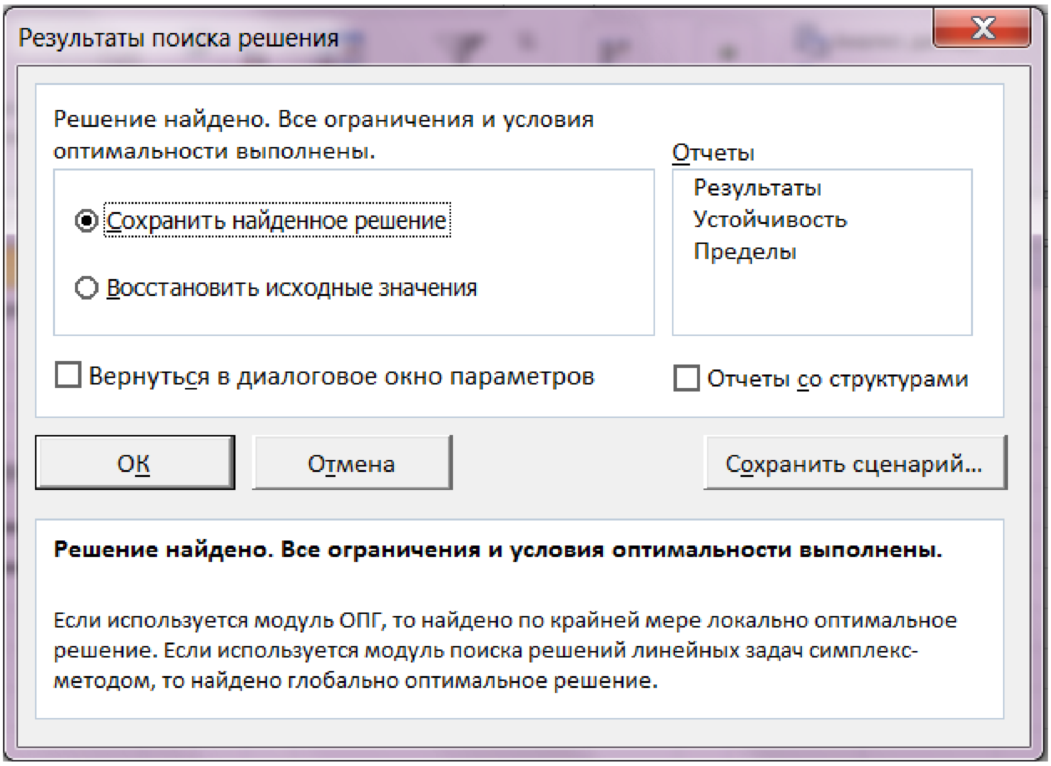


Рис. 3.12. Диалоговое окно *Результаты поиска решения*

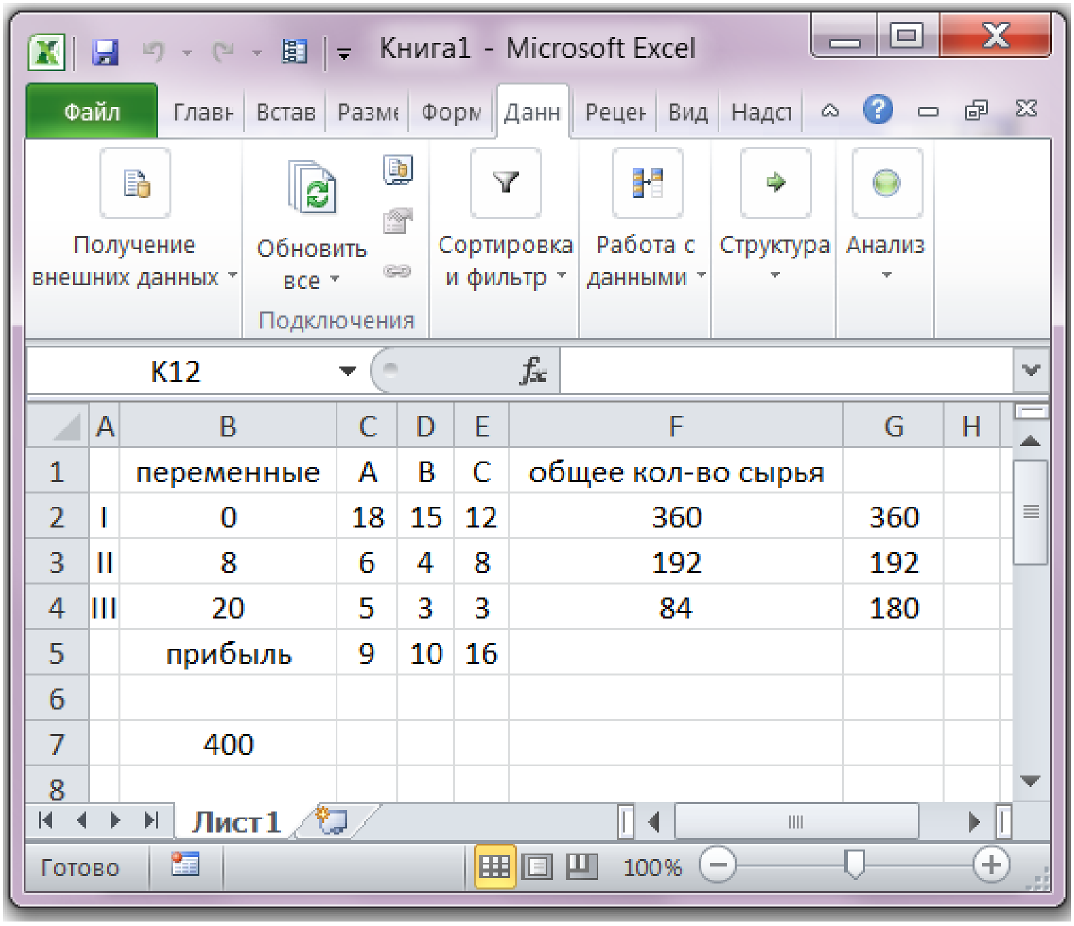


Рис. 3.13. Результаты решения задачи

3.4. Задача об оптимальном составе смеси (задача составления рациона, задача о диете)

В студенческом кафе имеются три компонента: яблочный сок, апельсиновый сок и газировка.

Цель: подобрать оптимальный состав коктейля из этих трёх компонентов, если известно:

- стоимости ингредиентов в рублях: $c_1 = 50,\ c_2 = 100,\ c_3 = 20$;
- содержание мякоти в граммах: $a_1 = 0.4,\ a_2 = 0.5,\ a_3 = 0$;
- вкусовые качества в баллах: $b_1 = 4,\ b_2 = 8,\ b_3 = 10$.

Условия:

- содержание мякоти в коктейля должно быть не меньше 0,2 гр;
- вкус должен иметь не менее 8 баллов.

<u>Математическая постановка задачи.</u>

Пусть x_1, x_2, x_3 – доля каждого компонента в коктейле ($x_i \geq 0, i = 1,2,3$).

Тогда ограничения примут вид:

$$\begin{cases} 0{,}4x_1 + 0{,}5\,x_2 \geq 0{,}2\,, \\ 4x_1 + 8\,x_2 + 10\,x_3 \geq 8, \\ x_1 + \ x_2 + \ x_3 = 1. \end{cases}$$

(третье условие отражает наличие в составе смеси только трёх компонентов, т. е. то, что все три компонента составляют весь коктейль в целом – 1).

При этом линейная функция (стоимость коктейля) будет иметь вид:

$$f = 50x_1 + 100\,x_2, +20x_3 \to min.$$

Решение задачи в программе *MS Excel*:

1. Надо заполнить ячейки *A1:A3* таблицы обозначениями x_1, x_2, x_3, а ячейку *A4 – min*.
2. В ячейку *B4* записать целевую функцию.
3. В диапазон ячеек *A7:C9* записать систему ограничений через адреса соответствующих ячеек (рис. 3.14).
4. Заполнить форму *Поиск решения* (рис. 3.15).

Решение задачи представлено на рис. 3.16.

<u>Замечание.</u> Для того чтобы в решении задачи после запятой отображалось только 2 знака, надо выделить соответствующий диапазон ячеек, щёлкнуть на выделенном фрагменте правой кнопкой мыши, в появившемся контекстном меню выбрать пункт *Формат ячеек* и на вкладке *Число* выбрать *Числовой*, указав *Число десятичных знаков 2*.

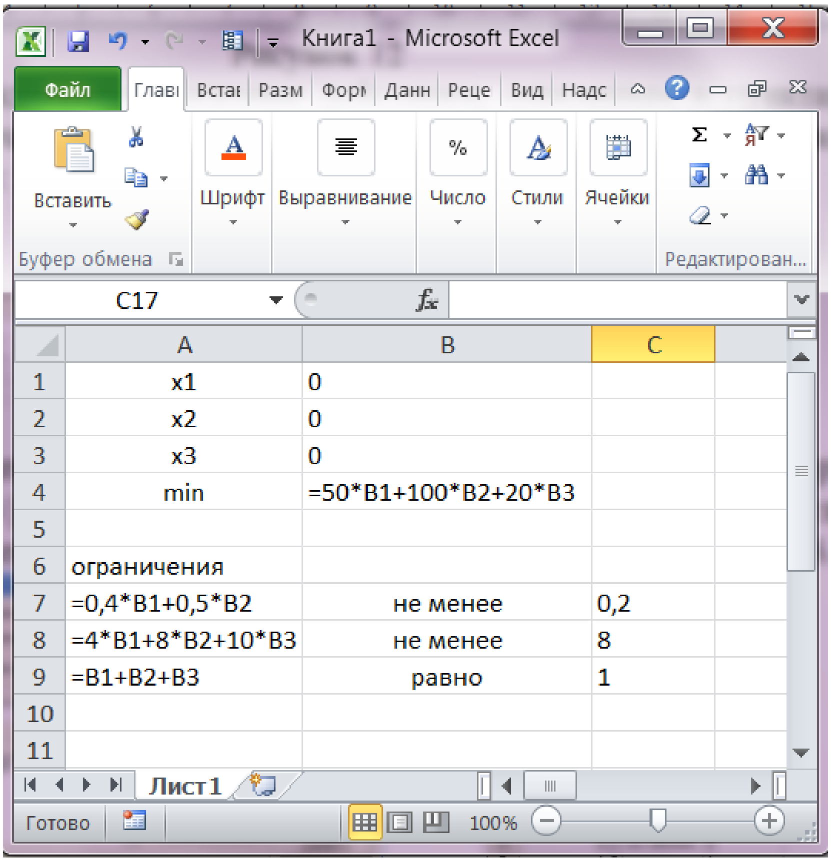


Рис. 3.14. Исходные данные

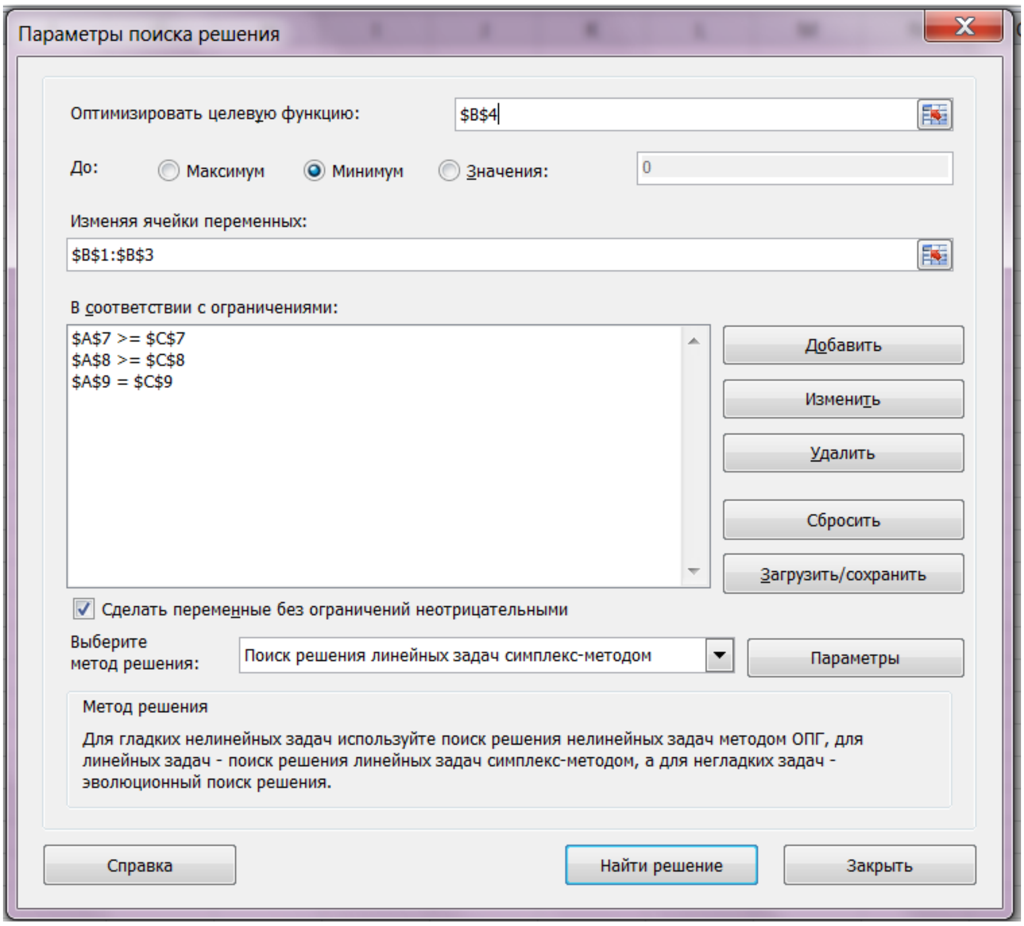
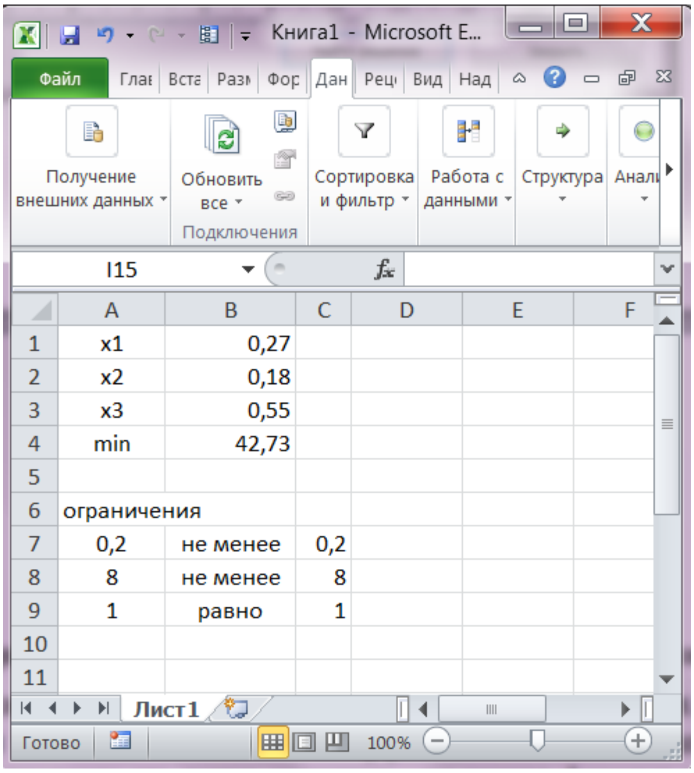


Рис. 3.15. Поиск решения



	A	B	C
1	x1	0,27	
2	x2	0,18	
3	x3	0,55	
4	min	42,73	
5			
6	ограничения		
7	0,2	не менее	0,2
8	8	не менее	8
9	1	равно	1

Рис. 3.16. Решение задачи

Решение задачи в среде *MathCad*:

1. Задать начальное приближение.
2. Записать все коэффициенты из ограничений и целевой функции в матричном виде.
3. Ввести целевую функцию.
4. Записать функцию Given.
5. Записать ограничения:

   знак «жирное равно» вставляется комбинацией клавиш *Ctrl* и = или выбирается на панели инструментов *Булева алгебра*.
6. Записать функцию минимизации. После чего получим оптимальный план.

   Для того чтобы *MathCad* отображал только 2 десятичных знака, надо выделить значения, для которых необходимо сменить формат вывода, в меню *Формат* выбрать подменю *Результат* – появится диалоговое окно *Формат результата* и указать *Число десятичных знаков 2*.
7. Найти значение целевой функции в точке минимума (рис. 3.17).

Оптимальные планы рассматриваемой задачи при решении в средах *Microsoft Excel* и *MathCad* получились одинаковые, а целевая функция при решении в среде *MathCad* имеет меньшее значение, таким образом, среда *MathCad*, в данном случае, даёт лучшее решение, чем *Microsoft Excel*.

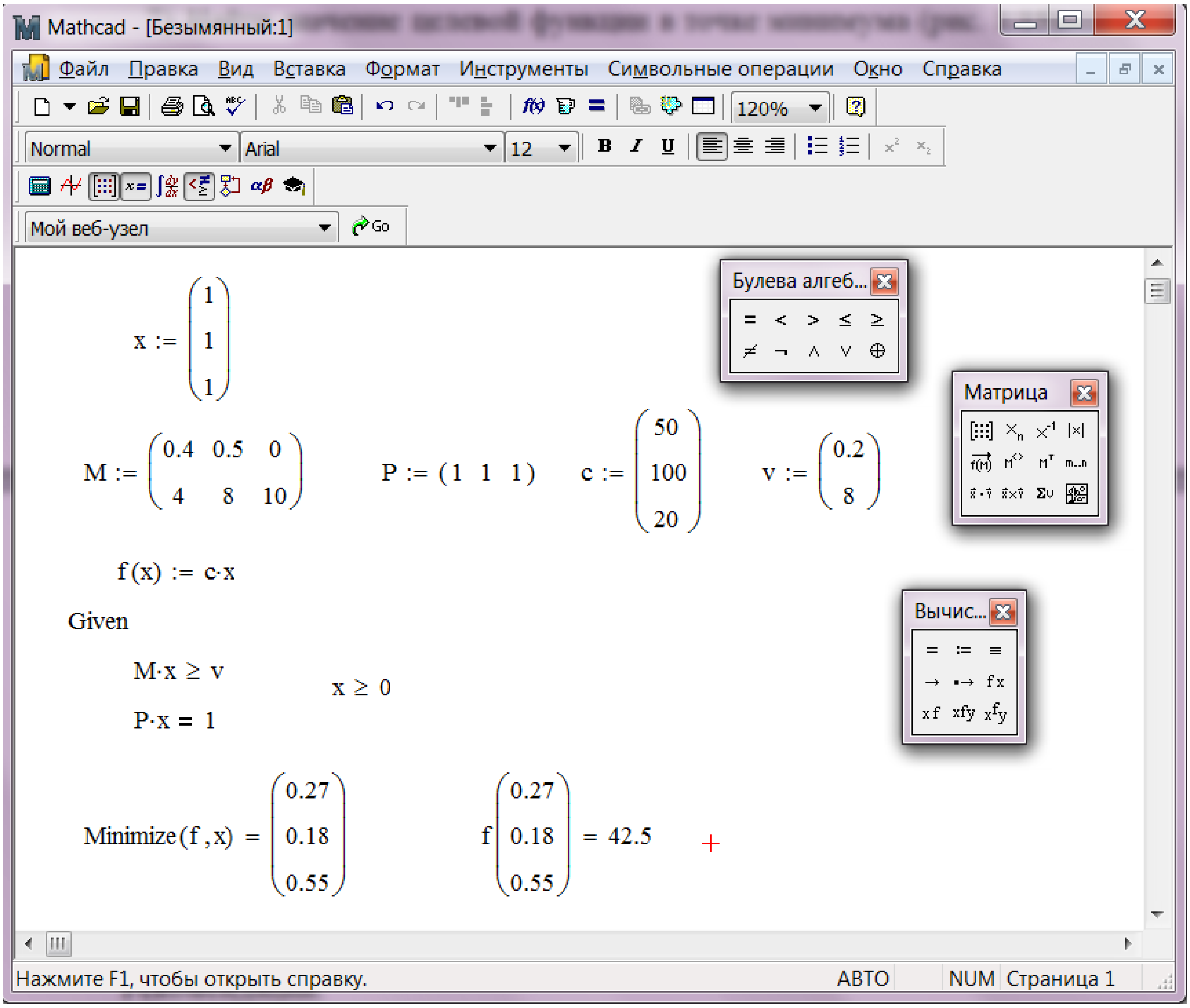


Рис. 3.17. Решение задачи в среде *Mathcad*

3.5. Задача производства

К группе задач о производстве относят задачи, целью которых является подбор наиболее выгодной производственной программы выпуска одного или нескольких видов продукции при использовании некоторого числа ограниченных источников сырья.

Фабрика производит мебель трёх типов:

- наборы пристенной мебели (далее «стенки»);
- шкафы для одежды (далее «шкафы»);
- кухонные гарнитуры (далее «гарнитуры»).

Для их производства в основном используются три типа сырья:

- древесина;
- стекло;
- зеркала.

Удельные коэффициенты расхода сырья, а также трудозатраты на единицу каждого типа мебели приведены в табл. 1.

Таблица 1

Коэффициенты расхода сырья

Ресурсы / Типы мебели	Древесина, м^3	Стекло, м^2	Зеркала, м^2	Трудозатраты, человеко-дней
«Стенка»	4	4	3	10
«Шкаф»	2	0	2	7
«Гарнитур»	2	5	1	8

Запасы сырья на складе обновляются ежемесячно и составляют 70 м^3 древесины, 90 м^3 стекла и 45 м^3 зеркал. Трудозатраты в месяц не должны превышать 200 человеко-дней. Чистая прибыль от продажи одной «стенки», «шкафа» и «гарнитура» составляет соответственно 2000 руб., 1250 руб. и 1500 руб. Найти оптимальный ассортимент продукции, максимизирующий общую прибыль за месяц.

Математическая постановка задачи.

Пусть x_1, x_2, x_3 – месячный выпуск продукции соответственно: «стенок», «шкафов» и «гарнитуров» ($x_i \geq 0,\ x_i$ – целые, $i = 1, 2, 3$). Тогда должно быть:

$$\begin{cases} 4x_1 + 2\,x_2 + 2\,x_3 \leq 70, \\ 4x_1 + 5\,x_3 \leq 90, \\ 3x_1 + 2\,x_2 + \ x_3 \leq 45, \\ 10x_1 + 7\,x_2 + 8\,x_3 \leq 200. \end{cases}$$

При этом линейная функция

$$f = 2000x_1 + 1250\,x_2 + 1500\,x_3 \to max.$$

Решение задачи в программе *MS Excel*:

1) Надо заполнить ячейки *A1:A3* таблицы обозначениями x_1, x_2, x_3, а ячейку *A4* – *max*.
2) В ячейку *B4* записать целевую функцию.
3) В диапазон ячеек *A7:C10* записать систему ограничений через адреса соответствующих ячеек (рис. 3.18).
4) Заполнить форму *Поиск решения*.

   Не забудьте ограничения на целочисленность переменных, так как предприятие не может выпустить часть изделия.

Решение задачи представлено на рис. 3.19.

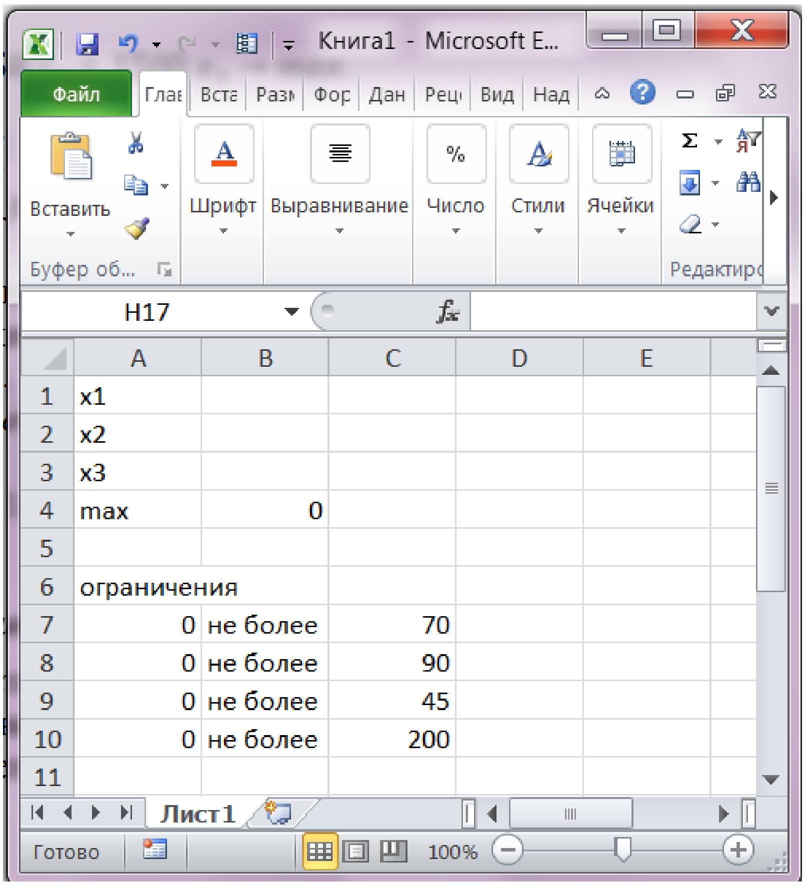


Рис. 3.18. Задача производства

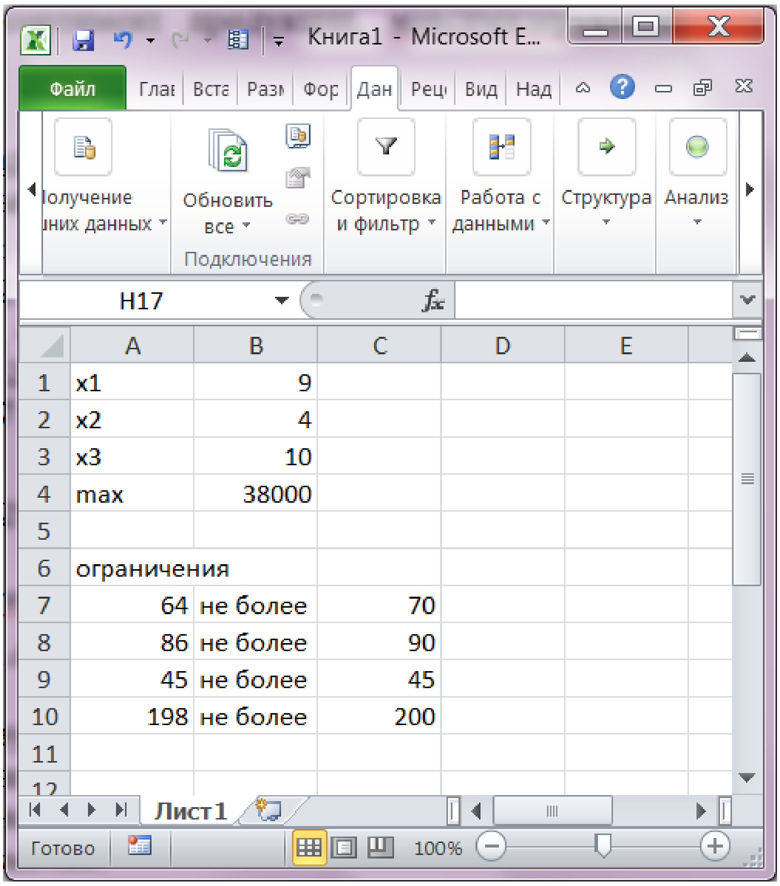


Рис. 3.19. Решение задачи производства

Решение задачи в среде *MathCad* приведено на рис. 3.20.

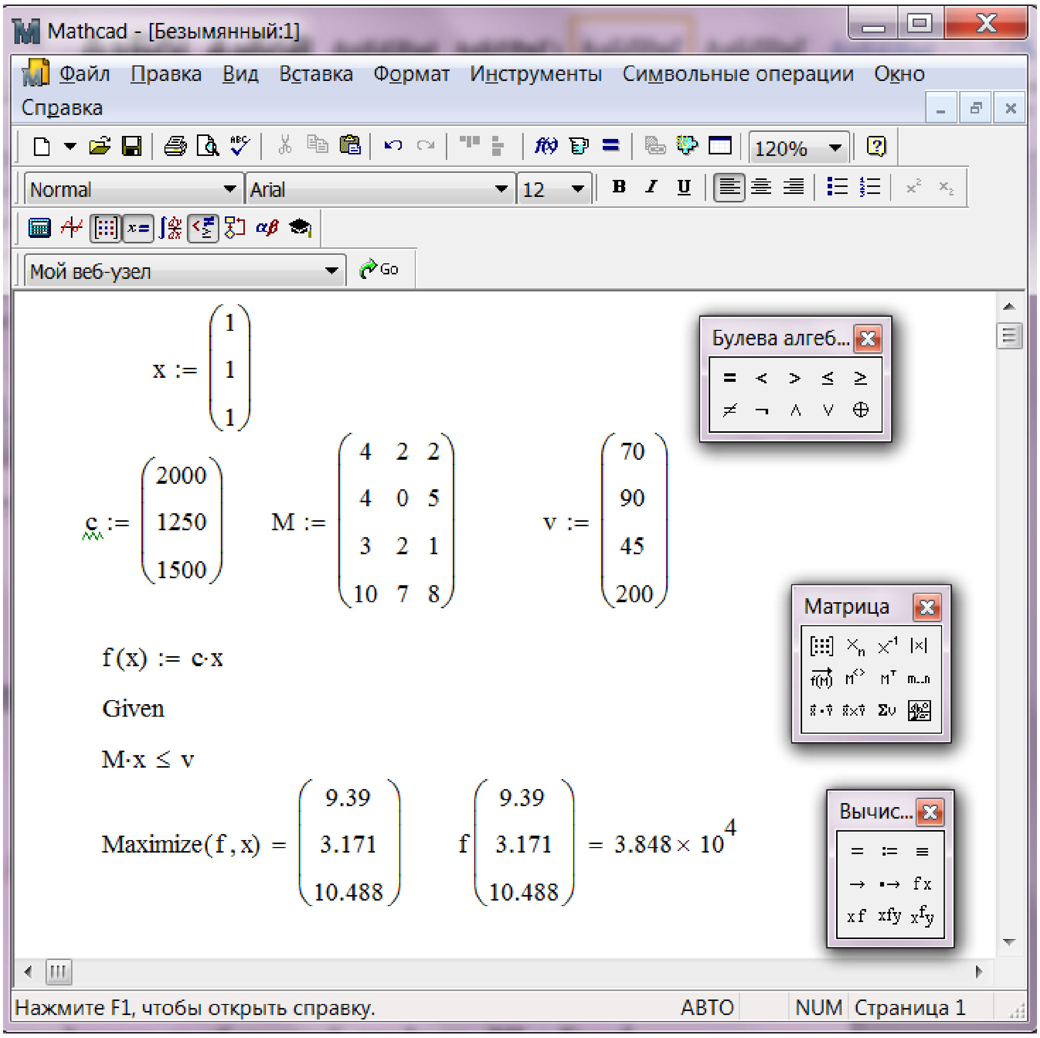


Рис. 3.20. Решение задачи производства в среде *Mathcad*

В классическом наборе функций в среде *MathCad* отсутствуют функции нахождения целочисленного решения.

3.6. Рекомендации по созданию табличной модели ЛП в Excel

Одним из важных результатов выполнения перечисленных ниже рекомендаций является то, что все основные коэффициенты модели содержатся в отдельных ячейках, поэтому их легко изменять, не меняя формул модели. Кроме того, группирование переменных решений и ограничений позволяет копировать формулы для создания аналогичных формул.

Благодаря группированию также упрощается работа со средством *Поиск решения*, поскольку для указания переменных решения или ограничений можно использовать диапазоны ячеек рабочего листа.

Рекомендации:

- каждая переменная решения располагается в отдельной ячейке, ячейки группируются по столбцам или строкам; каждому ограничению отводится отдельная строка или столбец таблицы (обычно переменные решения расположены в столбцах, а ограничения – в строках);
- переменные решения группируются в отдельный блок столбцов/строк; аналогично ограничения группируются в свой блок строк/столбцов;
- все ячейки, содержащие переменные решения и целевую функцию, имеют заголовки в верхней части своего столбца, а все ограничения имеют заголовки в крайней слева ячейке своей строки;
- коэффициенты целевой функции хранятся в отдельной строке, располагаясь непосредственно под или над соответствующими переменными решения; формула для вычисления целевой функции находится в соседней ячейке;
- чтобы модель была понятней, ячейки с переменными решения и целевой функцией выделяются рамкой по границе ячеек или заливкой ячеек;
- коэффициент перед определенной переменной решения в каком-либо ограничении записывается в ячейку на пересечении столбца (строки), содержащего данную переменную решения, и строки (столбца), содержащей это ограничение;
- в каждой строке ограничений за ячейками, содержащими коэффициенты данного ограничения, следует ячейка, в которую записано вычисленное значение функции ограничения (значение левой части неравенства), за ней следует ячейка, в которой стоит соответствующий знак неравенства, а затем ячейка, содержащая значение правой части неравенства;
- ячейки, содержащие правые части ограничений, должны включать константы или формулы, в которые не входят переменные решения, т.к. все формулы в правой части, прямо или косвенно связанные с переменными решения, должны быть перенесены в левую часть с помощью алгебраических преобразований данного неравенства;
- не следует использовать в формулах модели ЛП функции *Excel ЕСЛИ, ABS, MAX, MIN* и другие нелинейные функции (подобные функции могут использоваться в формулах рабочего листа, но только в том случае, если они не влияют прямо или косвенно на вычисление целевой функции);
- условия неотрицательности необязательно включать в табличную модель; как правило, они опускаются и указываются непосредственно в диалоговом окне средства *Поиск решения*.

3.7. Задание к лабораторной работе

Решить задачу в Microsoft Excel и MathCad, сравнить полученные решения.

Имеется n продуктов $P_1, P_2, \ldots, P_n$, содержащих m видов питательных веществ $S_1, S_2, \ldots, S_m$. Пусть a_{ij}, где $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, m$ – количество единиц j-го питательного вещества в единице i-го продукта; b_j – суточная потребность (минимальная норма) организма в j-м питательном веществе; C_i – стоимость единицы i-го продукта. Требуется выбрать такой суточный рацион питания (т. е. назначить количества продуктов $P_1, P_2, \ldots, P_n$, входящих в него), чтобы условия по питательным веществам были выполнены, а стоимость рациона была минимальной.

Таблица 2

Варианты 1-5

Вариант	Виды питательных веществ	Количество единиц питательных веществ в единице продукта				Минимальная норма питательных веществ	Стоимость единицы продукта			
		P_1	P_2	P_3	P_4		C_{P_1}	C_{P_2}	C_{P_3}	C_{P_4}
1	S_1	3	1	-	-	9	4	6	-	-
	S_2	1	2	-	-	8				
	S_3	1	6	-	-	12				
2	S_1	1,2	1,4	0,8	-	1,6	3	4	5	-
	S_2	80	280	240	-	200				
	S_3	5	5	100	-	10				
3	S_1	26,5	7,8	0	0	21	14,4	16	12,8	10,5
	S_2	51	26	45,7	0	30				
	S_3	0	0	5	72,5	500				
4	S_1	1	5	-	-	10	2	3	-	-
	S_2	3	2	-	-	12				
	S_3	2	4	-	-	16				
	S_4	2	2	-	-	10				
	S_5	1	0	-	-	1				
5	S_1	0,18	0,24	1,2	-	12	1	1,1	7,5	-
	S_2	10	8	200	-	100				
	S_3	15	1	1,5	-	450				

Для изготовления n видов продукции $P_1, P_2, \ldots, P_n$ предприятие использует m видов ресурсов $S_1, S_2, \ldots, S_m$ (сырьё, топливо, материалы и т. д.). Запасы ресурсов каждого вида ограничены и равны $b_1, b_2, \ldots, b_m$. На изготовление единицы продукции j-го вида ($j = 1, 2, \ldots, n$) расходуется a_{ij} единиц i-го ресурса ($i = 1, 2, \ldots, m$). При реализации единицы j-й продукции предприятие получает C_j единиц прибыли. Необходимо составить такой план выпуска продукции, чтобы при её реализации получить максимальную прибыль.

Таблица 3

Варианты 6-24

<table>
<tr><th rowspan="2">Вариант</th><th rowspan="2">Виды ресурсов</th><th colspan="3">Расход ресурсов на единицу продукции</th><th rowspan="2">Запасы ресурсов</th><th colspan="3">Доходы от реализации единицы продукции</th></tr>
<tr><th>P_1</th><th>P_2</th><th>P_3</th><th>C_{P_1}</th><th>C_{P_2}</th><th>C_{P_3}</th></tr>
<tr><td rowspan="4">6</td><td>S_1</td><td>2</td><td>1</td><td>1</td><td>25</td><td rowspan="4">6</td><td rowspan="4">5</td><td rowspan="4">5</td></tr>
<tr><td>S_2</td><td>1</td><td>1</td><td>1</td><td>14</td></tr>
<tr><td>S_3</td><td>0</td><td>4</td><td>2</td><td>19</td></tr>
<tr><td>S_4</td><td>3</td><td>0</td><td>1</td><td>24</td></tr>
<tr><td rowspan="4">7</td><td>S_1</td><td>2</td><td>5</td><td>-</td><td>300</td><td rowspan="4">5</td><td rowspan="4">8</td><td rowspan="4">-</td></tr>
<tr><td>S_2</td><td>4</td><td>5</td><td>-</td><td>400</td></tr>
<tr><td>S_3</td><td>3</td><td>0</td><td>-</td><td>100</td></tr>
<tr><td>S_4</td><td>0</td><td>4</td><td>-</td><td>200</td></tr>
<tr><td rowspan="3">8</td><td>S_1</td><td>2</td><td>5</td><td>-</td><td>20</td><td rowspan="3">50</td><td rowspan="3">40</td><td rowspan="3">-</td></tr>
<tr><td>S_2</td><td>8</td><td>5</td><td>-</td><td>40</td></tr>
<tr><td>S_3</td><td>5</td><td>6</td><td>-</td><td>30</td></tr>
<tr><td rowspan="4">9</td><td>S_1</td><td>2</td><td>3</td><td>-</td><td>19</td><td rowspan="4">7</td><td rowspan="4">5</td><td rowspan="4">-</td></tr>
<tr><td>S_2</td><td>2</td><td>1</td><td>-</td><td>13</td></tr>
<tr><td>S_3</td><td>0</td><td>3</td><td>-</td><td>15</td></tr>
<tr><td>S_4</td><td>3</td><td>0</td><td>-</td><td>18</td></tr>
<tr><td rowspan="3">10</td><td>S_1</td><td>2</td><td>5</td><td>7</td><td>60</td><td rowspan="3">32</td><td rowspan="3">13</td><td rowspan="3">61</td></tr>
<tr><td>S_2</td><td>22</td><td>14</td><td>18</td><td>500</td></tr>
<tr><td>S_3</td><td>10</td><td>14</td><td>8</td><td>328</td></tr>
<tr><td rowspan="4">11</td><td>S_1</td><td>1</td><td>20</td><td>0</td><td>150</td><td rowspan="4">40</td><td rowspan="4">71</td><td rowspan="4">13</td></tr>
<tr><td>S_2</td><td>5</td><td>15</td><td>5</td><td>160</td></tr>
<tr><td>S_3</td><td>1</td><td>3</td><td>1</td><td>50</td></tr>
<tr><td>S_4</td><td>14</td><td>3</td><td>6</td><td>250</td></tr>
</table>

Вариант	Виды ресурсов	Расход ресурсов на единицу продукции					Запасы ресурсов	Доходы от реализации единицы продукции				
		P_1	P_2	P_3	P_4	P_5		C_{P_1}	C_{P_2}	C_{P_3}	C_{P_4}	C_{P_5}
12	S_1	2	5	1	0	5	300	32	37	11	15	50
	S_2	3	1	1	0	5	250					
	S_3	1	5	0	1	5	200					
	S_4	0	3	0	1	3	100					
13	S_1	8	5	9	0	-	152	8	12	15	5	-
	S_2	1	3	5	7	-	123					
	S_3	7	5	3	1	-	130					
	S_4	0	2	2	0	-	30					
	S_5	4	2	6	8	-	200					
14	S_1	1	2	1	2	1	50	12	15	11	7	9
	S_2	2	0	1	1	1	40					
	S_3	0	2	1	2	2	52					
	S_4	1	1	2	2	2	60					
	S_5	2	2	1	1	2	66					
	S_6	1	1	0	0	0	15					
	S_7	0	1	1	0	0	20					
15	S_1	15	12	18	14	10	300	4	6	8	7	11
	S_2	2	3	2	2	1	50					
	S_3	6	4	3	1	0	90					
	S_4	1	1	3	3	5	45					
16	S_1	2	5	7	1	0	60	32	13	61	15	3
	S_2	22	14	18	5	2	500					
	S_3	10	14	8	3	0	328					
17	S_1	5	4	1	0	5	100	95	40	32	21	18
	S_2	3	1	2	5	1	95					
	S_3	7	3	4	2	0	50					
18	S_1	1	3	4	0	-	34	5	3	3	2	-
	S_2	7	5	2	5	-	80					
	S_3	2	2	1	1	-	28					
	S_4	0	1	1	7	-	36					

Вариант	Виды ресурсов	Расход ресурсов на единицу продукции				Запасы ресурсов	Доходы от реализации единицы продукции			
		P_1	P_2	P_3	P_4		C_{P_1}	C_{P_2}	C_{P_3}	C_{P_4}
19	S_1	1	20	0	2	150	40	71	13	37
	S_2	5	15	5	4	160				
	S_3	1	3	1	1	50				
	S_4	14	3	6	10	250				
	S_5	0	0	1	0	10				
	S_6	1	3	0	2	50				
20	S_1	2	1	1	-	25	6	5	5	8
	S_2	1	1	1	-	14				
	S_3	0	4	2	-	19				
	S_4	3	0	1	-	24				
	S_5	0	0	1	-	4				
21	S_1	1	1	0	0	10	7	5	9	8
	S_2	0	2	1	0	15				
	S_3	0	1	2	1	20				
	S_4	3	1	1	1	30				
	S_5	1	2	2	3	35				
	S_6	1	1	3	2	35				
	S_7	3	0	3	3	24				
22	S_1	15	20	13	-	170	2	5	9	-
	S_2	17	19	21	-	200				
	S_3	0	5	7	-	50				
	S_4	60	23	0	-	600				
23	S_1	1	1	2	0	16	16	23	19	10
	S_2	2	1	1	1	25				
	S_3	0	3	2	0	20				
24	S_1	80	15	20	34	500	4	2	3	3
	S_2	42	54	30	51	600				
	S_3	15	19	32	12	540				
	S_4	47	74	23	32	800				

Таблица 4

Варианты 25-30

Вариант	Математическая модель
25	$3x_1 - 8x_2 - 2x_3 + 2x_4 - 4x_5 \to max,$ $\begin{cases} x_1 - 4x_2 + x_3 - x_4 - x_5 = -22, \\ -6x_1 + 3x_2 + x_3 + x_4 = 6, \\ 2x_1 + 2x_2 - x_3 + x_5 = 17, \\ x_i \geq 0, i = 1, 2, 3, 4, 5. \end{cases}$
26	$11x_1 - 4x_2 + 2x_3 - 5x_4 + 2x_5 \to min,$ $\begin{cases} -x_1 + 7x_2 - 2x_3 - 2x_4 + x_5 = 5, \\ 2x_1 + 2x_2 - x_3 + x_5 = 18, \\ 2x_1 - 4x_2 + x_3 + x_4 = 8, \\ x_i \geq 0, i = 1, 2, 3, 4, 5. \end{cases}$
27	$x_1 + 2x_2 + 2x_3 \to min,$ $\begin{cases} x_1 + x_2 - 4x_3 \geq 1, \\ x_1 - 2x_2 + 2x_3 = 2, \\ x_1 + 2x_2 - 2x_3 \leq 6, \\ x_i \geq 0, i = 1,2,3. \end{cases}$
28	$3x_1 + 4x_2 + 5x_3 \to max,$ $\begin{cases} x_1 - x_2 + x_3 = 4, \\ -x_1 + x_2 + 2x_3 \leq 8, \\ -3x_1 + x_2 + 2x_3 \geq 12, \\ x_i \geq 0, i = 1,2,3. \end{cases}$
29	$-x_1 - 7x_2 - 8x_3 + x_4 + 4x_5 \to max,$ $\begin{cases} x_1 + x_2 - x_3 - x_4 - x_5 = 1, \\ -x_1 + x_2 + 2x_3 - x_4 - 2x_5 = 4, \\ x_{1,2,3,4,5} \geq 0. \end{cases}$
30	$19x_1 + 4x_2 - 2x_3 - 7x_4 + x_5 \to min,$ $\begin{cases} -x_1 + 7x_2 - 2x_3 - 2x_4 + x_5 \geq 15, \\ 2x_1 - x_3 + x_5 = 18, \\ 2x_1 - 4x_2 + 11x_3 + 9x_4 \leq 5, \\ x_i \geq 0, i = 1, 2, 3, 4, 5. \end{cases}$

3.8. *Дополнительное задание*

Мебельный комбинат выпускает книжные полки A из натурального дерева со стеклом, полки B_1 из полированной ДСП (древесно-стружечной плиты) без стекла и полки B_2 из полированной ДСП со стеклом. Габариты полок A, B_1 и B_2 следующие: длина d мм, ширина w мм, высота h мм. Размер листа ДСП $2x3$ м.

При изготовлении полок A выполняются следующие работы: столярные, покрытие лаком, сушка, резка стекла, упаковка. Все операции, производимые в ходе столярных работ и упаковки, выполняются вручную. Полки B_1 и B_2 поставляются в торговую сеть в разобранном виде. За исключением операции упаковки, все остальные операции (производство комплектующих полки, резка стекла) при изготовлении полок B_1 и B_2, выполняются на специализированных автоматах.

Трудоемкость столярных работ по выпуску одной полки A составляет Тр_1 ч. Производительность автомата, покрывающего полки A лаком – Пр_1 полок в час, автомата, режущего стекло – Пр_2 стекол в час. Сменный фонд времени автомата для покрытия лаком – ФВ_1 ч, автомата для резки стекла – ФВ_2 ч. Сушка полок, покрытых лаком, происходит в течение суток в специальных сушилках, вмещающих V_1 полок. На упаковку полки A требуется Тр_2 минуты. В производстве полок заняты Р_1 столяров и Р_2 упаковщиков.

Производительность автомата, производящего комплектующие полок B_1 и B_2, равна Пр_3 полки в час, а его сменный фонд времени равен ФВ_3 ч, трудоемкость упаковочных работ составляет Тр_3 мин для полки B_1 и Тр_4 мин для полки B_2.

От поставщиков комбинат получает в месяц Z_1 листов полированной ДСП, Z_2 листов ДВП (древесно-волокнистой плиты), а также Z_3 листов стекла. Из каждого листа ДВП можно выкроить К_1 задних стенок полок B_1 и B_2, а из каждого листа стекла – К_2 стекол для полок B_1 и B_2.

Склад готовой продукции может разместить не более V_2 полок и комплектов полок, причем ежедневно в торговую сеть вывозится в среднем N полок и комплектов. На начало текущего месяца на складе осталось Ост полок, произведенных ранее. Себестоимость полки A равна C_1 руб., полки B без стекла – C_2 руб., со стеклом – C_3 руб.

Маркетинговые исследования показали, что доля продаж полок обоих видов со стеклом составляет не менее Д в общем объеме продаж, а емкость рынка полок производимого типа составляет около V_3 штук в месяц. Мебельный комбинат заключил договор на поставку заказчику З полок типа B_2 в текущем месяце.

Составьте план производства полок на текущий месяц. Известны цены реализации полок: полка A – Ц_1 руб., полка B без стекла – Ц_2 руб., полка B со стеклом – Ц_3 руб.

Таблица 5

Варианты

Вариант	1	2	3	4	5	6	7	8
d	1100	1070	1180	990	1220	950	1300	870
w	250	240	270	240	260	230	260	230
h	300	290	260	250	240	310	340	350
Тр_1	4	4,4	3,2	5,2	2,8	5,6	2	6,4
Тр_2	4	10	6	8	7	5	9	7
Тр_3	8	15	9	13	10	8	15	14
Тр_4	10	16	10	14	11	9	18	16
Р_1	40	22	27	16	9	25	30	14
Р_2	14	16	7	5	13	3	10	2
Пр_1	10	4	2	6	4	7	5	8
Пр_2	100	150	180	130	190	120	210	140
Пр_3	3	4	7	8	9	10	13	14
ФВ_1	7	7,1	7,4	7,5	7,6	7,7	7,0	7,3
ФВ_2	7,5	7,6	7,1	7,2	7,3	7,4	7,7	7,1
ФВ_3	7,4	7,5	7,8	7,4	7,5	7,6	7,4	7,5
Z_1	400	390	415	370	405	350	385	420
Z_2	230	240	215	200	195	180	175	140
Z_3	260	200	240	180	230	290	210	270
К_1	14	15	6	17	7	12	18	11
К_2	10	11	13	6	14	7	16	9
V_1	50	20	55	75	45	60	25	30
V_2	350	400	370	310	380	320	410	340
V_3	5300	2000	1100	4000	2500	1500	4300	3100
N	40	45	72	55	44	60	30	70
Ост	100	110	80	160	70	150	50	120
Д	60 ($\text{А},\text{В}_2$)	15 (А)	43 ($\text{А},\text{В}_1$)	72 (А)	12 (В_2)	16 ($\text{В}_1,\text{В}_2$)	59 (В_1)	13 ($\text{В}_1,\text{В}_2$)
З	50 (В_2)	30(А)	5(А), 12(В_2)	40(В_1), 3(В_2)	60(В_2)	24(А)	38(А), 62(В_2)	23(В_1), 20(В_2)
C_1	205	210	150	215	170	220	180	230
C_2	142	150	120	187	125	176	143	207
C_3	160	170	134	205	148	197	162	214
Ц_1	295	256	192	243	198	274	224	276
Ц_2	182	202	154	230	175	246	214	287
Ц_3	220	224	147	243	180	242	202	246
3 варианта раскроя листов ДСП; работа в 1 смену; 8 ч в смене; 22 рабочих дня в месяце.								

Контрольные вопросы

1. Что такое математическое и линейное программирование?
2. Какова общая форма записи модели ЛП?
3. Что такое допустимое и оптимальное решения?
4. Каковы основные этапы построения математической модели ЛП?
5. Каковы основные этапы решения задач ЛП в *MS Excel*?
6. Каков вид и способы задания формул для целевой ячейки и ячеек левых частей ограничений?
7. В чем смысл использования символа $ в формулах *MS Excel*?
8. В чем различие использования в формулах *MS Excel* символов ; и :?
9. Почему при вводе формул в ячейки ЦФ и левых частей ограничений в них отображаются нулевые значения?
10. Каким образом в *MS Excel* задается направление оптимизации ЦФ?
11. Какие ячейки экранной формы выполняют иллюстративную функцию, а какие необходимы для решения задачи?
12. Как наглядно отобразить в экранной форме ячейки, используемые в конкретной формуле, с целью проверки ее правильности?
13. Поясните общий порядок работы с окном *Поиск решения*.
14. Каким образом можно изменять, добавлять, удалять ограничения в окне *Поиск решения*?
15. Какие сообщения выдаются в *MS Excel* в случаях: успешного решения задачи ЛП; несовместности системы ограничений задачи; неограниченности ЦФ?
16. Объясните смысл параметров, задаваемых в окне *Параметры поиска решения*.
17. Что такое распределительная задача, общая распределительная задача?

4. Транспортные модели

4.1. Общие сведения

Среди задач линейной оптимизации могут быть выделены два класса задач со специальной структурой: транспортная задача и задача о назначениях.

Транспортные задачи (модели) – специальный класс задач линейного программирования. Эти модели часто описывают перемещение какого-либо товара из пункта отправления в пункт назначения. Назначение транспортной задачи – определить объем перевозок из пунктов отправления в пункты назначения с минимальной суммарной стоимостью перевозок. При этом должны учитываться ограничения, налагаемые на объемы грузов, имеющихся в пунктах отправления (предложения), и ограничения, учитывающие потребность грузов в пунктах назначения (спрос). В транспортной модели предполагается, что стоимость перевозки по какому-либо маршруту прямо пропорциональна объему груза, перевозимого по этому маршруту.

В общем случае транспортную модель можно применять для описания ситуаций, связанных с управлением запасами, управлением движением капиталов, составлением расписания, назначением персонала, определения оптимальной специализации предприятий, рабочих участков и станков, оптимального назначения кандидатов на работы, оптимального использования торговых агентов и др.

Критерием эффективности в данных задачах является линейная функция, ограничения также линейны, поэтому для их решения могут применяться методы линейной оптимизации, например симплекс-метод. Хотя транспортная задача может быть решена как обычная задача линейного программирования, ее специальная структура позволяет разобрать алгоритм с упрощенными вычислениями, основанный на симплексных отношениях двойственности.

Мы рассматриваем случай, когда перевозится один продукт, такая задача называется *однопродуктовой*. Однако транспортную задачу можно сформулировать и для многопродуктового варианта.

Решать транспортную задачу можно любым универсальным методом линейного программирования, например симплексным. При этом нужно иметь в виду, что реальные задачи имеют большую размерность. Поэтому имеет смысл разрабатывать более эффективные методы, учитывающие специфический вид условий транспортной задачи.

4.2. Свойства транспортной задачи

Свойство 1 (разрешимость). Сбалансированная транспортная задача всегда имеет решение.

Доказательство. Для разрешимости любой задачи линейного программирования необходимо доказать:
а) непротиворечивость условий;
б) ограниченность целевой функции.

а) для доказательства непротиворечивости условий достаточно указать хотя бы один план.

Пусть $\sum a_i = \sum b_j = D$. Возьмем $x_{ij} = \frac{a_i b_j}{D} \geq 0$. Тогда

$$\sum_j x_{ij} = \sum_j \frac{a_i b_j}{D} = \frac{a_i}{D} \sum b_j = a_i,$$

$$\sum_i x_{ij} = \sum_i \frac{a_i b_j}{D} = \frac{b_j}{D} \sum a_i = b_j;$$

б) целевая функция ограничена в сторону минимизации, так как

$$f(x) = \sum_i \sum_j c_{ij} x_{ij} \geq 0.$$

Свойство доказано.

Свойство 2 (размерность). Ранг матрицы условий равен $m + n - 1$, т. е. опорные планы транспортной задачи могут содержать не более $m + n - 1$ ненулевых компонент.

Доказательство. Для доказательства необходимо показать:
а) что все миноры порядка $m + n$ равны 0;
б) что существует минор порядка $m + n - 1$, не равный 0.

а) возьмем любой минор порядка $m + n$. Его матрица имеет $m + n$ строк, причем из структуры векторов условий следует, что в каждом столбце обязательно будет по единице в первой группе из m строк и во второй группе из n строк.

Если мы в первой группе строк прибавим все строки к первой, а во второй все строки к последней, то получим матрицу, у которой первая и последняя строки состоят из единиц. Определитель такой матрицы равен нулю, поскольку он имеет две одинаковые строки:

$$
\begin{array}{l}
m\left\{\begin{array}{c}\\ \\ \\ \\ \\ \\ \\ \end{array}\right.\\
n\left\{\begin{array}{c}\\ \\ \\ \\ \\ \\ \\ \end{array}\right.
\end{array}
\underbrace{\left\|\begin{array}{ccc}
0 & \cdots & 0\\
\cdots & \cdots & \cdots\\
1 & \cdots & 0\\
\cdots & \cdots & \cdots\\
0 & \cdots & 1\\
\cdots & \cdots & \cdots\\
0 & \cdots & 0\\
\hline
0 & \cdots & 0\\
\cdots & \cdots & \cdots\\
0 & \cdots & 1\\
\cdots & \cdots & \cdots\\
1 & \cdots & 0\\
\cdots & \cdots & \cdots\\
0 & \cdots & 0
\end{array}\right\|}_{m+n}
\qquad
\underbrace{\left\|\begin{array}{ccc}
1 & \cdots & 1\\
\cdots & \cdots & \cdots\\
1 & \cdots & 0\\
\cdots & \cdots & \cdots\\
0 & \cdots & 1\\
\cdots & \cdots & \cdots\\
0 & \cdots & 0\\
\hline
0 & \cdots & 0\\
\cdots & \cdots & \cdots\\
0 & \cdots & 1\\
\cdots & \cdots & \cdots\\
1 & \cdots & 0\\
\cdots & \cdots & \cdots\\
1 & \cdots & 1
\end{array}\right\|}_{m+n}
\qquad 0.
$$

б) возьмем специфический минор порядка $m+n$:

$$
|\vec{P}_{1n}, \vec{P}_{2n}, \ldots, \vec{P}_{mn}, \vec{P}_{11}, \vec{P}_{12}, \ldots, \vec{P}_{1n}| =
$$

$$
\left|\begin{array}{cccccccc}
1 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1\\
0 & 1 & \cdots & 0 & 0 & 0 & \cdots & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0\\
\hline
0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0\\
0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
1 & 1 & \cdots & 1 & 0 & 0 & \cdots & 1
\end{array}\right\| \qquad 0.
$$

После вычеркивания последней строки и последнего столбца получится треугольный минор порядка $m+n-1$, равный 1.

Этот результат является вполне предсказуемым, так как не все ограничения являются независимыми из-за условия баланса производства и потребления.

Свойство 3 (унимодулярность). Любой минор матрицы условий транспортной задачи равен 0 либо ± 1. Такая матрица называется унимодулярной.

Доказательство индукцией по размерности k.

При $k\ = 1$ утверждение очевидно.

Пусть утверждение справедливо для $1, \ldots, k-1$. Докажем его для k.

Возьмем минор k-го порядка. Возможны три варианта.

- Все столбцы его матрицы содержат точно по две единицы. Этот минор равен нулю.
- Найдется столбец, не содержащий ни одной единицы. Такой минор равен 0.
- Найдется столбец, содержащий одну единицу. Раскладываем минор по этому столбцу. В силу индуктивного предположения утверждение справедливо.

Свойство 4 (целочисленность). Если объемы производства и потребления целочисленны или кратны Q, то все опорные планы транспортной задачи, в том числе оптимальный план, также целочисленны или кратны Q.

Доказательство. Пусть имеется опорный план X транспортной задачи. Для $m+n-1$ его ненулевых компонент справедливо разложение

$$\sum_{ij} x_{ij}\vec{P}_{ij} = \vec{P}_0,$$

причем векторы $\vec{P}_{ij}$ линейно независимы.

Это – векторная запись системы m + n линейных уравнений с $m+n-1$ неизвестными. Ее можно решить по правилу Крамера и найти компоненты плана $X = \left(x_{ij}\right)$.

Выберем в матрице условий базисный минор B размерности $m+n-1$ и заменим в нем столбец P_{ij} вектором свободных членов. Тогда, раскладывая определитель по столбцу свободных членов, получаем

$$x_{ij} = \frac{\left\| \begin{array}{ccccc} 0 & \dots & a_1 & \dots & 0 \\ 1 & \dots & a_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & a_m & \dots & 1 \\ \hline 0 & \dots & b_1 & \dots & 0 \\ 1 & \dots & b_2 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & b_n & \dots & 1 \end{array} \right\|}{\|B\|} = \frac{\sum_i \alpha_i a_i + \sum_j \beta_j b_j}{\pm 1},$$

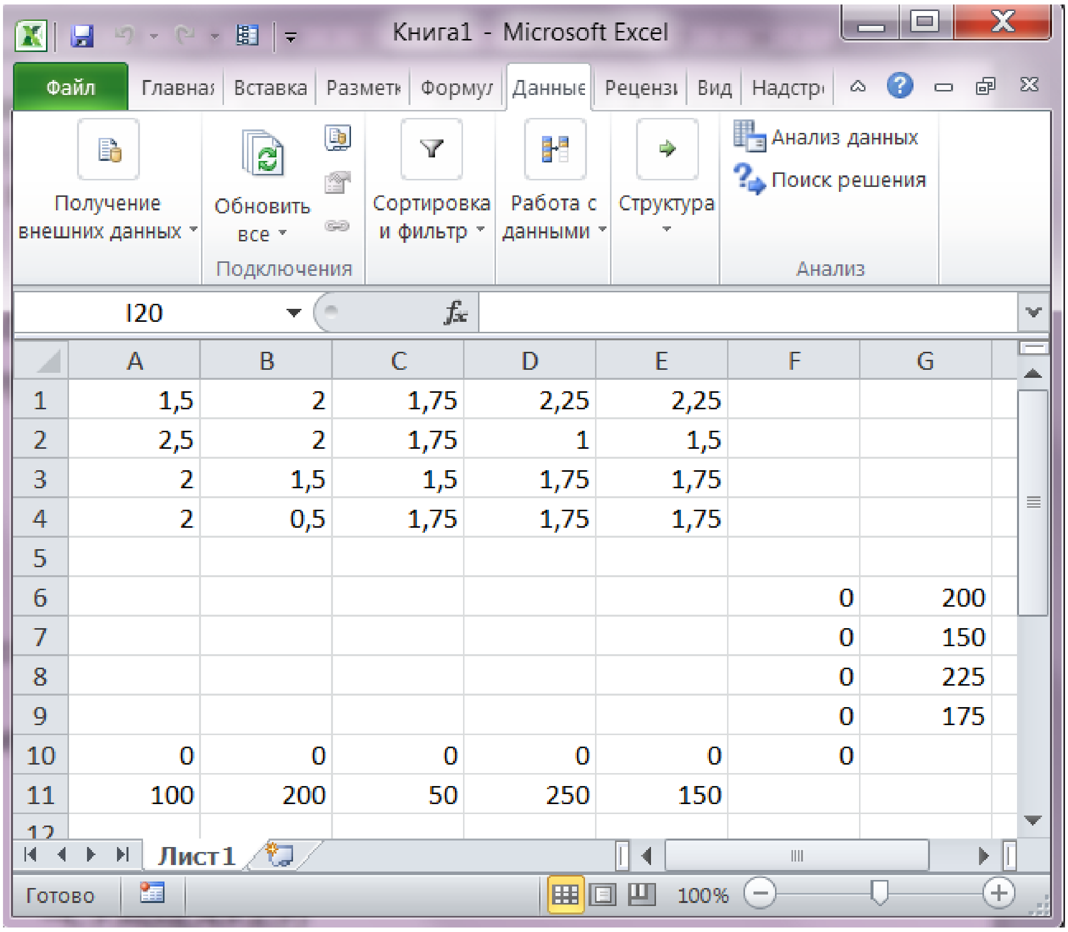
где $\alpha_i, \beta_j \in \{0, +1, -1\}$.

4.3. Пример решения транспортной задачи в Microsoft Excel с использованием модуля Поиск решения

Для решения транспортной задачи с помощью средства *Поиск решений* введем данные.

В ячейки *A1:E4* введены стоимости перевозок. Ячейки *A6:E9* отведены под значения неизвестных (объемы перевозок). В ячейки *G6:G9* введены объемы производства на предприятиях, а в ячейки *A11:E11* введена потребность в продукции в пунктах распределения. В ячейку *F10* введена целевая функция

=СУММПРОИЗВ(A1:E4;A6:E9) (рис. 4.1).



	A	B	C	D	E	F	G
1	1,5	2	1,75	2,25	2,25		
2	2,5	2	1,75	1	1,5		
3	2	1,5	1,5	1,75	1,75		
4	2	0,5	1,75	1,75	1,75		
5							
6						0	200
7						0	150
8						0	225
9						0	175
10	0	0	0	0	0	0	
11	100	200	50	250	150		

Рис. 4.1. Исходные данные

В ячейки *A10:E10* введены формулы:

=СУММ(A6:A9)
=СУММ(B6:B9)
=СУММ(C6:C9)
=СУММ(D6:D9)
=СУММ(E6:E9)

Эти формулы определяют объем продукции, ввозимой в центры распределения.

В ячейки *F6:F9* введены формулы:

=СУММ(A6:E6)
=СУММ(A7:E7)
=СУММ(A8:E8)
=СУММ(A9:E9)

По этим введенным формулам вычисляют объем продукции, вывозимой с предприятий.

Теперь заполним диалоговое окно *Поиск решения*. Находим оптимальный план поставок продукции и соответствующие ему транспортные расходы (рис. 4.2).

5							
6	100	25	50	25	0	200	200
7	0	0	0	150	0	150	150
8	0	0	0	75	150	225	225
9	0	175	0	0	0	175	175
10	100	200	50	250	150	975	
11	100	200	50	250	150		
12							

Лист1

Готово 100%

Рис. 4.2. Оптимальный план

4.4. Пример решения транспортной задачи с помощью Mathcad

Постановка транспортной задачи. Пусть имеется m источников финансирования $A_1, A_2, \ldots, A_m$ и n периодов финансирования $B_1, B_2, \ldots, B_n$. Известны затраты, связанные с выделением единицы денежных ресурсов C_{ij} из i-го источника в j-м периоде, а также объемы финансирования из каждого i-го источника в течение всего времени – a_i. Известны суммарные объемы финансирования из всех источников в каждый j-й промежуток времени – b_j. Требуется определить объемы финансирования X_{ij} из i-го источника в j-м периоде, чтобы:

1. Ресурсы всех источников были реализованы.
2. Обеспечить финансирование в полном объеме в каждом периоде.
3. Достигнуть экстремума выбранного критерия оптимизации.

Введите в рабочем листе поясняющий текст.

Далее введите критерий оптимизации – целевую функцию. Для этого вначале разместите курсор (визир – красный крестик) в месте ввода математического выражения. Затем с помощью соответствующих клавиш начните ввод; в первую очередь – имени критерия оптимизации с аргументами в скобках через запятые:

Y (X11, X12, X13, X21, X22, X23, X31, X32, X33).

Затем нажмите комбинацию клавиш <Shift>+<:> для ввода знака присваивания :=. На месте правой метки поместите все выражение критерия оптимизации. Начальные приближения вводятся аналогично.

Для решения задачи используем блок функций *Given ... Minimize*, выполнив следующие операции:

- ✓ введите, если необходимо, комментарии, воспользовавшись клавишей с двойной кавычкой;
- ✓ введите ключевое слово *Given*;
- ✓ введите систему ограничений, используя при вводе знак равенства, вызванный нажатием комбинации клавиш <Ctrl>+<=>;
- ✓ введите граничные значения;
- ✓ введите вектор-столбец искомых параметров, используя диалоговое окно *Вставить матрицу*. Для этого нужно выбрать левую верхнюю кнопку на панели инструментов *Матрица* или нажмите комбинацию клавиш <Ctrl>+<M>. В появившемся диалоговом окне в поле *Строки* число строк должно быть равно 9, а в поле *Столбцы* – 1;
- ✓ введите знак присваивания, нажав комбинацию клавиш <Shift>+<:>;
- ✓ используя диалоговое окно *Вставить функцию*, введите функцию *Minimize* с искомыми параметрами, нажав для этого комбинацию клавиш <Ctrl>+<E>;
- ✓ скопируйте и вставьте вектор-столбец искомых параметров и введите знак «равно».

На рис. 4.3 показан процесс оптимизации распределения однородных ресурсов с помощью *Mathcad*.

Это математическое описание конкретной транспортной задачи в нотации системы. Представлен критерий оптимизации, начальные приближения и граничные условия. В описании двух первых пунктов использован знак присваивания “:=”. Он вводится щелчком мыши по второй кнопке в первой строке панели *Вычисление*, если она была заранее выведена на рабочий лист.

Следует обратить внимание на представление системы ограничений в *Mathcad*. При ее написании используется жирный знак равенства, вызываемый щелчком по кнопке с аналогичным знаком – второй в первом столбце панели инструментов *Вычисление*.

Оптимальное распределение однородных ресурсов зафиксировано в векторе (X11 X12 X13 ...). Из полученного решения видно, что X11 = 4, X12 = 2, X13 = 8, X21= 0, X22 = 20, X23 = 0, X31 = 26, X32 =0, X33 = 0. Это означает, что источник 1 должен профинансировать в первом периоде 4 единицы, во втором и в третьем – 8 единиц. Источник 2: в первом периоде 0 единиц, во втором 20 и в третьем – 0 единиц. Источник 3: в первом периоде 26 единиц, во втором и в третьем финансирование отсутствует. Первая цифра в переменной X определяет источник, а вторая – период финансирования. Такое распределение денежных средств из источников обеспечит минимальные суммарные затраты Y, которые составят 1402000 единиц.

Критерий оптимизации – целевая функция (транспортная задача)

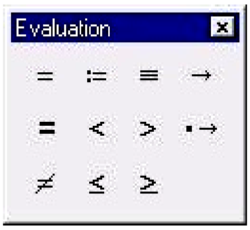
$$Y(X11,X12,X13,X21,X22,X23,X31,X32,X33) := 70\cdot X11 + 38\cdot X12 + 24\cdot X13 \;\ldots$$
$$58\cdot X21 + 18\cdot X22 + 56\cdot X23 \;\ldots$$
$$19\cdot X31 + 10\cdot X32 + 100\cdot X33$$

Начальные приближения

$X11 := 0 \quad X12 := 0 \quad X13 := 0 \quad X21 := 0 \quad X22 := 0 \quad X23 := 0 \quad X31 := 0 \quad X32 := 0 \quad X33 := 0$

Given

Система ограничений



$X11 + X12 + X13 = 14 \qquad X11 + X21 + X31 = 30$

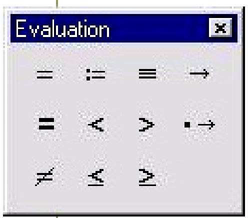
$X21 + X22 + X23 = 20 \qquad X12 + X22 + X32 = 22$

$X31 + X32 + X33 = 26 \qquad X13 + X23 + X33 = 8$

Граничные условия

$X11 \ge 0 \; X12 \ge 0 \; X13 \ge 0 \; X21 \ge 0 \; X22 \ge 0 \; X23 \ge 0 \; X31 \ge 0 \; X32 \ge 0 \; X33 \ge 0$

Найти оптимальное решение



$$\begin{bmatrix} X11 \\ X12 \\ X13 \\ X21 \\ X22 \\ X23 \\ X31 \\ X32 \\ X33 \end{bmatrix} := \mathrm{Minimize}(Y, X11, X12, X13, X21, X22, X23, X31, X32, X33)$$

$$\begin{bmatrix} X11 \\ X12 \\ X13 \\ X21 \\ X22 \\ X23 \\ X31 \\ X32 \\ X33 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \\ 8 \\ 0 \\ 20 \\ 0 \\ 26 \\ 0 \\ 0 \end{bmatrix}$$

Определить величину целевой функции для оптимального решения

$$Y(X11,X12,X13,X21,X22,X23,X31,X32,X33) = 1.402\cdot 10^{3}$$

Рис. 4.3. Процесс оптимизации

4.5. *Задача об оптимальном распределении ресурсов при выпуске продукции на предприятии (об ассортименте)*

На товарных станциях C_1 и C_2 имеется по 30 станков. Известно, что перевозка одного станка со станции C_1 в университеты Y_1, Y_2, Y_3 стоит 1000 руб., 3000 руб., 5000 руб. соответственно, а стоимость перевозки со станции C_2 в те же университеты – 2000 руб., 5000 руб., 4000 руб. необходимо доставить в каждый университет по 20 станков. Составить план перевозок так, чтобы затраты на транспортировку станков были наименьшими.

Математическая постановка задачи.

Обозначим количество станков, перевозимых со станции C_1 в университеты Y_1, Y_2, Y_3 через x_1, x_2, x_3 а со станции C_2 – через x_4, x_5, x_6. Тогда схема перевозок буде выглядеть следующим образом:

	В Y_1	В Y_2	В Y_3	Всего отправлено
Из C_1	x_1	x_2	x_3	30
Из C_2	x_4	x_5	x_6	30
Всего получено	20	20	20	60

В соответствии с условием задачи ($x_i \geq 0$, x_i – целые, $i = 1, \dots, 6$). Задача сводится к тому, чтобы найти такое неотрицательное целочисленное решение системы

$$\begin{cases} x_1 + x_2 + x_3 = 30, \\ x_4 + x_5 + x_6 = 30, \\ x_1 + x_4 = 20, \\ x_2 + x_5 = 20, \\ x_3 + x_6 = 20, \end{cases}$$

при котором линейная функция (стоимость перевозок)

$$f = 1000x_1 + 3000x_2 + 5000x_3 + 2000x_4 + 5000x_5 + 4000x_6 \to min$$

имеет наименьшее значение.

Решение задачи в программе *MS Excel*:

1. Надо заполнить ячейки *A1:A6* таблицы обозначениями x_1, x_2, x_3, x_4, x_5, x_6, а ячейку *A7* – *min*.
2. В ячейку *B7* записать целевую функцию.
3. В диапазон ячеек *A10:C14* записать систему ограничений через адреса соответствующих ячеек.
4. Заполнить форму *Поиск решения*.

Решение задачи представлено на рис. 4.4.

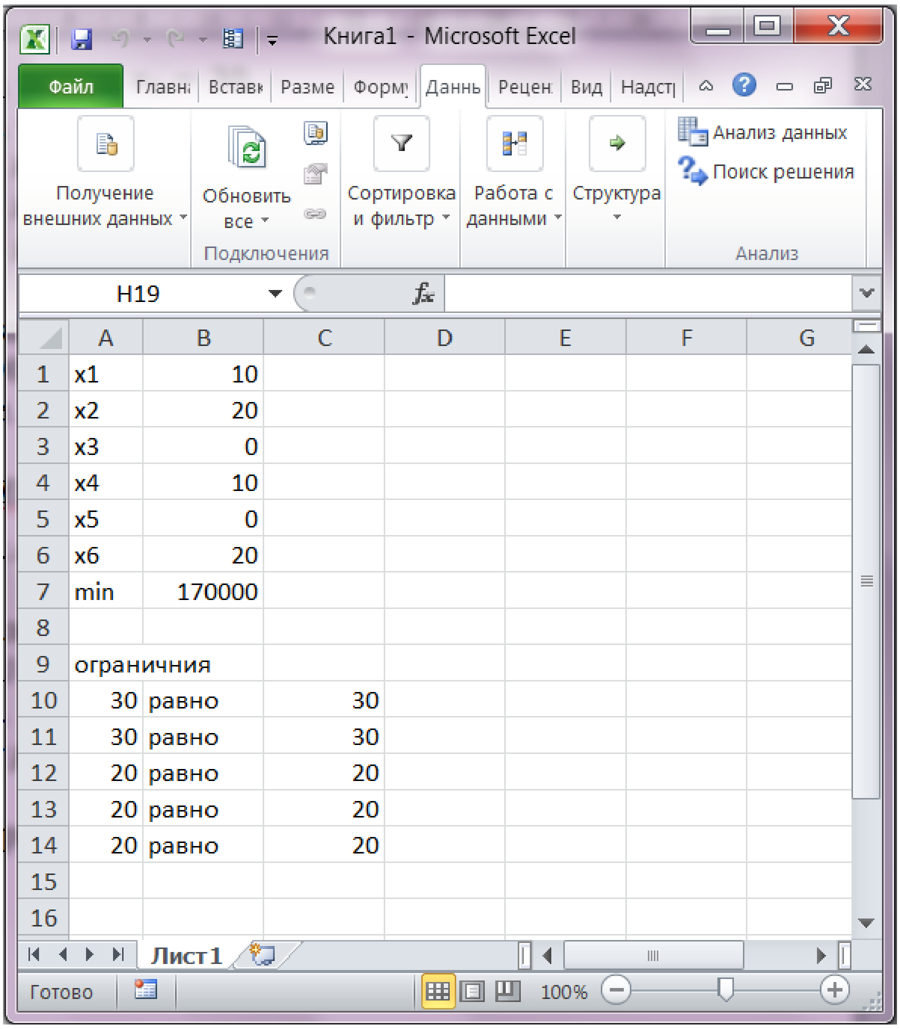


Рис. 4.4. Решение задачи в *MS Excel*

Решение задачи в среде *MathCad* приведено на рис. 4.5.

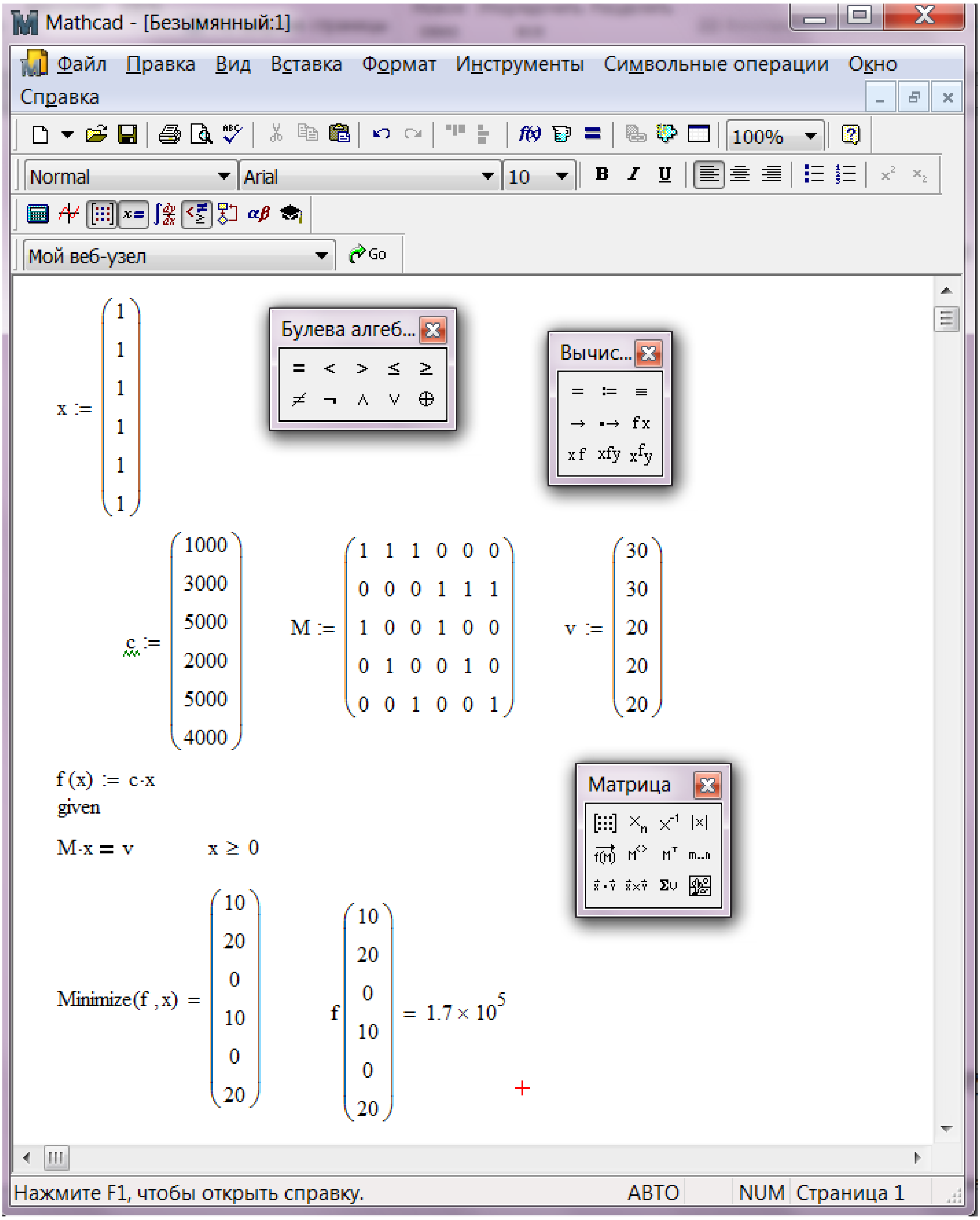


Рис. 4.5. Решение задачи в *Mathcad*

4.6. Задание к лабораторной работе

Некоторый однородный продукт, сосредоточенный у трёх поставщиков A_1, A_2 и A_3 в количестве a_1, a_2 и a_3 тонн соответственно, необходимо доставить потребителям B_1, B_2, B_3, B_4 и B_5 в количестве b_1, b_2, b_3, b_4 и b_5 тонн. Стоимость C_{ij} перевозки тонны груза от i-го поставщика j-му потребителю задана матрицей D. Составить план перевозок, имеющий минимальную стоимость и позволяющий вывести все грузы и полностью удовлетворить потребности.

Вариант 1

a_i	b_j				
	150	50	250	50	200
100	2	10	8	8	5
250	9	17	15	14	11
350	10	20	15	20	13

Вариант 2

a_i	b_j				
	300	200	50	150	100
250	5	3	15	1	10
250	10	10	20	6	15
300	13	10	22	8	7

Вариант 3

a_i	b_j				
	200	50	200	50	100
50	3	1	1	1	2
200	5	3	3	3	6
350	17	16	15	16	16

Вариант 4

a_i	b_j				
	350	200	100	50	50
100	2	1	5	1	3
350	4	3	7	5	5
300	6	6	8	6	8

Вариант 5

a_i	b_j				
	160	70	90	80	100
150	8	20	7	11	16
200	4	14	12	15	17
150	15	22	11	12	19

Вариант 6

a_i	b_j				
	20	20	30	30	20
20	2	4	8	2	2
30	4	6	10	3	4
50	2	5	9	7	2

Пусть на предприятии имеется m видов станков, максимальное время работы которых соответственно равно a_i часов. Каждый из станков может выполнять n видов операций. Суммарное время выполнения каждой операции соответственно b_j. Известна производительность C_{ij} i-го станка при выполнении j-й операции. Определить, сколько времени и на какой операции нужно использовать каждый из станков, чтобы разработать максимальное количество деталей.

Вариант 7

a_i	b_j			
	30	25	35	10
10	7	2	4	6
20	5	1	3	7
30	1	9	4	8
25	2	11	7	1
15	6	4	3	5

Вариант 8

a_i	b_j				
	5	5	10	10	5
5	3	4	6	5	13
5	6	3	7	6	10
10	10	5	2	2	6
15	9	4	4	9	5
10	4	6	2	3	4

Вариант 9

a_i	b_j			
	20	15	45	40
30	8	7	3	5
50	4	2	1	6
40	9	7	8	4

Вариант 10

a_i	b_j				
	10	10	5	8	7
7	4	6	8	3	2
13	5	3	4	6	4
20	3	2	5	7	5

Вариант 11

a_i	b_j			
	20	30	30	20
23	4	3	6	5
38	3	4	5	6
39	2	5	4	7

Вариант 12

a_i	b_j			
	100	100	150	150
100	2	1	3	4
150	4	3	1	7
250	5	8	9	15

Вариант 13

a_i	b_j			
	11	7	8	4
9	2	5	8	1
16	8	3	9	2
5	7	4	6	3

Вариант 14

a_i	b_j				
	10	10	5	8	7
7	4	6	8	3	2
13	5	3	4	6	4
20	3	2	5	7	5

Вариант 15

a_i	b_j			
	100	200	200	300
100	1	3	4	1
200	5	2	2	7
400	4	4	3	6
200	7	2	5	3

Вариант 16

a_i	b_j				
	10	15	15	10	10
5	3	4	5	4	6
10	1	5	7	1	5
15	4	6	6	3	4
10	2	7	4	7	2

Вариант 17

a_i	b_j			
	200	400	400	800
200	1	6	9	3
400	3	2	2	4
600	4	5	4	7
200	1	4	3	9

Вариант 18

a_i	b_j				
	30	90	60	90	30
30	1	3	4	3	1
60	9	5	2	4	8
90	3	4	7	4	3
60	5	7	2	6	6

Вариант 19

a_i	b_j			
	300	200	300	100
300	3	4	3	1
200	2	3	5	6
100	1	2	3	3
200	4	5	7	9

Вариант 20

a_i	b_j				
	200	300	200	300	100
100	2	3	4	5	1
200	2	4	2	6	7
300	6	5	4	5	4
400	4	6	7	6	9

Вариант 21

a_i	b_j			
	200	300	400	200
200	1	3	4	2
200	1	2	4	1
300	3	4	5	9
300	6	3	7	6

Вариант 22

a_i	b_j				
	40	60	40	60	20
20	4	3	4	2	3
40	1	2	1	5	3
60	4	8	2	9	12
40	5	7	1	3	6

Вариант 23

a_i	b_j			
	1000	500	1500	2000
500	3	1	2	5
1500	1	3	4	2
500	3	6	5	6
1500	2	8	5	7
500	4	3	9	8

Вариант 24

a_i	b_j				
	5	10	15	15	15
10	2	1	3	5	7
5	4	3	4	4	3
5	5	2	3	6	2
10	3	6	5	2	4
15	1	9	7	3	4

Вариант 25

a_i	b_j			
	50	25	50	75
25	3	1	8	1
50	2	5	2	3
75	9	4	6	5
25	7	3	10	3
75	4	6	7	4

Вариант 26

a_i	b_j				
	5	5	10	10	5
5	3	4	6	5	13
5	6	3	7	6	10
10	10	5	2	2	6
15	9	4	4	9	5
10	4	6	2	3	4

Вариант 27

a_i	b_j			
	400	200	150	250
500	6	5	4	-
300	8	8	2	6
100	9	-	7	6

Вариант 28

a_i	b_j			
	60	40	36	14
92	5	1	2	4
45	2	5	-	3
63	-	2	2	5

Вариант 29

a_i	b_j			
	40	30	20	10
30	-	5	4	2
50	2	5	-	3
120	3	2	-	5

Вариант 30

a_i	b_j			
	250	100	150	50
80	5	6	1	4
320	8	-	6	5
100	5	4	3	-

Вариант 31

a_i	b_j			
	50	70	130	150
140	3	-	-	6
160	5	2	3	1
150	1	1	2	4

Вариант 32

a_i	b_j			
	10	80	90	20
100	4	7	1	1
50	5	-	3	4
70	3	-	2	8

Вариант 33

a_i	b_j			
	20	40	80	60
200	3	-	2	1
70	2	3	-	4
80	5	8	7	3

Вариант 34

a_i	b_j			
	300	150	100	200
40	4	3	2	-
200	10	10	4	7
100	12	-	11	5

Вариант 35

a_i	b_j			
	40	60	100	50
100	4	1	2	3
200	3	6	-	4
150	-	2	3	5

Вариант 36

a_i	b_j			
	10	20	40	60
40	2	7	4	3
30	5	-	12	7
50	8	1	-	13

4.7. Дополнительное задание

На складах хранится мука, которую необходимо завезти в хлебопекарни. Номера складов и номера хлебопекарен выбирают в соответствии с вариантами табл.6. Текущие тарифы перевозки муки [руб./т], ежемесячные запасы муки [т/мес.] на складах и потребности хлебопекарен в муке [т/мес.] указаны в табл.7.

При этом необходимо учитывать, что из-за ремонтных работ временно нет возможности перевозить муку с некоторых складов в некоторые хлебопекарни. В табл.6 это показано в графе *Запрет перевозки* в формате № склада x № хлебопекарни. Например, «2x3» обозначает, что нельзя перевозить муку со склада №2 в хлебопекарню №3.

Кроме того, необходимо учесть, что некоторые хлебопекарни имеют договоры на гарантированную поставку муки с определенных складов. В табл.6 это показано в графе *Гарантированная поставка* в формате № склада x № хлебопекарни = объем поставки. Например, «1x4=40» обозначает, что между складом №1 и магазином №4 заключен договор на обязательную поставку 40 т муки.

Необходимо организовать поставки наилучшим образом, учитывая, что мука хранится и транспортируется в мешках массой 50 кг.

Таблица 6

Номера складов, хлебопекарен, запрещенные и гарантированные поставки

№ варианта	№ складов	№ хлебопекарен	Запрет перевозки	Гарантированная поставка, т/мес.
1	1, 2, 3	1, 2, 3, 4	2x2, 3x4	3x3=50
2	2, 3, 4, 5	1, 2, 5	2x2, 3x5	3x2=40
3	1, 2, 4	1, 2, 3, 5	1x5, 2x3	4x3=45
4	1, 2, 3, 4	3, 4, 5	3x3, 4x5	3x5=40
5	1, 2, 5	2, 3, 4, 5	1x4, 5x3	1x5=60
6	1, 2, 3, 5	2, 3, 5	5x5, 2x2	3x5=30
7	2, 3, 4	2, 3, 4, 5	3x3, 2x5	4x3=45
8	1, 2, 3, 5	1, 2, 4	1x2, 5x4	3x2=20
9	2, 3, 5	1, 2, 3, 5	5x1, 3x5	5x2=30
10	2, 3, 4, 5	2, 3, 4	5x4, 3x2	4x3=35
11	3, 4, 5	1, 2, 3, 4	3x4, 5x1	4x1=40
12	1, 2, 3, 4	1, 2, 3	3x2, 4x1	2x2=50

Таблица 7

Запасы, потребности и тарифы перевозок

Склады	Хлебопекарни					
	1	2	3	4	5	Запас, т/мес.
1	400	600	800	200	200	80
2	300	100	500	600	500	70
3	500	200	100	600	300	60
4	300	700	200	400	900	55
5	200	500	800	200	400	65
Спрос, т/мес.	77,86	56,78	58,88	62,44	73,92	

Контрольные вопросы

1. Что такое задача о размещении?
2. Какова постановка стандартной ТЗ?
3. Запишите математическую модель ТЗ.
4. Перечислите исходные и искомые параметры модели ТЗ.
5. Какова суть каждого из этапов построения модели ТЗ?
6. Раскройте понятие сбалансированности ТЗ.
7. Что такое фиктивные и запрещающие тарифы?

5. Задача о назначениях

5.1. Общие сведения

Рассмотрим ситуацию, когда требуется распределить m работ (или исполнителей) по n станкам. Работа i ($i = 1, \dots, m$), выполняемая на станке j ($j = 1, \dots, n$), связана с затратами c_{ij}. Задача состоит в таком распределении работ по станкам (одна работа выполняется на одном станке), которое соответствует минимизации суммарных затрат.

Эту задачу можно рассматривать как частный случай транспортной задачи. Здесь работы представляют «исходные пункты», а станки – «пункты назначения». Предложение в каждом исходном пункте равно 1, т.е. $a_i = 1$ для всех i. Аналогично спрос в каждом пункте назначения равен 1, т.е. $b_j = 1$ для всех j. Стоимость «перевозки» (прикрепления) работы i к станку j равна c_{ij}. Если какую-либо работу нельзя выполнять на некотором станке, то соответствующая стоимость берут равной очень большому числу. Матрицу стоимостей C определяют следующим образом:

$$
\begin{array}{cc}
 & \text{станки} \\
 & \begin{array}{cccccc}
 & 1 & 2 & \dots & n & a_i \\
1 & \left(c_{11}\right. & c_{12} & \dots & \left.c_{1n}\right) & 1 \\
2 & c_{21} & c_{22} & \dots & c_{2n} & 1 \\
\dots & \dots & \dots & \dots & \dots & \dots \\
m & c_{m1} & c_{m2} & \dots & c_{mn} & 1 \\
b_j & 1 & 1 & \dots & 1 &
\end{array} \\
\text{виды работ} &
\end{array}
$$

Прежде чем решать такую задачу необходимо ликвидировать дисбаланс, добавив фиктивные работы или станки в зависимости от начальных условий. Поэтому без потери общности можно положить $m = n$.

5.2. Пример решения задачи о назначениях в Microsoft Excel с использованием модуля Поиск решения

Рассмотрим пример решения задачи о назначениях. Четверо рабочих могут выполнять четыре вида работ. Стоимости c_{ij} выполнения i-м рабочим j-й работы приведены в ячейках диапазона *A1:D4*.

В этой таблице строки соответствуют рабочим, а столбцы – работам. Необходимо составить план выполнения работ так, чтобы все работы были выполнены, каждый рабочий был загружен только на одной работе, а суммарная стоимость выполнения всех работ была минимальной.

Отметим, что данная задача является сбалансированной, т. е. число работ совпадает с числом рабочих. Если задача не сбалансирована, то перед началом решения ее необходимо сбалансировать, введя недостающее число фиктивных строчек или столбцов с достаточно большими штрафными стоимостями работ.

Для решения этой задачи с помощью средства «Поиск решений» отведем под неизвестные диапазон ячеек *F2:I5*. В ячейку *J1* введем целевую функцию

=СУММПРОИЗВ (F2 :I5 ; A1 : D4), вычисляющую стоимость работ.

Введем формулы, задающие левые части ограничений.

Результат работы модуля *Поиск решения* показан на рис. 5.1.

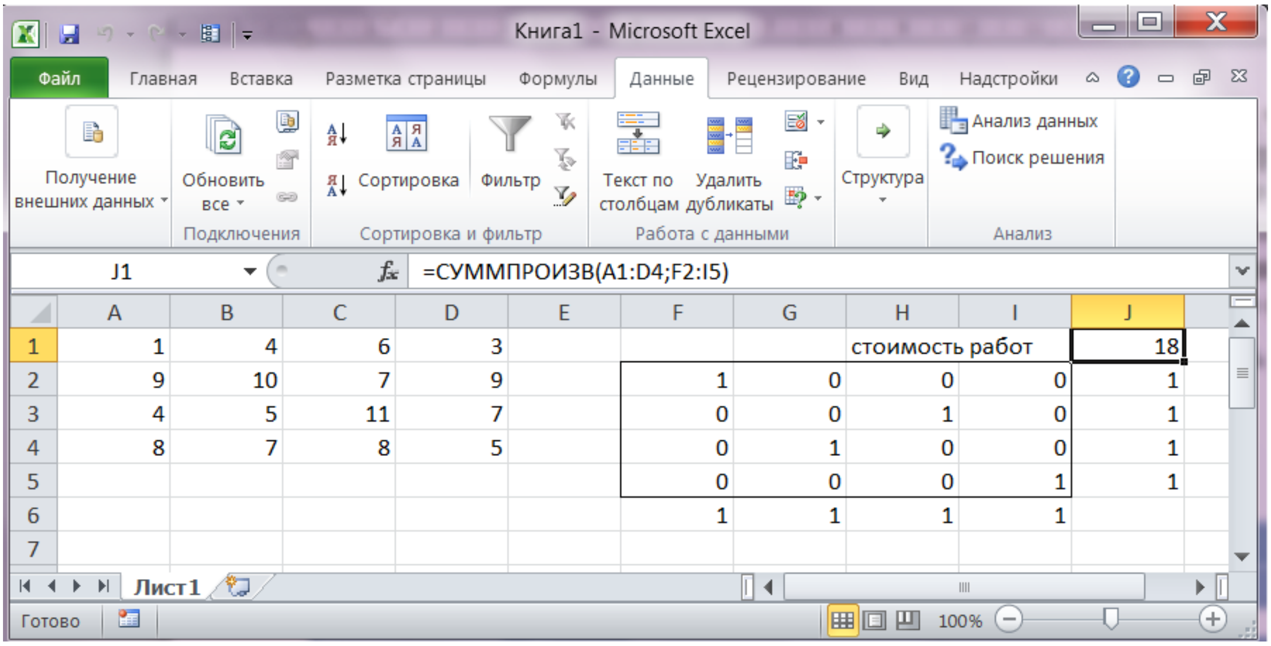


Рис. 5.1. Результат работы модуля *Поиск решения*

5.3. Пример решения задачи организации проверок в компании

Модель назначений часто встречается в разнообразных управленческих ситуациях. В моделях этого типа решается задача нахождения оптимального распределения n *неделимых* агентов или объектов по n заданиям. Это может быть распределение рабочих по цехам, мастеров по вызовам, компьютеров по сетям, назначение консультантов для клиентов. Распределяемые агенты или объекты являются неделимыми, т.е. один агент не может заниматься несколькими задачами. Важное ограничение для агента заключается в том, что он (она или оно) может быть назначен одной и только одной задаче.

В качестве примера модели назначений рассмотрим следующую ситуацию. Штаб-квартира отделения некоторой компании находится в Москве. В этом году президент Европейского отделения компании решил, что в рамках ежегодной ревизии каждый из четырех вице-президентов должен посетить с проверкой один из сборочных заводов компании в течение первых двух недель июня. Сборочные заводы расположены в четырех городах, которые условно назовем: Город 1, Город 2, Город 3, Город 4.

Существует множество моментов, которые нужно учесть, выбирая, какой вице-президент будет проверять конкретный завод. Необходимо принять во внимание следующее.

1. Важно обеспечить соответствие специализации вице-президента и наиболее важных проблем конкретного завода.
2. Нужно соотнести время, которое потребуется на проведение проверки, с другими обязанностями каждого вице-президента в двухнедельном интервале проверок.
3. Вице-президент должен владеть языком, который является основным на проверяемом заводе.

Удачно организовать проверку заводов с учетом всех этих факторов – достаточно сложная задача. Президент решил начать с оценки затрат на командировки. Эти данные представлены в табл. 8.

Таблица 8

Затраты на командировку, тыс. руб.

Специализация вице-президента	**Город (1)**	**Город (2)**	**Город (3)**	**Город (4)**
Финансы (Ф)	24	10	21	11
Маркетинг (М)	14	22	10	15
Производство (П)	15	17	20	19
Персонал (Пл)	11	19	14	13

Исходя из этих данных, президент может оценить любой конкретный вариант командировки вице-президентов на заводы. Например, если он выберет вариант, представленный в табл. 9, общие затраты составят 79 000 руб.

Таблица 9

Вариант командировки вице-президентов на заводы

Вице-президент	**Завод**	**Стоимость**
Ф	1	24
М	2	22
П	3	20
Пл	4	13
Суммарные затраты		79

Решение с помощью полного перебора

Поскольку вице-президентов и заводов всего по 4 и возможных назначений конечное число, можно попытаться осуществить полный перебор, т.е. вычислить затраты для каждого возможного варианта и выбрать вариант с наименьшими затратами. Однако полный перебор всех возможных вариантов очень скоро становится слишком обременительным.

Посмотрим, сколько возможных вариантов назначений в нашей модели. Предположим, что назначения осуществляются в таком порядке: Ф, М, П, Пл. Тогда процесс перебора состоит из следующих этапов.

- Ф можно направить на любой из четырех заводов.
- После того, как назначение Ф состоялось, М можно направить на один из трех оставшихся заводов.
- П можно направить на один из двух оставшихся заводов.
- Пл придется направить на последний завод.

Таким образом, число возможных решений 4x3x2x1 = 24. В общем случае, когда имеется n вице-президентов и n заводов, возможных решений будет $n \cdot (n-1) \cdot (n-2) \cdot (n-3) \cdot \cdots \cdot 2 \cdot 1 = n!$ (n факториал). По мере увеличения n, $n!$ возрастает чрезвычайно быстро. Ниже приводятся значения $n!$ для значений n в пределах от 3 до 20:

n	3	4	5	6	7	8	9	10	20
$n!$	6	24	120	720	5040	40320	362880	3628900	$2.4\text{x}10^{18}$

Предположим, что президенту нужно решить вопрос о назначении 15 торговых представителей. Очевидно, что осуществить полный перебор в этом случае может оказаться не под силу даже с помощью *Excel*.

Данный пример показывает, насколько быстро увеличивается количество вариантов в комбинаторных задачах.

Поскольку небольшие модели иногда проще решать с помощью перебора (или интуиции), то даже при незначительном увеличении их размерности необходимы точные и эффективные методы оптимизации.

Формализация и решение задачи назначения в EXCEL.

Чтобы построить модель для задачи назначений, используем такое же обозначение переменных, как и в транспортной модели.

Пусть x_{ij} – число вице-президентов специализации i, направленных на завод j ($i = \text{Ф}, \text{М}, \text{П}, \text{Пл}$; $j = 1, 2, 3, 4$).

Модель назначений для отделения рассматриваемой в нашей модели компании представлена на рис. 5.2, где также показаны минимальные затраты на командировки.

Поскольку имеется только один вице-президент каждой специализации (предложение) и на каждом заводе требуется только один вице-президент (спрос), модель сбалансирована, так как спрос равен предложению. Следовательно, вместо неравенств можно использовать ограничения в виде равенств.

В табличном редакторе *Excel* числа в ячейках *C18:F21* отражают затраты компании при выборе варианта назначений.

В представленной модели первое ограничение указывает, что число вице-президентов по финансам не должно превышать 1. Аналогичные ограничения касаются и других вице-президентов.

Поэтому все переменные решения в ячейках *C10:F13* могут принимать только значения 0 или 1 (с учетом условия неотрицательности). Ограничение для завода 1 показывает, что на этот завод необходимо прислать по меньшей мере одного вице-президента. Аналогичные ограничения существуют для других заводов.

На рис. 5.2 показано оптимальное решение: все переменные решения принимают значения 0 или 1, а оптимальным является назначение, представленное в табл. 10.

	A	B	C	D	E	F	G	H
1				Модель назначиния				
2								
3		Удельные затраты	Город 1	Город 2	Город 3	Город 4		
4		по финансам	24	10	21	11		
5		по маркетингу	14	22	10	15		
6		по производству	15	17	20	19		
7		по персоналу	11	19	14	13		
8								
9		Назначение	Город 1	Город 2	Город 3	Город 4	всего	имеется
10		по финансам	0	1	0	0	1	<=1
11		по маркетингу	0	0	1	0	1	<=1
12		по производству	1	0	0	0	1	<=1
13		по персоналу	0	0	0	1	1	<=1
14		всего	1	1	1	1		
15		необходимо	>=1	>=1	>=1	>=1		
16								
17		Суммарные затраты	Город 1	Город 2	Город 3	Город 4	всего	
18		по финансам	0	0	0	0	10	
19		по маркетингу	0	0	10	0	10	
20		по производству	15	0	0	0	15	
21		по персоналу	0	10	0	13	13	
22		всего	15	10	10	13	48	
23								

Лист1

Готово 100%

Рис. 5.2 Модель назначений

Таблица 10

Оптимальное назначение

Вице-президент	Завод	Стоимость
Ф	2	10
М	3	10
П	1	15
Пл	4	13
Суммарные затраты, тыс.руб.		48

Связь между моделью назначений и транспортной моделью

Модель назначений является разновидностью транспортной модели. Она отличается только тем, что в ней единица предложения не может распределяться по нескольким местам назначения (в данном примере каждый вице-президент должен отправиться только на один завод).

Напомним, что если в транспортной модели все значения спроса и предложения являются целыми, то оптимальные значения переменных решения также будут целыми.

В модели назначений все значения спроса и предложения равны единице, т.е. целые, поэтому можно не сомневаться, что назначения, найденные средством «Поиск решения», не окажутся дробными.

В общем случае справедливо следующее утверждение:

Модель назначений можно построить в виде транспортной модели, в которой предложение в каждой исходной точке и спрос в каждом конечном пункте равны 1.

Варианты модели назначений

Модель назначений отделения компании – это модель (задача) минимизации, в которой число вице-президентов равно числу заводов и все возможные назначения допустимы. Рассмотрим модели, в которых эти условия не выполняются. В частности, рассмотрим модели, в которых:

- ✓ число назначаемых персон не равно количеству мест назначений (несбалансированная модель);
- ✓ надо максимизировать целевую функцию (а не минимизировать, как в предыдущем примере).

Несбалансированная модель

Рассмотрим два случая. Сначала предположим, что предложение превышает спрос.

Например, президент компании сам решил провести инспекцию завода в Городе №1. Теперь ему нужно решить, кого из четырех вице-президентов направить на три оставшихся завода. Затраты показаны в табл. 11.

Таблица 11

Несбалансированная модель 1

Специализация вице-президента	**Завод**			**Количество вице-президентов**
	1	**2**	**3**	
Ф	24	10	21	1
М	14	22	10	1
П	15	17	20	1
Пл	11	19	14	1
Требуемое число вице-президентов	1	1	1	3/4

Чтобы построить новую модель, исключим ограничение, требующее направить вице-президента на завод 4. В результате в оптимальном решении резерв для одного из четырех ограничений по вице-президентам будет равен 1, т.е. один вице-президент не будет отправлен в командировку на завод.

Теперь рассмотрим ситуацию, когда спрос превышает предложение. Например, вице-президент, отвечающий за работу с персоналом, должен в первые две недели июня находиться в штаб-квартире компании.

В таком случае в задаче будут затраты, представленные в табл. 12.

Таблица 12

Несбалансированная модель 2

Специализация вице-президента	**Завод**				**Количество вице-президентов**
	1	**2**	**3**	**4**	
Ф	24	10	21	11	1
М	14	22	10	15	1
П	15	17	20	19	1
Требуемое число вице-президентов	1	1	1	1	4/3

Задача в такой постановке не имеет допустимых решений.

Очевидно, что невозможно удовлетворить потребность в четырех вице-президентах, имея только трех. Если все же надо выяснить, какие три завода следует проверить, чтобы минимизировать затраты, нужно изменить неравенства, когда спрос превышал предложение, или внести данные о фиктивном вице-президенте в таблицу затрат.

В последнем случае на какой-нибудь завод будет направлен фиктивный вице-президент. В действительности данный завод не будет инспектироваться. Как и в примере транспортной задачи, преимущество второго подхода заключается в том, что можно определить ущерб, если данный завод не будет проинспектирован, а значения ущерба для различных заводов могут отличаться. Новую строку в табличном представлении модели можно обозначить как "Не проверяется" и ввести соответствующие значения в ее ячейки.

Теперь благодаря введению фиктивного предложения предложение равняется спросу. Также отметим, что фиктивному вице-президенту соответствуют нулевые затраты. В любом случае, если спрос превышает предложение, необходимо добавить к модели одну или несколько новых строк с подходящими значениями затрат, чтобы можно было найти допустимое решение.

Задачи максимизации

Рассмотрим модель назначений, в которой каждому назначению поставлена в соответствие определенная прибыль, а не затраты.

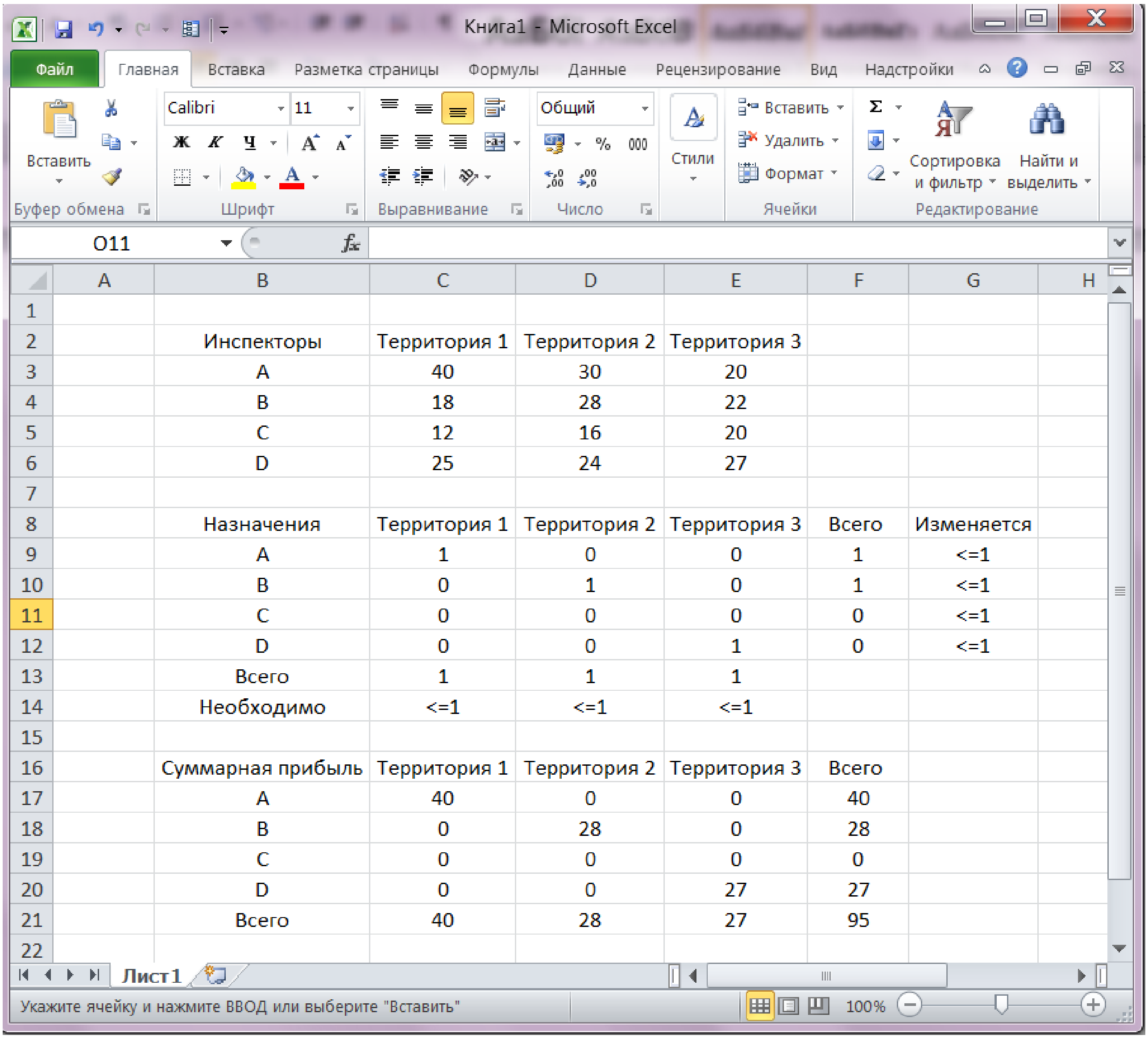
Например, предположим, что компании нужно направить инспекторов в новые районы сбыта. Имеется три вакансии и четыре кандидата. Одному из кандидатов придется подождать, пока появится еще одна вакансия. Модель назначений в этом случае показана на рис. 5.3.

Результат назначения кандидата на определенное место измеряется ожидаемым увеличением удельной прибыли. Естественно, что компания заинтересована в максимизации общей прибыли. Прибыль, которую могут дать конкретные инспекторы, показана в табл. 13.

Таблица 13

Постановка задачи

Инспекторы	**Территория**			**Количество кандидатов**
	1	**2**	**3**	
А	40	30	20	1
Б	18	28	22	1
В	12	16	20	1
Г	25	24	27	1
Число вакансий	1	1	1	3 / 4



	A	B	C	D	E	F	G	H
1								
2		Инспекторы	Территория 1	Территория 2	Территория 3			
3		A	40	30	20			
4		B	18	28	22			
5		C	12	16	20			
6		D	25	24	27			
7								
8		Назначения	Территория 1	Территория 2	Территория 3	Всего	Изменяется	
9		A	1	0	0	1	<=1	
10		B	0	1	0	1	<=1	
11		C	0	0	0	0	<=1	
12		D	0	0	1	0	<=1	
13		Всего	1	1	1			
14		Необходимо	<=1	<=1	<=1			
15								
16		Суммарная прибыль	Территория 1	Территория 2	Территория 3	Всего		
17		A	40	0	0	40		
18		B	0	28	0	28		
19		C	0	0	0	0		
20		D	0	0	27	27		
21		Всего	40	28	27	95		
22								

Рис. 5.3. Модель задачи максимизации

5.4. Пример решения задачи о назначениях с помощью Mathcad

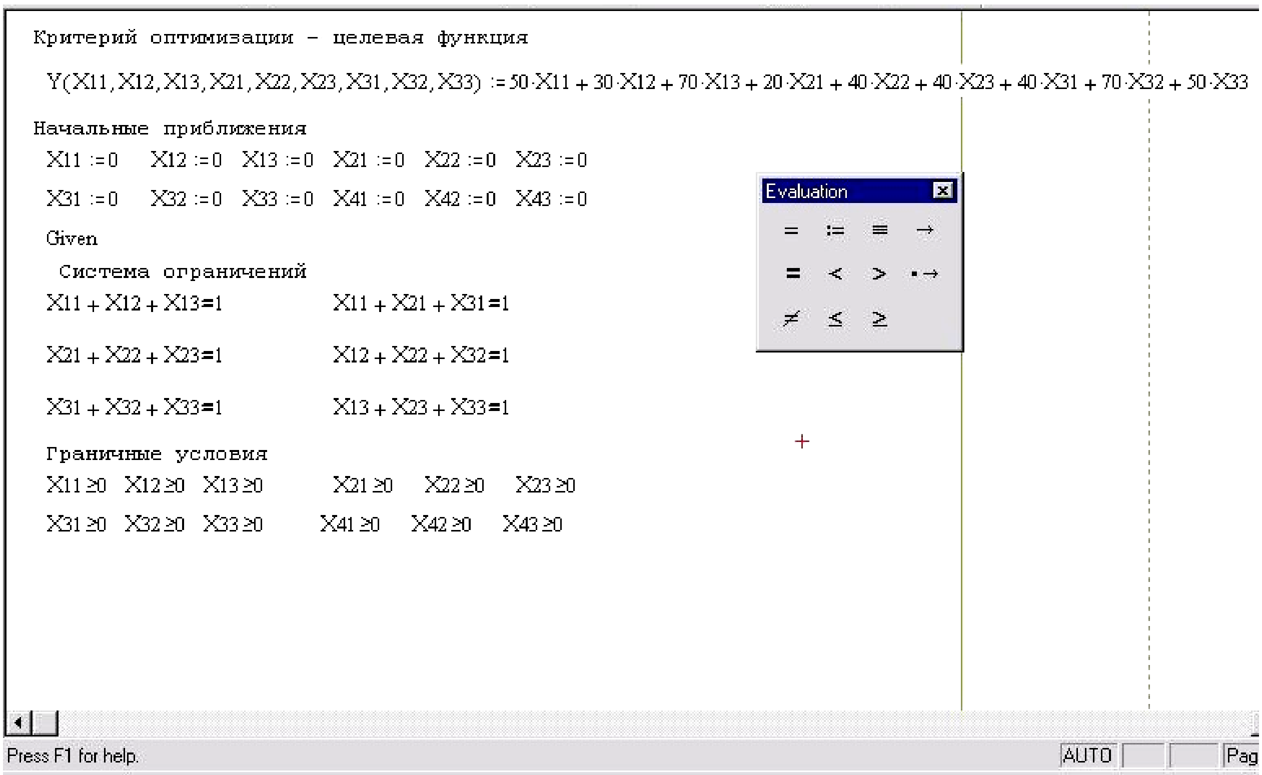


Рис. 5.4. Начальные приближения

На рис. 5.5 показан процесс и результаты решения задачи о назначении. Оптимальное распределение зафиксировано в векторе (X11 X12 X21 ...). Из полученного решения видно, что $X_{12} = 1$, $X_{21} = 1$ и $X_{33} = 1$.

Следовательно, для того, чтобы оптимально распределить три крана на три объекта, необходимо первый кран направить на второй объект, второй на первый, а третий – на третий. Первая цифра в переменной X определяет машину, а вторая – объект работы. При таком распределении кранов по объектам минимальные суммарные затраты Y составят 100 условных единиц.

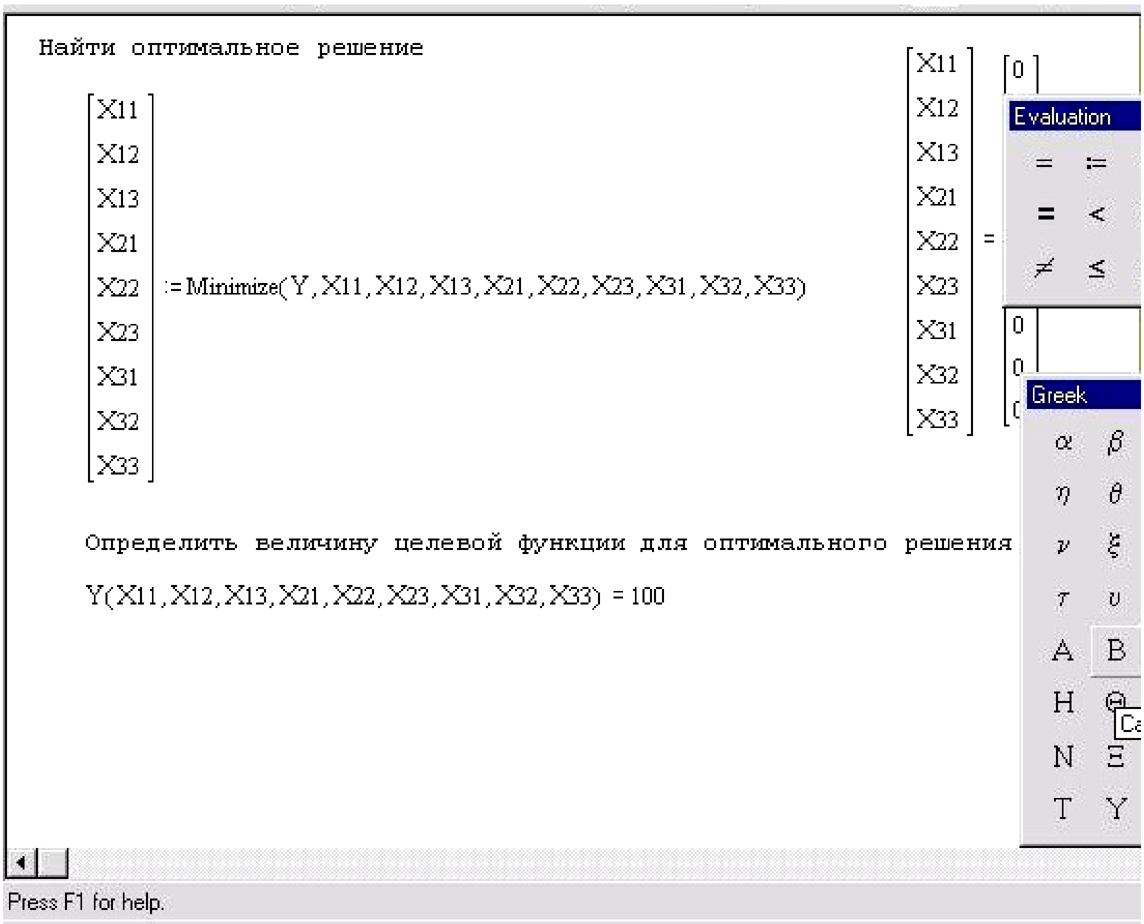


Рис. 5.5. Процесс и результаты решения задачи о назначении

5.5. *Задание к лабораторной работе*

Требуется расставить n рабочих по технологической цепочке так, чтобы время выполнения всего цикла операций было минимальным. Время, затрачиваемое каждым рабочим при выполнении любой операции, приведено в таблице.

Вариант 1

3	7	6	5
7	4	3	3
4	3	3	4
5	6	4	2

Вариант 2

4	6	9	7
13	10	14	14
9	9	16	13
12	10	12	10

Вариант 3

9	20	60	15	21
38	71	69	49	60
28	13	80	28	34
58	34	13	37	25
30	3	53	20	21

Вариант 4

10	5	9	18	11
13	19	6	12	14
3	2	4	4	5
18	9	12	17	15
11	6	14	19	10

Вариант 5

44	74	35	49	30	45
22	28	42	59	83	41
28	39	54	47	35	24
49	53	45	50	43	38
27	37	30	18	30	22
70	27	21	32	31	9

Вариант 6

4	6	9	7	10	6
2	3	7	6	5	10
9	7	4	3	3	9
6	4	3	3	4	10
7	5	6	4	2	16
4	9	12	10	12	10

Вариант 7

93	93	91	94	99	99	90	92
96	93	90	94	98	96	97	91
96	90	91	90	92	90	93	96
93	94	95	96	97	10	92	93
94	93	95	91	90	97	96	92
94	93	96	90	93	89	88	91
94	96	91	90	95	93	92	94
93	94	6	95	91	99	91	96

Вариант 8

34	39	15	18	45	27	37	24
26	53	44	74	35	49	30	45
85	50	22	28	42	59	83	41
15	18	28	39	54	47	35	24
96	32	49	53	45	50	43	38
15	45	27	37	30	18	30	22
63	41	70	27	21	32	31	19
64	24	47	35	24	24	28	42

Университет получил гранты на выполнение n исследовательских проектов. В качестве научных руководителей проектов рассматриваются кандидатуры n ученых. Каждый ученый оценил время, необходимое ему для реализации проекта. Таблица времен приведена ниже. Требуется выбрать руководителя для выполнения каждого проекта так, чтобы суммарное время выполнения всех проектов было минимальным.

Вариант 9

15	10	9
9	15	10
10	12	8

Вариант 10

3	7	6
7	4	3
4	3	3

Вариант 11

3	8	2	10	3
8	7	2	9	7
6	4	2	7	5
8	4	2	3	5
9	10	6	9	10

Вариант 12

3	9	2	3	7
6	1	5	6	6
9	4	7	10	3
2	5	4	2	1
9	6	2	4	5

Вариант 13

24	36	32	24	30	39	43
24	53	58	24	47	56	63
24	33	42	24	40	46	56
28	40	39	24	52	58	65
25	34	33	24	39	45	52
24	25	24	19	24	14	24
32	44	34	24	29	35	63

Вариант 14

15	15	23	35	36	39	34
16	15	41	41	45	48	46
19	15	44	47	51	51	46
15	13	15	18	15	15	15
30	15	38	38	63	63	61
30	15	32	38	36	63	61
28	15	30	39	34	37	54

Вариант 15

25	16	15	14	13
25	17	18	23	15
30	15	20	19	14
27	20	22	25	12
29	19	17	32	10

Вариант 16

2	4	6	8	3
1	3	2	7	6
7	2	4	5	8
9	1	3	4	7
3	5	1	2	4

Вариант 17

7	8	3	4	5
2	3	5	6	7
9	10	11	3	4
5	6	8	7	3
3	7	1	5	2

Вариант 18

56	42	35	28	14
48	36	30	24	12
40	30	25	20	10
36	27	22	18	9
32	24	20	16	8

Требуется расставить n рабочих по работам так, чтобы суммарная эффективность выполнения работ была максимальной. Значения эффективности выполнения i-й работы j-ым исполнителем приведены в таблице.

Вариант 19

9	20	60	15	21
38	71	69	49	60
28	13	80	28	34
58	34	13	37	25
30	3	53	20	21

Вариант 20

10	5	9	18	11
13	19	6	12	14
3	2	4	4	5
18	9	12	17	15
11	6	14	19	10

Вариант 21

25	16	15	14	13
25	17	18	23	15
30	15	20	19	14
27	20	22	25	12
29	19	17	32	10

Вариант 22

2	4	6	8	3
1	3	2	7	6
7	2	4	5	8
9	1	3	4	7
3	5	1	2	4

Вариант 23

7	8	3	4	5
2	3	5	6	7
9	10	11	3	4
5	6	8	7	3
3	7	1	5	2

Вариант 24

56	42	35	28	14
48	36	30	24	12
40	30	25	20	10
36	27	22	18	9
32	24	20	16	8

Вариант 25

93	93	91	94	99	99	90	92
96	93	90	94	98	96	97	91
96	90	91	90	92	90	93	96
93	94	95	96	97	10	92	93
94	93	95	91	90	97	96	92
94	93	96	90	93	89	88	91
94	96	91	90	95	93	92	94
93	94	6	95	91	99	91	96

Вариант 26

34	39	15	18	45	27	37	24
26	53	44	74	35	49	30	45
85	50	22	28	42	59	83	41
15	18	28	39	54	47	35	24
96	32	49	53	45	50	43	38
15	45	27	37	30	18	30	22
63	41	70	27	21	32	31	19
64	24	47	35	24	24	28	42

Вариант 27

24	36	32	24	30	39	43
24	53	58	24	47	56	63
24	33	42	24	40	46	56
28	40	39	24	52	58	65
25	34	33	24	39	45	52
24	25	24	19	24	14	24
32	44	34	24	29	35	63

Вариант 28

15	15	23	35	36	39	34
16	15	41	41	45	48	46
19	15	44	47	51	51	46
15	13	15	18	15	15	15
30	15	38	38	63	63	61
30	15	32	38	36	63	61
28	15	30	39	34	37	54

Вариант 29

3	8	2	10	3
8	7	2	9	7
6	4	2	7	5
8	4	2	3	5
9	10	6	9	10

Вариант 30

3	9	2	3	7
6	1	5	6	6
9	4	7	10	3
2	5	4	2	1
9	6	2	4	5

Контрольные вопросы

1. Какова постановка задачи о назначениях?
2. В чем отличие модели задачи о назначениях от модели ТЗ?
3. Каковы исходные и искомые параметры задачи о назначениях?
4. Запишите математическую модель задачи о назначениях.
5. Как записать модель задачи о назначениях, подразумевающую максимизацию ЦФ?
6. Каким образом в модели задачи о назначениях можно запретить конкретное назначение?
7. В чем особенности процесса приведения задачи о назначениях к сбалансированному виду?

6. Сетевые модели

6.1. Общие сведения

В рамках исследования операций рассматривается большое количество практических задач, которые можно сформулировать и решать как сетевые модели. Около 70% реальных задач математического программирования можно представить в виде сетевых моделей. Например:

1. Проектирование газопровода, соединяющего буровые скважины с приемной станцией.
2. Поиск кратчайшего маршрута между двумя объектами по существующей транспортной системе.
3. Определение максимальной пропускной способности.
4. Поиск схемы транспортировки нефти с минимальной стоимостью.
5. Составление временного графика строительных работ.

Решение приведенных задач требует применения различных сетевых оптимизационных алгоритмов:

1. Алгоритм нахождения минимального остовного дерева.
2. Алгоритм поиска кратчайшего пути.
3. Алгоритм определения максимального потока.
4. Алгоритм минимизации стоимости потока в сети с ограниченной пропускной способностью.
5. Алгоритм определения критического пути.

Задачи, вытекающие из приведенных примеров, можно решать как задачи линейного программирования. Однако специфическая структура задач позволяет использовать алгоритмы более эффективные, чем симплекс-метод.

Сеть состоит из множества узлов, связанных дугами (или ребрами). Таким образом, сеть описывается парой множеств (N, A), где N – множество узлов, а A – множество ребер. Например, сеть, показанная на рис. 6.1., описывается следующим образом:

$N = \{1, 2, 3, 4, 5\}$, $A = \{(1,2), (1,3), (1,5), (2,5), (3,4), (3,5), (4,2), (5,4)\}$.

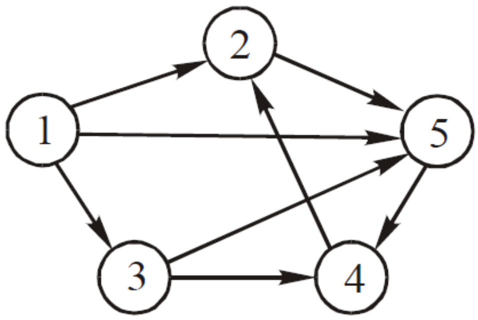


Рис. 6.1. Пример сети

Дуга – направленное ребро (в одном направление возможен только положительный поток, а в противоположном – только нулевой).

Путем называется последовательность различных ребер, соединяющих два узла, независимо от направления потока в каждом ребре. Путь формирует цикл, если начальный и конечный узлы совпадают. Например, на рис. 6.1 дуги $(2,5),(5,4),(4,2)$ составляют цикл.

Деревом называется сеть, содержащая подмножество узлов исходной сети и не имеющая циклов. Остовное дерево – это дерево, содержащее все узлы сети. На рис. 6.2 показано оставное дерево для сети из рис. 6.1.

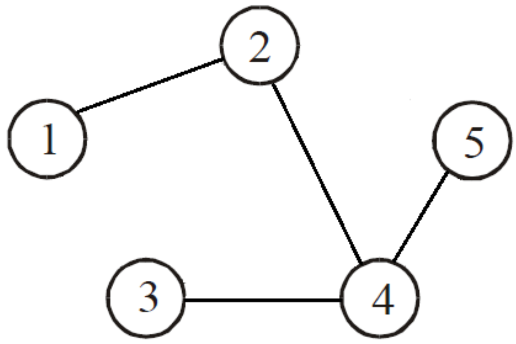


Рис. 6.2. Остовное дерево

6.2. Алгоритм построения минимального остовного дерева

Минимальное остовное дерево – это остовное дерево, имеющее минимальный возможный вес, где под весом дерева понимается сумма весов входящих в него рёбер.

Задача о нахождении минимального остовного дерева часто встречается в подобной постановке: допустим, есть n городов, которые необходимо соединить дорогами так, чтобы можно было добраться из любого города в любой другой (напрямую или через другие города). Разрешается строить дороги между заданными парами городов и известна стоимость строительства каждой такой дороги. Требуется решить, какие именно дороги нужно строить, чтобы минимизировать общую стоимость строительства.

Эта задача может быть сформулирована как задача о нахождении минимального остовного дерева, вершины которого представляют города, рёбра – это пары городов, между которыми можно проложить прямую дорогу, а вес ребра равен стоимости строительства соответствующей дороги.

Введем новые обозначения:

C_k – множество узлов сети, соединенных алгоритмом после выполнения k-й итерации,

$\overline{C_k}$ – множество узлов сети, не соединенных с узлами множества C_k.

Шаг 1. $C_0 = \emptyset,\ \overline{C_0} = N$.

Шаг 2. Выбираем любой узел j из множества $\overline{C_0}$.

$C_1 = \{j\}$, тогда $\overline{C_1} = N - j.\ k = 2.$

Шаг 3. В множестве $\overline{C_{k-1}}$ выбираем узел i, который соединен самым коротким ребром с каким-либо узлом из множества C_{k-1}.

$C_k = C_{k-1} + \{i\},\ \overline{C_k} = \overline{C_{k-1}} - \{i\}$.

Если множество $\overline{C_k}$ пусто, то выполнение алгоритма заканчивается, в противном случае $k = k + 1$ и повторяем 3 шаг.

В модульных перевозках груженые трейлерные платформы перевозятся по железной дороге между специальными перевалочными железнодорожными терминалами. На рис. 6.3 показаны железнодорожные терминалы и существующие железнодорожные пути между ними. Необходимо выделить сегменты железных дорог так, чтобы связать все железнодорожные терминалы и минимизировать суммарную стоимость перевозок трейлерных платформ.

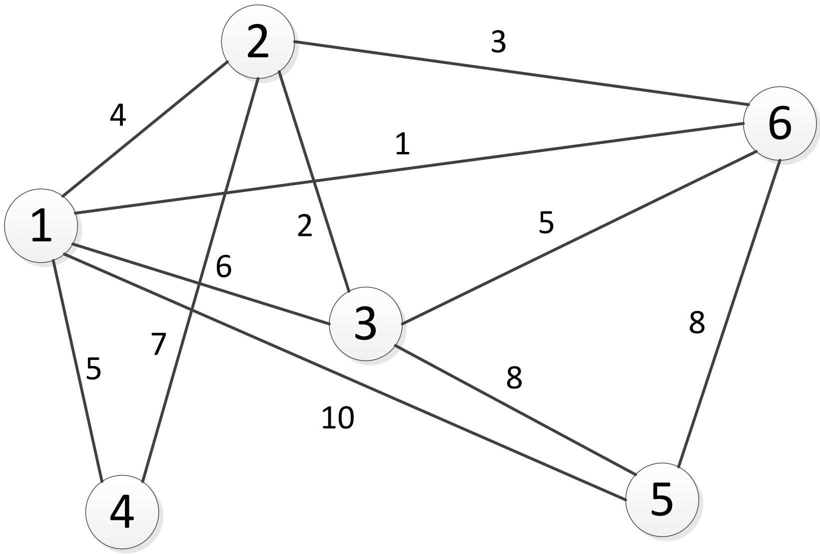


Рис. 6.3. Железнодорожные пути

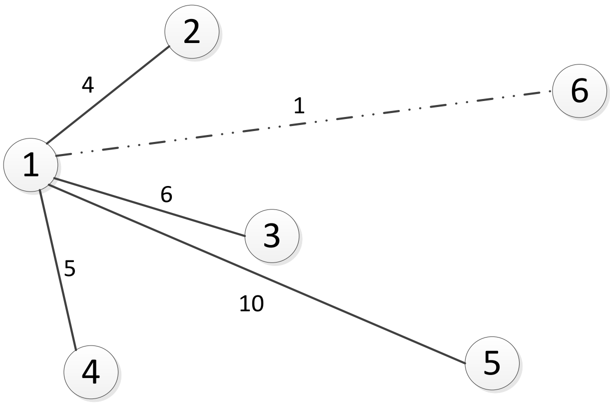
Чтобы начать выполнение алгоритма построения минимального остовного дерева, выберем узел 1 (или любой другой). Тогда

$$C_1 = \{1\}, \qquad \overline{C_1} = \{2, 3, 4, 5, 6\}.$$

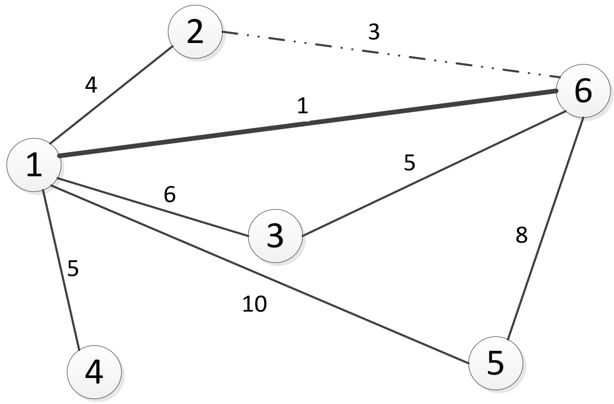
Последовательные итерации выполнения алгоритма представлены на рис. 6.4. Здесь тонкими линиями показаны ребра, соединяющие узлы, принадлежащие C_k и $\overline{C_k}$, среди которых нужно найти ребро с минимальной стоимостью. Это ребро показано пунктирной линией. Толстыми линиями обозначены ребра, соединяющие узлы множества C_k.

Решение в виде минимального остовного дерева получено на шестой итерации (рис. 6.5). Минимальная стоимость перевозок равна

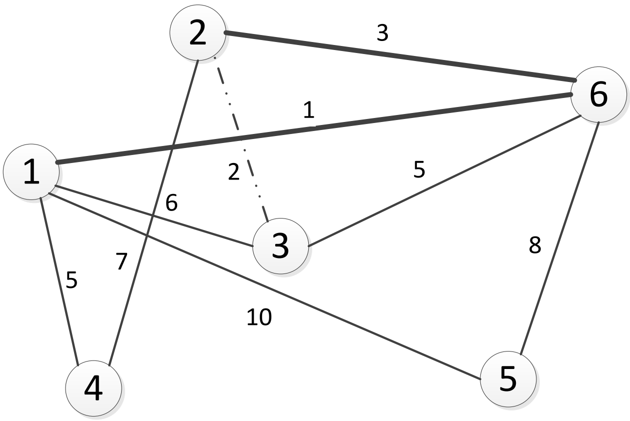
$5 + 1 + 3 + 2 + 8 = 19.$



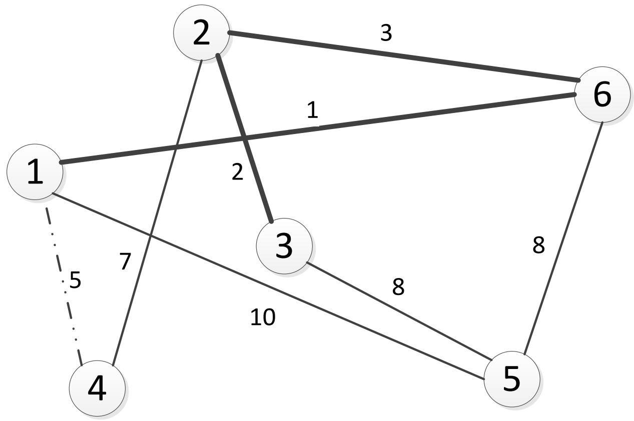
Итерация 1



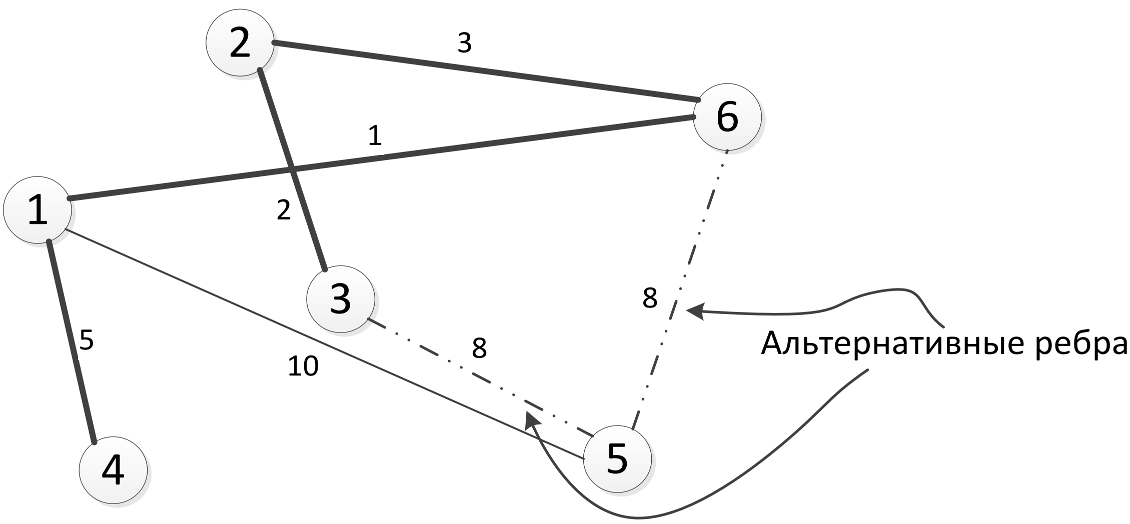
Итерация 2



Итерация 3

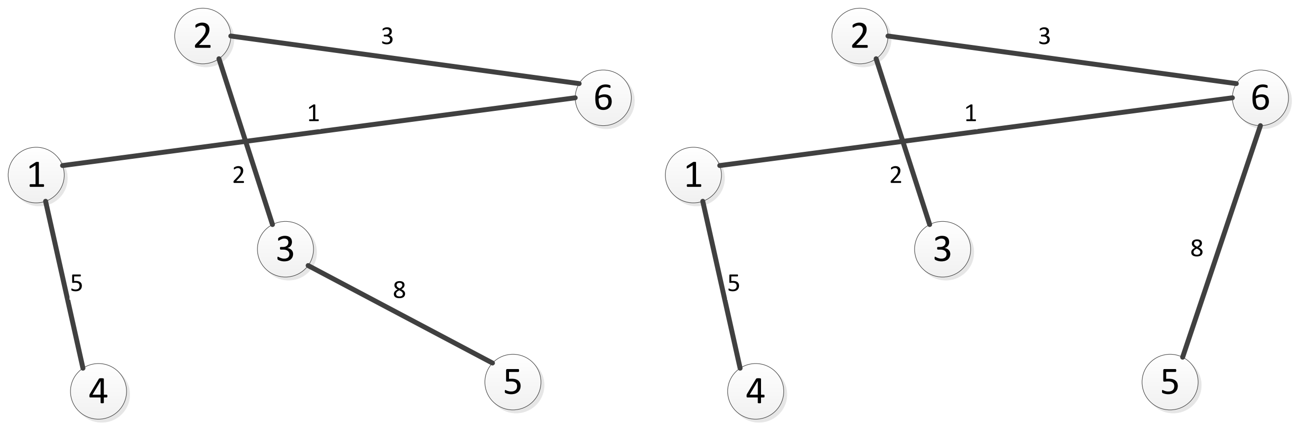


Итерация 4



Итерация 5

Рис. 6.4. Последовательные итерации выполнения алгоритма построения минимального остовного дерева



Минимальное остовное дерево 1 Минимальное остовное дерево 2

Рис. 6.5. Итерация 6

Телевизионная компания планирует подключение к своей кабельной сети пяти новых районов. На рис. 6.6 показана структура планируемой сети и расстояния между районами и телецентрами. Необходимо спланировать наиболее экономичную кабельную сеть.

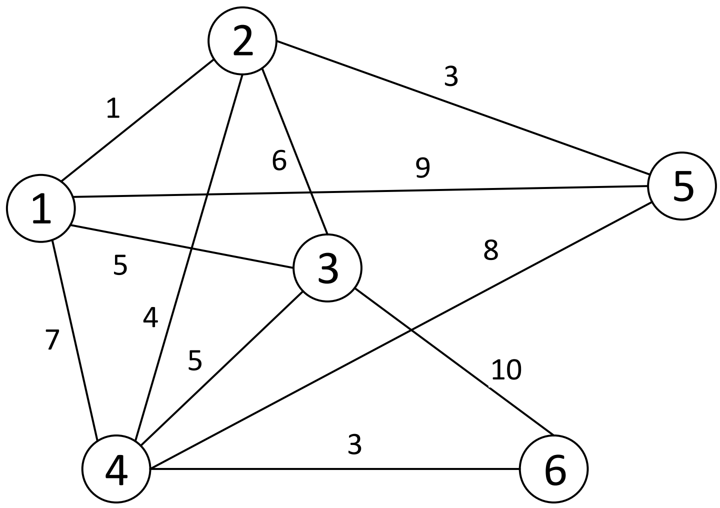


Рис. 6.6. Кабельная сеть

Чтобы начать выполнение алгоритма построения минимального остовного дерева, выберем узел 1. Тогда

$$C_1 = \{1\}, \qquad \overline{C_1} = \{2, 3, 4, 5\}.$$

Решение в виде минимального остовного дерева получено на шестой итерации (рис. 6.8). Минимальная стоимость перевозок равна

$$5 + 1 + 3 + 4 + 3 = 16.$$

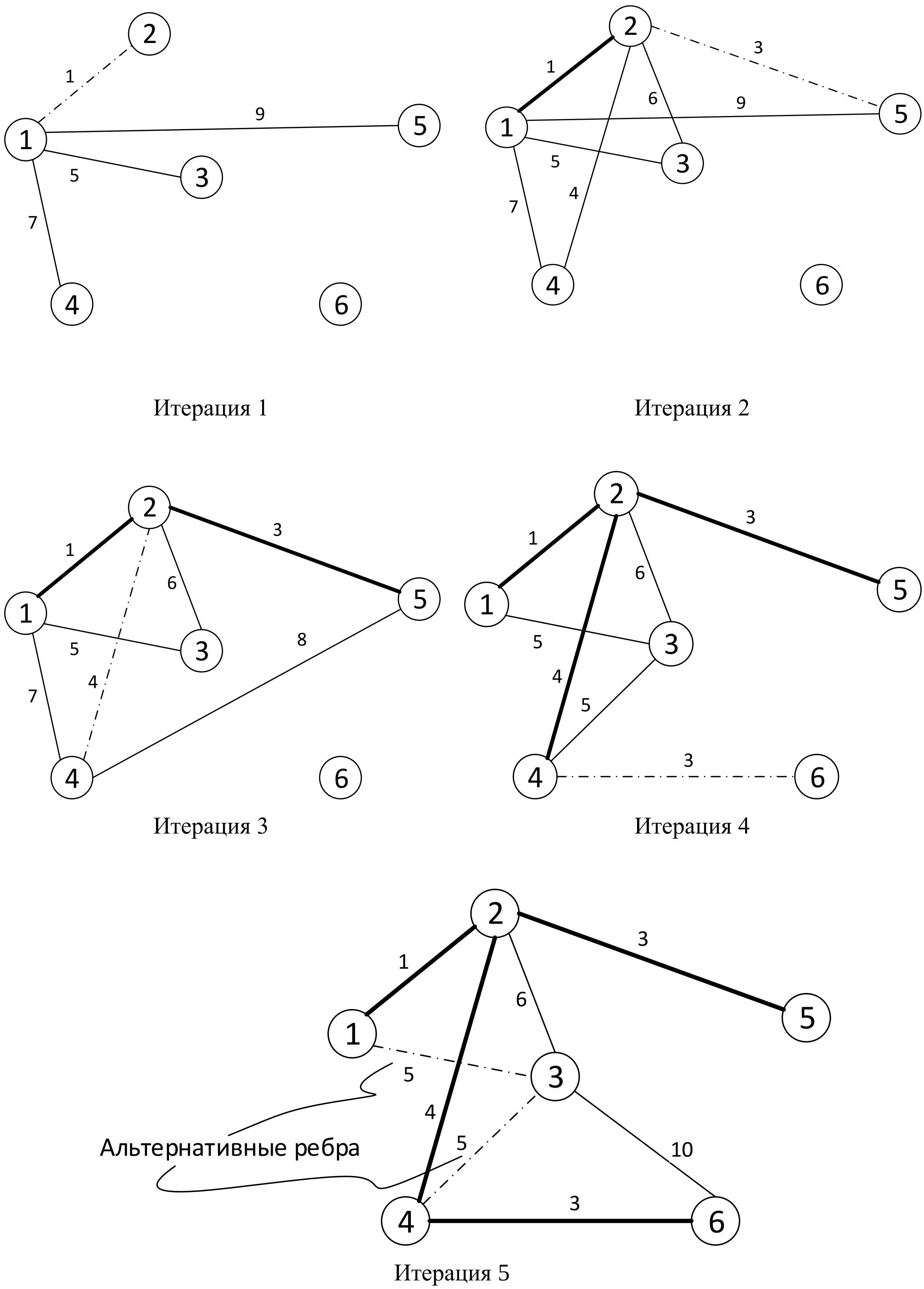


Рис. 6.7. Последовательные итерации

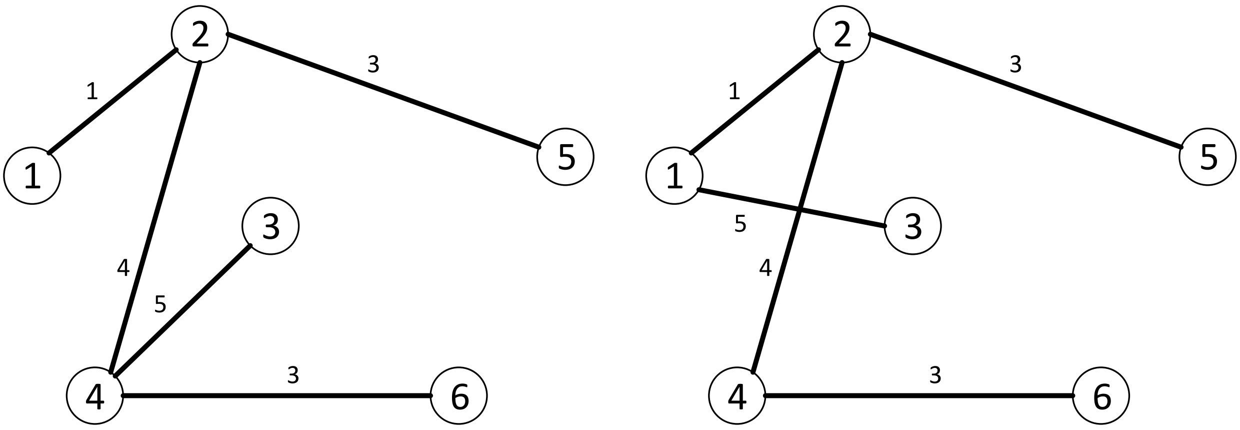


Рис. 6.8. Варианты минимального остовного дерева кабельной сети

6.3. Нахождение кратчайшего пути

Задача состоит в нахождении связанных между собой дорог на транспортной сети, которые в совокупности имеют минимальную длину от исходного пункта до пункта назначения.

Введем обозначения:

d_{ij} – расстояние на сети между смежными узлами i и j;

U_j – кратчайшее расстояние между узлами i и j, $U_1 = 0$.

Формула для вычисления U_j:

$$U_j = \min_i \left\{ \begin{matrix} \text{кратчайшее расстояние до} \\ \text{предыдущего узла } i \text{ плюс} \\ \text{расстояние между текущим} \\ \text{узлом } j \text{ и предыдущим узлом } i \end{matrix} \right\} = \min_i \{U_i + d_{ij}\}.$$

Определить кратчайшее расстояние между узлами 1 и 7 (рис. 6.9).

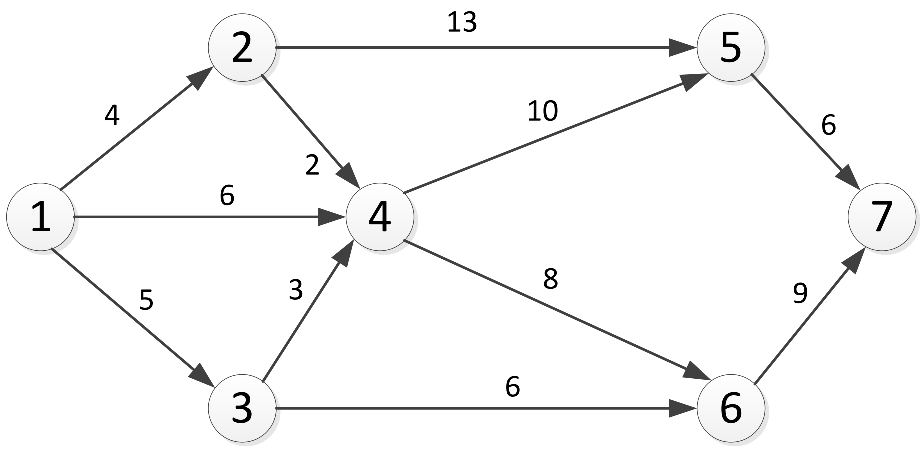


Рис. 6.9. Сеть

Найдем минимальные расстояния:

$U_1 = 0,$

$U_2 = U_1 + d_{12} = 0 + 4 = 4,$

$U_3 = U_1 + d_{13} = 0 + 5 = 5,$

$U_4 = \min\{U_1 + d_{14};\ U_2 + d_{24};\ U_3 + d_{34}\} = \min\{0 + 6; 4 + 2; 5 + 3\} = 6,$

$U_5 = \min\{U_2 + d_{25};\ U_4 + d_{45}\} = \min\{4 + 13; 6 + 10\} = 16,$

$U_6 = \min\{U_3 + d_{36};\ U_4 + d_{46}\} = \min\{5 + 6; 6 + 8\} = 11,$

$U_7 = \min\{U_5 + d_{57};\ U_6 + d_{67}\} = \min\{16 + 6; 11 + 9\} = 20.$

Минимальное расстояние между узлами 1 и 7 равно 20, а соответствующий путь: $1 \to 3 \to 6 \to 7$.

6.3.1. Алгоритм Дейкстры

Алгоритм Дейкстры разработан для поиска кратчайшего пути между заданными исходным узлом и любым другим узлом сети.

В процессе выполнения алгоритма при переходе от узла i к следующему узлу j используется специальная процедура пометки ребер. Для узла j определим метку $[U_j, i] = [U_i + d_{ij}, i]$.

Метка исходного узла $(i = 1)$ полагается равной $[0, -]$, метки остальных узлов – бесконечности. Все узлы сети помечаются как непосещённые.

Вычисляются метки для всех узлов j, которые можно достичь непосредственно из узла i и которые помечены как непосещенные. Если узел j уже имеет метку $[U_j, k]$ и если $U_i + d_{ij} < [U_j, k]$, тогда метка $[U_j, k]$ заменяется на $[U_i + d_{ij}, i]$.

Если все узлы посещены, алгоритм завершается. В противном случае из ещё непосещённых узлов выбирается узел i, имеющий минимальную метку. Для каждого узла, кроме отмеченных как посещённые, рассматривают новую длину пути, равную сумме значений текущей метки i и длины ребра, соединяющего i с этим узлом. Если полученное значение длины меньше значения метки, его заменяют на полученное значением длины.

Узел i помечается как посещенный. Если все узлы посещены, то процесс вычисления заканчивается. Иначе, выбирается метка $[U_r, s]$ с наименьшим значением U_r среди всех непосещённых узлов и полагаем $i = r$.

Пример. На рис. 6.10 показана транспортная сеть, состоящая из 5 городов. Необходимо найти кратчайшее расстояние от города 1 до всех остальных городов.

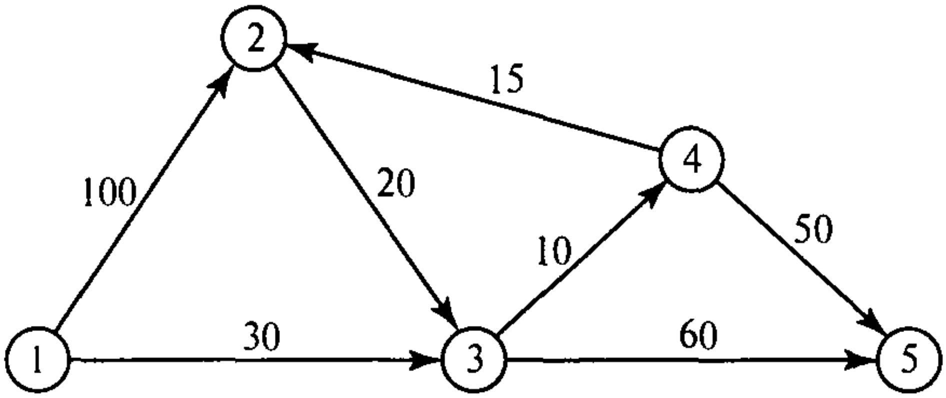


Рис. 6.10. Транспортная сеть, состоящая из 5 городов

<u>Шаг 1.</u> Назначаем узлу 1 метку $[0, -]$.

<u>Шаг 2.</u> Из узла 1 можно достичь узлов 2 и 3. Получаем таблицу меток:

Узел	**Метка**	**Статус**
1	$[0, -]$	посещён
2	$[0 + 100, 1] = [100, 1]$	
3	$[0 + 30, 1] = [30, 1]$	

<u>Шаг 3.</u> Из узла 3 (имеет наименьшую метку среди непосещённых узлов) можно попасть в узлы 4 и 5. Получаем таблицу меток:

Узел	**Метка**	**Статус**
1	$[0, -]$	посещён
2	$[100, 1]$	
3	$[30, 1]$	посещён
4	$[30 + 10, 3] = [40, 3]$	
5	$[30 + 60, 3] = [90, 3]$	

<u>Шаг 4.</u> Из узла 4 можно достичь узлов 2 и 5. Получаем таблицу меток:

Узел	**Метка**	**Статус**
1	$[0, -]$	посещён
2	$[40 + 15, 4] = [55, 4]$	
3	$[30, 1]$	посещён
4	$[40, 3]$	посещён
5	$[90, 3]$ или $[40 + 50, 4] =$ $= [90, 4]$	

Метка 2 узла $[100, 1]$ заменена на $[55, 4]$. Это указывает на то, что был найден более короткий путь к этому узлу.

<u>Шаг 5.</u> Из узла 2 можно перейти только в узел 3, но он уже имеет статус *посещён*. Поэтому на данном шаге получается такой же список меток, как и на предыдущем, но с единственным изменением: статус узла 2 становится *посещён*.

<u>Шаг 6.</u> Из узла 5 никуда нельзя попасть. Процесс вычислений закончен.

Алгоритм позволяет проводить вычисления непосредственно на схеме сети, как показано на рис. 6.11.

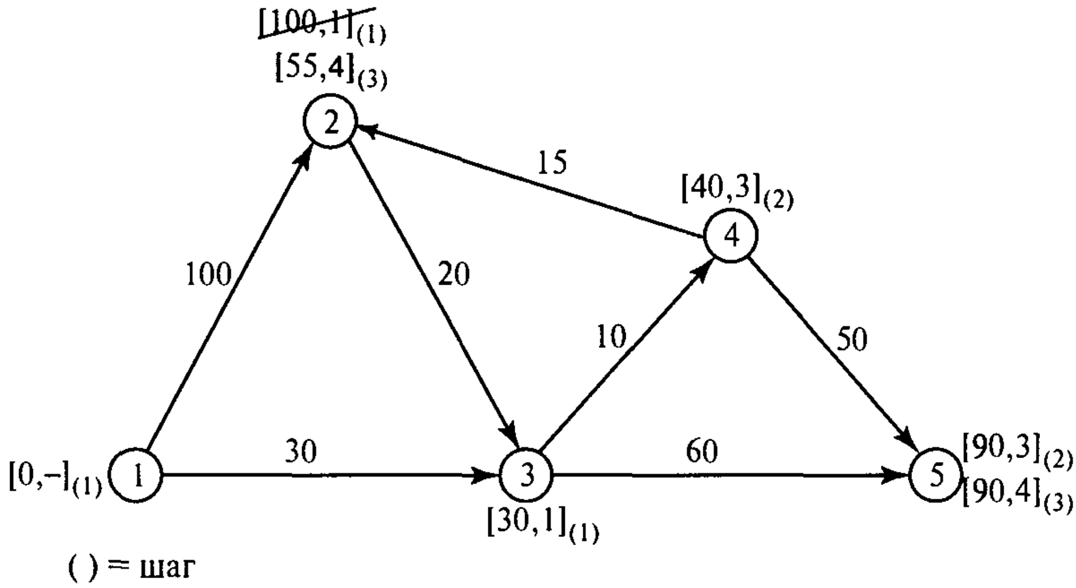


Рис. 6.11. Применение алгоритма Дейкстры

Для определения кратчайшего маршрута между узлами 1 и 2 получаем последовательность:

$$(2) \to [55,4] \to (4) \to [40,3] \to (3) \to [30,1] \to (1).$$

Таким образом, получаем путь $1 \to 3 \to 4 \to 2$ общей длиной 55.

6.3.2. Формализация задачи поиска кратчайшего пути к задаче линейного программирования

Пусть сеть состоит из n узлов и нужно найти кратчайший путь между некоторыми узлами s и t этой сети.

Формализация 1. В этой формализации предполагается, что в узел s входит одна единица внешнего потока и этот поток выходит через узел t этой сети. Обозначим:

x_{ij} – величина потока, проходящего по дуге (i,j), $x_{ij} = 0$ или 1,

c_{ij} – длина дуги (i,j).

Целевая функция примет вид

$$\sum_{\substack{\text{по всем} \\ \text{существующим дугам } (i,j)}} c_{ij}x_{ij} \to min.$$

Для каждого узла определяется только одно ограничение, задающее баланс потока, проходящего через данный узел:

общий входной поток = общий выходной поток.

Формализация 2. Эта формализация практически определяет двойственную задачу к прямой задаче, формализованной первым способом. Двойственная задача будет иметь столько же переменных, сколько узлов в сети. Эти переменные будут свободными, так как в прямой задаче все ограничения выражаются в виде равенств.

Пусть y_i – переменная двойственной задачи, ассоциированная с узлом j.

Считая узлы s и t начальным и конечным узлами сети, двойственная задача примет вид:

$$z = y_t - y_s \to max,$$

при ограничениях

$$y_j - y_i \le c_{ij} \text{ для всех возможных пар } i \text{ и } j.$$

Рассмотрим сеть на рис. 6.10. Предположим, что необходимо определить кратчайший путь из узла 1 в узел 2. Таким образом, $s = 1$ и $t = 2$.

Первая формализация дает следующую задачу ЛП.

	x_{12}	x_{13}	x_{23}	x_{34}	x_{35}	x_{42}	x_{45}	
$min\, z =$	100	30	20	10	60	15	50	
Узел 1	−1	−1						= −1
Узел 2	1		−1			1		= 1
Узел 3		1	1	−1	−1			= 0
Узел 4				1		−1	−1	= 0
Узел 5					1		1	= 0

При использовании второй формализации двойственная к представленной выше задаче ЛП имеет вид:

$$z = y_2 - y_1 \to max,$$

$$\begin{cases} y_2 - y_1 \le 100, \\ y_3 - y_1 \le 30, \\ y_3 - y_2 \le 20, \\ y_4 - y_3 \le 10, \\ y_5 - y_3 \le 60, \\ y_2 - y_4 \le 15, \\ y_5 - y_4 \le 50, \\ y_i \ge 0. \end{cases}$$

Определение кратчайшего пути из решения двойственной задачи не очевидно. Любое ограничение, которое включает ненулевую двойственную переменную, должно выполняться в виде равенства, что и приведет к построению кратчайшего пути.

Шаблоны *Excel* и *MathCad* для решения транспортной задачи можно немного модифицировать для решения задачи нахождения кратчайшего пути.

6.4. Задача о максимальном потоке

Рассмотрим сеть трубопроводов для транспортировки сырой нефти от буровых скважин до нефтеперегонных заводов. Для перекачки нефти предусмотрены магистральные насосные станции. Каждый сегмент трубопровода имеет свою пропускную способность. Сегменты трубопровода могут быть как однонаправленные (осуществляют перекачку нефти только в одном направлении), так и в двунаправленные. В однонаправленных сегментах положительная пропускная способность предполагается в одном направлении и нулевая – в другом. На рис. 6.12 показана типовая сеть нефтепроводов.

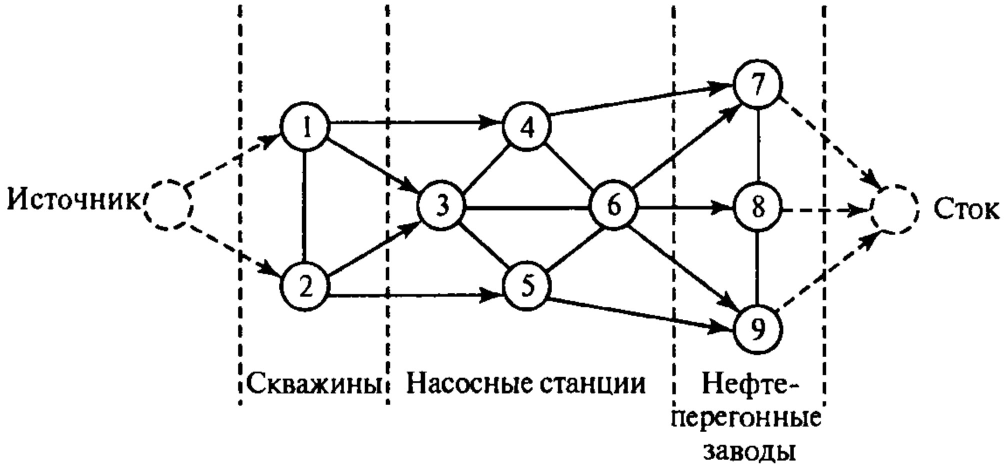


Рис. 6.12. Сеть, связывающая скважины и заводы

При решении данной задачи исходную сеть необходимо свести к сети с одним источником и одним стоком. Этого можно достигнуть путем введения дополнительных дуг с бесконечной пропускной способностью от источника к скважинам и от нефтеперегонных заводов к стоку (на рис. 6.12 эти дуги показаны пунктирными линиями).

Для ребра (i, j), где $i < j$, используем запись $(\overline{C_{ij}}, \overline{C_{ji}})$ для представления пропускных способностей в направлениях $i \to j$ и $j \to i$ соответственно. Во избежание недоразумений на схеме сети $\overline{C_{ij}}$ будем располагать на ребре (i, j) ближе к узлу i, а $\overline{C_{ji}}$ ближе к узлу j.

Разрез определяет множество ребер, при удалении которых из сети полностью прекращается поток от источника к стоку. Пропускная способность разреза равна сумме пропускных способностей "разрезанных" ребер. Среди всех разрезов сети разрез с минимальной пропускной способностью определяет максимальный поток в сети.

Рассмотрим сеть, показанную на рис. 6.13. На этом рисунке при обозначении пропускных способностей двунаправленных ребер придерживались соглашения, принятого ранее. Например, для ребра $(3, 4)$ пропускная способность в направлении $3 \to 4$ равна 10, а в направлении $4 \to 3$ равна 5.

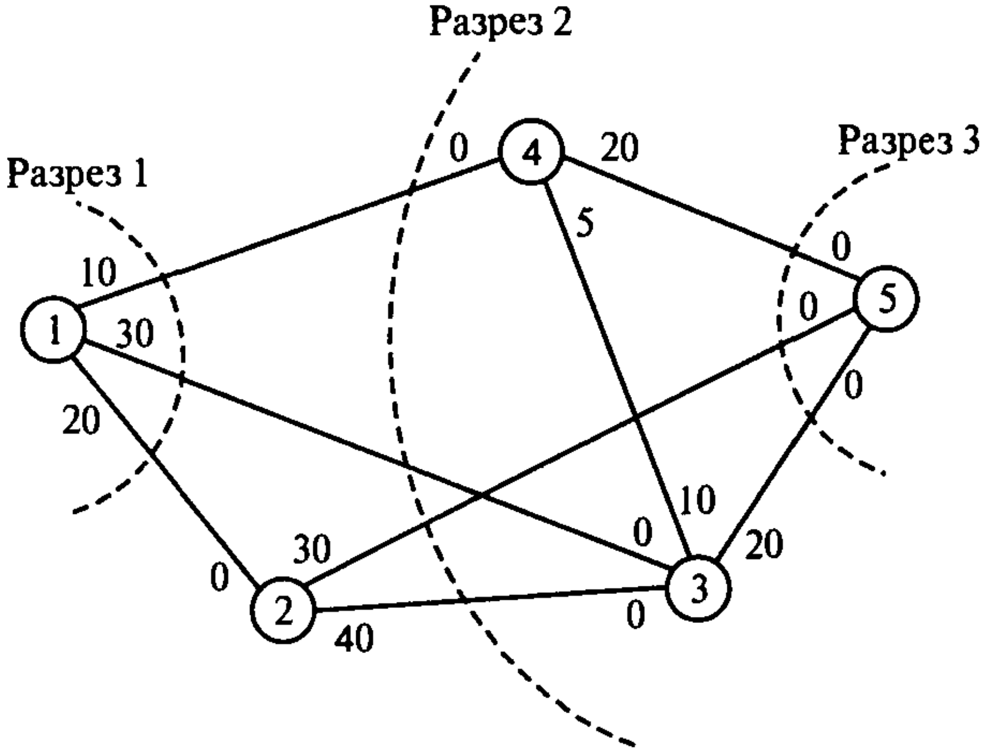


Рис. 6.13. Пример разрезов сети

Разрезы, представленные на рис. 6.13, имеют следующие пропускные способности:

Разрез	"Разрезанные" ребра	Пропускная способность
1	$(1,2),(1,3),(1,4)$	$10 + 30 + 20 = 60$
2	$(1,3),(1,4),(2,3),(2,5)$	$30 + 10 + 40 + 30 = 110$
3	$(2,5),(3,5),(4,5)$	$30 + 20 + 20 = 70$

Вывод, который можно сделать из этих трех разрезов, заключается в том, что максимальный поток не может превышать 60 единиц. Но нельзя сказать, какой максимальный поток на самом деле, так как были перебраны не все возможные разрезы сети. К сожалению, перебор всех разрезов является непростой задачей. Поэтому для определения максимального потока в сети не используются алгоритмы, основанные на полном переборе разрезов.

6.4.1. Формализация задачи поиска максимального потока как задачи линейного программирования

Обозначим через x_{ij} величину потока, проходящего по дуге (i,j), пусть c_{ij} – пропускная способность этой же дуги. Предположим, что нужно найти максимальный поток между начальным узлом s и конечным узлом t.

Ограничениями задачи линейного программирования будут уравнения баланса входного и выходного потоков для каждого узла. Целевая функция максимизирует либо величину общего потока, выходящего из начального узла, либо величину общего потока, входящего в конечный узел.

В задаче вычисления максимального потока в сети, показанной на рис. 6.13, $s = 1$ и $t = 5$.

Формализация дает следующую задачу ЛП.

	x_{12}	x_{13}	x_{14}	x_{23}	x_{25}	x_{34}	x_{35}	x_{42}	x_{45}	
$max\ z_1$	1	1	1							
$max\ z_2$					1		1		1	
Узел 2	1			-1	-1					$= 0$
Узел 3		1		1		-1	-1	1		$= 0$
Узел 4			1			1		-1	-1	$= 0$
Пропускная способность	20	30	10	40	30	10	20	5	20	

Шаблоны *Excel* и *MathCad* для решения транспортной задачи можно немного модифицировать для нахождения максимального потока.

6.5. Задача нахождения потока наименьшей стоимости

Задача нахождения потока наименьшей стоимости в сети с ограниченной пропускной способностью обобщает задачу определения максимального потока по следующим направлениям:

1. Все ребра допускают только одностороннее направление потока, т.е. являются дугами.
2. Каждой дуге поставлена в соответствие (неотрицательная) стоимость прохождения единицы потока по данной дуге.
3. Дуги могут иметь положительную нижнюю границу пропускной способности.
4. Любой узел сети может выступать как в качестве источника, так и стока.

Дана сеть $G = (N, A)$ с ограниченной пропускной способностью, где N – множество узлов, A – множество дуг. Обозначим:

- x_{ij} – величина потока, протекающего от узла i к узлу j;
- u_{ij} (l_{ij}) – верхняя (нижняя) граница пропускной способности дуги (i, j);
- c_{ij} – стоимость прохождения единицы потока по дуге (i, j);
- f_i – величина "чистого" результирующего потока, протекающего через узел i.

На рис. 6.14 показано, как на схемах сетей отображаются определенные параметры дуг. Метка $[f_i]$ указывает положительное (отрицательное) значение предложения (спроса), соответствующего узлу i.

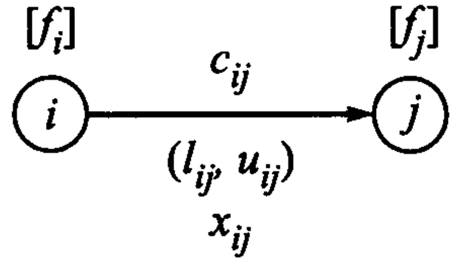


Рис. 6.14. Обозначение параметров дуг

Пример. Компания снабжает сырьем из трех рудников три медеплавильных завода. Предложение рудников составляет 100, 200 и 50 тысяч тонн, а спрос заводов – 150, 80 и 120 тысяч тонн соответственно. Компания может транспортировать сырье по железной дороге, за исключением трех маршрутов, где используется автомобильный транспорт.

На рис. 6.15 показаны возможные маршруты между заводами и рудниками. Рудники представлены узлами 1, 2 и 3; их предложения указаны метками $[100]$, $[200]$ и $[50]$ соответственно. Заводы обозначены узлами 4, 5 и 6 с величинами спроса $[-150]$, $[-80]$ и $[-120]$. Маршруты транспортировки сырья показаны на рис. 6.12 дугами, соединяющими узлы сети. Дуги $(1,4)$, $(3,4)$ и $(4,6)$ соответствуют автомобильным маршрутам. Эти маршруты имеют верхние и нижние границы пропускных способностей. Например, по маршруту $(1,4)$ можно провести от 50 до 80 тысяч тонн сырья. Другие маршруты соответствуют железнодорожному транспорту, пропускная способность которого практически не ограничена. Стоимость транспортировки одной тонны груза показана возле каждой дуги.

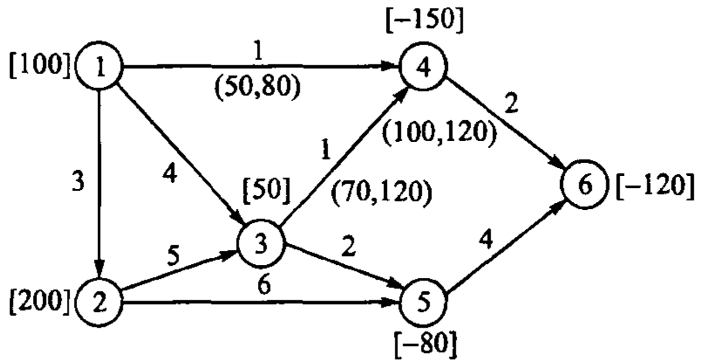


Рис. 6.15. Схема транспортировки грузов

6.5.1. Формализация задачи нахождения потока наименьшей стоимости как задачи линейного программирования

Определение модели сети с ограниченной пропускной способностью как задачи линейного программирования необходимо для разработки симплексного алгоритма решения задач данного типа. Можно записать задачу линейного программирования для сети с ограниченной пропускной способностью следующим образом.

$$z = \sum_{(i,j)} \sum_{\in A} c_{ij} x_{ij} \rightarrow min,$$

при ограничениях

$$\sum_{(i,j)\in A} x_{jk} - \sum_{(i,j)\in A} x_{ij} = f_i, \qquad j \in N,$$

$$l_{ij} \le x_{ij} \le u_{ij}.$$

Узел j выступает в качестве источника, если $f_j > 0$, и как исток при $f_j < 0$.

Нижнюю границу пропускной способности l_{ij} можно удалить из ограничений с помощью подстановки $x_{ij} = x'_{ij} + l_{ij}$. Для нового потока верхней границей пропускной способности будет величина $u_{ij} - l_{ij}$. В этом случае результирующий поток через узел i будет равен $f_i - l_{ij}$, а через узел j – $f_i + l_{ij}$. На рис. 6.16 показаны преобразования характеристик дуги.

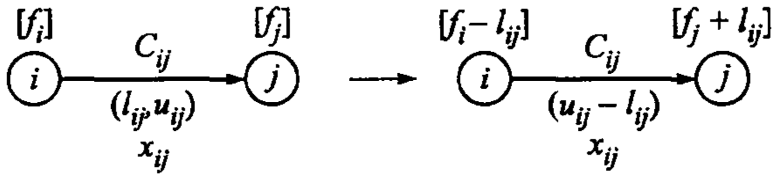


Рис. 6.16. Исключение нижних границ пропускных способностей

Основные ограничения формулируемой задачи (рис. 6.15) линейного программирования связаны с определением входных и выходных потоков, протекающих через каждый узел, что порождает следующую задачу линейного программирования.

	x_{12}	x_{13}	x_{14}	x_{23}	x_{25}	x_{34}	x_{35}	x_{46}	x_{56}	
min	3	4	1	5	6	1	2	2	4	
Узел 1	1	1	1							$= 100$
Узел 2	-1			1	1					$= 200$
Узел 3		-1		-1		1	1			$= 50$
Узел 4			-1			-1		1		$= -150$
Узел 5					-1		-1		1	$= -80$
Узел 6								-1	-1	$= -120$
нижние границы	0	0	50	0	0	70	0	100	0	
верхние границы	∞	∞	80	∞	∞	120	∞	120	∞	

В столбце, соответствующем переменной x_{ij}, всегда в строке i стоит $+1$, а в строке j находится -1. Остальные коэффициенты равны нулю. Такая структура коэффициентов типична для сетевых моделей.

Для переменных, представляющих потоки через дуги, имеющие ненулевые нижние границы пропускных способностей, выполняется замена:

$$x_{14} = x'_{14} + 50,$$
$$x_{34} = x'_{34} + 70,$$
$$x_{46} = x'_{46} + 100.$$

В результате получается следующая задача линейного программирования.

	x_{12}	x_{13}	x_{14}	x_{23}	x_{25}	x_{34}	x_{35}	x_{46}	x_{56}	
min	3	4	1	5	6	1	2	2	4	
Узел 1	1	1	1							$= 50$
Узел 2	-1			1	1					$= 200$
Узел 3		-1		-1		1	1			$= -20$
Узел 4			-1			-1		1		$= -130$
Узел 5					-1		-1		1	$= -80$
Узел 6								-1	-1	$= -20$
верхние границы	∞	∞	30	∞	∞	50	∞	20	∞	

Соответствующая сеть после исключения нижних границ пропускных способностей дуг показана на рис. 6.17. Данную сеть можно получить непосредственно из сети, представленной на рис. 6.15, с помощью преобразований, показанных на рис. 6.16, причем без необходимости записи в виде задачи линейного программирования.

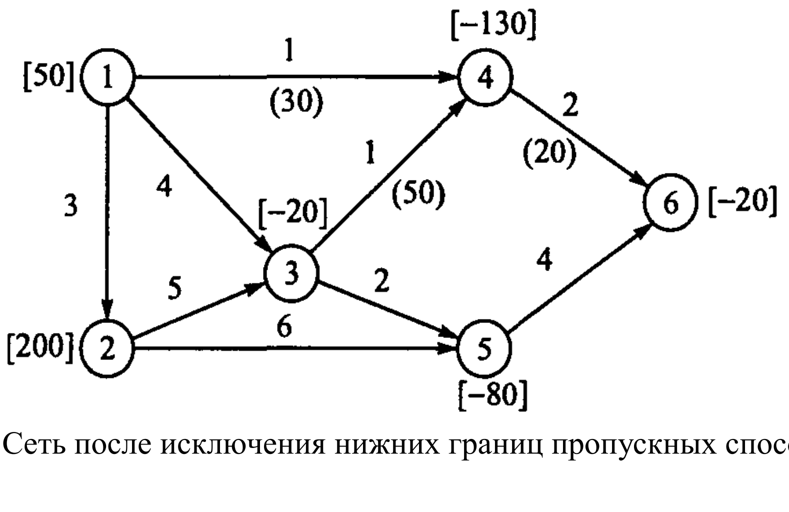


Рис. 6.17. Сеть после исключения нижних границ пропускных способностей

Шаблоны *Excel* и *MathCad* для решения транспортной задачи можно немного модифицировать для нахождения максимального потока.

6.5.1. Симплекс-метод для сетей с ограниченной пропускной способностью

Используется обычный симплекс-метод с учетом специальной структуры сетевой модели. Необходимо использовать условие равенства суммарного объема предложения суммарному объему спроса: $\sum_{i=1}^{n} f_i = 0$. Всегда можно удовлетворить данное условия, введя фиктивный источник или сток, которые можно связать с остальными узлами сети дугами с нулевой стоимостью и бесконечной пропускной способностью.

Сети с n узлами и нулевым результирующим потоком соответствуют $n - 1$ независимым ограничениям в виде равенств. Поэтому базисное решение должно содержать $n - 1$ переменных. Базисное решение соответствует остовному дереву данной сети.

Вводимая переменная (дуга) определяется путем вычисления разностей $z_{ij} - c_{ij}$ для всех текущих небазисных дуг (i, j). Если для всех разностей $z_{ij} - c_{ij} \leq 0$, тогда текущее базисное решение оптимально. Иначе в качестве вводимой в базис переменной выбираем дугу, которой соответствует наибольшее положительное значение разности $z_{ij} - c_{ij}$.

Вычисление разностей $z_{ij} - c_{ij}$ основано на соотношениях двойственности, точно так же как в транспортной модели. Обозначим через w_i переменную задачи, двойственной к задаче линейного программирования, которая (переменная) соответствует ограничению узла i. Тогда данная двойственная задача формулируется следующим образом:

$$z = \sum_{i=1}^{n} f_i \, w_i$$

при выполнении условий

$$w_i - w_j \leq c_{ij}, \qquad (i, j) \in A.$$

Из теории линейного программирования следует, что $w_i - w_j = c_{ij}$ для любой базисной дуги (i, j). Поскольку исходная задача линейного программирования по определению имеет одно избыточное ограничение, можно присвоить произвольное значение одной из переменной двойственной задачи. Для определенности положим $w_1 = 0$ (сравните с алгоритмом решения транспортной задачи). Затем следует решить (базисную) систему уравнений $w_i - w_j = c_{ij}$ для нахождения остальных переменных двойственной задачи. Далее вычисляем разности $z_{ij} - c_{ij}$ для небазисных переменных согласно формуле

$$z_{ij} - c_{ij} = w_i - w_j - c_{ij}.$$

Пример. Сеть трубопроводов связывает две станции опреснения воды с двумя городами. Ежедневное предложение опреснительных станций составляет 40 и 50 миллионов литров воды, города ежедневно потребляют 30 и 60 миллион литров воды. Каждая станция трубопроводами соединена с каждым городом непосредственно, однако они могут также перекачивать воду в города через специальную насосную станцию. Кроме того, станция 1 может транспортировать воду на станцию 2, а город 1 – в город 2. Данная сеть сбалансирована, так как в ней суммарный спрос равен суммарному предложению. Описанная сеть показана на рис. 6.18.

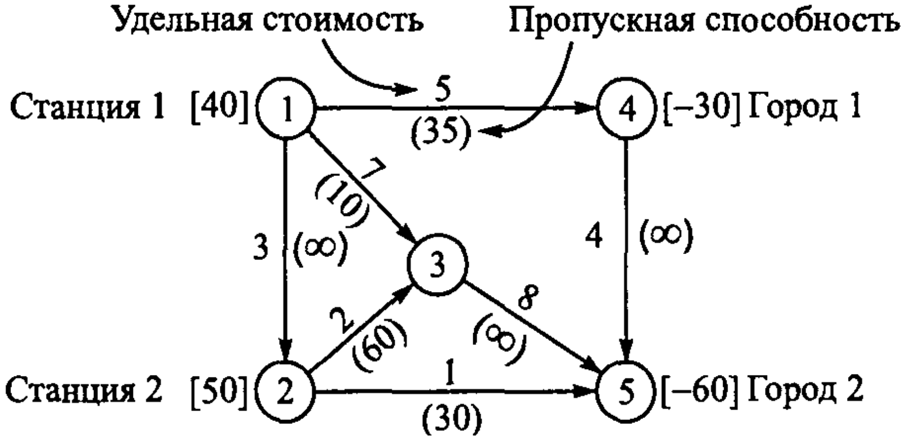


Рис. 6.18. Сеть трубопроводов

Определение начального допустимого базисного решения. На рис. 6.19 показано, что базисному решению соответствуют дуги (1, 3), (1,4) и (3, 5) с потоками 30, 10, 50 и 60 единиц соответственно. Оставшиеся дуги (показаны пунктиром) представляют небазисные переменные.

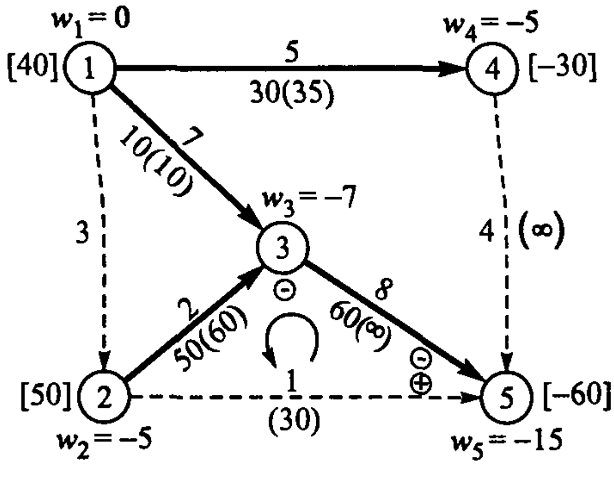


Рис. 6.19. Базисное решение

$$\underline{w_i - w_j = c_{ij}} \qquad w_1 = 0$$
$$w_1 - w_3 = 7 \qquad w_3 = -7$$
$$w_1 - w_4 = 5 \qquad w_4 = -5$$
$$w_2 - w_3 = 2 \qquad w_2 = -5$$
$$w_3 - w_5 = 8 \qquad w_5 = -15$$
$$\underline{z_{ij} - c_{ij} = w_i - w_j - c_{ij}}$$
$$z_{12} - c_{12} = 0 - (-5) - 3 = 2$$
$$z_{25} - c_{25} = -5 - (-15) - 1 = \mathbf{9}$$
$$z_{45} - c_{45} = -5 - (-15) - 4 = 6$$

Итерация 1. Дуга $(2,5)$ будет введена в базис. На рис. 6.19 видно, что дуга $(2,5)$ совместно с базисными дугами $(2,3)$ и $(3,5)$ образуют цикл. По определению остовное дерево не может содержать циклов. Поскольку поток через дугу $(2,5)$ должен возрасти, необходимо выровнять потоки через дуги, составляющие цикл таким образом, чтобы новое решение осталось допустимым. Для этого поток через дугу $(2,5)$ пометим знаком "+", потоки через другие дуги цикла – знаком "+" или "-", в зависимости от того, будут ли совпадать направления потоков в этих дугах с направлением потока в дуге $(2,5)$ при обходе цикла. Направление обхода дуг цикла всегда совпадает с направлением потока, протекающего через дугу, вводимую в базис. В данном случае это направление совпадает с направлением против часовой стрелки.

Поток через дугу $(2,5)$ равен 30, через дугу $(2,3)$ $50-30=20$, а через дугу $(3,5)$ $60-30=30$.

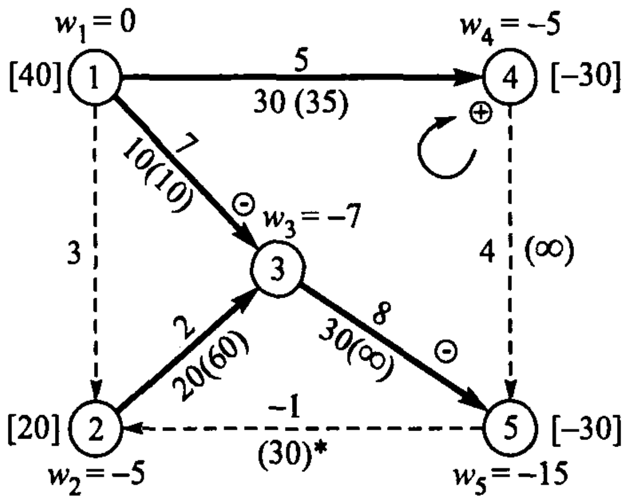
Поскольку никакая из текущих базисных переменных не приняла нулевого значения, дуга $(2,5)$ должна остаться небазисной, но с ненулевым значением в 30 единиц. Чтобы выполнить требование равенства нулю небазисных переменных, сделаем подстановку

$$x_{25} = 30 - x_{52}, 0 \le x_{52} \le 30.$$

Эта подстановка изменит уравнения для потоков, протекающих через узлы 2 и 5:

$$50 + x_{12} = x_{23} + x_{25} \;\; \Rightarrow \;\; 20 + x_{12} + x_{52} = x_{23}$$
$$x_{25} + x_{35} + x_{45} = 60 \;\; \Rightarrow \;\; x_{35} + x_{45} = x_{52} + 60.$$

Результаты этих изменений показаны на рис. 6.20. Направление потока через дугу $(2,5)$ изменилось на обратное (от узла 5 к узлу 2), причем, как и ожидалось, $x_{52}=0$. Описанная подстановка также требует изменения стоимости прохождения потока по дуге $(2,5)$ до -1. Те дуги, направления потоков которых изменены на противоположные, помечены в сети звездочкой.



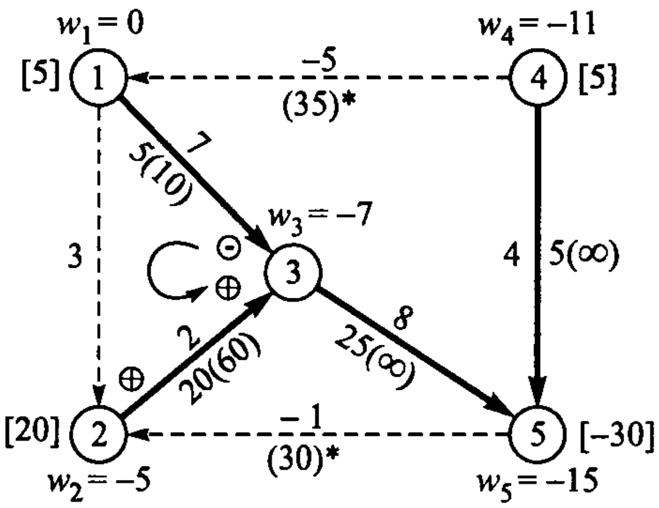
$$z_{12} - c_{12} = 0 - (-5) - 3 = 2$$
$$z_{52} - c_{52} = -15 - (-5) - (-1) = -9$$
$$z_{45} - c_{45} = -5 - (-15) - 4 = \mathbf{6}$$

Рис. 6.20. Итерация 1

<u>Итерация 2.</u> В базис следует ввести дугу $(4,5)$. Введение в базис этой дуги также приводит к образованию цикла.

Поток через дугу $(4,5)$ можно увеличить до 5 единиц (*Максимальное увеличение* потока через дугу $(1,4)$ равно $35-30=5$ единиц); эта дуга входит в базис, а дуга $(1,4)$ с потоком в 35 единиц исключается из базиса.

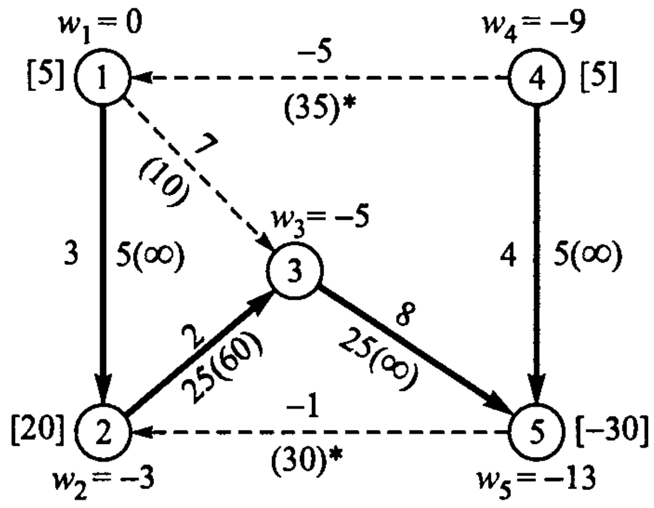
Выполнив подстановку $x_14 = 35 - x_{41}$, получим сеть, показанную на рис. 6.21. Для дуги $(1, 4)$ с обратным направлением потока изменена стоимость прохождения потока до -5.



$$z_{12} - c_{12} = 0 - (-5) - 3 = \mathbf{2}$$
$$z_{41} - c_{41} = -11 - 0 - (-5) = -6$$
$$z_{52} - c_{52} = -15 - (-5) - (-1) = -9$$

Рис. 6.21. Итерация 2

Итерация 3. В базис следует ввести дугу $(1, 2)$ с потоком в 5 единиц, тогда как дуга $(1, 3)$ исключается из базиса с нулевым значением потока. Новое решение представлено на рис. 6.22.



$$z_{13} - c_{13} = 0 - (-5) - 7 = -2$$
$$z_{41} - c_{41} = -9 - 0 - (-5) = -4$$
$$z_{52} - c_{52} = -13 - (-3) - (-1) = -9$$

Рис. 6.22. Итерация 3

Итерация 4. Из новых значений разностей $z_{ij} - c_{ij}$ видно, что последнее решение оптимально. Значения исходных переменных получаем путем обратной подстановки.

$$x_{12} = 5, \qquad x_{13} = 0$$
$$x_{14} = 35 - 0 = 35$$
$$x_{23} = 25$$
$$x_{25} = 30 - 0 = 30$$
$$x_{35} = 25, \qquad x_{45} = 5$$
$$\text{Суммарная стоимость} = 490.$$

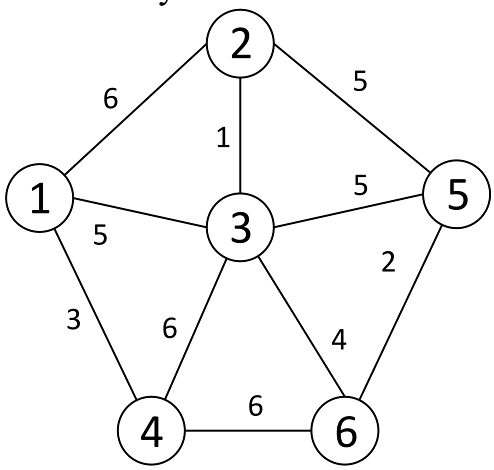
6.6. Задачи для самостоятельного решения

1. Нарисуйте сеть, заданную множествами

$$N = \{1, 2, 3, 4, 5, 6\},$$

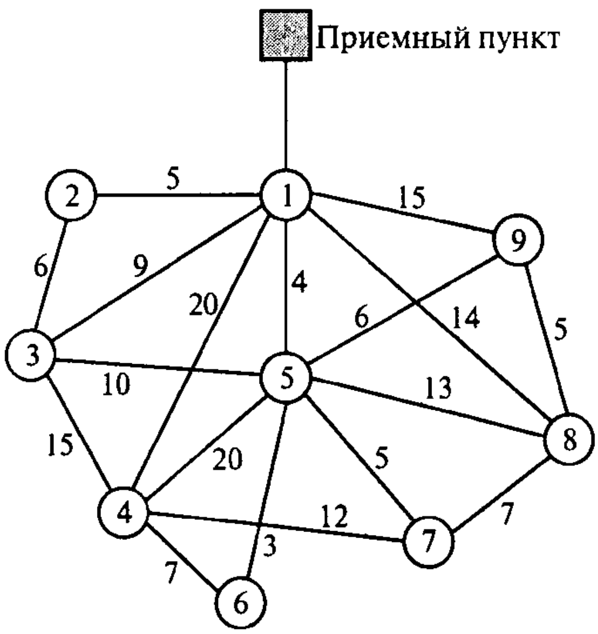
$$A = \{(1, 2), (1, 5), (2, 3), (2, 4), (3, 5), (3, 4), (4, 3), (4, 6), (5, 2), (5, 6)\}.$$

2. Телевизионная компания планирует подключение к своей кабельной сети пяти новых районов. На рисунке показана структура планируемой сети и расстояния между районами и телецентрами. Необходимо спланировать наиболее экономичную кабельную сеть.



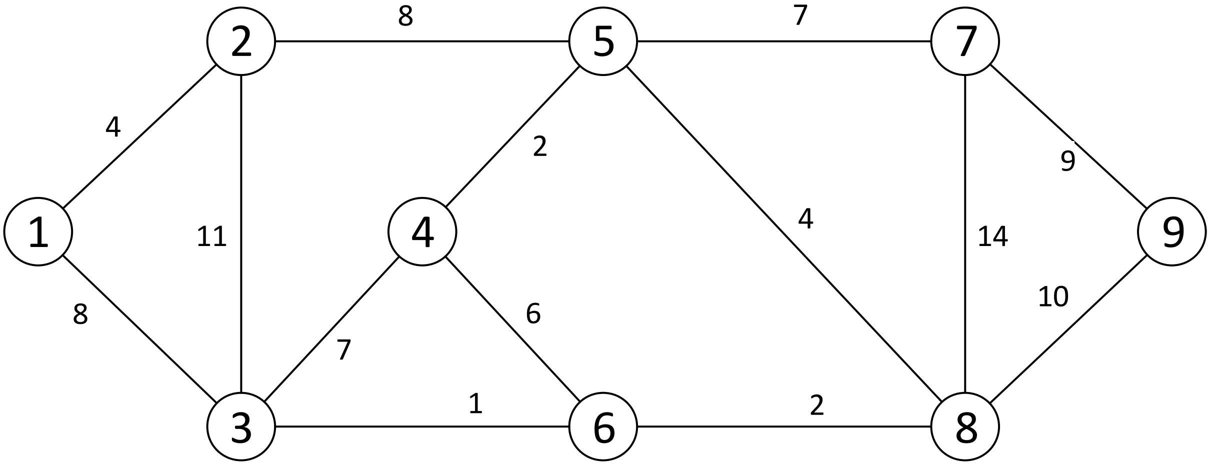
(Ответ: 15)

3. На рисунке показаны расстояния между платформами, добывающими газ, и приемным пунктом. Спроектируйте сеть трубопроводов минимальной длины, соединяющую приемный пункт со всеми добывающими платформами.



(Ответ: 41)

4. На рисунке показаны железнодорожные терминалы и существующие железнодорожные пути между ними. Необходимо выделить сегменты железных дорог так, чтобы связать все железнодорожные терминалы и минимизировать суммарную стоимость перевозок трейлерных платформ.



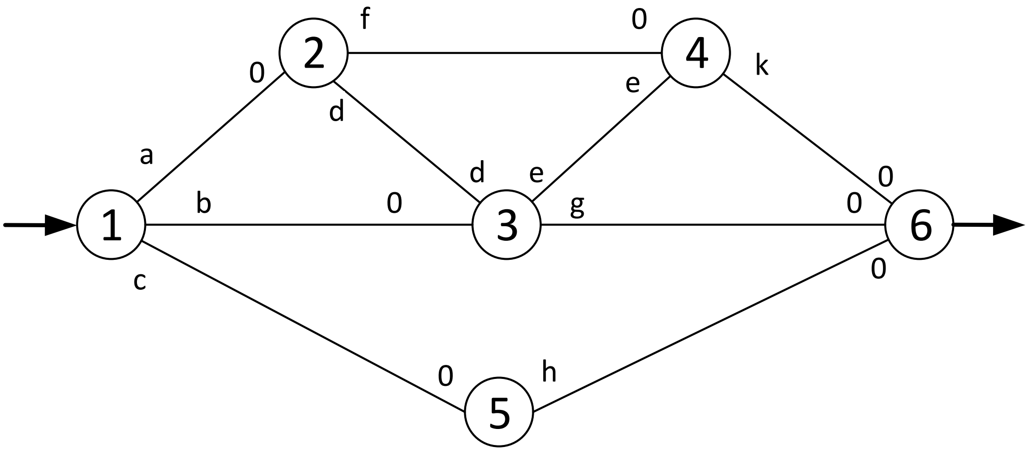
(Ответ: 37)

5. Компания по прокату автомобилей разрабатывает план по обновлению парка своих машин на следующие 5 лет. Каждый автомобиль должен проработать не менее одного года и не боле трех лет.

	Стоимость замены в зависимости от срока эксплуатации		
Год покупки	1	2	3
2015	4000	5400	9800
2016	4300	6200	8700
2017	4800	7100	-
2018	4900	-	-

(Ответ: 12500)

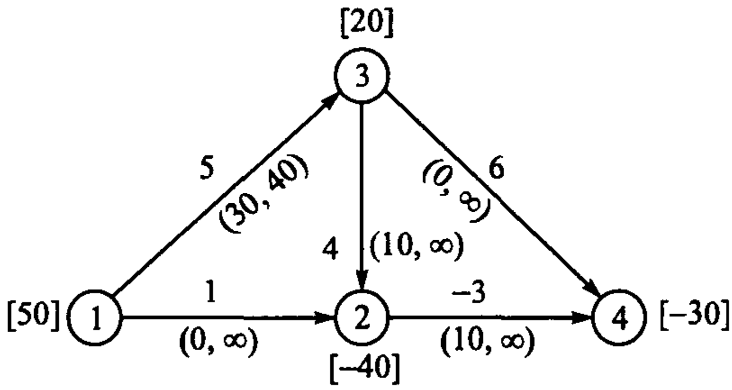
6. Чему равен максимальный поток между пунктами 1 и 6?



Вариант	1	2	3	4	5	6	7	8	9	10
a	9	8	5	2	1	7	7	6	7	1
b	4	9	5	6	8	7	1	6	7	2
c	2	1	4	9	5	5	8	6	9	7
d	9	3	4	3	3	1	1	8	3	4
e	5	5	4	3	1	8	9	8	8	2
f	7	7	3	2	5	7	2	5	6	8
g	2	4	8	2	9	4	5	2	4	2
h	1	8	3	4	5	2	9	9	6	3
k	7	4	6	1	8	8	6	8	5	4

7. Химическая компания владеет двумя заводами, которые производят определенные химические компоненты для двух клиентов; потребности этих клиентов составляют 660 и 800 тонн продукции ежемесячно. Первый завод может производить от 400 до 800 тонн продукции в месяц, а второй – от 450 до 900 тонн ежемесячно. На первом заводе расходы на производство одной тонны продукции составляют 25, на втором – 28. Сырье для заводов поставляют два поставщика, которые могут поставить не менее 500 тонн сырья по цене 200 за тонну первому заводу и не менее 750 тонн по цене 210 второму заводу. Химическая компания предполагает самостоятельно транспортировать и сырье, и готовую продукцию. Стоимость транспортировки одной тонны сырья от первого поставщика к заводам составляет 10 для первого завода и 12 – для второго. Аналогичные стоимости перевозок от второго поставщика равны 9 и 13, соответственно для первого и второго заводов. Стоимость перевозки одной тонны продукции от первого завода к клиенту1 и 2 составляет 3 и 4, от второго завода – 5 и 2 соответственно. Допустим, из одной тонны сырья можно получить одну тонну готовой продукции. Сформулируйте задачу в виде сетевой модели.

8. Сформулируйте задачу линейного программирования, минимизирующую стоимость потока в сети до и после исключения нижних границ пропускных способностей дуг.



9. Электрическая компания транспортирует по трубопроводам угольную пульпу от трех шахт (1, 2 и 3) к трем электростанциям (4, 5 и 6). Каждый трубопровод может транспортировать не более 10 тонн пульпы в час. Стоимость транспортировки одной тонны пульпы, а также предложение шахт и спрос электростанций представлены в следующей таблице.

	4	5	6	Предложение
1	5	8	4	8
2	6	9	12	10
3	3	1	5	10
Спрос	16	6	14	

Найдите оптимальную схему транспортировки угля к электростанциям.

Вопросы для дополнительного изучения

1. Алгоритм нахождения максимального потока
2. Алгоритм определения критического пути
3. Алгоритмы определения кратчайшего пути. Алгоритм поиска А*.
4. Алгоритмы определения кратчайшего пути. Алгоритм Флойда.
5. Алгоритмы определения кратчайшего пути. Волновой алгоритм.
6. Сетевая модель «самый надежный маршрут».
7. Сетевая модель «головоломка о трех бидонах».
8. Сетевая модель распределения рабочих.

Приложение 1

Список вопросов, позволяющих выявить ошибки ввода условия задачи в Excel

	Вопрос	Месторасположение в Excel
1	Правильно ли Вы ввели численные значения и знаки (+, –) коэффициентов целевой функции и ограничений, правых частей ограничений ?	Экранная форма
2	Сбалансирована ли двухиндексная задача?	Экранная форма
3	Правильны ли формулы в целевой ячейке и в ячейках левых частей ограничений? Для наглядности проверки поставьте курсор на ячейку с формулой и сделайте двойной щелчок левой клавишей мыши. Рамкой в экранной форме будут выделены ячейки, участвующие в данной формуле.	Экранная форма
4	Правильно ли указан адрес целевой ячейки?	Окно *Поиск решения*
5	Правильно ли указано направление оптимизации ЦФ?	Окно *Поиск решения*
6	Правильно ли указаны адреса ячеек переменных?	Окно *Поиск решения* Поле *Изменяя ячейки*
7	Правильно ли введены знаки ограничений (<=, >=, =) ?	Экранная форма, Окно *Поиск решения* Поле *Ограничения*
8	Правильно ли указаны адреса ячеек левых и правых частей ограничений?	Окно *Поиск решения* Поле *Ограничения*
9	Не забыли ли Вы задать требование неотрицательности переменных?	Окно *Поиск решения* Поле *Ограничения*
10	Не забыли ли Вы задать требования по единичному значению верхней границы переменных (для задач с булевыми переменными)	Окно *Поиск решения* Поле *Ограничения*
11	Не забыли ли Вы задать условие целочисленности переменных (согласно условию задачи)?	Окно *Поиск решения* Поле *Ограничения*
12	Проверьте правильность установки параметров	Окно *Параметры поиска решения*

В отчет о лабораторной работе входят:

1. **Титульный лист** установленного образца (Приложение 3).
2. **Оглавление.**
3. На третьем листе отчета повторяется название лабораторной работы, после которого в обязательном порядке указывается **общая информация, использованная в ходе выполнения работы** (задание, определения и краткое описание представленных в отчете данных, особенности реализации и другая дополнительная информация, индивидуально характеризующая работу).
4. **Полный текст отчета**, включающий поясняющие рисунки, графики и программный код, оформленные с соблюдением общих требований и рекомендаций к их содержанию и стилям форматирования.
5. **Выводы.**

Отчет о лабораторной работе выполняют с помощью компьютерных средств реализации (*MS Office* 2010 (или *MS Office* 2013/2016), *Mathcad*) на листах формата А4 книжной ориентации. Допускается оформление рисунков большого размера в альбомном режиме. При этом колонтитулы страницы должны быть сохранены по меньшим сторонам листа, а правильное расположение рисунка для его просмотра должно обеспечиваться одним поворотом листа на 90° по часовой стрелке.

Отчет оформляют с использованием следующих стилей:

- размеры полей страницы:
  - ✓ левое – 3 см;
  - ✓ правое – 2 см;
  - ✓ верхнее – 2 см;
  - ✓ нижнее – 2 см;

- текстовая информация:
  - ✓ шрифт – Times;
  - ✓ начертание шрифта – обычный;
  - ✓ размер шрифта – 14 пунктов;
  - ✓ межстрочный интервал – 1,5 строки;
  - ✓ метод выравнивания – по ширине;
  - ✓ отступ первой строки – 1,25 см;
- программный код:
  - ✓ шрифт – Courier (моноширинный);
  - ✓ начертание шрифта – обычный;
  - ✓ размер шрифта – 12 пунктов;
  - ✓ межстрочный интервал – 1,5 строки;
  - ✓ метод выравнивания – по левому краю;
  - ✓ отступ первой строки – отсутствует;
- комментарии к отчету:
  - ✓ шрифт – Monotype Corsiva;
  - ✓ начертание шрифта – полужирный;
  - ✓ размер шрифта – 12 пунктов;
  - ✓ межстрочный интервал – 1,5 строки;
  - ✓ отступ первой строки – отсутствует;
- встречающиеся в отчете заголовки выравнивают по центру, печатают с прописной (заглавной) буквы без точки в конце, используемый шрифт – Times, 16 пунктов, полужирный;
- верхний колонтитул отчета выравнивают по правому краю, используемый шрифт – Times, 10 пунктов, обычный;
- в верхнем колонтитуле указывают номер и название лабораторной работы, номер группы, фамилия и инициалы студента;
- нижний колонтитул оформляют в том же стиле, что и верхний;

- в нижнем колонтитуле указывают дату оформления отчета и приводят сквозную нумерацию страниц;
- титульный лист учитывают в нумерации, но номера страниц и колонтитулы на нем не проставляют.

Для сдачи лабораторной работы отчет распечатывают на листах белой односортной бумаги. Рабочей является только одна сторона листа. После распечатки отчет скрепляют степлером в левом верхнем углу таким образом, чтобы обеспечить надежное соединение всех листов, не препятствующее свободному просмотру отдельных страниц.

Рукописные тексты, рисунки, оформленные вручную (в том числе, частично), а также напечатанные с двух сторон, на черновиках или неправильно скрепленные отчеты **не допускаются**.

Приложение 3

Шаблон титульного листа для отчетов о лабораторных работах

МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технологический университет «СТАНКИН»
(ФГБОУ ВО МГТУ «СТАНКИН»)

Институт информационных систем и технологий

Кафедра информационных технологий и вычислительных систем

Исследование операций

Лабораторная работа № __

____

(наименование лабораторной работы)

Вариант № __

Выполнил: студент группы ____ _____________

Проверил: _________ _____________

Москва 201_

Библиографический список

1. Балашевич В.А. Основы математического программирования: учебник/ В.А. Балашевич. – Минск, 1985. – 173 с.
2. Вентцель Е.С. Исследование операций/ Е.С. Вентцель. – М.: Советское радио, 1972. – 552 с.
3. Ефромеев Н.М., Ефромеева Е.В. Информатика и информационно-телекоммуникационные технологии : учебное пособие/ Н.М. Ефромеев, Е.В. Ефромеева – М.: ФГБОУ ВО «МГТУ «СТАНКИН», 2015. – 175 с.
4. Ефромеева Е.В. Задачи линейного программирования: Методические указания к практическим занятиям/ Е.В. Ефромеева. – М.: ГОУ ВПО МГТУ «Станкин», 2009. – 50 с.
5. Ефромеева Е.В. Методы исследования операций в машиностроении: примеры, задачи/ Е.В. Ефромеева. – М.: ГОУ ВПО МГТУ «Станкин», 2010. – 142 с.
6. Ефромеева Е.В. Основы исследований операций в машиностроении : учебное пособие/ Е.В. Ефромеева. – М.: ГОУ ВПО МГТУ «Станкин», 2009. – 104 с.
7. Правила оформления курсовой работы по дисциплине «Модели и методы анализа проектных решений»: методические указания/ сост. Е.В. Ефромеева. – М.: ФГБОУ ВПО МГТУ «Станкин», 2013. – 31 с.
8. Сборник задач по высшей математике для экономистов: Учебное пособие/ Под ред. В.И. Ермакова. – М.: ИНФРА-М, 2003. – 575 с.
9. Таха, Хемди А. Введение в исследование операций. – 7-е изд. – М.: Вильямс, 2005. – 912 с.
10. Шикин Е. В., Шикина Г. Е. Исследование операций: учебник/ Е. В. Шикин, Г.Е. Шикина. – М.: ТК Велби, Проспект, 2006. – 280 с.

Учебное издание

Ефромеев Николай Максимович, **Ефромеева** Елена Валентиновна

МЕТОДЫ ИССЛЕДОВАНИЯ ОПЕРАЦИЙ В МАШИНОСТРОЕНИИ

Учебное пособие

Подписано в печать 24.12.2017 г.
Формат 60×90 $^{1}/_{16}$. Бумага 80 г.
Усл. печ. л. 9,75. Тираж 200 экз. Заказ 5.

Отпечатано в Издательском центре
ФГБОУ ВО «Московский государственный
технологический университет ‹‹СТАНКИН››.
127055, Москва, Вадковский пер., 3а.
Тел.: 8(499) 973-31-93