

**Набережночелнинский институт (филиал)  
федерального государственного автономного образовательного  
учреждения  
высшего профессионального образования  
«Казанский (Приволжский) федеральный  
университет»**



**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ  
В ЭКОНОМИКО-МАТЕМАТИЧЕСКОМ  
МОДЕЛИРОВАНИИ  
ДЛЯ СТУДЕНТОВ ВУЗОВ**

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки:

- 01.03.02 (010400.62) – «Прикладная математика и информатика»,
- 09.03.02 – «Информационные системы и технологии»,
- 09.03.03 (230700.62) – «Прикладная информатика (по отраслям)»,
- 38.03.01 (080100.62) – «Экономика»,
- 38.03.02 (080200.62) – «Менеджмент»
- 38.03.05 (080500.62) – «Бизнес-информатика»,

Протокол № 513 от «19» мая 2015 г.

**Набережные Челны, 2015**

УДК 004.94 (075.8)  
ББК 22.19я73  
Г 20

**Печатается по решению учебно-методической комиссии  
Экономического отделения Набережночелнинского филиала  
федерального государственного автономного образовательного учреждения  
высшего профессионального образования  
«Казанский (Приволжский) федеральный университет»**

**протокол № 5 от « 20 » октября 2014 г.**

Рецензенты:

- Т.Г. Макусева** Нижнекамский институт (филиал) ФГБОУ ВПО «Казанский национальный исследовательский технологический университет», зав. кафедры математики, кандидат педагогических наук, доцент.
- А.Г. Исавнин** Набережночелнинский институт (филиал) ФГАОУ ВПО «Казанский (Приволжский) федеральный университет», докт. ф-м. наук, профессор кафедры Бизнес-информатики и математических методов в экономике.

**Г.А. Гареева, И.И. Еремина, А.Г. Файзуллина** Информационные технологии в экономико-математическом моделировании: учебное пособие – Набережные Челны: Издательско-полиграфический центр НЧИ КФУ, 2015. – 180 с.: ил.

Круг проблем, изучаемых в Экономико-математическом моделировании (ЭММ), еще недостаточно определен. В учебном пособии излагаются методы и алгоритмы решения линейных, целочисленных, транспортных и других оптимизационных задач, а также их реализация на ЭВМ. Наиболее изученными являются задачи с линейными функциями и ограничениями. Для нахождения оптимальных решений этих задач разработаны эффективные методы, алгоритмы и программы для ЭВМ.

Учебное пособие является обязательным дополнением лекционного курса по дисциплине Экономико-математическое моделирование и помогает получить навыки работы со специализированными программами для принятия управленческих решений.

В учебном пособии изложены экономико-математические методы и модели для решения прикладных задач управления экономическими процессами. Рассмотрены некоторые вопросы применения ЭВМ для принятия управленческих решений. В пособие включено достаточное количество примеров задач прикладного характера. В примерах даны подробные решения с нужными пояснениями.

При написании учебного пособия преследовалась цель – доступно и понятно изложить материал, использовались разнообразные источники, а также сведения накопленные авторами в результате опыта педагогической работы в Набережночелнинском институте КФУ.

Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки:

01.03.02 (010400.62) – «Прикладная математика и информатика»,  
09.03.02 – «Информационные системы и технологии»,  
09.03.03 (230700.62) – «Прикладная информатика (по отраслям)»,  
38.03.01 (080100.62) – «Экономика»,  
38.03.02 (080200.62) – «Менеджмент»  
38.03.05 (080500.62) – «Бизнес-информатика»,

Протокол №513 от «19» мая 2015 г.

© Г.А.Гареева, И.И. Еремина, А.Г. Файзуллина 2015  
© Набережночелнинский институт (филиал) КФУ,  
кафедра Бизнес-информатики и математических  
методов в экономике, 2015

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
РАБОТА С МАССИВАМИ КОНСТАНТ .....	6
РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ MS EXCEL И MATHCAD .....	24
РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ С ПОМОЩЬЮ MS EXCEL И MATHCAD .....	47
ЦЕЛОЧИСЛЕННАЯ ОПТИМИЗАЦИЯ.....	65
СИМПЛЕКС-МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ.....	77
ПРОГРАММНЫЕ СРЕДСТВА РЕШЕНИЯ ЗАДАЧ СИМПЛЕКС-МЕТОДОМ	95
1. Программа Simplex-M, БГУИР, кафедра ИИТ, 1997 г.....	96
2. Программа Simplex, С-ППТ, Жердин Д.А. (5 курс), 2002 г.....	98
3. Программа SASIMPL - ДВГТУ Богдановский А. (4 курс), 1996 г.....	99
4. Руководство по использованию программы Simplexwin 3.1 .....	102
5. Решение задач линейного программирования с помощью программы Simplexwin 3.1.....	104
6. Разработка своей программы Simplex.....	112
ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ.....	116
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭЛЕМЕНТАХ ТЕОРИИ ГРАФОВ И СЕТЕВОМ УПРАВЛЕНИИ .....	138
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ MS EXCEL РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ .....	148
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ MS EXCEL В ИГРАХ С ПРИРОДОЙ	159
СПИСОК ЛИТЕРАТУРЫ.....	164
ПРИЛОЖЕНИЯ.....	167
Приложение 1. Текст программы на языке Турбо Паскаль.....	167
Приложение 2. Результаты работы программы на примере задачи из 2 раздела	175

## ВВЕДЕНИЕ

Современные требования к знаниям вызвали поиски различных путей построения учебного предмета:

- на основе системных идей;
- на основе восхождения от абстрактного к конкретному;
- на основе теории поэтапного формирования умственных действий и понятий.

При реализации в обучении все эти способы обнаруживают высокую эффективность и позволяют достичь достаточно высокого качества знаний, если внедрить в учебный процесс информационные технологии.

Для успешного овладения курсов «Математическое программирование», «Экономико-математическое моделирование», «Исследование операций» и пр. студентам необходимы глубокие знания математического анализа, теории вероятностей, линейной алгебры.

В отличие от чисто теоретических дисциплин курсы «Математическое программирование», «Экономико-математическое моделирование», «Исследование операций» имеют ярко выраженную прикладную направленность.

Данные курсы готовят студентов к изучению курса «Компьютерное моделирование», а также к прослушиванию в дальнейшем спецкурсов, связанных с математическим моделированием.

Под исследованием операций понимается применение математических, количественных методов для обоснования решений во всех областях целенаправленной человеческой деятельности. Среди методов принятия оптимального решения выделены математическое программирование, теория игр и теория массового обслуживания.

Для достижения цели обучения этот курс был разбит на составляющие:

1. изложение теоретических основ, методов решения всех классов задач оптимизации: линейного, целочисленного, нелинейного и стохастического программирования;
2. рассмотрена постановка, решение и анализ всех классов задач оптимизации. Разработана система заданий решения типовых задач принятия оптимальных решений и контрольных вопросов по предложенным темам;
3. разработана система лабораторных работ применения информационных технологий при решении задач.

В результате изучения студент получает следующие знания:

- ✓ знание основных принципов математического моделирования, способов задания цели операции;
- ✓ знание основных понятий теории игр;
- ✓ знание основных свойств матричных игр и методов их решения;
- ✓ знание основных понятий теории массового обслуживания (ТМО).

В процессе изучения курса студенты овладевают следующими умениями:

- ✓ строить простейшие модели исследования операций, формировать цель операции;
- ✓ решать матричные игры, применять различные методы (аналитический, графический, сведение к линейному программированию);
- ✓ определять основные характеристики систем массового обслуживания (на примере марковских моделей систем).

А так же приобретают твердые навыки решения математических задач с доведением решения до практически приемлемого результата и развивают на этой базе логическое и алгоритмическое мышление; вырабатывают первичные навыки математического исследования прикладных вопросов и развивают необходимую интуицию в вопросах приложения математики; самостоятельно разрабатывают и реализовывают алгоритмы решения задач оптимизации для ЭВМ.

Особое внимание авторы уделяют построению лабораторных работ, так как именно они позволяют изучить и закрепить теоретический материал, сопоставить результаты решения задач в ручном режиме и при помощи применения НИТ. Студенты, работая над созданием своих проектов, овладевают новыми информационными технологиями и приобретают опыт творческой деятельности, который предусматривает активную познавательную деятельность. В целом информационные технологии обогащают образовательный процесс, наполняют его новым смыслом.

Построение учебного предмета – всегда исследовательская задача. Трансформация системы научного знания в учебное – одна из актуальных проблем педагогики. Основная трудность возникает при сочетании в учебном процессе двух требований: следовать в учебном изложении, с одной стороны, логике научного знания, а с другой – психолого-педагогической логике, то есть требованию оптимально организованного обучения, обеспечивающего интерес к знаниям, доступность изложения и развивающий характер обучения, а также отражение системы знаний в сознании обучаемых. Построение учебного предмета, в котором информационные технологии выступают, как средство решения проблемы соотношения теоретического и практического материала при обучении позволяет повысить эффективность процесса обучения и достичь достаточно высокого качества знаний.

## РАБОТА С МАССИВАМИ КОНСТАНТ

### Формулы массива и их ввод

*Формула массива* может выполнить несколько вычислений, а затем вернуть одно значение или группу значений. Формула массива воздействует на несколько наборов значений, называемых аргументами массива. При этом все аргументы массива должны иметь одинаковое количество строк и столбцов. Каждый аргумент массива должен иметь соответствующий номер строки и столбца. Формула массива создается так же, как и простая формула. Выделяется ячейка или группа ячеек, в которых необходимо создать формулу, вводится формула, а затем нажимаются клавиши



Если необходимо вычислить одно значение, Microsoft Excel может понадобиться выполнить несколько действий для возврата такого значения. *Например*, следующая формула вычисляет среднее значение только тех ячеек, принадлежащих диапазону D5:D15, которым в столбце А поставлена в соответствие строка «авиалиния небеса». Функция ЕСЛИ находит ячейки в диапазоне A5:A15, содержащие строку «авиалиния небеса», и возвращает значения, соответствующие этой строке в диапазоне D5:D15, функции СРЗНАЧ.

{=СРЗНАЧ(ЕСЛИ(A5:A15="авиалиния небеса", D5:D15))}

Для вычисления нескольких значений в формуле массива, необходимо ввести массив в диапазон ячеек, имеющих соответствующее число строк или столбцов, как аргументы массива.

### Константы в формулах массива

Обычно формула при обработке нескольких аргументов возвращает одно значение; в качестве аргумента формулы может при этом выступать либо ссылка на ячейку, содержащую значение, либо само значение. Для создания

ссылки на диапазон ячеек используется формула массива, позволяющая ввести в одну ячейку массив значений.

Этот массив значений называется массивом констант; удобен он тем, что при этом не требуется заполнять значениями вспомогательные ячейки. Чтобы создать массив констант, выполните следующие действия:

- ✓ весь массив заключается в фигурные скобки ({ });
- ✓ значения столбцов разделяются точками с запятой (;);
- ✓ значения строк разделяются двоеточиями (:).

*Например*, вместо ввода четырех чисел (10, 20, 30, 40) в отдельные ячейки их можно ввести в массив, непосредственно в используемой функции в фигурных скобках: {10; 20; 30; 40}.

Чтобы записать массив  $\begin{pmatrix} 10 & 20 & 30 & 40 \\ 50 & 60 & 70 & 80 \end{pmatrix}$  непосредственно в используемой функции необходимо представить значения «10, 20, 30, 40» и «50, 60, 70, 80», находящиеся в расположенных друг под другом ячейках, следующим образом: {10;20;30;40:50;60;70;80}.

### **Элементы массива констант**

- ✓ Массив констант может состоять из чисел, текста, логических значений (например, ИСТИНА или ЛОЖЬ) или значений ошибок (например, #Н/Д).
- ✓ Числа в массиве могут быть целыми, с десятичной точкой или экспоненциальными.
- ✓ Текст должен быть взят в двойные кавычки, например «Четверг».
- ✓ Массив констант может состоять из элементов разного типа, *например*, {1,3,4;ИСТИНА, ЛОЖЬ, ИСТИНА}.
- ✓ Элементы массива должны быть константами, но не формулами.
- ✓ Массив констант не может содержать знаки доллара (\$), в круглых скобках и процента (%).
- ✓ Массив констант не может содержать ссылок.

- ✓ Массив констант не может иметь столбцы или строки разного размера.

## Вычисление определителя матрицы

**МОПРЕД(массив)** - возвращает определитель матрицы (матрица хранится в массиве). *Массив* - это числовой массив с равным количеством строк и столбцов.

- ✓ Массив может быть задан как интервал ячеек, например, A1:C3 или как массив констант, например {1;2;3;4;5;6;7;8;9} или как имя, именуемое интервал или массив.
- ✓ Если какая-либо ячейка в массиве пуста или содержит текст, то функция МОПРЕД возвращает значение ошибки #ЗНАЧ!.
- ✓ МОПРЕД также возвращает значение ошибки #ЗНАЧ!, если массив имеет неравное количество трок и столбцов.

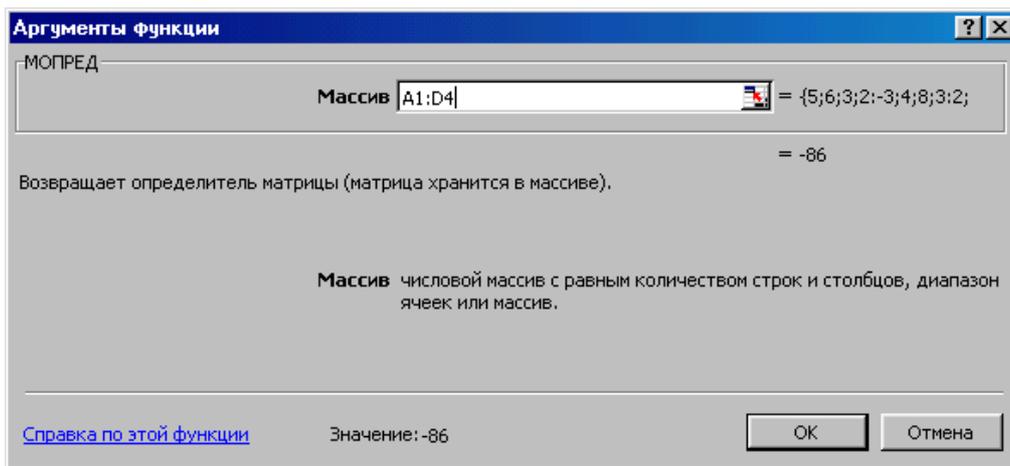
*Замечания.*

1. Определитель матрицы - это число, вычисляемое на основе значений элементов массива. Для массива A1:C3, состоящего из трех строк и трех столбцов, определитель вычисляется следующим образом:  
$$\text{МОПРЕД}(A1:C3) = A1*(B2*C3-B3*C2) + A2*(B3*C1-B1*C3) + A3*(B1*C2-B2*C1)$$
2. Определители матриц обычно используются при решении систем уравнений с несколькими неизвестными.
3. МОПРЕД производит вычисления с точностью примерно 16 значащих цифр, что может в некоторых случаях приводить к небольшим численным ошибкам.

## Создание формулы вычисления определителя массива

1. Занесите исходную матрицу в таблицу.
2. В свободном месте таблицы выделите ячейку, в которой будет выведен результат.

3. Вызовите функцию МОПРЕД(). На экране появится диалоговое окно:



4. Нажмите красную стрелочку в строке Массив, при этом на экране появится лист с исходными данными.
5. Выделите матрицу, для которой надо найти определитель.
6. Нажмите красную стрелочку для возврата в диалоговое окно:



Нажмите клавишу  .

Кроме этого, можно в данной функции ввести массивы непосредственно:

**Примеры:**

МОПРЕД({1; 3; 8; 5; 1; 3; 6; 1; 1; 1; 1; 0; 7; 3; 10; 2}) равняется 88.

МОПРЕД({3; 6; 1; 1; 1; 1; 0; 3; 10; 2}) равняется 1.

МОПРЕД({3; 6; 1; 1}) равняется -3.

МОПРЕД( { 1; 3; 8; 5 : 1; 3; 6; 1 }) равняется #ЗНАЧ!, так как массив имеет неравное количество строк и столбцов.

**Вычисление обратной матрицы**

**МОБР(массив)** - возвращает обратную матрицу для матрицы, хранящейся в массиве. *Массив* - это числовой массив с равным количеством строк и столбцов.

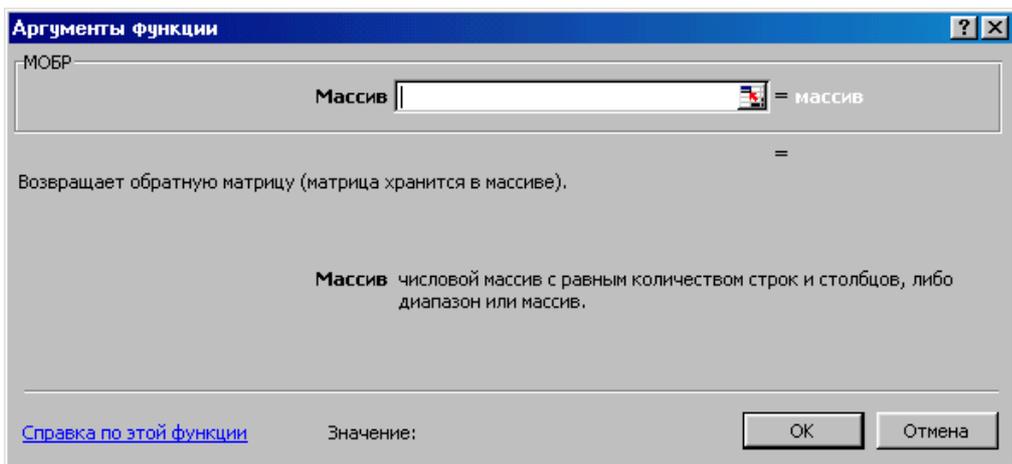
1. Массив может быть задан как диапазон ячеек, например A1:C3; как массив констант, например {1;2;3: 4;5;6: 7;8;9} или как имя диапазона или массива.
2. Если какая-либо из ячеек в массиве пуста или содержит текст, то функция МОБР возвращает значение ошибки #ЗНАЧ!.
3. МОБР также возвращает значение ошибки #ЗНАЧ! если массив имеет неравное число строк и столбцов.

#### *Замечания.*

1. Формулы, которые возвращают массивы, должны быть введены как формулы массива.
2. Обратные матрицы, как и определители, обычно используются для решения систем уравнений с несколькими неизвестными. Произведение матрицы на ее обратную - это единичная матрица, то есть квадратный массив, у которого диагональные элементы равны 1, а все остальные элементы равны 0.
3. МОБР производит вычисления с точностью до 16 значащих цифр, что может привести к небольшим численным ошибкам округления.
4. Некоторые квадратные матрицы не могут быть обращены, в таких случаях функция МОБР возвращает значение ошибки #ЧИСЛО!. Определитель такой матрицы равен 0.

#### **Создание формулы вычисления обратного массива**

1. Занесите исходную матрицу в таблицу.
2. В свободном месте таблицы выделите блок для обратной матрицы.
3. Вызовите функцию МОБР() в классе математических функций. На экране появится диалоговое окно:



4. Нажмите красную стрелочку в строке Массив, при этом на экране появится лист с исходными данными.
5. Выделите матрицу, для которой надо найти обратную.
6. Нажмите красную стрелочку для возврата в диалоговое окно



7. Нажмите комбинацию клавиш Ctrl + ↑ Shift + Enter. На экране появится обратная матрица.

Кроме этого, можно в данной функции ввести массивы непосредственно:

*Примеры:*

МОБР({4; -1; 2; 0}) равняется {0; 0,5; -1; 2}

МОБР({1; 2; 1; 3; 4; -1; 0; 2; 0}) равняется

{0,25; 0,25; -0,75 : 0; 0; 0,5 : 0,75; -0,25; -0,25}

## Умножение массивов

**МУМНОЖ(массив1;массив2)** - возвращает произведение матриц (матрицы хранятся в массивах). Результатом является массив с таким же числом строк, как массив 1 и с таким же числом столбцов, как массив 2.

*Массив1, массив2* - это перемножаемые массивы; могут быть заданы как интервалы, массивы констант или ссылки.

Количество столбцов аргумента *массив1* должно быть таким же, как количество строк аргумента *массив2*, и оба массива должны содержать только числа.

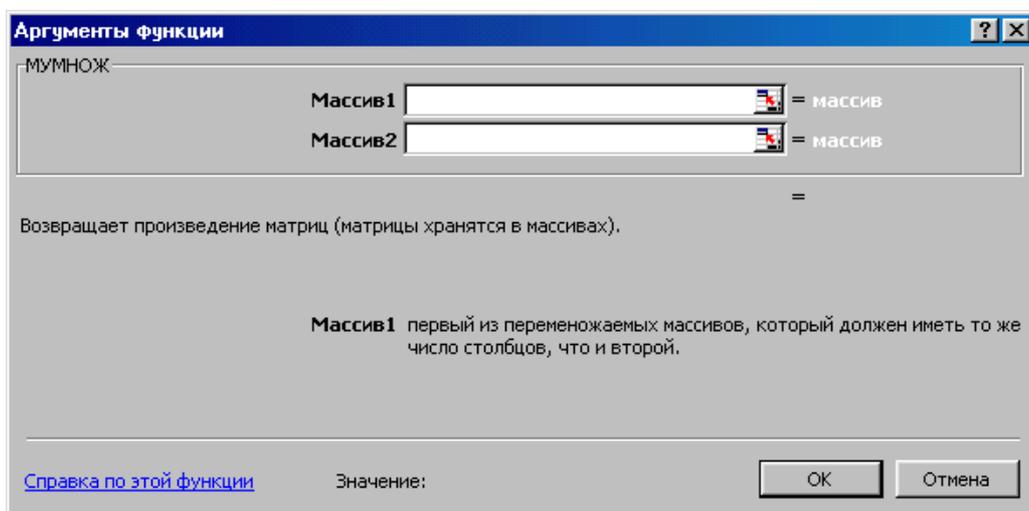
Если хотя бы одна ячейка в аргументах пуста или содержит текст, или если число столбцов в аргументе *массив1* отличается от числа строк в аргументе *массив2*, то функция МУМНОЖ возвращает значение ошибки #ЗНАЧ!.

*Замечания.*

1. Массив *A*, который является произведением двух массивов *B* и *C* определяется следующим образом: где *i* - это номер строки, а *j* - это номер столбца.
2. Формулы, которые возвращают массивы, должны быть введены как формулы массива.

## Создание формулы перемножения массивов

1. Занесите исходные матрицы в таблицу.
2. В свободном месте таблицы выделите блок для результирующей матрицы.
3. Вызовите функцию МУМНОЖ() в классе математических функций. На экране появится диалоговое окно:



4. Нажмите красную стрелочку в строке Массив1, при этом на экране появится лист с исходными данными.
5. Выделите первую матрицу.
6. Нажмите красную стрелочку для возврата в диалоговое окно:



7. Нажмите красную стрелочку в строке Массив2, при этом на экране появится лист с исходными данными.
8. Выделите вторую матрицу.
9. Нажмите красную стрелочку для возврата в диалоговое окно:



10. Нажмите комбинацию клавиш  +  +  . На экране появится обратная матрица.

Кроме этого, можно в данной функции ввести массивы непосредственно:

*Примеры.*

МУМНОЖ({1; 3 : 7; 2}; {2; 0: 0; 2}) равняется {2; 6: 14; 4}

МУМНОЖ({3; 0 : 2; 0}; {2; 0: 0; 2}) равняется {6; 0 : 4; 0}

МУМНОЖ({1; 3; 0 : 7; 2; 0: 1; 0; 0}; {2; 0: 0; 2}) равняется #ЗНАЧ!,

поскольку первый массив имеет три столбца, а второй массив имеет только две строки.

## Транспонирование массива

**ТРАНСП(массив)** - возвращает транспонированный массив.

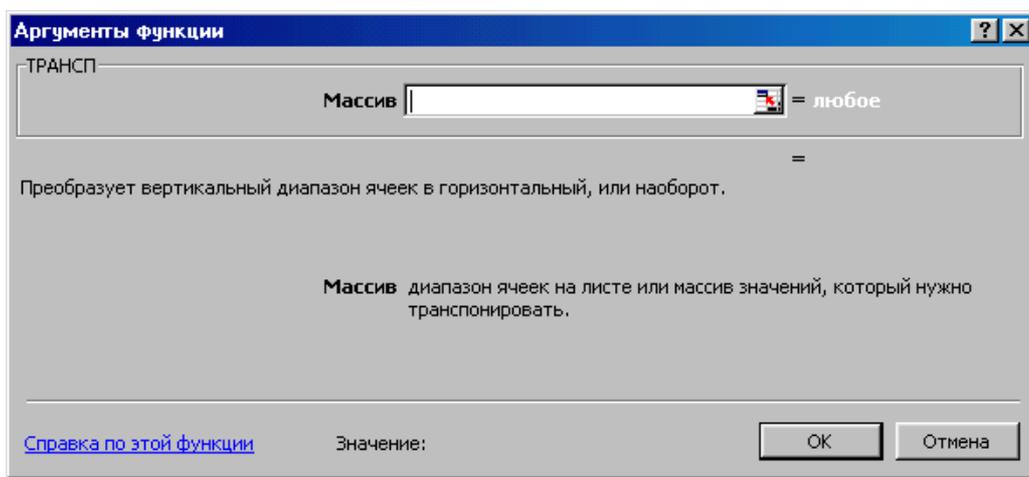
Функция ТРАНСП должна быть введена как формула массива в интервал, который имеет столько же строк и столбцов, соответственно, сколько столбцов и строк имеет аргумент массив. Функция ТРАНСП используется для того, чтобы поменять ориентацию массива на рабочем листе с вертикальной на горизонтальную и наоборот.

*Массив* - это транспонируемый массив или диапазон ячеек на рабочем листе. Массив может быть интервалом ячеек. Транспонирование массива

заключается в том, что первая строка массива становится первым столбцом нового массива, вторая строка массива становится вторым столбцом нового массива, и так далее.

### Создание формулы для транспонирования массива

1. Занесите исходную матрицу в таблицу.
2. В свободном месте таблицы выделите блок для результирующей матрицы.
3. Вызовите функцию ТРАНСП() в классе математических функций. На экране появится диалоговое окно:



4. Нажмите красную стрелочку в строке Массив, при этом на экране появится лист с исходными данными.
5. Выделите исходную матрицу.
6. Нажмите красную стрелочку для возврата в диалоговое окно



7. Нажмите комбинацию клавиш **Ctrl** + **⇧ Shift** + **Enter**. На экране появится транспонированная матрица.

*Существует ещё один способ нахождения транспонированного массива:*

1. Выделите ячейки, которые необходимо транспонировать.
2. Нажмите кнопку **Копировать**.

3. Укажите левую верхнюю ячейку области вставки, при этом область вставки не должна совпадать с областью копирования.
4. В меню **Правка** выберите команду **Специальная вставка**.
5. Установите флажок Транспонировать.

### Нахождение $k$ -го наибольшего элемента массива

**НАИБОЛЬШИЙ** (*массив;  $k$* ) - возвращает  $k$ -ое наибольшее значение из множества данных.

Эта функция используется, чтобы выбрать значение по его относительному местоположению. Например, функцию **НАИБОЛЬШИЙ** можно использовать, чтобы определить наилучший, второй или третий результат в баллах, показанный при тестировании.

*Массив* - это массив или интервал данных, для которых определяется  $k$ -ое наибольшее значение.

$k$  - это позиция (начиная с наибольшей) в массиве или интервале ячеек данных.

*Замечание.*

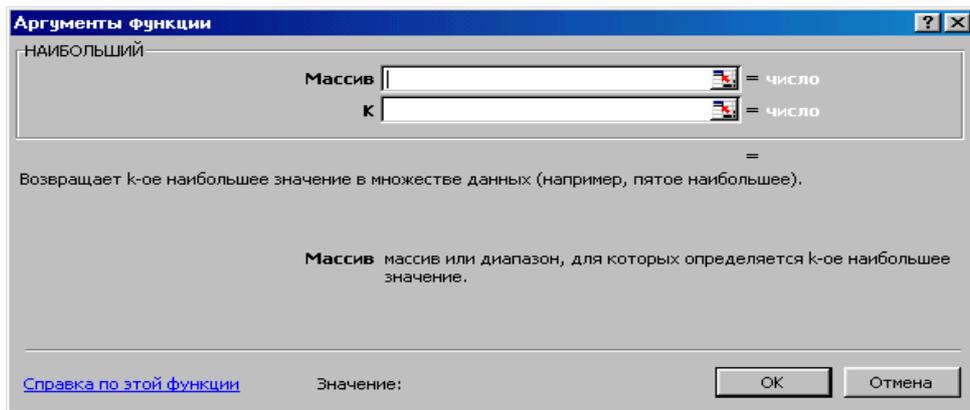
1. Если массив пуст, то функция **НАИБОЛЬШИЙ** возвращает значение ошибки #ЧИСЛО!.
2. Если  $k < 0$  или если  $k$  больше, чем число точек данных, то функция **НАИБОЛЬШИЙ** возвращает значение ошибки #ЧИСЛО!.

Если  $n$  - это число точек данных в интервале, то функция **НАИБОЛЬШИЙ**(*массив;1*) возвращает наибольшее значение, а **НАИБОЛЬШИЙ**(*массив;n*) возвращает наименьшее значение.

### Создание формулы вычисления $k$ -го наибольшего элемента массива

1. Занесите исходную матрицу в таблицу.
2. В свободном месте таблицы выделите ячейку, в которой будет выведен результат.

3. Вызовите функцию **НАИБОЛЬШИЙ()**. На экране появится диалоговое окно:



4. Нажмите красную стрелочку в строке *Массив*, при этом на экране появится лист с исходными данными.

5. Выделите матрицу, в которой надо найти наибольший элемент.

6. Нажмите красную стрелочку для возврата в диалоговом окне:



7. Введите в строке ниже число, которое указывает, каким по счету наибольшим должен быть элемент.

8. Нажмите клавишу  .

Кроме этого, можно в данной функции ввести массивы непосредственно:

*Пример:*

**НАИБОЛЬШИЙ**({3;4;5;2;3;4;5; 6;4;7};3) равняется 5

**НАИБОЛЬШИЙ**({3;4;5;2;3;4;5;6;4;7};7) равняется 4

### Нахождение *k*-го наименьшего элемента массива

**НАИМЕНЬШИЙ** (*массив; k*) - Возвращает *k*-ое наименьшее значение в множестве данных.

Эта функция используется для определения значения, занимающего определенное относительное положение в множестве данных.

Работа с этой функцией аналогична функции **НАИБОЛЬШИЙ**.

*Примеры:*

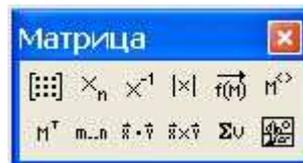
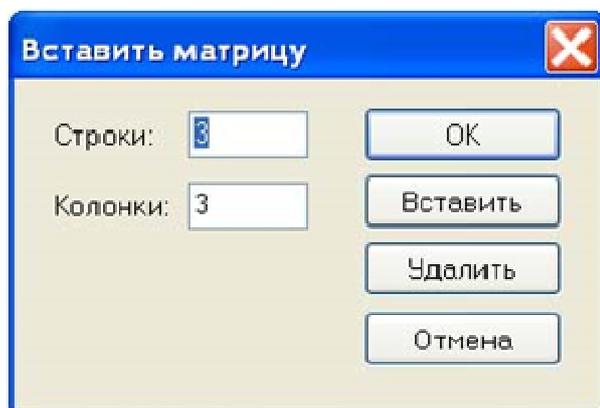
НАИМЕНЬШИЙ( $\{3;4;5;2;3;4;5;6;4;7\};4$ ) равняется 4

НАИМЕНЬШИЙ( $\{1; 4; 8; 3;7; 12; 54; 8; 23\}; 2$ ) равняется 3

## Простейшие операции с матрицами в Mathcad

Простейшие операции матричной алгебры реализованы в Mathcad в виде операторов. Написание операторов по смыслу максимально приближено к их математическому действию. Каждый оператор выражается соответствующим символом. Векторы и матрицы рассматриваются в программе Mathcad как одномерные и двумерные массивы данных.

Матрица создается с помощью диалогового окна «Вставить матрицу», которое открывают командой **Вставка** → **Матрица** (Insert → Matrix) или первой кнопкой на панели «Матрица». Вектор задается как матрица, имеющая один столбец. В диалоговом окне задается число строк и столбцов, а после вставки на рабочий лист матрица содержит вместо элементов заполнители, вместо которых следует вставить число, переменную или выражение.



Для матриц определены сложение, умножение на число, перемножение и другие операции.

Для работы с элементами матрицы используют индексы элементов. Нумерация строк и столбцов матрицы начинается с нуля, что задается системной переменной **ORIGIN**, и может быть изменено пользователем (например, блок **ORIGIN:=1** устанавливает начальный индекс равным единице). Индекс элемента задается числом, переменной или выражением и отображается

как нижний индекс. Он вводится после щелчка на кнопке **Индекс** (*Subscript,  $x_n$* ) на панели инструментов **Матрица** (*Matrix*) или нажатием символа «[» после имени массива. Индексы двумерных матриц записываются через запятую.

Иногда (например, при построении графиков) требуется выделить вектор, представляющий собой столбец матрицы. Номер столбца матрицы отображается как верхний индекс, заключенный в угловые скобки:  $s^{(3)}$ . Для его ввода используется кнопка **Столбец** (*Matrix Column*) на панели инструментов **Матрица** или соответствующая "горячая" клавиша.

Для задания некоторой последовательности элементов матрицы используют диапазоны. Диапазон фактически представляет собой вектор, содержащий арифметическую прогрессию, определенную первым, вторым и последним элементами. Чтобы задать диапазон, следует указать значение первого элемента, через запятую значение второго и через точку с запятой значение последнего элемента. Точка с запятой при задании диапазона отображается как две точки (..). Диапазон можно использовать как значение переменной, например  $x:=0,0.01..1$ . Если разность прогрессии равна единице (то есть, элементы просто нумеруются), значение второго элемента и соответствующую запятую опускают.

Ввод символа транспонирования (transpose) осуществляется с помощью панели инструментов **Матрица** (*Matrix*) или нажатием клавиш <Ctrl>+<1>.

В Mathcad можно как складывать матрицы, так и вычитать их друг из друга. Для этих операторов применяются символы <+> или <-> соответственно. Матрицы должны иметь одинаковую размерность.

Определитель матрицы обозначается стандартным математическим символом. Чтобы ввести оператор нахождения определителя матрицы, можно нажать кнопку **Determinant** на панели инструментов **Matrix** или набрать на клавиатуре <|> (нажав клавиши <Shift>+<|>). В результате любого из этих действий появляется местозаполнитель, в который следует поместить матрицу.

Ниже приведены примеры реализации некоторых операций с матрицами.

Mathcad - [matrix.mcd]

Файл Правка Вид Вставить Формат Инструменты Символика Окно Помощь

Normal Arial 10 B I U

### Некоторые операции с матрицами

ORIGIN = 0      ORIGIN := 1

Исходные матрицы

$$A := \begin{pmatrix} 5 & -1 \\ -7 & 3 \\ 4 & 0 \end{pmatrix} \quad B := \begin{pmatrix} 0 & 2 & 3 \\ 1 & 0 & 7 \end{pmatrix} \quad S := \begin{pmatrix} 2 & 1 & 4 \\ 1 & 2.5 & 6 \\ 3 & 6 & 1 \end{pmatrix} \quad T := \begin{pmatrix} 16 \\ 24 \\ 18 \end{pmatrix}$$

Умножение на скаляр:

$$C := B \cdot 0.5 \quad C = \begin{pmatrix} 0 & 1 & 1.5 \\ 0.5 & 0 & 3.5 \end{pmatrix}$$

Перемножение матриц      Транспонирование

$$D := A \cdot B \quad D = \begin{pmatrix} -1 & 10 & 8 \\ 3 & -14 & 0 \\ 0 & 8 & 12 \end{pmatrix} \quad D^T = \begin{pmatrix} -1 & 3 & 0 \\ 10 & -14 & 8 \\ 8 & 0 & 12 \end{pmatrix}$$

+ Вычисление определителя      Вычисление обратной матрицы

$$|S| = -56 \quad F := S^{-1} \quad F = \begin{pmatrix} 0.598 & -0.411 & 0.071 \\ -0.304 & 0.179 & 0.143 \\ 0.027 & 0.161 & -0.071 \end{pmatrix}$$

Замена одного столбца на вектор

$$Y := S \quad Y^{(2)} := T \quad Y = \begin{pmatrix} 2 & 16 & 4 \\ 1 & 24 & 6 \\ 3 & 18 & 1 \end{pmatrix} \quad x := \frac{|Y|}{|S|} \quad x = 2$$

Векторное произведение

$$V1 := \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} \quad V2 := \begin{pmatrix} 0.2 \\ 5 \\ 12.6 \end{pmatrix} \quad V1 \cdot V2 = 78.2$$

Выравнивание влево      AUTO      NUM      Лист 1

## Практическое занятие

1. Откройте новую книгу Microsoft Excel.
2. Сохраните её под именем *Работа с массивами*.
3. Переименуйте лист 1, задан имя *Функции для работы с массивами*.

4. Выполните упражнения в Excel (упражнения 1-6) и Mathcad (упражнения 1-4):

**Упражнение 1.**

Вычислите определитель квадратной матрицы (по варианту).

**Упражнение 2.**

На этом же листе вычислите обратную матрицу данной квадратной матрицы.

**Упражнение 3.**

Умножьте исходную матрицу на обратную ей. *Какой получился результат?*

*Почему?*

**Упражнение 4.**

Найдите транспонированную матрицу для исходной.

**Упражнение 5.**

Найдите первый, второй, третий, четвертый, пятый наименьшие элементы.

*Поясните полученный результат.*

**Упражнение 6.**

Найдите первый, второй, третий, четвертый, пятый наибольшие элементы.

*Поясните полученный результат.*

Вычисления оформите следующим образом:

Исходная матрица А					Обратная матрица А <sup>-1</sup>			
10	2	3	4		0,11	-0,22	0,11	0
2	3	4	5		-0,22	-4,22	3,11	0,33
3	4	5	6		0,11	3,11	-1,56	-0,67
4	5	6	10		0	0,33	-0,67	0,33
<b>Определитель матрицы:</b>				<b>-27</b>				
Матрица А*А <sup>-1</sup>					Транспонированная матрица А <sup>Т</sup>			
1	0	0	0		10	2	3	4
0	1	0	0		2	3	4	5
0	0	1	0		3	4	5	6
0	0	0	1		4	5	6	10
Наибольший первый		10			Наименьший первый		2	
Наибольший второй		6			Наименьший второй		2	

Наибольший третий	6		Наименьший третий	3
Наибольший четвёртый	4		Наименьший четвёртый	3
Наибольший пятый	5		Наименьший пятый	3

### Варианты задач для самостоятельной работы:

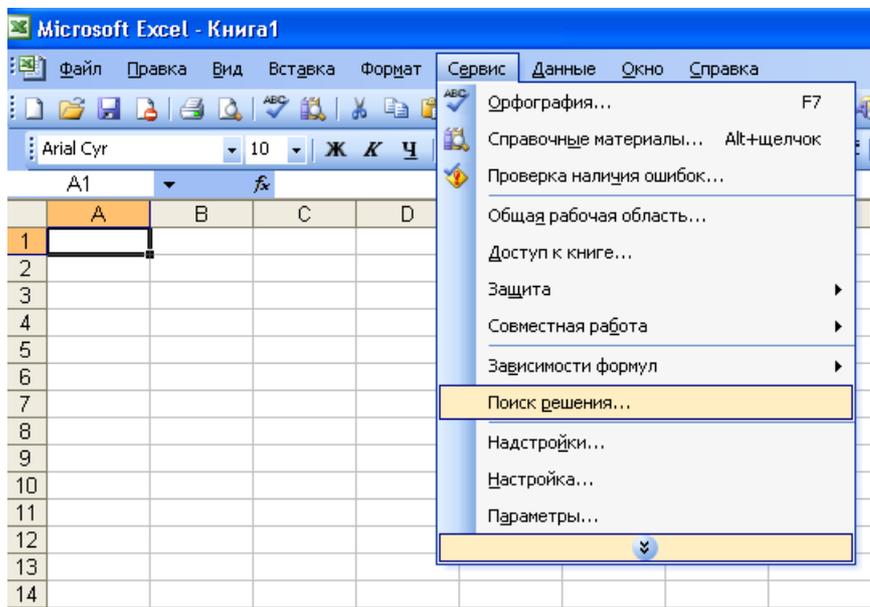
1	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & 4 & 8 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 6 & -7 & 8 \end{pmatrix}$	9	$\begin{pmatrix} 3 & 6 & 1 & 2 \\ 3 & -4 & 7 & 3 \\ 0 & 4 & -2 & 1 \\ 5 & 2 & -3 & 8 \end{pmatrix}$	17	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 6 & -1 & 7 & 1 \\ 2 & 4 & -2 & 1 \\ 5 & 2 & -7 & 5 \end{pmatrix}$	24	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 4 & -4 & 7 & 3 \\ 2 & 4 & -2 & 0 \\ 5 & -1 & -7 & 8 \end{pmatrix}$
2	$\begin{pmatrix} 8 & 6 & 1 & 2 \\ 13 & 4 & 8 & 5 \\ 2 & -4 & -2 & 1 \\ 15 & 6 & -7 & 7 \end{pmatrix}$	10	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & -3 \\ 2 & 4 & 6 & 1 \\ 5 & 7 & -7 & 8 \end{pmatrix}$	18	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & 3 \\ 2 & 2 & -2 & 7 \\ 5 & 1 & -7 & 8 \end{pmatrix}$	25	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 8 & -4 & 4 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 2 & -1 & 8 \end{pmatrix}$
3	$\begin{pmatrix} 2 & 6 & -6 & 2 \\ 7 & 8 & -2 & 7 \\ 2 & 4 & -2 & 1 \\ 6 & 6 & -3 & 9 \end{pmatrix}$	11	$\begin{pmatrix} 5 & 8 & 3 & -1 \\ 3 & -4 & 14 & 3 \\ 2 & 4 & -2 & 7 \\ 3 & -2 & -7 & 8 \end{pmatrix}$	19	$\begin{pmatrix} 5 & 6 & 3 & -2 \\ -3 & 4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 3 & -7 & -8 \end{pmatrix}$	26	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & 0 & 7 & 3 \\ 2 & 4 & 4 & 1 \\ -7 & 2 & -7 & 8 \end{pmatrix}$
4	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & 4 & 8 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 6 & -7 & 8 \end{pmatrix}$	12	$\begin{pmatrix} 5 & 6 & 3 & 1 \\ 3 & -4 & 7 & -3 \\ 2 & 4 & -4 & 1 \\ 5 & -7 & -7 & 8 \end{pmatrix}$	20	$\begin{pmatrix} 5 & 2 & 3 & 2 \\ 4 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 3 & -7 & 6 \end{pmatrix}$	27	$\begin{pmatrix} 5 & 6 & 3 & 9 \\ 3 & -4 & 7 & 3 \\ -5 & 4 & 2 & 1 \\ 5 & 2 & 0 & 8 \end{pmatrix}$
5	$\begin{pmatrix} 5 & -6 & 3 & 2 \\ 3 & 4 & 5 & 3 \\ -2 & 4 & -2 & 1 \\ 5 & 6 & -7 & 8 \end{pmatrix}$	13	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	21	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	28	$\begin{pmatrix} -2 & 6 & 3 & 2 \\ 3 & 4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 4 & -7 & 0 \end{pmatrix}$
6	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	14	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 5 & -3 \\ 2 & -2 & -2 & 1 \\ 5 & 9 & -7 & 4 \end{pmatrix}$	22	$\begin{pmatrix} 3 & 6 & 0 & 2 \\ 3 & -4 & 7 & 3 \\ 7 & 4 & -2 & 1 \\ 5 & 2 & -7 & 4 \end{pmatrix}$	29	$\begin{pmatrix} 5 & 6 & -3 & 2 \\ 3 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & -1 & 7 & 8 \end{pmatrix}$
7	$\begin{pmatrix} 5 & 6 & 3 & 2 \\ 3 & -4 & 7 & 3 \\ -2 & 4 & 1 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	15	$\begin{pmatrix} 5 & 6 & 13 & 2 \\ 3 & 2 & 7 & 3 \\ -6 & 4 & -2 & 11 \\ 5 & 1 & -7 & 8 \end{pmatrix}$	23	$\begin{pmatrix} 1 & 6 & 3 & -1 \\ 3 & -4 & 7 & 3 \\ 2 & 0 & -2 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	30	$\begin{pmatrix} 5 & 6 & -6 & 2 \\ 9 & -4 & 7 & 3 \\ 2 & 4 & -2 & 1 \\ 5 & 0 & -7 & 8 \end{pmatrix}$
8	$\begin{pmatrix} 5 & 1 & 3 & 2 \\ 1 & -4 & 7 & 3 \\ -2 & 4 & 1 & 1 \\ 5 & 2 & -7 & 8 \end{pmatrix}$	16	$\begin{pmatrix} -1 & 6 & 13 & 2 \\ 3 & 2 & 7 & 3 \\ -6 & 4 & -2 & 11 \\ 5 & 1 & -7 & 5 \end{pmatrix}$				



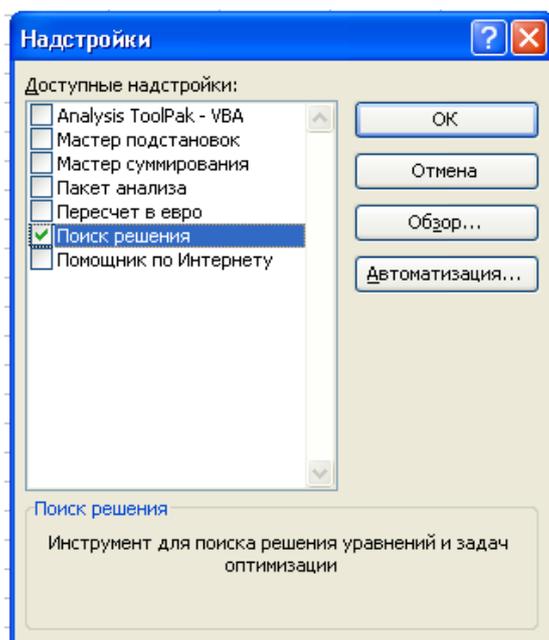
- e) Нахождение наибольшего элемента в массиве.
- f) Нахождение наименьшего элемента в массиве.
- 6. В чем заключается метод **Обратной матрицы** решения СЛАУ?

# РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ MS EXCEL И MATHCAD

Для решения задач оптимизации в MS Excel используют надстройку **Поиск решения**, которая вызывается из пункта главного меню «Сервис».



Если в версии Excel, установленной на Вашем компьютере, отсутствует данный подпункт меню «Сервис», необходимо вызвать пункт меню «Настройки» и предложенном списке дополнительных модулей выбрать «Поиск решения»



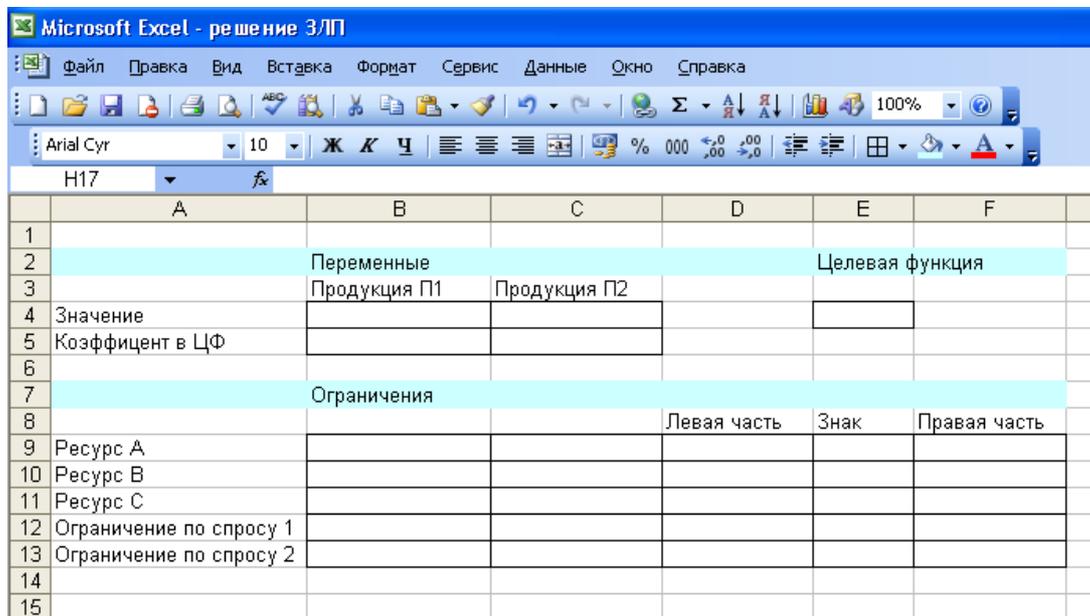
Рассмотрим на примере использование данной надстройки. Решим с её помощью задачу, математическая модель которой имеет вид:

целевая функция:  $F(x) = 3 \cdot x_1 + 2 \cdot x_2 \rightarrow \max$

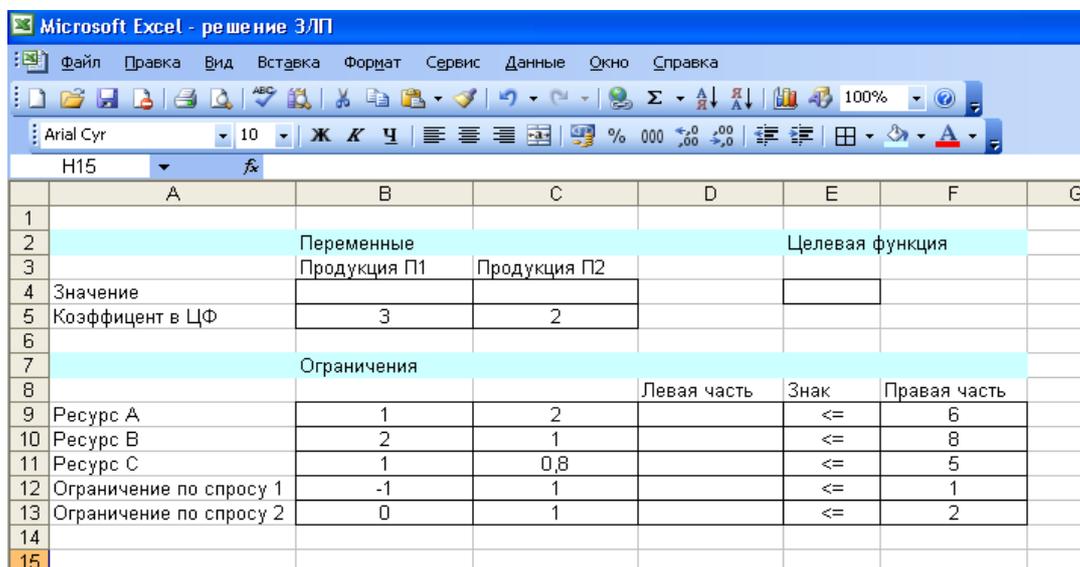
при ограничениях:

$$\begin{cases} x_1 + 2x_2 \leq 6 \\ 2x_1 + x_2 \leq 8 \\ x_1 + 0.8x_2 \leq 5 \\ -x_1 + x_2 \leq 1 \\ x_2 \leq 2 \\ x_1 \geq 0, \quad x_2 \geq 0 \end{cases}$$

Составим шаблон в редакторе Excel, как показано далее.



Теперь занесем данную числовую информацию в задачу.

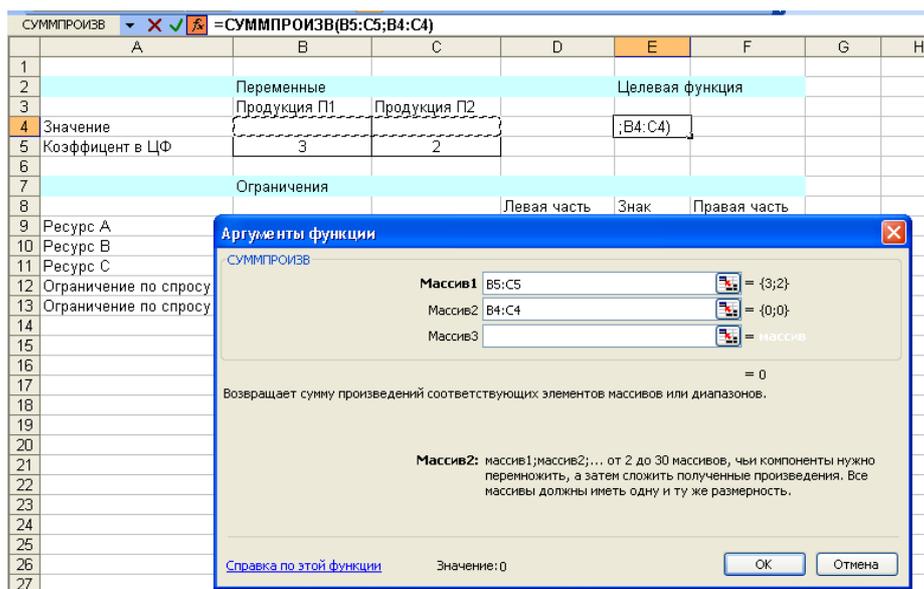


В выделенные пустые ячейки (значения целевой функции и левых частей неравенств) необходимо занести формулы, отображающие связи и отношения между числами на рабочем листе.

Ячейки В4-С4 называются в Excel *изменяемыми* (в нашей модели это неизвестные переменные), т.е., изменяя их **Поиск решения** будет находить оптимальное значение целевой функции. Значения, которые первоначально вводят в эти ячейки, обычно нули (незаполненные клетки трактуются по умолчанию как содержащие ненулевые значения).

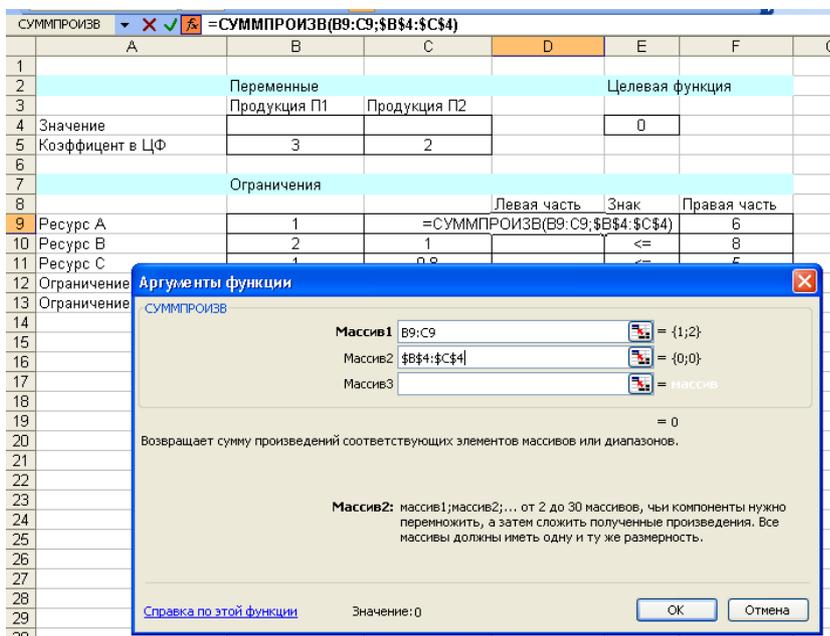
Теперь необходимо ввести формулы. В нашей математической модели, целевая функция представляет собой произведение вектора коэффициентов на вектор неизвестных. Действительно выражение  $3 \cdot x_1 + 2 \cdot x_2$  можно рассматривать как произведение вектора (3, 2) на вектор  $(x_1, x_2)$ .

В Excel существует функция СУММПРОИЗВ, которая позволяет найти скалярное произведение векторов. В ячейку Е4 необходимо вызвать данную функцию, а в качестве перемножаемых векторов задать адреса ячеек, содержащих коэффициенты уравнений (в данном случае, это В5:С5) и ячеек, которые в результате решения будут помещены значения  $x_1$  и  $x_2$  (ячейки В4:С4).



Каждая левая часть ограничения тоже представляет собой произведение двух векторов: соответствующей строки матрицы затрат и вектора неизвестных. То есть, выражение  $x_1 + 2x_2$  (для первого ограничения  $x_1 + 2x_2 \leq 6$ ) будем рассматривать как произведение вектора коэффициентов (1,2) и вектора пока переменных  $(x_1, x_2)$ .

В ячейке, отведенной для формулы левой части первого ограничения (D4), вызовем функцию СУММПРОИЗВ. В качестве адресов перемножаемых векторов занесем адрес строки коэффициентов B9:C9 и адрес значений переменных B4:C4.



В четыре оставшиеся ячейки графы «Левая часть» вводим аналогичные формулы, используя соответствующую строку матрицы затрат. Фрагмент экрана с введенными формулами показан ниже:

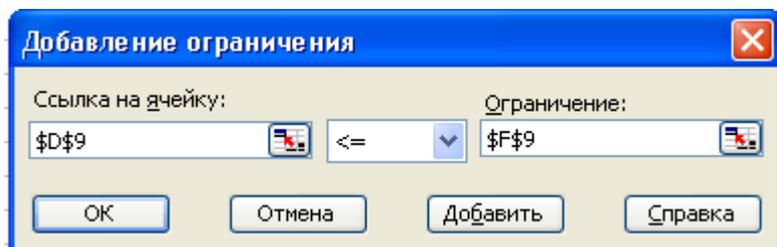
	A	B	C	D	E	F
1						
2		Переменные			Целевая функция	
3		Продукция П1	Продукция П2			
4	Значение				0	
5	Кoeffициент в ЦФ	3	2			
6						
7		Ограничения				
8				Левая часть	Знак	Правая часть
9	Ресурс А	1	2	0	<=	6
10	Ресурс В	2	1	0	<=	8
11	Ресурс С	1	0,8	0	<=	5
12	Ограничение по спросу 1	-1	1	0	<=	1
13	Ограничение по спросу 2	0	1	0	<=	2
14						
15						
16						

Важно! К моменту вызова сервиса «Поиска решения» на рабочем листе с задачей должны быть занесены **формулы для левых частей ограничений и формула для значения целевой функции.**

В меню **Сервис** выбираем **Поиск решения**. В появившемся окне задаём следующую информацию:

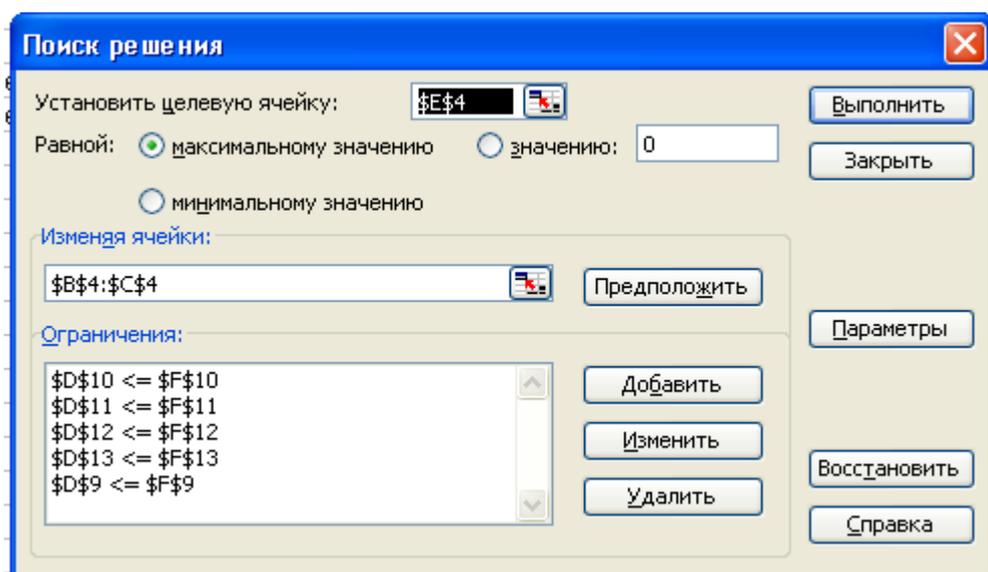
- ✓ в качестве целевой ячейки устанавливаем адрес ячейки для значения целевой функции E4;
- ✓ «флажок» устанавливаем на вариант «максимальному значению», т.к. в данном случае целевая функция подлежит максимизации;
- ✓ в качестве изменяемых ячеек заносится адрес строки значений переменных B4:C4;
- ✓ справа от окна, предназначенного для занесения ограничений, нажимаем кнопку «Добавить», появится форма для занесения ограничения;

✓ в левой части формы «Ссылка на ячейку» заносится адрес формулы для левой части первого ограничения D9, выбирается требуемый знак неравенства (в нашем случае, <=), в поле «Ограничение» заносится ссылка на правую часть ограничения F9;



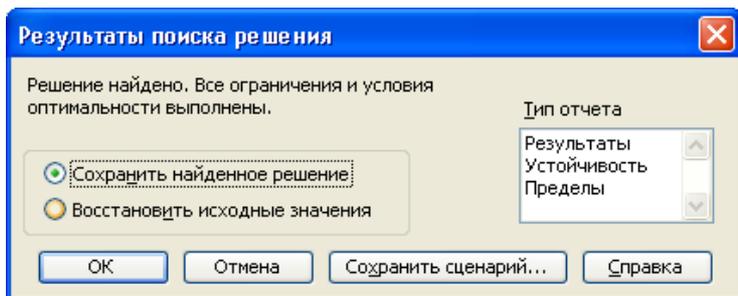
✓ аналогично заносятся все ограничения задачи, после чего нажимается кнопка «ОК».

Таким образом, окно «Поиск решения» с занесенной информацией выглядит следующим образом:



Далее необходимо нажать кнопку **Параметры**, установить «флажки» «**Линейная модель**» и «**Неотрицательные значения**», поскольку в данном случае задача является ЗЛП, а ограничения требуют неотрицательности переменных.

Затем следует нажать «ОК», «Выполнить», после чего появиться окно результатов решения.



Если в результате всех действий получено окно с сообщением «Решение найдено», то Вам предоставляется возможность получения трех типов отчета, которые полезны при анализе модели на чувствительность. В данном примере достаточно сохранить найденное решение, нажав «ОК». В результате получено решение задачи.

	A	B	C	D	E	F	G
1							
2		Переменные				Целевая функция	
3		Продукция П1	Продукция П2				
4	Значение	3,333333333	1,333333333			12,66667	
5	Кoeffициент в ЦФ	3	2				
6							
7		Ограничения					
8				Левая часть	Знак	Правая часть	
9	Ресурс А	1	2	6	<=	6	
10	Ресурс В	2	1	8	<=	8	
11	Ресурс С	1	0,8	4,4	<=	5	
12	Ограничение по спросу 1	-1	1	-2	<=	1	
13	Ограничение по спросу 2	0	1	1,333333333	<=	2	
14							
15							
16							

Если в результате решения задачи выдано окно с сообщением о невозможности нахождения решения, это означает, что при оформлении задачи была допущена ошибка (не заполнены формулы для ограничений, неправильно установлен «флажок» максимизации/минимизации и т. д.).

## Решение задач линейного программирования в MathCAD

1. Пример решения задачи линейного программирования в Mathcad.

Найти минимальное значение функции:

$$Z(x) = x_1 - x_2 \rightarrow \min$$

$$\text{при ограничениях: } \begin{cases} x_1 + x_2 \leq 10 \\ x_1 + x_2 \geq 2 \\ 3x_1 + x_2 \leq 5 \end{cases} \quad x_1 \geq 0, \quad x_2 \geq 0$$

Решение.

$$\text{ORIGIN} := 1$$

$$Z(x_1, x_2) := x_1 - x_2$$

$$x_2 := 0 \quad x_1 := 0$$

Given

$$x_1 + x_2 \leq 10$$

$$x_1 + x_2 \geq 2$$

$$3x_1 + x_2 \leq 5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$\text{Opt} := \text{Minimize} (Z, x_1, x_2)$$

$$\text{Opt} = \begin{pmatrix} 0 \\ 5 \end{pmatrix}$$

$$x_1 := \text{Opt}_1$$

$$x_2 := \text{Opt}_2$$

$$Z(x_1, x_2) = -5$$

Специальной переменной ORIGIN присваивают значение 1. Значением ORIGIN является номер первого элемента строки или столбца в матрице. По умолчанию ORIGIN=0.

2. Пример решения задачи линейного программирования в Mathcad.

Найти максимальное значение функции:

$$Z(x) = x_1 - x_2 \rightarrow \max$$

$$\text{при ограничениях: } \begin{cases} x_1 + x_2 \leq 10 \\ x_1 + x_2 \geq 2 \\ 3x_1 + x_2 \leq 5 \end{cases} \quad x_1 \geq 0, \quad x_2 \geq 0$$

Решение.

$$\text{ORIGIN} := 1$$

$$Z(x_1, x_2) := x_1 - x_2$$

$$x_2 := 0 \quad x_1 := 0$$

Given

$$x_1 + x_2 \leq 10$$

$$x_1 + x_2 \geq 2$$

$$3x_1 + x_2 \leq 5$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$\text{Opt} := \text{Maximize} (Z, x_1, x_2)$$

$$\text{Opt} = \begin{pmatrix} 1.5 \\ 0.5 \end{pmatrix}$$

$$x_1 := \text{Opt}_1$$

$$x_2 := \text{Opt}_2$$

$$Z(x_1, x_2) = 1$$

### 3. Пример производственно-экономической задачи в Mathcad

$$\text{ORIGIN} := 1$$

$$Q := 10000 \quad \text{ðèíàíñù íà çàèóíéó ðáñóðñíâ}$$

$$A := \begin{pmatrix} 0.4 & 0 & 0.4 & 0.3 \\ 0.3 & 0.3 & 0 & 0.1 \\ 0 & 0.5 & 0.3 & 0.2 \end{pmatrix} \quad \begin{array}{l} \text{íðìà çàòðàò ðáñóðñíâ} \\ (\text{ñòðíèè - àèàù ïðíáóéèè, ñòíèáóù- àèàù ðáñóðñíâ}) \end{array}$$

$$B0 := ( 300 \ 150 \ 200 \ 200 ) \quad \text{óðíááíù çàíàñà ðáñóðñíâ}$$

$$Y := ( 17 \ 20 \ 15 \ 18 ) \quad \text{öáíà íà ðáñóðñù}$$

$$C := \begin{pmatrix} 40 \\ 35 \\ 45 \end{pmatrix} \quad \text{öáíà ïðíáóéèè}$$

$$d := \begin{pmatrix} 10 \\ 500 \\ 400 \end{pmatrix} \quad \text{ìèíèìàèüíúé íáúàì çàèàçíâ}$$

$$D := \begin{pmatrix} 100 \\ 1000 \\ 800 \end{pmatrix} \quad \text{ïðááàèüíàÿ àíèíñòù ðúíèà}$$

$$Z(X, B) := \sum (C^T \cdot X - Y \cdot B^T) \quad \text{öàèáááÿ óóíéèÿ}$$

$$X := \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad B := ( 0 \ 0 \ 0 \ 0 ) \quad \text{ííèñàíéà íàèçáñòíúõ ïàðàìàòðíâ}$$

Given

$$A^T \cdot X \leq B0^T + B^T \quad \text{íãðàíè÷áíèà íà ðáñóðñù}$$

$$\sum (Y \cdot B^T) \leq Q \quad \text{íãðàíè÷áíèà íà ðèíàíñù}$$

$$X \geq d \quad X \leq D \quad \text{íãðàíè÷áíèà íà óñèíàèÿ ðúíèà}$$

$$B \geq 0 \quad X \geq 0 \quad \text{óñèíàèÿ íáòðèèòàèèíñòè ïàðàìáííúõ}$$

Opt := Maximize(Z, X, B)

οοίεοεϋ ιιòèìèçàöèè ìàðàìáòðîá

$$\text{Opt} = \begin{pmatrix} \{3,1\} \\ \{1,4\} \end{pmatrix}$$

X := Opt<sub>1</sub>

B := Opt<sub>2</sub>

Z(X, B) = 54231.508

ιιòèìàèüíàÿ ιðèáúèü

$$X = \begin{pmatrix} 12 \\ 985 \\ 651 \end{pmatrix}$$

ιιòèìàèüíúé ìèàì ιðìèçâîãñòâà

B = ( 0 471 0 32 )

ιιòèìàèüíúé ìèàì çàèóíèè ðãñóðñîá

## Варианты задач для самостоятельной работы

### Вариант 1

$$\begin{aligned} z &= 2x_1 + 3x_2 \rightarrow \max \\ \text{a)} \quad &\begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned}$$

$$\begin{aligned} z &= 2x_1 + 4x_2 \rightarrow \min \\ \text{б)} \quad &\begin{cases} x_1 + x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + 2x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned}$$

$$\begin{aligned} z &= 2x_1 + x_2 \rightarrow \max \\ \text{в)} \quad &\begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + 3x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned}$$

$$\begin{aligned} z &= 2x_1 + x_2 \rightarrow \min \\ \text{г)} \quad &\begin{cases} x_1 + 2x_2 \leq 4, \\ 2x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 8, \\ x_1, x_2 \geq 0. \end{cases} \end{aligned}$$

### Вариант 2

$$z = 2x_1 + 2x_2 \rightarrow \max$$

$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + x_2 \geq 8, \\ 3x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 7x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 5x_2 \leq 5, \\ 2x_1 + x_2 \leq 4, \\ 6x_1 + 7x_2 \geq 5, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 7x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 3x_2 \leq 9, \\ 2x_1 + x_2 \leq 6, \\ x_1 + 2x_2 \geq 16, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 3

$$z = 2x_1 + 5x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 6x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 9, \\ x_1 + 3x_2 \geq 6, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 2x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} 6x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + x_2 \geq 1, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 25x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 2x_1 + 3x_2 \leq 8, \\ x_1 + 2x_2 \leq 6, \\ x_1 + x_2 \geq 7, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 4

$$z = 2x_1 + 6x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ x_1 + 3x_2 \geq 7, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 8x_1 + 6x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 4x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 2x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + 3x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + x_2 \geq 7, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 5

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + x_2 \leq 5, \\ x_1 + 3x_2 \leq 7, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 3x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 7x_2 \leq 14, \\ 3x_1 + x_2 \leq 6, \\ 7x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ 3x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 4x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ x_1 + 2x_2 \geq 13, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 6

$$z = x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 9, \\ x_1 + 4x_2 \leq 8, \\ 2x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 14x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 7x_2 \leq 14, \\ 3x_1 + x_2 \leq 16, \\ x_1 + 2x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + x_2 \geq 5, \\ x_1 + 3x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 6x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 5x_1 + 2x_2 \leq 10, \\ 6x_1 + x_2 \leq 6, \\ 2x_1 + x_2 \geq 11, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 7

$$z = x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 3x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 10, \\ 2x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 4x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 4x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 6x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 8

$$z = 2x_1 + x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ 3x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 13, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 9

$$z = 2x_1 + x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ 10x_1 + 6x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 5, \\ 3x_1 + x_2 \geq 7, \\ 2x_1 + 3x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 5, \\ x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 9, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 10

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} 4x_1 + x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + 4x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 2x_1 + x_2 \leq 6, \\ 2x_1 + 5x_2 \geq 5, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 7x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 7, \\ 7x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 17, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 11

$$z = 7x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 2x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 4, \\ 2x_1 + x_2 \leq 6, \\ 2x_1 + x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 8x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + x_2 \geq 16, \\ 3x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 2x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 6x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + x_2 \geq 16, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 12

$$z = 2x_1 + 5x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 7, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ 3x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 7x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 3x_2 \leq 9, \\ 2x_1 + x_2 \leq 6, \\ 2x_1 + 7x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 - 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 7, \\ 2x_1 + x_2 \leq 6, \\ x_1 + 3x_2 \geq 18, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 13

$$z = 3x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 10, \\ 2x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ 2x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 5, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 6x_2 \geq 9, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = -x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 6, \\ x_1 + 3x_2 \leq 9, \\ 2x_1 + x_2 \geq 23, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 14

$$z = x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 4x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 4x_1 + 3x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + 3x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 5x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 5x_1 + 2x_2 \leq 15, \\ x_1 + 3x_2 \leq 9, \\ x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 15

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 4x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 5x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 4x_2 \geq 8, \\ x_1 + x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 13, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 16

$$z = 2x_1 + 5x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 9, \\ x_1 + 4x_2 \geq 8, \\ 2x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = -5x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 2x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 16, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 17

$$z = 2x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + 3x_2 \geq 6, \\ x_1 + 3x_2 \geq 5, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 9x_1 + 2x_2 \leq 9, \\ x_1 + 2x_2 \leq 7, \\ 3x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + 4x_2 \geq 18, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 18

$$z = 4x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 4, \\ x_1 + 6x_2 \geq 2, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 7x_2 \leq 14, \\ 3x_1 + x_2 \leq 6, \\ x_1 + 3x_2 \geq 21, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 19

$$z = 2x_1 + x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 4x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 7x_2 \leq 27, \\ 2x_1 + x_2 \leq 3, \\ 7x_1 + 3x_2 \geq 7, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ 3x_1 + x_2 \geq 6, \\ 2x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 4x_1 + 2x_2 \leq 6, \\ x_1 + 3x_2 \leq 8, \\ x_1 + 4x_2 \geq 20, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 20

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 7x_2 \leq 28, \\ 2x_1 + x_2 \leq 10, \\ 4x_1 + 3x_2 \geq 12, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 7, \\ 3x_1 + x_2 \geq 9, \\ 2x_1 + 3x_2 \geq 5, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 4x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 3x_2 \leq 7, \\ 3x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 14, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 21

$$z = 5x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 6x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 4x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 7x_2 \leq 14, \\ 2x_1 + x_2 \leq 12, \\ x_1 + 4x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 5x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 15, \\ x_1 + 6x_2 \geq 7, \\ 2x_1 + 3x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 5x_2 \leq 8, \\ x_1 + x_2 \leq 10, \\ x_1 + x_2 \geq 22, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 22

$$z = 3x_1 + 2x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 4x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} 9x_1 + 2x_2 \geq 9, \\ x_1 + 2x_2 \geq 7, \\ x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 6x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 5x_2 \leq 10, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = -3x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 4x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ x_1 + 2x_2 \geq 23, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 23

$$z = 2x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 5x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + 2x_2 \geq 8, \\ x_1 + 2x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 3x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 5x_2 \leq 15, \\ x_1 + x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 7x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} 7x_1 + 2x_2 \leq 7, \\ x_1 + x_2 \leq 9, \\ 4x_1 + x_2 \geq 38, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 24

$$z = 2x_1 + 2x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 2x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 3x_1 + 2x_2 \leq 12, \\ x_1 + 3x_2 \leq 10, \\ 2x_1 + x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + x_2 \geq 8, \\ x_1 + 2x_2 \geq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 5x_2 \leq 10, \\ 3x_1 + x_2 \geq 33, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 25

$$z = x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 4x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 3x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} 8x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ 6x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 6x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} x_1 + x_2 \geq 8, \\ x_1 + 2x_2 \geq 7, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 5x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 4x_1 + 2x_2 \leq 8, \\ x_1 + 4x_2 \leq 8, \\ 2x_1 + x_2 \geq 18, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 26

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 3x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = x_1 + 6x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} x_1 + 4x_2 \leq 8, \\ 2x_1 + x_2 \leq 6, \\ x_1 + 6x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 8x_1 + 3x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} 3x_1 + x_2 \geq 4, \\ 4x_1 + x_2 \geq 14, \\ 2x_1 + x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 8x_1 + 2x_2 \leq 8, \\ 3x_1 + x_2 \leq 6, \\ x_1 + 3x_2 \geq 21, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 27

$$z = 2x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ 4x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 7x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} x_1 + x_2 \geq 11, \\ 4x_1 + 3x_2 \geq 16, \\ 5x_1 + 3x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 6x_1 + 5x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 8x_1 + 2x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ 6x_1 + 5x_2 \geq 6, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 2x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} 3x_1 + 2x_2 \leq 6, \\ 2x_1 + x_2 \leq 10, \\ x_1 + x_2 \geq 13, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 28

$$z = 5x_1 + 3x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + 2x_2 \leq 8, \\ x_1 + 2x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 3x_2 \rightarrow \max$$
$$\text{в) } \begin{cases} 3x_1 + 2x_2 \geq 14, \\ 3x_1 + 4x_2 \geq 12, \\ 2x_1 + 3x_2 \geq 5, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} 9x_1 + 2x_2 \leq 18, \\ x_1 + 3x_2 \leq 7, \\ 3x_1 + x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 7x_1 + 3x_2 \rightarrow \min$$
$$\text{г) } \begin{cases} x_1 + 7x_2 \leq 7, \\ 2x_1 + x_2 \leq 3, \\ 3x_1 + x_2 \geq 33, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 29

$$z = 2x_1 + x_2 \rightarrow \max$$
$$\text{a) } \begin{cases} x_1 + x_2 \leq 8, \\ x_1 + 2x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 4x_1 + 3x_2 \rightarrow \min$$
$$\text{б) } \begin{cases} x_1 + 2x_2 \leq 6, \\ x_1 + 3x_2 \leq 7, \\ 4x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} 5x_1 + x_2 \geq 5, \\ 3x_1 + x_2 \geq 6, \\ 4x_1 + 3x_2 \geq 4, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + 5x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} x_1 + 5x_2 \leq 5, \\ 2x_1 + x_2 \leq 6, \\ x_1 + x_2 \geq 8, \\ x_1, x_2 \geq 0. \end{cases}$$

### Вариант 30

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\text{а) } \begin{cases} 4x_1 + x_2 \leq 8, \\ x_1 + x_2 \leq 6, \\ 2x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 8x_1 + 4x_2 \rightarrow \min$$

$$\text{б) } \begin{cases} 9x_1 + x_2 \leq 9, \\ x_1 + 2x_2 \leq 4, \\ 2x_1 + x_2 \geq 1, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\text{в) } \begin{cases} 3x_1 + x_2 \geq 7, \\ x_1 + 2x_2 \geq 5, \\ 2x_1 + x_2 \geq 2, \\ x_1, x_2 \geq 0. \end{cases}$$

$$z = 9x_1 + 3x_2 \rightarrow \min$$

$$\text{г) } \begin{cases} 9x_1 + 2x_2 \leq 9, \\ x_1 + 2x_2 \leq 7, \\ x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0. \end{cases}$$

## РЕШЕНИЕ ТРАНСПОРТНОЙ ЗАДАЧИ С ПОМОЩЬЮ MS EXCEL И MATHCAD

### *Постановка транспортной задачи*

Имеется  $m$  поставщиков (пунктов отправления)  $A_1, \dots, A_m$ , у которых имеется однородный груз в количествах  $a_1, \dots, a_m$  у.е. Этот груз необходимо доставить потребителям  $B_1, \dots, B_n$ , спрос которых в указанном грузе составляет  $b_1, \dots, b_n$ .

Обозначим через  $c_{ij}$  стоимость перевозки единицы груза от поставщика  $A_i$  к потребителю  $B_j$ .

Исходные данные внесены в распределительную таблицу:

<div style="display: flex; justify-content: center; align-items: center;"> <div style="writing-mode: vertical-rl; transform: rotate(180deg);">поставщик</div> <div style="margin: 0 10px;">\</div> <div style="writing-mode: vertical-rl;">потребитель</div> </div>		$B_1$	$\dots$	$B_n$
		$b_1$	$\dots$	$b_n$
$A_1$	$a_1$	$c_{11}$ $x_{11}$	$\dots$	$c_{1n}$ $x_{1n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$A_m$	$a_m$	$c_{m1}$ $x_{m1}$	$\dots$	$c_{mn}$ $x_{mn}$

Необходимо найти такой план перевозок, чтобы:

- 1) мощность всех поставщиков была реализована;
- 2) спрос всех потребителей был удовлетворен;
- 3) суммарные затраты на перевозку были минимальными.

*Экономико-математическая модель транспортной задачи.*

$x_{ij}$  - количество единиц груза, которое необходимо доставить от поставщика  $A_i$  к потребителю  $B_j$ .

Матрица, составленная из  $x_{ij}$ , называется *матрицей перевозок*.

Матрица, составленная из коэффициентов  $c_{ij}$ , называется *матрицей тарифов*.

По первому пункту транспортной задачи мощности всех поставщиков должны быть реализованы. По второму пункту спрос всех потребителей должен быть удовлетворен, т.е. должна выполняться следующая система уравнений:

$$\begin{cases} \sum_{j=1}^n x_{ij} = a_i, & i = \overline{1, m} \\ \sum_{i=1}^m x_{ij} = b_j, & j = \overline{1, n} \end{cases} \quad (1)$$

Перевозка не может быть отрицательной, т.е. должно выполняться:

$$x_{ij} \geq 0, \quad i = \overline{1, m}, \quad j = \overline{1, n} \quad (2)$$

Суммарные затраты на перевозку:

$$F = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (3)$$

Транспортная задача заключается в нахождении  $x_{ij}$ , удовлетворяющих системам (1), (2), при котором (3) принимает наименьшее значение. Это есть каноническая задача линейного программирования.

План перевозок  $x_{ij}$  называется *допустимым планом* (*допустимым распределением поставок*), если он удовлетворяет (1) и (2).

Допустимое распределение поставок называется *оптимальным*, если оно доставляет минимум (3).

*Теорема (о существовании допустимого плана).*

Для того чтобы транспортная задача имела допустимый план, необходимо и достаточно выполнение:

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (4)$$

(суммарный спрос потребителей должен совпадать с суммарной мощностью поставщиков).

*Доказательство:*

Пусть  $x_{ij}$  – допустимый план. Просуммировав в распределительной таблице элементы  $x_{ij}$  отдельно по  $j$  и по  $i$ , получается равенство (1).

Суммируются все элементы  $x_{ij}$ , как по строчкам, так и по столбцам. Различие лишь в перестановке элементов. От перестановки слагаемых сумма конечных слагаемых не меняется, откуда и следует доказательство.

Покажем, что если выполняется (4), то всегда имеется допустимый план.

$$\text{Обозначим } \sum_{i=1}^m a_i = \sum_{j=1}^n b_j = A$$

$$x_{ij} = \frac{a_i \cdot b_j}{A} \quad (5)$$

По смыслу задачи  $a_i \geq 0, b_j \geq 0$ . Следовательно,  $x_{ij} \geq 0$ , т.е.  $x_{ij}$  удовлетворяет (2).

Покажем, что  $x_{ij}$  удовлетворяет (1). Вначале просуммируем (5) по  $i = \overline{1, m}$ , затем по  $j = \overline{1, n}$ :

$$\sum_{i=1}^m x_{ij} = \sum_{i=1}^m \frac{a_i \cdot b_j}{A} = \frac{b_j}{A} \sum_{i=1}^m a_i = \frac{b_j \cdot A}{A} = b_j, \quad j = \overline{1, n}$$

$$\sum_{j=1}^n x_{ij} = \sum_{j=1}^n \frac{a_i \cdot b_j}{A} = \frac{a_i}{A} \sum_{j=1}^n b_j = \frac{a_i \cdot A}{A} = a_i, \quad i = \overline{1, m}$$

Таким образом, значения  $x_{ij}$ , определенные формулой (5) удовлетворяют (1). Теорема доказана.

### *Открытая и закрытая транспортные задачи*

Транспортная задача называется *закрытой*, если суммарная мощность поставщиков совпадает с суммарным спросом потребителей, т.е.  $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$ .

Если условие не выполняется, то задача называется *открытой*.

### *Сведение открытой задачи к закрытой*

Для решения открытой задачи необходимо ее свести к закрытой.

Пусть суммарная мощность поставщиков превышает суммарный спрос потребителей, т.е.  $\sum_{i=1}^m a_i > \sum_{j=1}^n b_j$ .

В этом случае необходимо ввести нового фиктивного потребителя  $B_{n+1}$  со спросом:  $b_{n+1} = \sum_{i=1}^m a_i - \sum_{j=1}^n b_j$ .

Очевидно, что новый фиктивный потребитель есть просто склад. В этом столбце коэффициенты затрат  $c_{ij}$  принимают равными нулю, т.к.  $c_{ij}$  есть стоимость перевозки от поставщика на склад. Но хранение стоит определенных денег. Поэтому если информация о затратах на хранение товара на складе имеется, то эти данные можно внести в последний столбец при простановке коэффициентов  $c_{ij}$ .

Если  $\sum_{i=1}^m a_i < \sum_{j=1}^n b_j$  (суммарная мощность поставщиков меньше суммарного спроса потребителей), то вводят фиктивного поставщика  $A_{m+1}$  с мощностью:  $a_{m+1} = \sum_{j=1}^n b_j - \sum_{i=1}^m a_i$ . Коэффициенты затрат также можно принять равными нулю.

*Теорема (о ранге матрицы транспортной задачи)*

Ранг матрицы  $A$  системы ограничений в транспортной задаче (1) равен  $m + n - 1$ .

*Доказательство:*

Матрица  $A$  коэффициентов в ограничениях системы (1) имеет следующий вид:

$$A = \begin{pmatrix} 1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & 0 & 0 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 & 1 & 0 & \dots & 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 & 0 & 1 & \dots & 0 & \dots & 0 & 1 & \dots & 0 \\ \cdot & \cdot \\ 0 & 0 & \dots & 1 & 0 & 0 & \dots & 1 & \dots & 0 & 0 & \dots & 1 \end{pmatrix}$$

(всего  $(m+n)$  строк и  $(m \cdot n)$  неизвестных столбцов)

Первые 4 строчки получаются из первого ограничения (коэффициенты при всех  $x$  – в первой строке).

Последние 4 строки получаются из второго ограничения.

В каждом столбце матрицы  $A$  только 2 элемента равны 1, остальные равны 0. Если сложить первые  $m$  строк, то получается строка со всеми элементами = 1.

То же самое получается, если сложить последние  $n$  строк.

$$P_1 + P_2 + \dots + P_m = P_{m+1} + P_{m+2} + \dots + P_{m+n}$$

В результате получается линейная зависимость всех строк. И если вычеркнуть, например, последнюю строку получается линейная независимая система. Теорема доказана.

### *Решение транспортной задачи*

Этапы решения транспортной задачи:

- 1) Нахождение первоначального допустимого распределения поставок
- 2) Целенаправленный перебор допустимых распределений поставок, последовательно приближаясь к оптимальному распределению поставок.

### *Нахождение первоначального распределения поставок*

Количество заполненных поставкой клеток должно быть  $m+n-1$ .

#### *1. Метод северо-западного угла*

В этом методе груз распределяется, начиная с верхней левой (северо-западной) клетки (1,1), двигаясь затем от нее по строке вправо или по столбцу

вниз. Пусть  $x_{11} = \min(a_1, b_1)$ . Если  $a_1 > b_1$ , то  $x_{11} = b_1$  и первый потребитель будет полностью удовлетворен.

В дальнейшем первый столбец в расчет не принимается и полагают, что  $x_{i1} = 0, i = \overline{2, m}$ . Клетка, в которую была поставлена поставка, называется *заполненной* и перечеркивается сплошной линией. Те клетки, которые выпали из дальнейшего рассмотрения, называются *свободными* и перечеркиваются пунктирными линиями.

Затем необходимо двигаться по первой строке таблицы. В клетку (1,2) записывается  $x_{12} = \min(a_1 - b_1, b_2)$ .

Если  $a_1 - b_1 < b_2$ , то запасы первого поставщика исчерпаны. Далее нужно перейти ко второму поставщику, при этом выпадает первая строка.

Если  $b_1 > a_1$ , то  $x_{11} = a_1$ . Запас первого поставщика исчерпан, т.е.  $x_{1j} = 0, j = \overline{2, n}$ . Следовательно, выпадает первая строка.

Распределение запасов второго поставщика. В клетку (2,1) записывается  $x_{21} = \min(a_2, b_1 - a_1)$ .

После заполнения клетки (1,2) или (2,1) нужно перейти ко второй строке или ко второму столбцу и т.д., пока не будут исчерпаны мощности (все запасы) поставщиков.

## 2. Метод наименьших затрат

Находится клетка с наименьшим коэффициентом затрат и в нее дается максимально возможная поставка. При этом выпадет какая-то строка или столбец. Затем из оставшихся клеток находится клетка с наименьшим коэффициентом затрат и процедура повторяется.

Если одновременно выпадает строка и столбец, то число заполненных клеток будет меньше, чем  $m+n-1$ . В этом случае необходимо вводить в любые свободные клетки фиктивные нулевые поставки, так чтобы число заполненных клеток равнялось  $m+n-1$ . При этом с нулевой поставкой в дальнейшем работают так же как с любой другой.

### 3. Метод Фогеля

Вначале определяется разность между двумя наименьшими коэффициентами затрат по строкам и столбцам.

Находится наибольшая разность. В строке (столбце) с наибольшей разностью заполняется клетка с наименьшим коэффициентом затрат. При этом выпадают какие-то клетки. С оставшимися клетками повторяют всё заново.

#### Теорема о потенциалах

Если  $x_{ij}^*$  - оптимальное распределение поставок, то ему соответствует система чисел:  $(u_i^*, v_j^*)$ ,  $i = \overline{1, m}$   $j = \overline{1, n}$  называемых *потенциалами строк и столбцов* (поставщиков и потребителей), удовлетворяющих условиям:

$$\left. \begin{array}{l} u_i^* + v_j^* = c_{ij} \text{ для } x_{ij}^* > 0 \\ u_i^* + v_j^* \leq c_{ij} \text{ для } x_{ij}^* = 0 \end{array} \right\} \quad (6)$$

*Доказательство:*

Пусть имеется транспортная задача:

$$F = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min$$

$$\left\{ \begin{array}{l} \sum_{j=1}^n x_{ij} = a_i, i = \overline{1, m} \\ \sum_{i=1}^m x_{ij} = b_j, j = \overline{1, n} \\ x_{ij} \geq 0, i = \overline{1, m}, j = \overline{1, n} \end{array} \right.$$

Эта задача является двойственной задачей к следующей задаче:

Ограничению  $i$  в двойственной задаче  $x_{i1} + \dots + x_{in} = a_i$  поставлена переменная  $u_i$ , а ограничению  $j$  -  $x_{1j} + \dots + x_{mj} = b_j$  - переменная  $v_j$ .

Тогда по соответствию переменных исходной и двойственной задач исходная задача запишется:

$$u_i + v_j \leq c_{ij}, i = \overline{1, m}, j = \overline{1, n}$$

$$Z = \sum_{i=1}^m a_i u_i + \sum_{j=1}^n b_j v_j \rightarrow \max$$

Для двойственной задачи оптимальным является решение  $x_{ij}^*$ , а для исходной задачи  $u_i^*, v_j^*$ .

По первой теореме двойственности  $F_{\min} = Z_{\max}$ , а по второй теореме двойственности следует (6). Теорема доказана.

### *Метод потенциалов*

С помощью метода потенциалов находят оптимальное распределение поставок. Согласно теореме о потенциалах в каждой занятой клетке выполняется  $u_i + v_j = c_{ij}$ . Число заполненных клеток равно  $m+n-1$ . Для определения потенциалов нужно решить систему из  $(m+n-1)$  уравнений с  $(m+n)$  неизвестными. Система неопределенная, поэтому для простоты нахождения частного решения один из потенциалов, например,  $u_1$  полагают равным нулю. Затем строится матрица оценок  $S$  с элементами  $S_{ij} = c_{ij} - (u_i + v_j)$ .

Если все потенциалы удовлетворяют условию  $u_i + v_j \leq c_{ij}$ , т.е. все элементы матрицы оценок неотрицательные, то найденное распределение поставок является оптимальным. Если же некоторые  $S_{ij} < 0$ , то распределение поставок не является оптимальным. Из отрицательных оценок  $S_{ij}$  выбирается наименьшее число, и строится цикл пересчета, в котором поставки перераспределяются. Цикл представляет собой замкнутую ломаную линию, состоящую из звеньев, пересекающихся под прямым углом. В каждой строке (столбце) находится только одно звено. Цикл начинается со свободной клетки, затем нужно двигаться по заполненным клеткам так, чтобы вернуться назад. В цикле не обязательно участвуют все заполненные клетки. В цикле всегда четное количество клеток. В свободной клетке цикла ставится «+», дальше

знаки чередуются. В клетках со знаком «-» находится наименьшая поставка, которая передается по циклу: из клеток с «-» - эта поставка вычитается, а в клетках с «+» она прибавляется. Если наименьшая поставка сразу достигалась в нескольких клетках с «-», то после вычитания свободными клетками сразу станут несколько клеток цикла и, следовательно, количество заполненных клеток будет  $< m + n - 1$ . В этом случае свободной в цикле оставляют одну клетку, а в остальные свободные клетки цикла, записывают нулевые поставки. После этого измененные значения цикла возвращаются в исходную таблицу и все начинается заново.

### *Алгоритм решения транспортной задачи*

1. Нахождение первоначального распределения поставок.
2. Нахождение потенциалов строк и столбцов.

Нахождение матрицы оценок. Если все элементы этой матрицы неотрицательные, то найденное распределение поставок оптимально. Если при этом все  $S_{ij}$  для свободных клеток  $> 0$ , то полученное оптимальное распределение поставок является единственным. Если хотя бы одна  $S_{ij}$  для свободной клетки в оптимальном решении  $= 0$ , то имеем бесконечное множество оптимальных распределений поставок (альтернативное решение). Если в матрице  $S$  имеются отрицательные оценки, то строится цикл пересчета и переход к шагу 2.

Применение информационных технологий Excel для решения транспортной задачи рассмотрим на примере, исходные данные которого представлены в таблице:

Поставщики	Потребители				Запас
	1	2	3	4	
1	4	3	2	1	600
2	2	1	7	9	800

3	3	6	8	4	1000
Спрос потребителя	900	600	800	600	2400 2900

$$\sum_{i=1}^m a_{ij} = 2400 < \sum_{j=1}^n b_j = 2900$$

Вводим фиктивного четвертого поставщика с объемом запаса  $a_4=2900-2400=500$  и стоимостями единицы груза  $c_{4j}=0, j=1,2,3,4$ . Тогда данные транспортной задачи можно представить в таблице:

Поставщики	Потребители				Запас
	1	2	3	4	
1	4	3	2	1	600
2	2	1	7	9	800
3	3	6	8	4	1000
4	0	0	0	0	500
Спрос потребителя	900	600	800	600	2900 2900

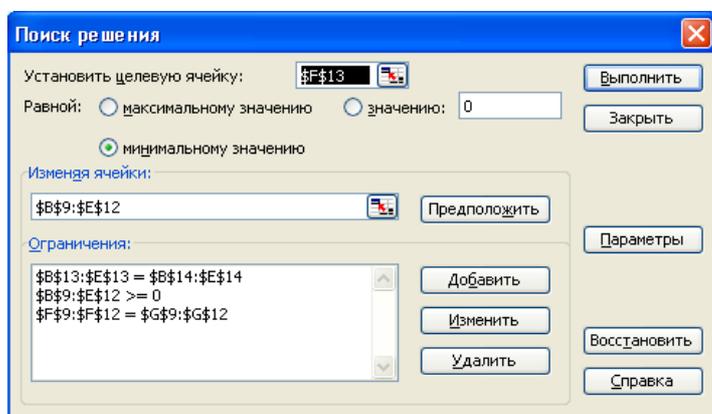
Чтобы решить задачу с помощью программного продукта Excel, командой **Поиск решения** введем исходные данные в ПЭВМ в виде, показанном на рисунке:

	A	B	C	D	E	F	G
1		ПОТРЕБИТЕЛИ					
2	Поставщики	1	2	3	4		
3	1	4	3	2	1		
4	2	2	1	7	9		
5	3	3	6	8	4		
6	4	0	0	0	0		
7		объемы перевозок потребителями					
8	Поставщики	1	2	3	4		Запас
9	1					=СУММ(B9:E9)	600
10	2					=СУММ(B10:E10)	800
11	3					=СУММ(B11:E11)	1000
12	4					=СУММ(B12:E12)	500
13		=СУММ(B9:B12)	=СУММ(C9:C12)	=СУММ(D9:D12)	=СУММ(E9:E12)	=СУММПРОИЗВ(B3:	
14	Спрос	900	600	800	600		2900
15							

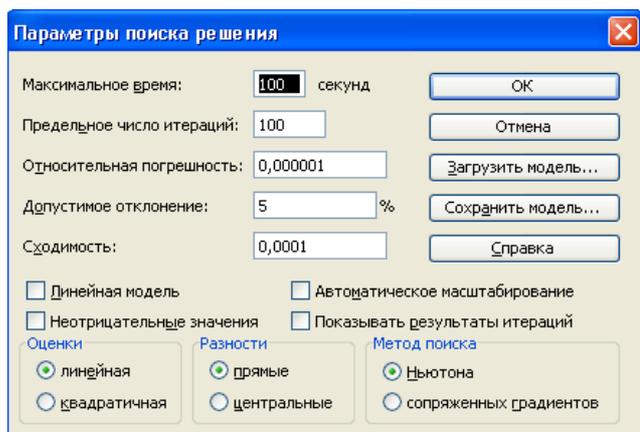
В массив В3:Е6 введены значения стоимости перевозок единицы груза. В ячейки В14:Е14 введены величины спроса потребителей, в G9:G12 – запасы поставщиков, а в ячейку G14 – суммарного запаса, равного суммарному спросу и составляющего 2900 единиц. Массив В9:Е12 отведен под значения неизвестных  $X_{ij}$  (объемы перевозок). Функция =СУММПРОИЗВ (В3:Е6; В9:Е12) введена в ячейку F13. Функция отражает сумму произведений

стоимости  $C_{ij}$  на объемы перевозок  $X_{ij}$  ( $i=1,2,3,4; j=1,2,3,4$ ). В массивы F9:F12 и B13:E13 введены левые части ограничений задачи  $\sum_{j=1}^4 x_{ij}$  ( $i=\overline{1,4}$ ) и  $\sum_{i=1}^4 x_{ij}$  ( $j=\overline{1,4}$ ) соответственно. Эти суммы и целевая функция введены с помощью **Мастера функций**.

После ввода данных вызывается диалоговое окно **Поиск решения** командной строкой **Поиск решения** из меню **Сервис**. В этом диалоговом окне заносится номер ячейки с целевой функцией (F13), номера изменяемых ячеек (B9:E12), устанавливается направление оптимизации, а также вводятся ограничения  $\$B\$13:\$E\$13=\$B\$14:\$E\$14$  и  $\$B\$9:\$E\$12 \geq 0$



В диалоговом окне **Параметры поиска решения** установим флажок **Линейная модель**, и, щелкнув по кнопке **ОК**, возвратимся в диалоговое окно **Поиск решения**.



Щелкнув по кнопке **Выполнить** в этом окне, получим на экране результат решения задачи

	A	B	C	D	E	F	G
1	Постав	ПОТРЕБИТЕЛИ					
2	щики	1	2	3	4		
3	1	4	3	2	1		
4	2	2	1	7	9		
5	3	3	6	8	4		
6	4	0	0	0	0		
7	Постав	объемы перевозок					
8	щики	1	2	3	4		Запас
9	1	0	0	300	300	600	600
10	2	200	600	0	0	800	800
11	3	700	0	0	300	1000	1000
12	4	0	0	500	0	500	500
13		900	600	800	600	5200	
14	Спрос	900	600	800	600		2900
15							

Таким образом,  $f_{min}=5200$ ,  $x_{13}=300$ ,  $x_{14}=300$ ,  $x_{21}=200$ ,  $x_{22}=600$ ,  $x_{31}=700$ ,  $x_{34}=300$ ,  $x_{43}=500$ .

Остальные объемы перевозок груза равны нулю.

При необходимости можно вывести на экран (на печать) отчеты: по результатам, по устойчивости решения и по разделам, которые содержат полезную информацию для целей экономико-математического анализа полученного решения.

## Применение информационных технологий Mathcad

Ниже предложен пример решения транспортной задачи открытого и закрытого типа в **Mathcad**.

## Решение задачи линейного программирования транспортного типа в MathCAD

ORIGIN:=1

m:=3    Определение количества баз-поставщиков

n:=5    Определение количества заводов-потребителей

A:=( 40 150 100)<sup>T</sup>    Матрица мощностей баз-поставщиков

B:=( 20 80 90 60 40)<sup>T</sup>    Матрица потребностей заводов-потребителей

C:= $\begin{pmatrix} 7 & 3 & 5 & 4 & 2 \\ 6 & 2 & 3 & 1 & 7 \\ 3 & 5 & 2 & 6 & 4 \end{pmatrix}$     Матрица транспортных затрат

i:=1..m    j:=1..n    X<sub>i,j</sub>:=0    Искомые переменные (объемы поставок)  
и их начальные значения

K<sub>1</sub>:=1    K<sub>2</sub>:=1    Вспомогательные вектора

F(X):= $\sum (C \cdot X) K_2$     Целевая функция - суммарные затраты транспортн

Given

X · K<sub>2</sub> = A    Условия загрузки мощности баз-поставщиков

X<sup>T</sup> · K<sub>1</sub> = B    Условия удовлетворения спроса заводов-потребителей

X ≥ 0    Условие неотрицательности объемов поставок

Result:=Minimize(F, X)    F(Result) = 550    Суммарные оптимальные затраты  
транспортной сети

Result= $\begin{pmatrix} 0 & 0 & 0 & 0 & 40 \\ 0 & 80 & 10 & 60 & 0 \\ 20 & 0 & 80 & 0 & 0 \end{pmatrix}$     Искомый план поставок

Ответ: оптимальное распределение поставок:

$$X^* = \begin{pmatrix} 0 & 0 & 0 & 0 & 40 \\ 20 & 80 & 10 & 60 & 0 \\ 0 & 0 & 80 & 0 & 0 \end{pmatrix}$$

Данное распределение поставок обеспечит оптимальные  
транспортные издержки в размере 550 усл. ден. ед.

## Упражнения

1. Решить задачу, исходные данные которой приведены в таблице

Поставщики	Потребители				Объемы производства
	1	2	3	4	
1	30	50	62	10	650
2	40	50	80	20	850
3	50	10	30	30	700
Спрос потребителя	500	800	300	600	2200

при следующих условиях:

Запрещена перевозка груза от 1-го поставщика к 1-му потребителю.

Определить, на сколько увеличилось значение функции из-за запрета перевозки по сравнению с оптимальным вариантом без запрета этой перевозки.

Поставка груза от 3-го поставщика к 1-му потребителю зафиксирована и равна 100 единицам. Требуется оценить удорожание перевозок груза по сравнению с оптимальным вариантом без этого условия.

От 2-го поставщика к 4-му потребителю необходимо поставить не менее 50 единиц груза (т.е. минимальная поставка груза от 2-го поставщика к 4-му потребителю равна 50 единиц) и оценить удорожание затрат на перевозку из-за этого условия.

Объем перевозки груза от 2-го поставщика к 1-му потребителю не должен превышать 300 единиц (максимальная поставка груза от 2-го поставщика к 1-му потребителю равна 300 единицам). Требуется определить удорожание затрат на перевозку груза из-за данного условия.

2. Имеются четыре участка земли для посева: 1) ржи, 2) пшеницы, 3) ячменя, 4) кукурузы. Площади участков соответственно равны 400, 200, 240, 220 га. Урожайность культур (ц/га) на соответствующих участках земли представлена матрицей  $C$ .

$$C = \begin{pmatrix} 20 & 25 & 20 & 22 \\ 26 & 22 & 30 & 28 \\ 22 & 14 & 16 & 18 \\ 40 & 42 & 47 & 50 \end{pmatrix}$$

Требуется определить, сколько гектаров земли засеять каждой из культур на каждом участке земли, чтобы суммарная стоимость собранного зерна была максимальной, если известно, что из-за ограниченности в семенном фонде можно засеять рожью, пшеницей, ячменем и кукурузой соответственно 240, 200, 300 и 320 га.

Стоимости центнера зерна соответствующих культур равны  $S_j$  единиц,  $j=1,2,3,4$  ( $S_1=3, S_2=5, S_3=4, S_4=7$ ).

### Варианты задач для самостоятельной работы

#### Вариант 1

$a_i \backslash b_j$	200	200	280
120	3	5	4
90	2	2	1
150	2	4	2
220	1	3	3

#### Вариант 2

$a_i \backslash b_j$	200	250	280
100	4	4	4
190	3	3	2
50	6	5	5
120	2	2	3

#### Вариант 3

$a_i \backslash b_j$	100	150	180
100	5	2	4
60	3	3	3
50	2	4	4
120	1	5	5

#### Вариант 4

$a_i \backslash b_j$	130	200	280
110	2	3	4
90	5	2	1
100	1	□	2
200	4	3	3

#### Вариант 5

$a_i \backslash b_j$	120	150	200
220	4	3	2
80	2	4	1
130	1	4	2
120	3	3	3

#### Вариант 6

$a_i \backslash b_j$	160	130	180
100	3	4	2
100	4	2	3
90	2	5	4
120	1	6	3

**Вариант 7**

$a_i \backslash b_j$	400	150	80
70	1	2	5
200	3	5	4
250	5	3	3
120	6	4	4

**Вариант 8**

$a_i \backslash b_j$	□00	350	200
320	3	1	3
100	2	3	2
130	2	5	2
120	4	2	3

**Вариант 9**

$a_i \backslash b_j$	120	150	180
140	3	5	4
130	2	3	3
50	1	4	5
200	5	3	2

**Вариант 10**

$a_i \backslash b_j$	100	150	250
80	3	4	1
190	6	3	4
50	2	4	4
320	1	5	6

**Вариант 11**

$a_i \backslash b_j$	100	50	200
150	3	5	4
90	2	3	□
170	2	3	3
140	4	1	2

**Вариант 12**

$a_i \backslash b_j$	210	150	140
100	3	5	1
70	5	1	3
130	2	4	2
220	2	3	3

**Вариант 13**

$a_i \backslash b_j$	120	150	80
220	3	5	5
190	3	2	3
50	2	3	4
70	4	1	3

**Вариант 14**

$a_i \backslash b_j$	100	150	100
120	3	5	3
70	1	3	1
50	2	4	5
120	4	1	3

**Вариант 15**

$a_i \backslash b_j$	100	180	200
120	3	3	4
80	4	2	1
150	2	4	2
210	4	1	3

**Вариант 16**

$a_i \backslash b_j$	170	150	80
130	3	5	4
90	4	2	1
250	2	3	3
120	1	3	2

**Вариант 17**

$a_i \backslash b_j$	100	150	200
120	3	6	4
100	3	3	1
150	2	1	3
130	3	2	4

**Вариант 18**

$a_i \backslash b_j$	200	240	210
100	3	5	4
110	4	2	3
150	3	4	2
160	2	1	3

**Вариант 19**

$a_i \backslash b_j$	250	150	180
130	3	4	3
70	2	5	1
120	3	4	2
210	4	3	3

**Вариант 20**

$a_i \backslash b_j$	100	150	80
100	2	2	4
100	2	3	3
50	4	4	2
130	1	4	2

**Вариант 21**

$a_i \backslash b_j$	80	100	130
80	5	1	4
90	3	5	3
120	2	3	5
20	1	4	6

**Вариант 22**

$a_i \backslash b_j$	100	130	150
120	3	4	5
110	5	1	4
250	2	5	3
170	1	4	2

**Вариант 23**

$a_i \backslash b_j$	100	150	180
100	3	2	4
80	4	5	3
130	5	4	2
140	2	3	5

**Вариант 24**

$a_i \backslash b_j$	100	120	150
140	3	5	1
170	1	2	4
50	2	6	2
120	3	3	3

**Вариант 25**

$a_i \backslash b_j$	200	150	140
140	1	5	5
190	4	2	1
50	3	2	3
120	2	3	3

**Вариант 26**

$a_i \backslash b_j$	170	50	150
200	4	1	2
130	3	2	4
40	5	5	1
80	2	3	4

**Вариант 27**

$a_i \backslash b_j$	250	50	150
100	3	4	5
150	5	3	4
50	3	4	3
200	4	1	4

**Вариант 28**

$a_i \backslash b_j$	300	50	150
300	5	1	2
100	1	5	3
0	3	4	3
200	4	4	2

**Вариант 29**

$a_i \backslash b_j$	150	100	120
280	4	1	2
300	3	4	3
100	2	3	4
40	5	2	1

**Вариант 30**

$a_i \backslash b_j$	100	60	80
130	4	2	4
200	3	1	4
50	3	3	2
40	2	5	1

## ЦЕЛОЧИСЛЕННАЯ ОПТИМИЗАЦИЯ

Задача линейного программирования, переменные которой принимают лишь целочисленные значения, называется задачей целочисленного ЛП.

### *Методы целочисленного программирования (ЦП)*

- точные;
- приближенные.

К *точным методам* относятся методы отсечения и комбинаторные методы. Точные методы имеют слабую сходимость. Приближенные методы построены на использовании особенностей конкретной задачи.

### *Методы отсечения*

Общая идея этих методов заключается в следующем:

Вначале решается задача без условия целочисленности. Если получилось целочисленное решение, то цель достигнута. Если же имеются нецелочисленные компоненты, то к ограничениям задачи добавляется новое ограничение, которое:

1. отсекает в оптимальном решении нецелочисленные компоненты;
2. не отсекает целочисленные компоненты.

С геометрической точки зрения это ограничение отсекает некоторые области с нецелыми координатами, не затрагивая область с целыми координатами.

Одним из методов отсечения является метод Гомори.

### *Метод Гомори*

В методе Гомори новые ограничения являются линейными и называются *правильным отсечением*.

Если в области решений есть целочисленные компоненты, то с помощью метода Гомори за конечное число итераций получается целочисленное решение.

Пусть имеется следующая задача целочисленного программирования.

$$\begin{cases} F = \sum_{j=1}^n c_j \cdot x_j \\ \sum_{j=1}^n a_{ij} \cdot x_j = b_i, \quad i = \overline{1, m} \\ x_j \geq 0, \quad j = \overline{1, n} \end{cases} \quad (1)$$

$$x_j \in Z \quad (Z - \text{целые числа}) \quad (2)$$

Пусть задача (1) решалась симплексным методом и на последнем шаге была получена следующая симплексная таблица:

БП	В	БП			СП		
		$x_1 \dots x_m$			$x_{m+1} \dots x_n$		
$x_1$	$b_1$	1	.	0	$a_{1, m+1}$	.	$a_{1n}$
$\vdots$	$\vdots$	$\vdots$	.	$\vdots$	$\vdots$	.	$\vdots$
$x_m$	$b_m$	0	.	1	$a_{m, m+1}$	.	$a_{mn}$
---	---	---			---		

Предположим, что оптимальное решение имеет вид:  $x^* = (b_1, \dots, b_m, 0, \dots, 0)$

Целой частью числа  $a$  называется наибольшее целое число  $[a]$ , не превосходящее  $a$ .

Дробной частью числа  $a$ , называется число, определяемое следующей формулой  $\{a\} = a - [a]$ .

Пример:

1.  $a = -3,7$       Целая: -4; дробная: 0,3

2.  $a = 3,7$       Целая: 3; дробная: 0,7

3.  $a = -\frac{3}{5}$       Целая: -1; дробная:  $\frac{2}{5}$

4.  $a = \frac{3}{5}$       Целая: 0; дробная:  $\frac{3}{5}$

Пусть среди компонент оптимального решения  $x^*$  есть нецелые числа. Среди нецелых компонент с наибольшей целой частью компонента  $b_i$ .

$$\text{Теорема. Неравенство вида: } \{b_i\} - \sum_{j=m+1}^n \{a_{ij}\}x_j \leq 0 \quad (3)$$

является правильным отсечением, т.е. оно:

- 1) линейно
- 2) отсекает найденное оптимальное нецелочисленное решение (1)
- 3) не отсекает ни одного целочисленного решения (1)-(2)

*Доказательство:*

1. Линейно, так как  $x$  в 1 степени.

2. Пусть  $b_i$  - нецелое. Покажем, что  $b_i$  не удовлетворяет (3).

По условию  $x^*$  - оптимальное решение (все свободные переменные равны 0).

$$\sum_{j=m+1}^n \{a_{ij}\}x_j = 0$$

После подстановки в (3), получается, что  $\{b_i\} \leq 0$ .

Так как по предположению  $b_i$  нецелое число, то его дробная часть отлична от нуля и кроме того должно быть  $< 0$ .

По определению дробной части числа, она не может быть отрицательной. Получается противоречие, что доказывает 2 часть теоремы.

3. Пусть целочисленные компоненты оптимального решения задачи (1)-(2) не удовлетворяют (3), т.е.

$$\{b_i\} - \sum_{j=m+1}^n \{a_{ij}\}x_j > 0 \quad (4)$$

Так как  $\sum_{j=m+1}^n \{a_{ij}\}x_j \geq 0$ ,  $0 \leq \{b_i\} < 1$ , то из (4) следует:

$$0 \leq \{b_i\} - \sum_{j=m+1}^n \{a_{ij}\}x_j < 1 \quad (5)$$

Обозначим компоненту

$$x_i = b_i - \sum_{j=m+1}^n a_{ij} x_j = \left( [b_i] - \sum_{j=m+1}^n [a_{ij}] x_j \right) + \left( \{b_i\} - \sum_{j=m+1}^n \{a_{ij}\} x_j \right) \quad (6)$$

Первое слагаемое в (6) - целое число. Пусть  $x_i$  - есть целая компонента. Для того чтобы она была целой второе слагаемое в (6) тоже должно быть целое. Получается противоречие с (5). Теорема доказана.

#### *Алгоритм метода Гомори*

- 1) Задача решается симплексным методом.
- 2) Определяется нецелочисленная компонента с наибольшей целой частью.
- 3) В ограничение (3) необходимо ввести дополнительную переменную  $x_{n+1}$  так, чтобы получилось равенство в (3). Добавляется новая строка в симплексной таблице с базисной переменной  $x_{n+1}$ , решается симплексным методом и получается оптимальное решение. Если оно целочисленное, то задача выполнена. Если нет, то переход к шагу 2.

#### *Замечание.*

Если в процессе решения появилась строка с нецелым свободным элементом и целыми остальными коэффициентами, то задача не имеет целочисленного решения.

Проиллюстрируем применение метода ветвей и границ на следующем примере.

Предприятие выпускает три вида продукции: I, II, III. Для производства продукции оно располагает ресурсами в следующих объемах (единиц):

Комплекующие изделия	3120
Сырье	3000
Материалы	3150

Расход ресурсов на единицу продукции каждого вида представлен в таблице:

Ресурсы	Вид продукции		
	I	II	III
Комплекующие изделия	4	6	8
Сырье	2	8	10
Материалы	6	9	4

Прибыль от единицы продукции I вида составляет 240 млн.руб., II – 210 млн.руб. и III – 180 млн.руб.

Требуется определить производственную программу предприятия, обеспечивающую максимальную прибыль.

Запишем математическую модель задачи:

$$Z = 240x_1 + 210x_2 + 180x_3 \rightarrow \max$$

$$\begin{cases} 4x_1 + 6x_2 + 8x_3 \leq 3120 \\ 2x_1 + 8x_2 + 10x_3 \leq 3000 \\ 6x_1 + 9x_2 + 4x_3 \leq 3150 \end{cases}$$

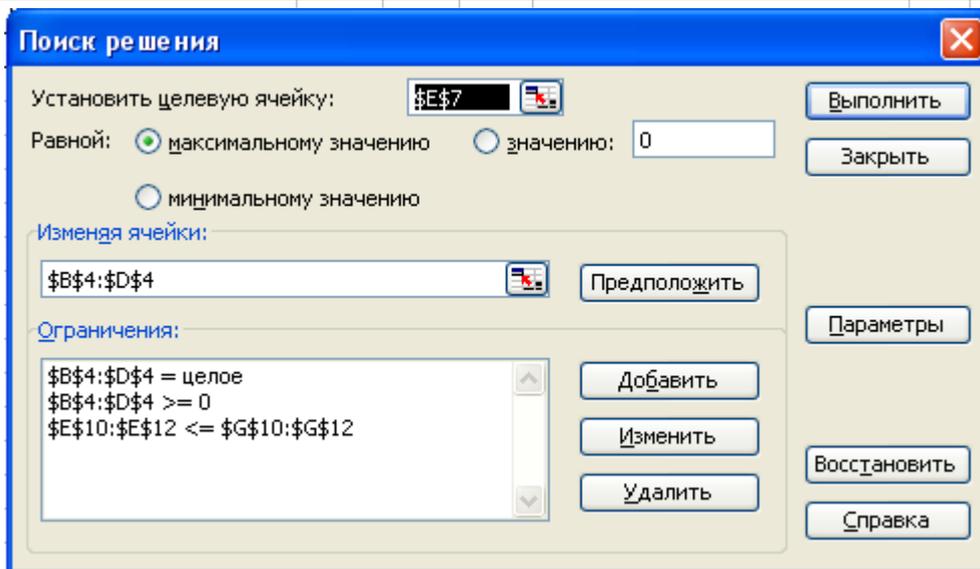
$$x_j \geq 0 \text{ и целые } (j=1,2,3)$$

Решение задачи целочисленной линейной оптимизации средствами **Excel** осуществляется командой **Поиск решения** из меню **Сервис**, так же как это делалось при решении задач без учета условия целочисленности. Однако при решении целочисленных задач необходимо в форму представления данных ввести требование целочисленности переменных и это требование занести с исходными данными в ЭВМ.

После ввода в ЭВМ исходной информации из примера вызываем диалоговое окно **Поиск решения** и заносим в этом окне необходимые данные. Закончив ввод ограничений, командой **Добавить** вводим условия целочисленности. Для этого в диалоговом окне **Добавление ограничения** вводим в окно **Ссылка на ячейку** адреса ячеек B4:D4, где должны быть изменяемые значения неизвестных  $x_1$ ,  $x_2$  и  $x_3$ . Далее курсор переводим в среднее окно, в котором находятся виды ограничений ( $\leq$ ,  $=$ ,  $\geq$ ) и требования (целое и

двоичное). Устанавливаем курсор на требование целочисленности и выполняем **М1**. Выполняем **М1** по **ОК**. На экране - диалоговое окно **Поиск решения**.

	A	B	C	D	E	F	G
1		ПЕРЕМЕННЫЕ					
2	<b>Имя</b>	<b>прод1</b>	<b>прод2</b>	<b>прод3</b>			
3	Переменные	X1	X2	X3			
4	Значение						
5	Нижняя граница						
6	Целочислен	целое	целое	целое			
7	Прибыль	240	210	180	=СУММПРОИЗВ(\$B\$4:\$D\$4;B7:D7)	макс	
8		ОГРАНИЧЕНИЯ					
9	<b>Вид</b>				левая часть	знак	объем ресурсов
10	Компл. Изд.	4	6	8	=СУММПРОИЗВ(\$B\$4:\$D\$4;B10:D10)	<=	3120
11	Сырье	2	8	10	=СУММПРОИЗВ(\$B\$4:\$D\$4;B11:D11)	<=	3000
12	Материалы	6	9	4	=СУММПРОИЗВ(\$B\$4:\$D\$4;B12:D12)	<=	3150



Выполнив щелчок мышью по команде **Параметры**, устанавливаем необходимые параметры в диалоговом окне **Параметры поиска решения**. Возвращаемся в диалоговое окно **Поиск решения** и, щелкнув мышью по команде **Выполнить**, получаем целочисленное решение:

	A	B	C	D	E	F	G
1		ПЕРЕМЕННЫЕ					
2	<b>Имя</b>	<b>прод1</b>	<b>прод2</b>	<b>прод3</b>			
3	Переменные	X1	X2	X3			
4	Значение	398	0	191			
5	Нижняя граница	0	0	0			
6	Целочислен	целое	целое	целое			
7	Прибыль	240	210	180	129825	макс	
8		ОГРАНИЧЕНИЯ					
9	<b>Вид</b>				левая часть	знак	объем ресурсов
10	Компл. Изд.	4	6	8	3120	<=	3120
11	Сырье	2	8	10	2707,5	<=	3000
12	Материалы	6	9	4	3150	<=	3150

## Упражнения

Найти целочисленные решения задач:

$$Z = 70x_1 + 80x_2 + 110x_3 + 60x_4 \rightarrow \max$$

$$1. \begin{cases} x_1 + 2x_2 + x_3 + 2x_4 \leq 30 \\ x_1 + x_2 + 2x_3 + x_4 \leq 28 \\ 2x_1 + 3x_2 + 5x_3 + x_4 \leq 50 \\ 3x_1 + x_2 + x_3 + x_4 \leq 45 \end{cases}$$

$$x_j \geq 0, \quad j = \overline{1,4}$$

$$Z = 2x_1 + 3x_2 + 4x_3 \rightarrow \min$$

$$2. \begin{cases} x_1 + 2x_2 + x_3 \geq 20 \\ 2x_1 + 3x_2 + 5x_3 \geq 50 \\ x_1 + x_2 + x_3 \geq 60 \end{cases}$$

$$x_j \geq 0, \quad j = \overline{1,3}$$

$$Z = 20x_1 + 25x_2 + 30x_3 + 15x_4 \rightarrow \max$$

$$3. \begin{cases} x_1 + x_2 + x_3 + 2x_4 \geq 18 \\ 2x_1 + 3x_2 + 5x_3 + x_4 \leq 40 \\ x_1 + 2x_2 + 3x_3 + 5x_4 \leq 60 \end{cases}$$

$$x_j \geq 0, \quad j = \overline{1,4}$$

4. Для перевозки пассажиров по трем маршрутам аэропорт может использовать три типа самолетов. Вместимость самолета  $i$ -го типа равна 150, 200, 300 пассажиров соответственно, а потребность в перевозке пассажиров по  $j$ -му маршруту ( $j=1,2,3$ ) за сезон составляет соответственно 5600, 7000 и 6500 человек. Затраты на эксплуатацию  $i$ -го типа самолета на  $j$ -м маршруте равны  $c_{ij}$  денежных единиц и представлены матрицей  $C$

$$\tilde{N} = \begin{pmatrix} 50 & 60 & 40 \\ 45 & 53 & 60 \\ 70 & 65 & 50 \end{pmatrix}$$

Парк самолетов каждого вида составляет 35, 38 и 25 единиц соответственно.

Требуется определить, сколько самолетов каждого типа использовать на каждом из маршрутов, чтобы затраты на перевозку пассажиров были минимальными.

5. Морское судно грузоподъемностью 20 тыс. т и вместимостью 28 тыс. м<sup>3</sup> может быть использовано для перевозки пяти видов груза. Данные о массе, объеме и стоимости единицы груза каждого вида приведены в таблице.

Параметры единицы груза	Номер груза				
	1	2	3	4	5
Масса, т	95	70	90	105	75
Объем, м <sup>3</sup>	125	90	110	100	120
Стоимость, млн.руб.	270	280	440	350	400

Определить сколько единиц груза каждого вида следует загрузить на судно, чтобы суммарная стоимость груза была максимальной и выполнялись ограничения по вместимости и грузоподъемности судна.

### Варианты задач для самостоятельной работы

#### Вариант 1

$$z = x_1 + 4x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 4x_2 \leq 17, \\ 10x_1 + 3x_2 \leq 15, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

#### Вариант 2

$$z = 2x_1 + 3x_2 \rightarrow \min$$

$$\begin{cases} x_1 + 2x_2 \geq 16, \\ 2x_1 + x_2 \geq 16, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

#### Вариант 3

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} -x_1 + 2x_2 \leq 14, \\ 7x_1 + 4x_2 \leq 28, \\ 4x_1 - 3x_2 \leq 12, \\ 10x_1 - 8x_2 \geq 2, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

#### Вариант 4

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} -x_1 + 2x_2 \leq 14, \\ 7x_1 + 4x_2 \leq 28, \\ 4x_1 - 3x_2 \leq 12, \\ 10x_1 - 8x_2 \geq 2, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

#### Вариант 5

#### Вариант 6

$$z = x_1 + 2x_2 \rightarrow \min$$

$$\begin{cases} -3x_1 + 4x_2 \leq 12, \\ x_1 + x_2 \geq 4, \\ 3x_1 + 3x_2 \leq 15, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 7

$$z = 3x_1 + 4x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + 2x_2 \leq 7, \\ x_1 + 4x_2 \leq 15, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 9

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} x_1 + 2x_2 \leq 16, \\ 2x_1 + x_2 \leq 16, \\ x_1 + 2x_2 \geq 2, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 11

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + x_2 \leq 10, \\ 3x_1 + x_2 \leq 6, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 13

$$z = 2x_1 - x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + x_2 \leq 8, \\ x_1 + 3x_2 \geq 10, \\ 3x_1 + x_2 \geq 3, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 8

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 4x_2 \leq 10, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 10

$$z = 2x_1 + 5x_2 \rightarrow \min$$

$$\begin{cases} x_1 + 3x_2 \leq 15, \\ 8x_1 + 4x_2 \leq 32, \\ 4x_1 + 3x_2 \leq 12, \\ 5x_1 + 2x_2 \geq 10, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 12

$$z = 8x_1 + 5x_2 \rightarrow \max$$

$$\begin{cases} 5x_1 + 2x_2 \leq 20, \\ x_1 + x_2 \leq 6, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 14

$$z = 4x_1 + 4x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + x_2 \leq 5, \\ 2x_1 + x_2 \leq 9, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 15

$$z = 2x_1 + x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 3x_2 \leq 12, \\ 4x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 17

$$z = 7x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 5x_1 + 2x_2 \leq 20, \\ 4x_1 + 2x_2 \leq 16, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 19

$$z = 4x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + 2x_2 \leq 8, \\ x_1 + 4x_2 \leq 10, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 21

$$z = x_1 + 2x_2 \rightarrow \max$$

$$\begin{cases} 4x_1 + 3x_2 \leq 24, \\ -x_1 + x_2 \leq 3, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 16

$$z = 2x_1 + 2x_2 \rightarrow \max$$

$$\begin{cases} -3x_1 + 4x_2 \leq 12, \\ 3x_1 + 4x_2 \leq 30, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 18

$$z = 8x_1 + 6x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 5x_2 \leq 12, \\ 4x_1 + x_2 \leq 10, \\ 2x_1 - 3x_2 \leq 12, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 20

$$z = 3x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} x_1 + 2x_2 \geq 5, \\ 3x_1 + 2x_2 \leq 25, \\ x_2 \leq 10, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 22

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 - 3x_2 \geq 12, \\ x_1 + x_2 \geq 4, \\ 3x_1 + 6x_2 \leq 24, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 23

$$z = x_1 + x_2 \rightarrow \max$$

$$\begin{cases} -4x_1 + x_2 \leq 1, \\ 2x_1 - 3x_2 \leq 6, \\ 2x_1 + x_2 \leq 8, \\ -x_1 + x_2 \leq 7, \\ x_1 + 2x_2 \geq 2, \\ x_1, x_2 \geq 0, \quad x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 25

$$z = 2x_1 + 5x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + 4x_2 \leq 25, \\ 2x_1 - x_2 \geq 1, \\ x_1 \leq 5, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 27

$$z = 7x_1 + 6x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 5x_2 \geq 10, \\ 5x_1 + 2x_2 \geq 10, \\ x_1 \leq 6, \\ x_2 \leq 5, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 29

$$z = 6x_1 + 4x_2 \rightarrow \min$$

$$\begin{cases} 3x_1 + 2x_2 \geq 6, \\ 2x_1 - 3x_2 \geq -6, \\ x_1 - x_2 \leq 4, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 24

$$z = x_1 + 2x_2 \rightarrow \min$$

$$\begin{cases} -3x_1 + 2x_2 \leq 9, \\ 3x_1 + 4x_2 \geq 28, \\ 2x_1 + x_2 \leq 8, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 26

$$z = x_1 + x_2 \rightarrow \max$$

$$\begin{cases} 3x_1 + x_2 \leq 20, \\ 2x_1 + 3x_2 \leq 30, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 28

$$z = 3x_1 - 3x_2 \rightarrow \max$$

$$\begin{cases} x_1 - 4x_2 \leq 4, \\ 3x_1 + 2x_2 \leq 6, \\ -x_1 + x_2 \geq 7, \\ x_1 + 2x_2 \leq 2, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

### Вариант 30

$$z = 2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 2x_2 \geq 1, \\ 4x_1 - x_2 \leq 15, \\ x_1 + 3x_2 \leq 16, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

$$z = 3x_1 + x_2 \rightarrow \min$$

$$\begin{cases} -4x_1 + x_2 \leq 29, \\ 3x_1 - x_2 \leq 15, \\ 5x_1 + 2x_2 \geq 38, \\ x_1, x_2 \geq 0, \\ x_1, x_2 - \text{целые.} \end{cases}$$

## СИМПЛЕКС-МЕТОД РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Пусть поставлена стандартная задача линейного программирования:

$$F = \sum_{j=1}^n c_j x_j \rightarrow \max(\min)$$

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = \overline{1, m}$$

$$x_j \geq 0, \quad j = \overline{1, n}.$$

Каждое из ограничений задачи определяет полупространство, ограниченное гиперплоскостью (плоскостью в  $k$ -мерном пространстве). Пересечение соответствующих полупространств образует выпуклый многогранник, в котором необходимо найти максимум (минимум) целевой функции.

Теоретически достаточно найти конечное число вершин (угловых точек) многогранника и вычислить в них значения целевой функции, затем выбрать из них наибольшее (наименьшее).

Однако простой перебор практически неосуществим из-за чрезвычайно большого числа вершин. В результате возникла необходимость применения метода целенаправленного перебора, который приводит к решению задачи за приемлемое время. Таким методом является симплекс-метод.

Симплекс это простейший выпуклый многогранник. Решение задачи ЛП симплекс-методом состоит в определении одной из вершин многогранника условий и последовательном переходе от одной вершины к другой, причем каждый такой переход приближает решение к оптимальному. В этом заключается геометрический смысл симплекс-метода.

Базисное решение, удовлетворяющее условиям неотрицательности всех переменных, называется *допустимым базисным решением (опорным планом)*.

Опорный план называется *невырожденным*, если он содержит ровно  $m$  положительных компонент, в противном случае, он называется *вырожденным*.

Идея симплексного метода состоит в:

1. умении находить первую угловую точку (начальное базисное решение);
2. наличие признака оптимальности базисного решения;
3. умении переходить к нехудшему базисному решению.

*Построение начального базисного решения*

Предположим, что имеется каноническая задача ЛП:

$$(1') \left\{ \begin{array}{l} F = \sum_{j=1}^n c_j x_j \\ \sum_{j=1}^n a_{ij} x_j = b_i \\ i = \overline{1, m}, \quad m < n \\ x_j \geq 0, j = \overline{1, n} \end{array} \right. \quad (1)$$

Предположим, что  $b_i \geq 0$ .

Ограничение в (1) имеет *предпочтительный вид*, если левая часть ограничения содержит переменную, входящую с коэффициентом =1, а остальные ограничения с коэффициентом 0.

Если каждое ограничение в (1) имеет предпочтительный вид, то говорят, что (1) представлена в предпочтительном виде.

В этом случае допустимое базисное решение находится путем приравнивания к 0 всех неpreferred переменных. Если при этом получили не более  $m$  нулевых координат, то согласно теореме об угловых точках многоугольника решений получили допустимое базисное решение. В этом случае предпочтительные переменные есть базисные, а неpreferred – свободные.

Предположим, что система ограничений имеет вид:

$$\sum_{j=1}^n a_{ij} x_j \leq b_i \\ b_i \geq 0 \quad i = \overline{1, m}$$

Приведем данное ограничение к предпочтительному виду, для этого просто введем дополнительные неотрицательные переменные  $x_{n+i} \geq 0$  так, чтобы

$$\sum_{j=1}^n a_{ij} x_j + x_{n+i} = b_i, i = \overline{1, m}.$$

Видно, что начальное допустимое базисное решение с базисными переменными равными дополнительным переменным будет иметь вид:

$$X = (\underbrace{0, \dots, 0}_n, \underbrace{b_1, \dots, b_m}_m).$$

### *Первый метод искусственного базиса*

Пусть ограничения имеют вид:

$$\sum_{j=1}^n a_{ij} x_j \geq b_i \tag{2}$$

$$b_i \geq 0 \quad i = \overline{1, m}$$

Введем дополнительные переменные так, чтобы неравенства превратились в равенства:

$$\sum_{j=1}^n a_{ij} x_j - x_{n+i} = b_i, \tag{3}$$

$$x_{n+i} \geq 0, \quad i = \overline{1, m}$$

Очевидно, что последние ограничения не имеют предпочтительный вид и базисное решение будет недопустимым:

$$X = (\underbrace{0, \dots, 0}_n, \underbrace{-b_1, \dots, -b_m}_m).$$

Избавиться от неpreferable вида можно путем введения так называемых искусственных переменных, используя первый метод искусственного базиса.

Для этого к левым частям (3) добавляем искусственные переменные  $w_i \geq 0$ . В функцию  $F$  переменные  $w_i$  вводятся с коэффициентом  $M$  в случае нахождения минимума, и  $-M$  в случае нахождения максимума, где  $M$  – большое

положительное число. С помощью этого действия получается так называемая «*M-задача*», соответствующая исходной. *M-задача* всегда имеет предпочтительный вид.

Предполагаем, что система ограничений имеет знак равенства вида (1) (если был знак неравенства, то путем введения дополнительных переменных этого вида можно добиться, как например, в (2), (3)).

Пусть в (1) ни одно из ограничений не имеет предпочтительного вида.

Записываем *M-задачу* к задаче (1'):

$$(2') \left\{ \begin{array}{l} F^* = \sum_{j=1}^n c_j x_j - (+) \sum_{i=1}^m M w_i \\ \sum_{j=1}^n a_{ij} x_j + w_i = b_i, i = \overline{1, m}, \quad b_i \geq 0 \text{ (-) - максимум, (+) - минимум} \\ x_j \geq 0, j = \overline{1, n}, \quad w_i \geq 0, i = \overline{1, m} \end{array} \right.$$

*Теорема.* Если в оптимальном решении *M-задачи*  $X^* = (x_1, \dots, x_n, w_1, \dots, w_m)$  все искусственные переменные  $w_i = 0$ , то решение  $X = (x_1, \dots, x_n)$  является оптимальным для исходной задачи (1').

*Доказательство:*

Если в задаче (2') подставить  $w_i = 0$ , то получим задачу (1'), при этом очевидно:

$$F^*(X^*) = F(X)$$

Докажем, что  $X$  – оптимальное решение для (1').

Допустим, что это не так. Тогда существует решение  $\hat{X}$  задачи (1') такое, что  $F(\hat{X}) \geq F(X)$ .

Этому решению  $\hat{X}$  соответствует решение *M-задачи*  $\hat{X}^*$ . Тогда

$$F^*(\hat{X}^*) = F(\hat{X}) \geq F(X) = F^*(X^*)$$

$$F^*(\hat{X}^*) = F^*(X^*)$$

Последняя строчка противоречит тому, что  $X^*$  - оптимальное решение задачи (2'). Получили противоречие, что доказывает теорему. Теорема доказана.

Алгоритм приведения задачи к предпочтительному виду:

1. Добиваются, чтобы все  $b_i \geq 0$ . В противном случае умножить на (-1).
2. Если были знаки неравенств, то ввести дополнительные переменные так, чтобы стали равенства (дополнительные переменные  $\geq 0$ ).
3. Если некоторые равенства не имеют предпочтительный вид, то ввести искусственные переменные и  $M$ -задачу.

Далее решают  $M$ -задачу, при этом способ решения называется *симплексным методом с первым искусственным базисом*.

### Симплексные таблицы

Любую задачу можно представить в предпочтительном виде, если же его нет, то решать  $M$ -задачу и добиваться, чтобы все  $w_i = 0$ .

Пусть имеем задачу в предпочтительном виде:

$$F = \sum_{j=1}^n c_j x_j \quad (4)$$

$$x_i + \sum_{j=m+1}^n a_{ij} x_j = b_i, b_i \geq 0 \quad i = \overline{1, m} \quad (5)$$

$$x_j \geq 0, \quad j = \overline{1, n} \quad (6)$$

Выражаем из (5) базисные переменные  $x_i$ :

$$x_i = b_i - \sum_{j=m+1}^n a_{ij} x_j$$

Подставляем в (4) и группируем слагаемые так, чтобы:

$$F = \sum_{j=1}^m c_j b_j - \sum_{j=m+1}^n \left( \sum_{i=1}^m c_i a_{ij} - c_j \right) x_j$$

Обозначим:

$$\Delta_0 = \sum_{j=1}^m c_j b_j = c_{\sigma} \cdot B$$

$$\Delta_j = \sum_{i=1}^m c_i a_{ij} - c_j = c_{\sigma} \cdot A_j - c_j$$

$c_{\bar{0}} = (c_1, \dots, c_m)$  - вектор коэффициентов в целевой функции  $F$  при базисных переменных.

$$B = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \quad A_j = \begin{pmatrix} a_{1j} \\ \vdots \\ a_{mj} \end{pmatrix}$$

Таким образом, задачу (4)-(6) можно переписать в удобном для применения симплексного метода виде:

$$\begin{cases} F = \Delta_0 - \sum_{j=m+1}^n \Delta_j x_j \\ x_i + \sum_{j=m+1}^n a_{ij} x_j = b_i, \\ b_i \geq 0 \quad i = \overline{1, m} \\ x_j \geq 0, j = \overline{1, n} \end{cases} \quad (7)$$

Задачу (7) удобно решать в виде так называемой «симплексной таблицы», т.е. таблицы следующего вида:

<i>Б.П.</i>	$c_{\bar{0}}$	$B$	$x_1$	...	$x_m$	$x_{m+1}$	...	$x_n$
			$c_1$	...	$c_m$	$c_{m+1}$	...	$c_n$
$x_1$	$c_1$	$b_1$	$1$	$0$	$0$	$a_{1m+1}$	...	$a_{1n}$
...	...	...	...	...	...	...	...	...
$x_m$	$c_m$	$b_m$	$0$	$0$	$1$	$a_{mm+1}$	...	$a_{mn}$
$F_j - c_j$		$\Delta_0$	$0$	...	$0$	$\Delta_{m+1}$	...	$\Delta_n$

$\Delta_j$  - оценки свободных переменных;

$\Delta_0$  - значения  $F$  для исходного базисного решения.

### *Признак оптимальности допустимого базисного решения*

*Теорема.* Если для некоторого допустимого базисного решения все оценки  $\Delta_j \geq 0$  ( $\Delta_j \leq 0$ ), то такое базисное решение доставляет максимум (минимум).

*Доказательство:*

$$F = \Delta_0 - \sum_{j=m+1}^n \Delta_j x_j$$

Пусть  $\Delta_j \geq 0$ .

Видно, что  $F$  достигает максимального значения при  $\sum_{j=m+1}^n \Delta_j x_j = 0$ . Это возможно лишь при  $x_{m+1} = 0, \dots, x_n = 0$ , т.е. опорный план оптимален. Теорема доказана.

*Переход к нехудшему допустимому базисному решению*

Рассмотрим задачу на максимум.

Пусть существует  $j_0$  такое, что  $\Delta_{j_0} < 0$ .

Вектор-столбец  $A_{j_0}$  называется разрешающим, а переменная  $x_{j_0}$  - перспективной. Можно не изменяя нулевых значений свободных переменных, не считая  $x_{j_0}$ , увеличить значение функции  $F$  за счет перевода  $x_{j_0}$  в базисные переменные, т.е. за счет увеличения его значения.

Однако при увеличении  $x_{j_0}$  необходимо учитывать условие неотрицательности базисных переменных. Из (5) при условии равенства нулю свободных переменных, а  $x_{j_0} > 0$  следует:

$$x_i = b_i - a_{i j_0} x_{j_0}, \quad i = \overline{1, m}$$

Если  $a_{i j_0} > 0$ , то при значительном увеличении  $x_{j_0}$  можно получить  $x_i < 0$ , что недопустимо (при  $a_{i j_0} < 0$  этого не произойдет).

Пусть, например, первые  $k$  коэффициентов  $a_{i j_0} > 0$ ,  $i = \overline{1, k}$ . Тогда  $x_{j_0}$  можно увеличивать до тех пор, пока  $b_i - a_{i j_0} x_{j_0} \geq 0$ .

$$x_{j_0} \leq \frac{b_i}{a_{i j_0}}, \quad a_{i j_0} > 0, \quad i = \overline{1, k}.$$

Выберем среди правых частей наименьшую дробь. Пусть она достигается в строке  $i_0$  (ее назовем разрешающей), а элемент  $a_{i_0 j_0}$  - разрешающим

элементом. Базисную переменную  $x_{i_0}$  переводим в свободные переменные, а свободную переменную  $x_{j_0}$  переводим в базисные. Получаем новое допустимое базисное решение, в котором  $\Delta_0 - \Delta_{j_0} x_{i_0} = \underbrace{F(x_0) - \Delta_{j_0} x_{i_0}}_{F(x_1)} \geq F(x_0)$ , т.е. значение целевой функции не уменьшилось и, следовательно, новое допустимое базисное решение  $x_1$  не хуже решения  $x_0$ .

*Замечание:*

Если имеется выбор между несколькими столбцами, то среди положительных выбирается наибольший элемент при нахождении минимума, а при нахождении максимума среди отрицательных выбирается наименьший. Другими словами всегда выбирается наибольший по модулю.

*Преобразование симплексной таблицы при переходе к новому допустимому базисному решению*

Выразим новую базисную переменную  $x_{j_0}$  из уравнения с номером  $i=i_0$  системы (5) через свободные переменные  $x_{m+1}, \dots, x_{j_0-1}, x_{j_0}, x_{j_0+1}, \dots, x_n$ :

$$x_{j_0} = \frac{b_{i_0}}{a_{i_0 j_0}} - \left( \frac{a_{i_0, m+1}}{a_{i_0 j_0}} x_{m+1} + \dots + \frac{a_{i_0, j_0-1}}{a_{i_0 j_0}} x_{j_0-1} + \frac{1}{a_{i_0 j_0}} x_{i_0} + \frac{a_{i_0, j_0+1}}{a_{i_0 j_0}} x_{j_0+1} + \dots + \frac{a_{i_0, n}}{a_{i_0 j_0}} x_n \right).$$

Подставляем данное выражение в остальные ограничения:

$$x_i = \frac{b_i a_{i_0 j_0} - b_{i_0} a_{i j_0}}{a_{i_0 j_0}} - \left( \frac{a_{i, m+1} a_{i_0 j_0} - a_{i_0, m+1} a_{i j_0}}{a_{i_0 j_0}} x_{m+1} + \dots + \frac{a_{i, j_0-1} a_{i_0 j_0} - a_{i_0, j_0-1} a_{i j_0}}{a_{i_0 j_0}} x_{j_0-1} + \frac{a_{i j_0}}{a_{i_0 j_0}} x_{i_0} + \frac{a_{i, j_0+1} a_{i_0 j_0} - a_{i_0, j_0+1} a_{i j_0}}{a_{i_0 j_0}} x_{j_0+1} + \dots + \frac{a_{i, n} a_{i_0 j_0} - a_{i_0, n} a_{i j_0}}{a_{i_0 j_0}} x_n \right)$$

$$F = \frac{\Delta_0 a_{i_0 j_0} - \Delta_{j_0} b_{i_0}}{a_{i_0 j_0}} - \left( \frac{\Delta_{m+1} a_{i_0 j_0} - \Delta_{j_0} a_{i_0, m+1}}{a_{i_0 j_0}} x_{m+1} + \dots + \frac{\Delta_{j_0}}{a_{i_0 j_0}} x_{i_0} + \dots + \frac{\Delta_n a_{i_0 j_0} - \Delta_{j_0} a_{i_0, n}}{a_{i_0 j_0}} x_n \right)$$

Из этих формул вытекают правила преобразования симплексной таблицы:

1. Элементы разрешающей строки  $i_0$  в новой симплексной таблице делятся на разрешающий элемент.
2. Элементы разрешающего столбца  $j_0$  равны 0, а на месте разрешающего элемента ставится 1.
3. Остальные элементы таблицы заполняются по правилу прямоугольника: вначале строится прямоугольник, одна вершина которого совпадает с элементом, требующего изменения, с противоположной вершиной, равной разрешающему элементу. Затем вместо старого элемента записывают разность между этим элементом и дробью, числитель в которой равен произведению двух других вершин, а знаменатель – разрешающий элемент.

Другими словами, из произведения разрешающего элемента и противоположной вершины вычитаются произведения двух других вершин, и вся разность делится на разрешающий элемент.

### *Признак альтернативного решения*

Если задача ЛП имеет бесконечное множество решений, то говорят о случае альтернативности решения.

*Теорема.* Если в последней строке симплексной таблицы, содержащей оптимальное решение, имеется хотя бы один нулевой в столбце свободных переменных, то задача имеет альтернативное решение.

*Доказательство:*

Пусть в оптимальном решении  $\Delta_{j_0} = 0$ . Переведем  $x_{j_0}$  в базисные переменные, а некоторую переменную  $x_{i_0}$  в свободные переменные. Тогда

$$\Delta'_0 = \Delta_0 - \frac{\overbrace{\Delta_{j_0}}^{=0} \cdot b_{i_0}}{a_{i_0 j_0}} = \Delta_0.$$

Следовательно, значение целевой функции  $F$  не изменилось при переходе к новому допустимому базисному решению. Получили два оптимальных решения  $x_1^*$ ,  $x_2^*$  (две угловые точки). Согласно второй части основной теоремы

линейного программирования оптимальным решением будет любая точка  $x^*$ , являющаяся выпуклой линейной комбинацией точек  $x_1^*$ ,  $x_2^*$ , т.е. все точки отрезка  $x_1^* x_2^*$ . Теорема доказана.

*Следствие:*

Если в последней строке оптимальной симплексной таблицы все элементы ненулевые, то оптимальное решение будет единственным.

### *Признак неограниченности целевой функции*

*Теорема.* Если в последней строке симплексной таблицы содержится  $\Delta_{j_0} < 0$  ( $> 0$ ), а в столбце  $j_0$  нет ни одного положительного элемента, то целевая функция на множестве допустимых решений неограниченна сверху (снизу).

*Доказательство:*

Пусть  $\Delta_{j_0} < 0$ . Переведем  $x_{j_0}$  в базисные переменные и все остальные свободные переменные равны 0.

Получим:

$$x_i = b_i - a_{i j_0} x_{j_0}, \quad i = \overline{1, m} \quad (8)$$

При переводе необходимо, чтобы правая часть (8) была  $\geq 0$ .

В силу предположения теоремы все  $a_{i j_0} \leq 0$ . Следовательно,  $x_{j_0}$  может быть сколь угодно большим, при этом все  $x_i$  будут всегда  $\geq 0$ .

Так как функция  $F$  представляется в виде:

$$F = \Delta_0 - (\Delta_{m+1} x_{m+1} + \dots + \Delta_{j_0} x_{j_0} + \dots + \Delta_n x_n)$$

Учитывая вид функции  $F$  и вышесказанное, функция может принимать сколь угодно большие значения. Теорема доказана.

### *Сходимость симплексного метода*

Каноническая задача ЛП называется *невыврожденной*, если все ее базисные решения невырождены, т.е. среди базисных переменных нет нулевых элементов.

Задача ЛП называется *вырожденной*, если хотя бы одно из базисных решений вырождено.

Известно, что при нахождении максимума, значения на двух последовательных итерациях связаны соотношением:

$$\Delta_0^{(k+1)} = \Delta_0^{(k)} - \frac{\Delta_{j_0} \cdot b_{i_0}}{a_{i_0 j_0}}$$

где  $b_{i_0} \geq 0$     $\Delta_{j_0} \leq 0$     $a_{i_0 j_0} > 0$

Если задача невырожденная, то  $b_{i_0} > 0$  и  $\Delta_0^{(k+1)} \geq \Delta_0^{(k)}$  при  $\Delta_{j_0} < 0$  и функция  $F$  монотонно возрастает (для минимума – монотонно убывает). В этом смысле и говорят о монотонности симплексного метода.

Если  $\Delta_{j_0} \leq 0$ , то  $\Delta_0^{(k+1)} \geq \Delta_0^{(k)}$  и в этом случае также говорят о монотонности симплексного метода, т.е. если в процессе решения симплексным методом произошло увеличение (уменьшение) функции  $F$  на итерации  $k$ , то в силу единственности допустимого базисного решения и монотонности симплексного метода решения от 1 до  $(k-1)$  включительно в дальнейших итерациях не будут встречаться.

*Теорема.* Если все допустимые базисные решения задачи линейного программирования невырождены, то симплексным методом за конечное число итераций либо будет найдено оптимальное решение задачи линейного программирования, либо будет установлена неограниченность целевой функции  $F$ .

*Доказательство:*

Пусть задача на максимум и дошли до решения с номером  $l$ .

От противного, пусть существует  $\Delta_{j_0} < 0$

Если  $F$  ограничена, то в столбце  $j_0$  существуют положительные элементы и в силу невырожденности базисного решения получится:

$$\min \frac{b_i}{a_{ij_0}} = \frac{b_{i_0}}{a_{i_0j_0}} > 0$$

Согласно симплексному методу два значения целевой функции на последовательных итерациях:

$$\Delta_0^{l+1} = \Delta_0^l - \frac{\Delta_{j_0} \cdot b_{i_0}}{a_{i_0j_0}}$$

Следовательно, новое значение  $F$  на  $(l+1)$  итерации будет больше значения  $F$  на итерации  $l$ .

Известно, что решения с первого по  $l$  включительно не встретятся в дальнейших итерациях. Так как  $l$  произвольно, то ни одно решение не встретится дважды. Так как число допустимых базисных решений ограничено, то получается противоречие. Теорема доказана.

#### *Защивание симплексного алгоритма*

Если задача линейного программирования вырождена, то некоторая  $b_{i_0} = 0$ .

Так как:  $\Delta_0^{l+1} = F(x^{l+1})$ ;  $\Delta_0^l = F(x^l)$ .

В силу  $b_{i_0} = 0$  мы переходим к нехудшему допустимому базисному решению, но значение  $F$  не изменится. Если такая ситуация будет повторяться при переходе к новому базисному решению, то в силу конечности числа допустимых базисных решений мы придем к решению, которое встречалось ранее, т.е. мы будем иметь повторяющиеся симплексные таблицы, получим *защивание*.

В реальных задачах защивание не наблюдается, но существуют теоретические примеры защивания. Так как теоретически защивание возможно, то разработаны правила его исключения. Защивание можно избежать, если взять на некотором этапе другие разрешающие элементы.

Вектор  $x = (x_1, \dots, x_n)$  лексикографически больше вектора  $y = (y_1, \dots, y_n)$ , если  $x \neq y$  и  $x_k > y_k$ , где  $k = \min \{i : x_i \neq y_i\}$ . Обозначение:  $x \succ y$

Строчку в симплексной таблице, соответствующей базисной переменной  $x_i$  обозначим через  $Q_i$ . Последнюю строчку симплексной таблицы обозначим через  $Q_{m+1} = (\Delta_0, \Delta_1, \dots, \Delta_n)$ .

Допустимое базисное решение  $x_0$  называется *обобщенно положительным*, если:  $Q_i \succ 0, \forall i \in I = \{i_1, \dots, i_m\}$ .

Если  $x_0$  является допустимым базисным решением, то добиться его обобщенно положительности можно перенумерацией переменных (столбцов) симплексной таблицы.

Для невырожденной задачи ЛП любое допустимое базисное решение является обобщенно положительным.

*Теорема.* Пусть решается задача на минимум и имеется допустимое базисное решение  $x_0$ . Пусть  $\Delta_{j_0} > 0$ , т.е. не удовлетворяет критерию оптимальности. Тогда, если номер разрешающей строки  $i_0$  выбрать из множества  $I' = \{i \in I : a_{ij_0} > 0\}$  так, чтобы

$$\frac{1}{a_{i_0 j_0}} Q_{i_0} \prec \frac{1}{a_{ij_0}} Q_i \quad (9)$$

$\forall i \in I', i \neq i_0$ , то новое допустимое базисное решение  $x_1$  будет обобщенно положительным и при этом  $\tilde{Q}_{m+1} \prec Q_{m+1}$  ( $\sim$  - новая симплексная таблица).

*Доказательство:*

Согласно симплексному методу в новой симплексной таблице  $i_0, j_0$  меняются местами и

$$\tilde{Q}_{j_0} = \frac{Q_{i_0}}{a_{i_0 j_0}} \quad (10)$$

Очевидно, что:

$$\tilde{Q}_{j_0} \succ 0$$

Для  $i \neq i_0$ , где  $i \in I$ , строки сохраняют свои номера и при этом

$$\tilde{a}_{ij} = a_{ij} - \frac{a_{i_0j} a_{ij_0}}{a_{i_0j_0}}$$

$$j = \overline{0, n}, \quad a_{i_0} = b_i$$

Перепишем последнюю строчку в векторном виде

$$\tilde{Q}_i = Q_i - \frac{a_{ij_0}}{a_{i_0j_0}} Q_{i_0} \quad (11)$$

$$i \neq i_0, \quad i \in I.$$

Для номеров  $i$ , из множества равному разности  $\{I - I'\}$  выполняется:  $a_{ij_0} \leq 0$  и в силу (11)  $\tilde{Q}_i \succ 0$ . Для номеров  $i \in I'$  последнее лексикографическое неравенство  $\tilde{Q}_i \succ 0$  выполняется в силу (9) и (11).

Таким образом, с учетом (10) новое допустимое базисное решение  $x_1$  будет обобщено положительным. Кроме того:

$$\tilde{Q}_{m+1} = Q_{m+1} - \frac{\Delta_{j_0}}{a_{i_0j_0}} Q_{i_0} \quad (12)$$

и неравенство  $Q_{m+1} \succ \tilde{Q}_{m+1}$  следует из (12), (10),  $\Delta_{j_0} > 0$ . Теорема доказана.

Если при подготовке к решению задачи произвести перенумерацию переменных  $x$  так, чтобы исходное допустимое базисное решение было обобщенно положительным, а во всех итерациях разрешающую строку  $i_0$  брать по правилу (9), то заикливание симплексного метода невозможно, и задача будет решена за конечное число шагов. Так как по доказанной теореме каждая из допустимых базисных решений будет обобщено положительным. Если существует  $\Delta_{j_0} > 0$  (при нахождении минимума) и  $a_{ij_0} > 0$  для некоторого  $i$ , то при переходе к следующему допустимому базисному решению строка  $Q_{m+1}$  будет лексикографически убывать. Поэтому ни одно из допустимых базисных

решений не будет совпадать с предыдущими допустимыми базисными решениями. В силу конечности допустимых базисных решений для некоторого решения получим либо все  $\Delta_j \leq 0$ , т.е. найдено оптимальное решение, либо все  $a_{ij_0} \leq 0$ , а  $\Delta_{j_0} > 0$ , т.е. функция  $F$  неограниченна снизу. Данный способ в силу редкой встречаемости зацикливания симплексного метода разумно применять в том случае, когда встретилось зацикливание.

### *Второй метод искусственного базиса*

Пусть дана каноническая задача ЛП:

$$\sum_{j=1}^n a_{ij} \cdot x_j = b_i \quad (13)$$

$$b_i \geq 0, \quad i = \overline{1, m}$$

$$x_j \geq 0, \quad j = \overline{1, n}$$

$$F = \sum_{j=1}^n c_j x_j \rightarrow \min \quad (14)$$

Пусть (13) не имеет предпочтительный вид. Вводим искусственные переменные  $w$ , так чтобы (13) приняло предпочтительный вид. Если некоторое равенство в (13) имеет предпочтительный вид, то искусственную переменную не вводим или считаем, что она тождественно равна 0. После введения дополнительных переменных получается следующая задача:

$$\sum_{j=1}^n a_{ij} x_j + w_{n+i} = b_i \quad (15)$$

$$w_{n+i} \geq 0, \quad i = \overline{1, m}$$

$$x_j \geq 0, \quad j = \overline{1, n}$$

$$F_1 = \sum_{i=1}^m w_{i+n} \rightarrow \min \quad (16)$$

Переход от задачи (13)-(14) к задаче (15)-(16) составляет суть второго метода искусственного базиса.

*Теорема.* Пусть задача (13)-(14) имеет допустимое решение. Решение  $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n, \bar{w}_{n+1}, \dots, \bar{w}_{n+m})$  является оптимальным для задачи (15)-(16) тогда и только тогда, когда все искусственные переменные  $\bar{w}_{n+i}$  ( $i = \overline{1, m}$ ) равны 0.

*Доказательство:*

Пусть  $\bar{x}$  - оптимальное решение задачи (15)-(16), а решение  $x = (x_1, \dots, x_n)$  допустимое решение задачи (13)-(14). Тогда  $(n+m)$ -мерный вектор  $x' = (x_1, \dots, x_n, 0, \dots, 0)$  является допустимым решением задачи (15)-(16). Коэффициенты функции  $F_1$  образуют следующий вектор:  $c = (0, \dots, 0, 1, \dots, 1)$ .

Найдем значение функции  $F_1$  на оптимальном решении  $\bar{x}$  и учитывая, что задача на  $\min$ , а  $x'$  - допустимое решение задачи (15)-(16):

$$c^T \cdot \bar{x} = \sum_{i=1}^m \bar{w}_{n+i} \leq c^T \cdot x' = 0 \quad (17)$$

Так как  $\bar{w}_{n+i} \geq 0$  и учитывая (17), получается:

$$\bar{w}_{n+i} = 0 \quad \forall i = \overline{1, m}.$$

Пусть для решения  $\bar{x}$  задачи (15)-(16) выполняется условие  $w_{n+i} = 0 \quad \forall i = \overline{1, m}$ .

Покажем, что оно оптимальное для задачи (15)-(16).

Учитывая равенство нулю искусственных переменных, имеем  $c^T \cdot \bar{x} = 0$ .

Решение  $x' = (x'_1, \dots, x'_n, w'_{n+1}, \dots, w'_{n+m})$  есть любое допустимое решение задачи (15)-(16). В силу его допустимости имеем:  $c^T \cdot x' = \sum_{i=1}^m w'_{n+i} \geq 0$

Получается, что  $c^T \cdot x' \geq c^T \cdot \bar{x}$ . Откуда следует, что  $\bar{x}$  является оптимальным решением. Теорема доказана.

Теорема доказывает равенство нулю всех искусственных переменных в оптимальном решении задачи (15)-(16)  $\bar{x}$ . Это означает, что решение  $x = (\bar{x}_1, \dots, \bar{x}_n)$  является допустимым решением задачи (13)-(14). Но для задачи (13)-(14) оптимальным не является, как это имело место в первом методе искусственного базиса.

Предположим, что оптимальное решение  $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n, \bar{w}_{n+1}, \dots, \bar{w}_{n+m})$  задачи (15)-(16) было найдено симплексным методом. Тогда ненулевым компонентам  $\bar{x}$  соответствуют линейно независимые векторы, являющиеся единичными  $A_j$ .

Тогда ненулевыми компонентами из теоремы могут оказаться только  $x_j$  ( $j = \overline{1, n}$ ) и, следовательно, решению  $\bar{x}$  соответствуют линейно независимые векторы из системы  $A_j$  ( $j = \overline{1, n}$ ). Если оптимальное решение  $\bar{x}$  не вырождено, то число линейно независимых векторов, соответствующих  $\bar{x}$ , равно  $m$ , т.е. они составляют базис. Этот базис и соответствующие ему переменные  $x_1, \dots, x_n$  принимаются в качестве первоначального решения задачи (13)-(14).

*Алгоритм второго метода искусственного базиса*

1. Составляется задача (15)-(16).
2. Задача (15)-(16) решается симплексным методом, находится  $\bar{x}$ .
3. Если  $\bar{x}$  невырожден, то в последней симплексной таблице вычеркиваются столбцы, соответствующие искусственным переменным, пересчитываются  $\Delta_j$ . Это будет исходная симплексная таблица без любых искусственных переменных для задачи (13)-(14).

*Случай вырожденного оптимального решения во втором методе искусственного базиса*

Второй метод видоизменяется, если оптимальное решение задачи (15)-(16) является вырожденным и в этом решении в базисных переменных присутствуют искусственные.

Эти переменные необходимо вывести из базисных. Пусть:

$$A_{i_1}, \dots, A_{i_m} \tag{18}$$

- это заключительный базис, полученный в последней симплексной таблице, при этом вектор  $A_{i_l}$ ,  $1 \leq l \leq m$ , является искусственным вектором.

В (18) эти векторы являются единичными векторами, т.к. (18) – базис, все остальные векторы могут быть выражены через них:

$$A_j = \sum_{k \neq l}^m a_{i_k j} A_{i_k} + a_{i_l j} A_{i_l} \quad (19)$$

$$j = \overline{0, n} \quad A_0 = B$$

Замечаем, что  $a_{i_k 0} \geq 0 \quad \forall k = \overline{1, n}$ , т.к. базису (18) соответствует допустимое решение и по доказанной теореме:  $a_{i_l 0} = 0$

Среди индексов  $j = \overline{1, n}$  найдется такой индекс, что  $a_{i_l j} \neq 0$ , иначе все векторы  $A_j \quad (j = \overline{1, n})$  выражались бы через векторы:  $A_{i_1}, \dots, A_{i_{l-1}}, A_{i_{l+1}}, \dots, A_{i_m}$  что невозможно, так как ранг по условию равен  $m$ . Тогда векторы

$$A_{i_1}, \dots, A_{i_{l-1}}, A_j, A_{i_{l+1}}, \dots, A_{i_m} \quad (20)$$

образуют базис.

Запишем равенство (19) при  $j=0$ :

$$A_0 = \sum_{k \neq l}^m a_{i_k 0} A_{i_k} \quad (21)$$

Векторы  $A_{i_k}$ , где  $k = 1, \dots, l-1, l+1, \dots, m$  принадлежат базису (20). Следовательно, (21) представляет собой разложение вектора  $A_0$  не только по базису (18), но и по базису (20). Это значит, что базису (20) соответствует базисное решение, совпадающее со старым базисным решением. Если при этом искусственных переменных в базисе (20) нет, то (20) используется для решения исходной задачи. Если есть, то (20) обозначаем за (18) и проделываем все по-новому.

## ПРОГРАММНЫЕ СРЕДСТВА РЕШЕНИЯ ЗАДАЧ СИМПЛЕКС-МЕТОДОМ

Необходимо автоматизировать процесс решения задачи симплекс-методом на компьютере. На сегодняшний день существует много программных средств для решения задач линейного программирования, написанные на разных языках программирования, для разных операционных систем. Программы в основном написаны студентами ВУЗов.

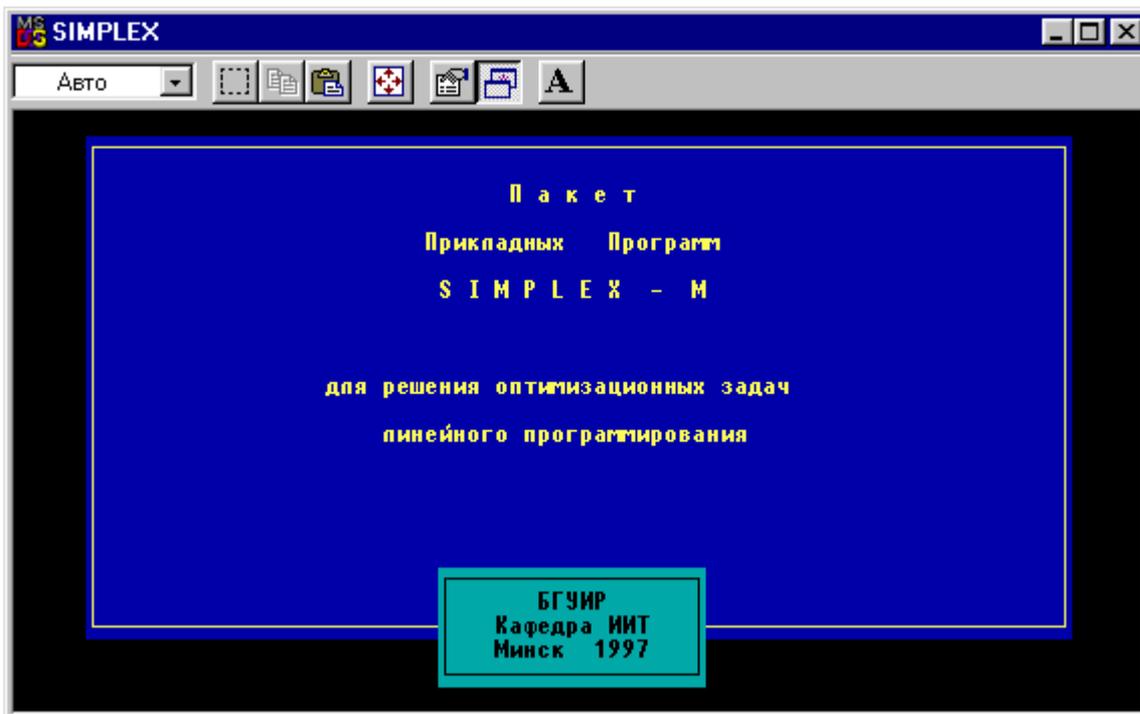
Остановимся на некоторых из них, написанных для операционной системы MS-Dos на примере решения следующей задачи:

**Задача.** Для изготовления двух видов продукции  $P_1$ ,  $P_2$  используют 4 вида ресурсов:  $S_1$ ,  $S_2$ ,  $S_3$ ,  $S_4$ . Запасы ресурсов и число единиц ресурсов, затрачиваемых на изготовление одного вида продукции приведено в таблице:

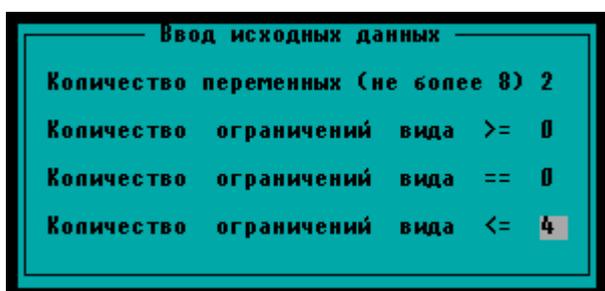
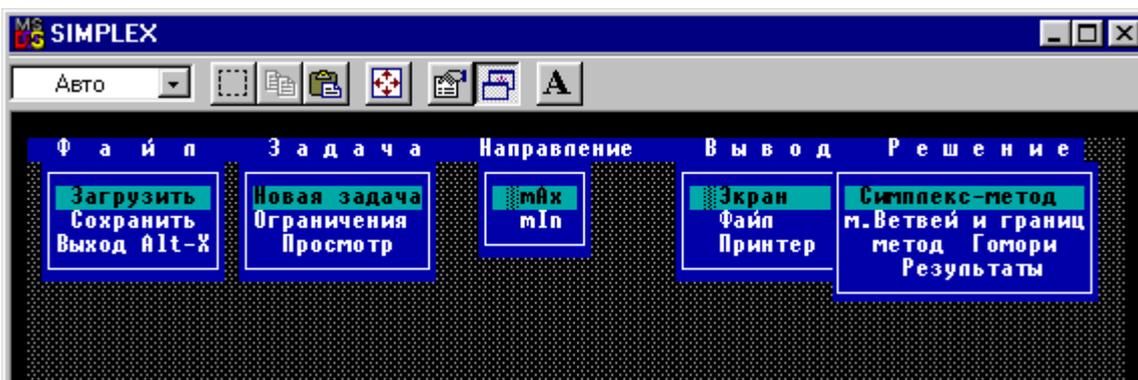
Вид ресурса	Запас ресурсов	Число единиц ресурсов	
		$P_1$	$P_2$
$S_1$	18	1	3
$S_2$	16	2	1
$S_3$	5	-	1
$S_4$	21	3	-

Прибыль, получаемая от единицы продукции соответственно равно 2 и 3 рубля. Необходимо составить такой план производства продукции, при котором прибыль от реализации будет максимальной.

# 1. Программа Simplex-M, БГУИР, кафедра ИИТ, 1997 г.



В программе интерфейс организован в виде меню. Имеется возможность ввода вручную начальных данных, а так же есть возможность сохранить и прочитать эти данные в файле.



целевой функции и уравнений.

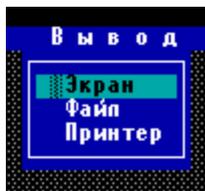
Программа работает с системой уравнений и неравенств. Сначала запрашивается количество переменных, уравнений и неравенств.

Затем вводятся коэффициенты

Можно указать на каком направлении нужно искать оптимальное решение (максимум или минимум).

	Ввод данных		Значен.
	X1	X2	
2	3	E	
1	3	<=	18
2	1	<=	16
0	1	<=	5
3	0	<=	21

Данные можно вывести на экран, на файл или на принтер.



Когда все входные данные введены, можно перейти к итерациям в симплекс таблице, в ходе чего программа выводит промежуточные результаты симплекс таблицы и указывает ведущий элемент.

**SIMPLEX**

Авто

Вывод промежуточных результатов

И	БП	X1	X2	X3	X4	X5	X6	БазРеш
1	E	-2	-3	0	0	0	0	0
	X3	1	3	1	0	0	0	18
	X4	2	1	0	1	0	0	16
	X5	0	0	0	0	1	0	5
	X6	3	0	0	0	0	1	21

ВЕДУЩИЙ ЭЛЕМЕНТ НАХОДИТСЯ  
В 2 СТОЛБЦЕ И 3 СТРОКЕ.

Для продолжения нажмите любую клавишу. ESC - выход.

Симплекс-ме

Вывод промежуточных результатов

И	БП	X1	X2	X3	X4	X5	X6	БазРеш
2	E	-2	0	0	0	3	0	15
	X3	1	0	1	0	-3	0	3
	X4	2	0	0	1	-1	0	11
	X2	0	1	0	0	1	0	5
	X6	3	0	0	0	0	1	21

ВЕДУЩИЙ ЭЛЕМЕНТ НАХОДИТСЯ  
В 1 СТОЛБЦЕ И 1 СТРОКЕ.

Вывод промежуточных результатов

И	БП	X1	X2	X3	X4	X5	X6	БазРеш
3	E	0	0	2	0	-3	0	21
	X1	1	0	1	0	-3	0	3
	X4	0	0	-2	1	5	0	5
	X2	0	1	0	0	1	0	5
	X6	0	0	-3	0	9	1	12

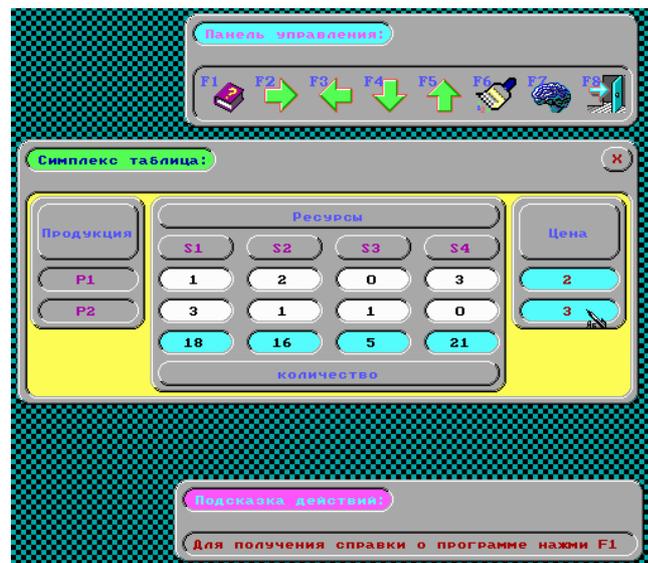
ВЕДУЩИЙ ЭЛЕМЕНТ НАХОДИТСЯ  
В 5 СТОЛБЦЕ И 2 СТРОКЕ.

Вывод промежуточных результатов								
И	БП	X1	X2	X3	X4	X5	X6	БазРеш
4	E	0	0	0.8	0.6	0	0	24
	X1	1	0	-0.2	0.6	0	0	6
	X5	0	0	-0.4	0.2	1	0	1
	X2	0	1	0.4	-0.2	0	0	4
	X6	0	0	0.6	-1.8	0	1	3

Результаты оптимизации	
Базис	Значение
X1	6
X5	1
X2	4
X6	3

Максимум функции равен 24

## 2. Программа Simplex, С-ППТ, Жердин Д.А. (5 курс), 2002 г.



В данной программе вполне заманчивый графический интерфейс, управление через мышку. Есть возможность подписывать симплекс таблицу.

Сразу можно заметить, что программа написана скорее не для использования ее при вычислениях, для демонстрации данного метода решения задачи линейного программирования, так как нет возможности указания направления вычислений (максимум или минимум), не учтены возможности работы с системой уравнений или неравенств со знаком « $\geq$ ».

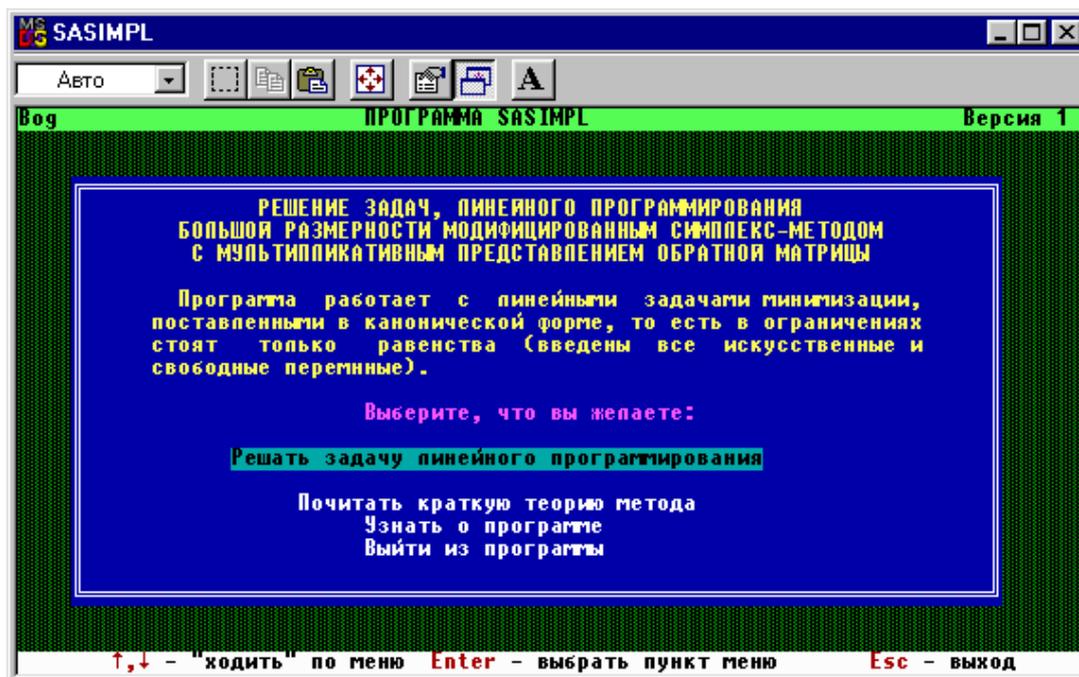


а

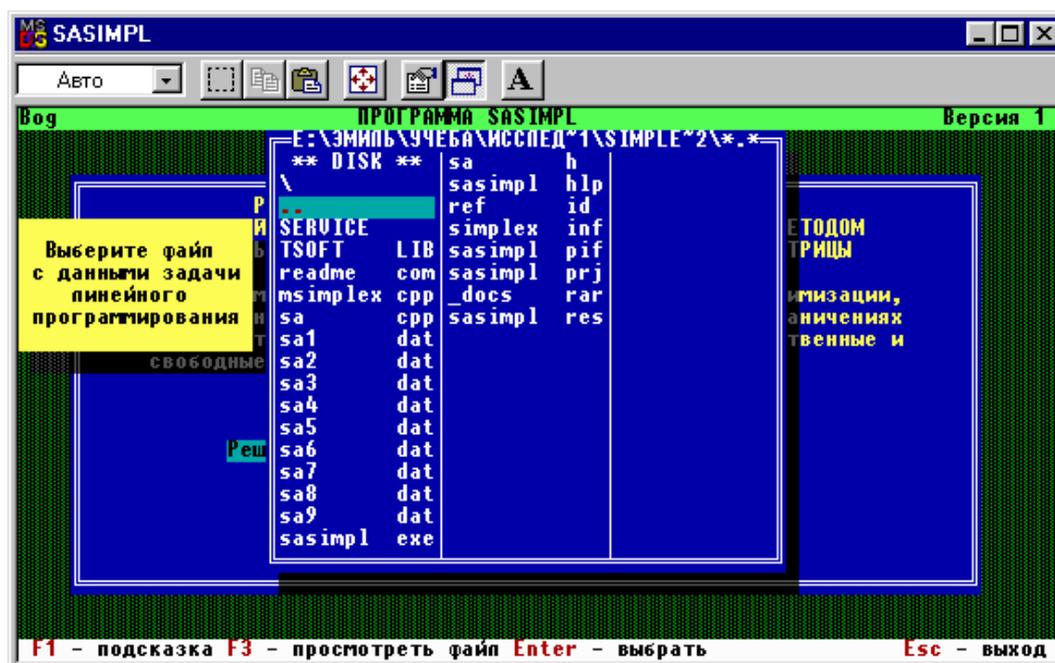
Еще нами было выявлено, что при вычислении в симплекс-таблице содержащем ноль (данный пример), программа «зависает».

### 3. Программа SASIMPL - ДВГТУ Богдановский А. (4 курс), 1996 г.

Программа SASIMPL имеет примитивный пользовательский интерфейс, который позволяет загружать данные о задачах из внешних файлов, решать их и просматривать результаты. Кроме того, программа предоставляет скромную справочную систему для удобства работы.



Программа работает лишь с входными данными в файле, что на наш взгляд не совсем удобно.



Следующим недостатком является то, что программа решает только модифицированным симплекс-методом, т.е. количество переменных больше количества ограничений и целевая функция обязательно минимизируется. Если вам нужно решить задачу, где например, функция минимизируется, или количество переменных не превышает ограничений, то придется «вручную» приводить ее (задачу) к нужному виду. Нам тоже пришлось в целевой функции дописывать минусы перед коэффициентами. И в ответ мы тоже получили, конечно, отрицательное число.

Хотя отмечается, почему автор выбрал именно модифицированный симплекс-метод:

При решении задач линейного программирования, в которых  $n$  (количество переменных) существенно больше  $m$  (количество ограничений), модифицированный симплекс-метод требует по сравнению с другими значительно меньшего количества вычислительных операций и объема памяти ЭВМ.

В модифицированном симплекс-методе реализуется та же основная идея, что и в обычном симплекс-методе, но здесь на каждой итерации пере-

считывается не вся матрица  $A_{.l}$ , обратная матрице ограничений  $A$ , а лишь та часть, которая относится к текущему базису  $A_x$ .

Ответ выводится в одно окно с возможностью перекрутки, выводятся все итерации, правда сама таблица не выводится, а выводятся лишь значения переменных и целевой функции:

```
===== Результаты вычислений =====
*****
Итерация N0
Значения базисных переменных:
X3 = 18
X4 = 16
X5 = 5
X6 = 21
Значение целевой функции: 0.000

Итерация N1
Значения базисных переменных:
X3 = 3
X4 = 11
X2 = 5
X6 = 21
Значение целевой функции: -15.000
===== Esc, ↑, PgUp, PgDn, Home, End =====
```

```
===== Результаты вычислений =====
Итерация N2
Значения базисных переменных:
X1 = 3
X4 = 5
X2 = 5
X6 = 12
Значение целевой функции: -21.000

Итерация N3
Значения базисных переменных:
X1 = 6
X5 = 1
X2 = 4
X6 = 3
Значение целевой функции: -24.000
===== Esc, ↑, PgUp, PgDn, Home, End =====
```

```
===== Результаты вычислений =====
X1 = 3
X4 = 5
X2 = 5
X6 = 12
Значение целевой функции: -21.000

Итерация N3
Значения базисных переменных:
X1 = 6
X5 = 1
X2 = 4
X6 = 3
Значение целевой функции: -24.000

Данное решение является оптимальным!
===== Esc, ↑, PgUp, PgDn, Home, End =====
```

#### 4. Руководство по использованию программы Simplexwin 3.1

Данная программа предназначена для решения задач линейного программирования симплексным методом. Программа создана Вартановым Сергеем (vartserg@mail.ru) и выложена для бесплатного пользования на ресурсе [www.simplexwin.narod.ru](http://www.simplexwin.narod.ru).

Входные данные можно загружать из файла и вводить на экране. Первый способ нужен, если необходимо вводить большую матрицу (максимальный размер входных данных - 100 уравнений и 100 переменных.).

Для того чтобы загрузить файл с исходными данными, он должен иметь определенный формат:

- ✓ первая строка содержит два числа через пробел - количество уравнений и количество переменных;
- ✓ следующие строки содержат коэффициенты уравнений, знак и правую часть (свободный член) через пробелы;
- ✓ далее следует строка, содержащая тип оптимизационной задачи (*max* или *min*);
- ✓ предпоследняя строка содержит коэффициенты целевой функции (тоже через пробелы);
- ✓ последняя строка содержит псевдонимы переменных и правых частей, разделенные пробелами.

Загружаем очень просто: надо выбрать из меню **Файл** пункт **Открыть**.

Если требуется сохранить матрицу, надо заполнить матрицу значениям и выбрать из меню **Файл** пункт **Сохранить**.

Ввод задачи на экране. Сначала необходимо установить требуемый размер матрицы ограничений (количество уравнений и количество переменных): из меню **Настройки** выбираем пункт **Размер матрицы**.

Если это необходимо, можно ввести псевдонимы для переменных и правых частей уравнений. Для этого добавлен редактор подписей столбцов, вызываемый соответствующей командой из меню **Настройки**.

Исходные данные - целые числа. Вводим коэффициенты элементов в первое поле, выбираем знак ( $\geq$ ,  $\leq$ ,  $=$ ), вводим свободный член. Во втором поле определяем, максимизирована или минимизирована должна быть целевая функция, в третье поле вводим целевую функцию. Для решения задачи надо нажать кнопку **Вычислить**.

Если задача вводится не в канонической форме, то дополнительные переменные и искусственные базисы (а также соответствующие им коэффициенты целевой функции) добавляются автоматически.

Добавлена возможность составления двойственной задачи при условиях ограничений  $x_i \geq 0$ . Для этого выбираем пункт **Составить двойственную** из меню **Задача**.

Опция настроек **Вывод в Excel** позволяет в целях экономии памяти отключать сохранения таблиц в памяти для их последующего вывода в Excel.

Исходные данные теперь могут быть представлены в любом виде: целые числа, обыкновенные дроби, обыкновенные дроби с целой частью, десятичные дроби. Переключать эту опцию можно в пункте **Прочие** меню **Настройки**.

Экран результатов содержит базисы, базисный план (БП), элементы матрицы и индексную строку (ИС).

Кнопка **Результат** позволяет вычислить задачу по шагам, при этом все шаги записываются в поле **Действия**.

Кнопка **Результат** позволяет произвести решение задачи в автоматическом режиме и отобразить на экране только последнюю симплексную таблицу и результат.

Кнопка **Вручную** позволяет самому выбрать ключевой элемент. Использование этой опции целесообразно в случаях, когда стандартный алгоритм не позволяет вывести из базисов искусственный базис, а также во всех остальных ситуациях, когда необходимо выбрать ключевой элемент, отличный от стандартного.

Рассмотрена ситуация бесконечного множества оптимальных планов. В этом случае ответ выводится в виде:  $x^* = k * (x1^*) + (1-k) * (x2^*)$ , где  $k$  число, лежащее в пределах от 0 до 1, а  $x1^*$  и  $x2^*$  - оптимальные планы задачи.

Добавлена возможность вывода результата в Excel. Для этого в форме **Результаты** нужно нажать соответствующую кнопку.

## 5. Решение задач линейного программирования с помощью программы Simplexwin 3.1

Прежде чем приступить к решению задач линейного программирования на компьютере, откройте файл «ReadMe.txt» и ознакомьтесь с руководством по использованию программы.

После этого откройте программу «Simplex\_Advanced.exe».

### Примеры решения задач линейного программирования с помощью программы Simplexwin 3.1.

#### Задача 1.

Решить симплекс-методом задачу:

$$F = x_1 + 2x_2 \rightarrow \max$$

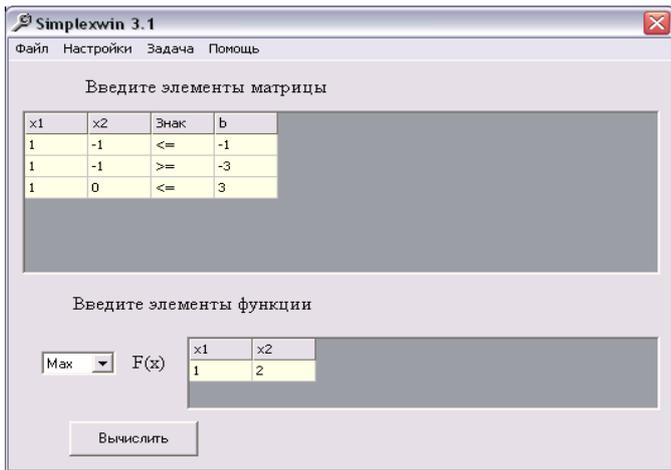
$$\begin{cases} x_1 - x_2 \leq -1 \\ x_1 - x_2 \geq -3 \\ x_1 \leq 3 \\ x_i \geq 0, i = \overline{1,2} \end{cases}$$

Решение.

Первую задачу решим, набирая ее на экране.

Решим задачу не в канонической форме, дополнительные переменные и искусственные базисы (а также соответствующие им коэффициенты целевой функции) добавятся автоматически.

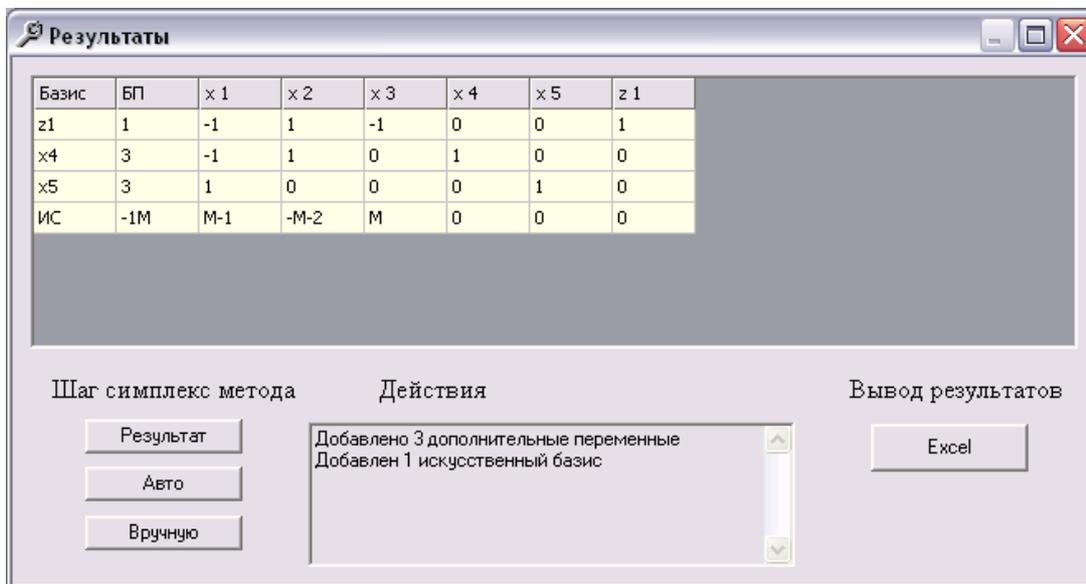
Сначала необходимо установить требуемый размер матрицы ограничений (количество уравнений - 3 и количество переменных - 2): для этого из меню **Настройки** выбираем пункт **Размер матрицы**.



Введем коэффициенты элементов в первое поле, выбираем знак ( $\geq$ ,  $\leq$ ,  $=$ ), далее - свободный член. Во втором поле определяем, что целевая функция должна быть максимизирована, в третье поле вводим целевую функцию. Для решения задачи надо нажать кнопку

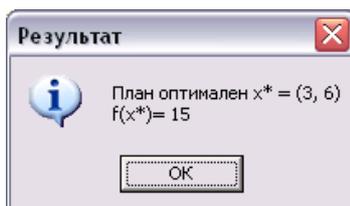
**Вычислить.**

Далее у нас выйдет новая форма:

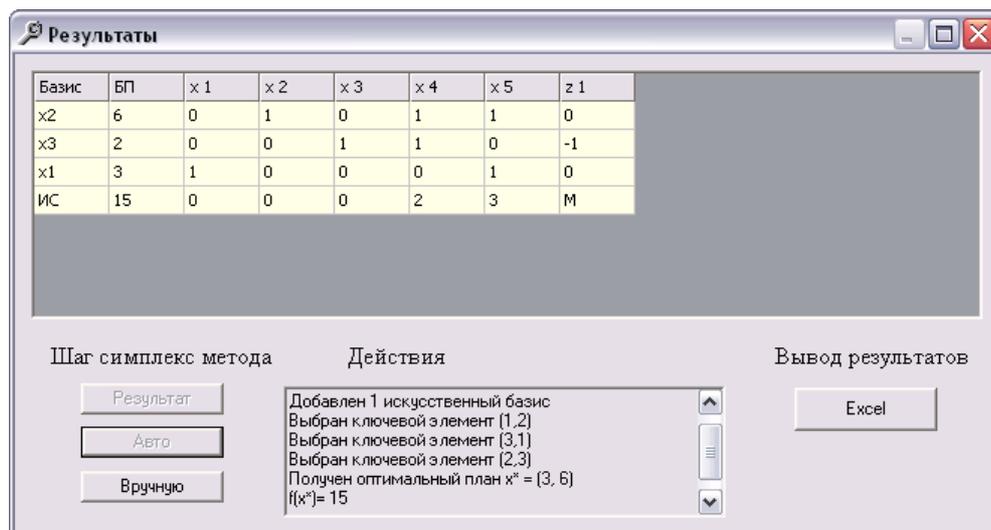


В текстовом поле **Действия** отображена служебная информация. В нашем случае там написано, что были добавлены три дополнительные переменные и один искусственный базис.

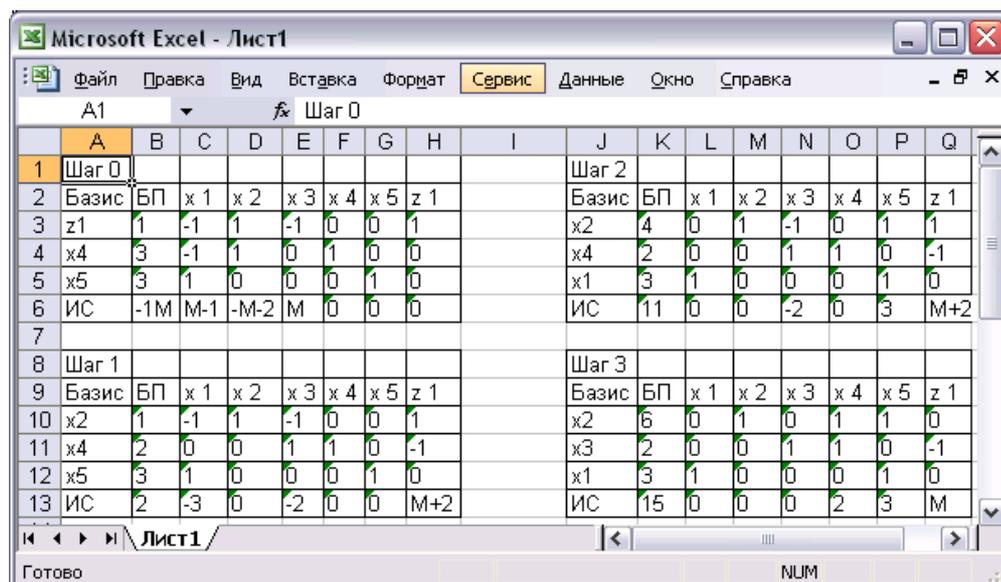
Далее нажимаем кнопку **Авто**, чтобы получить подробный план решения нашей задачи. В конце мы получим оптимальный план решения:



В текстовом поле **Действия** записаны все наши итерации (шаги решения) и оптимальное решение:



Результат вычислений можно вывести в Excel. Для этого в форме **Результаты** нужно нажать кнопку **Excel**. Внимание! Эта кнопка работает лишь в том случае, если решение задачи проводилось с помощью кнопки **Авто**.



Закроем форму **Результаты** и сохраним матрицу под именем «zadacha1.txt» с помощью пункта **Сохранить** из меню **Файл**.

## Задача 2.

Решить симплекс-методом задачу, двойственную к данной:

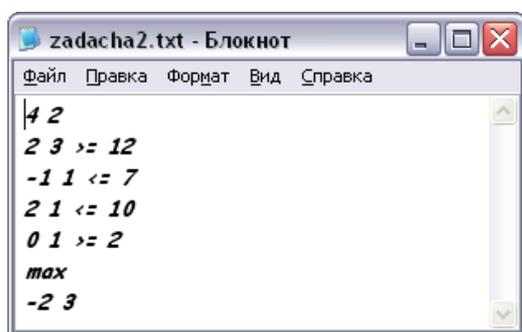
$$F = -2x_1 + 3x_2 \rightarrow \max$$

$$\begin{cases} 2x_1 + 3x_2 \geq 12 \\ -x_1 + x_2 \leq 7 \\ 2x_1 + x_2 \leq 10 \\ x_2 \geq 2 \\ x_i \geq 0, i = \overline{1,2} \end{cases}$$

Решение.

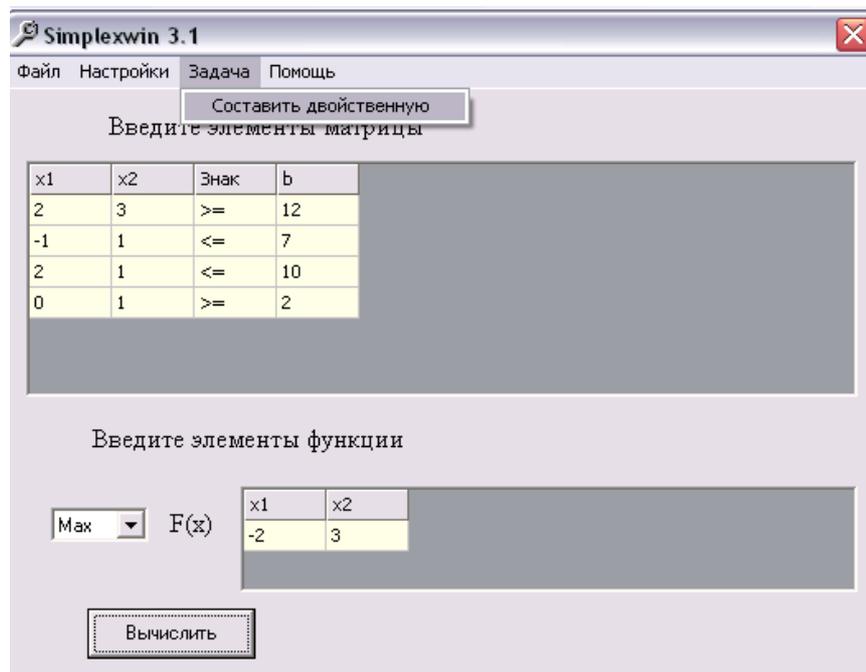
Решим эту задачу, введя данные с помощью файла.

Откроем **Блокнот** и сохраним файл в папке «Simplex» на рабочем столе под именем «zadacha2.txt». В файле напишем:

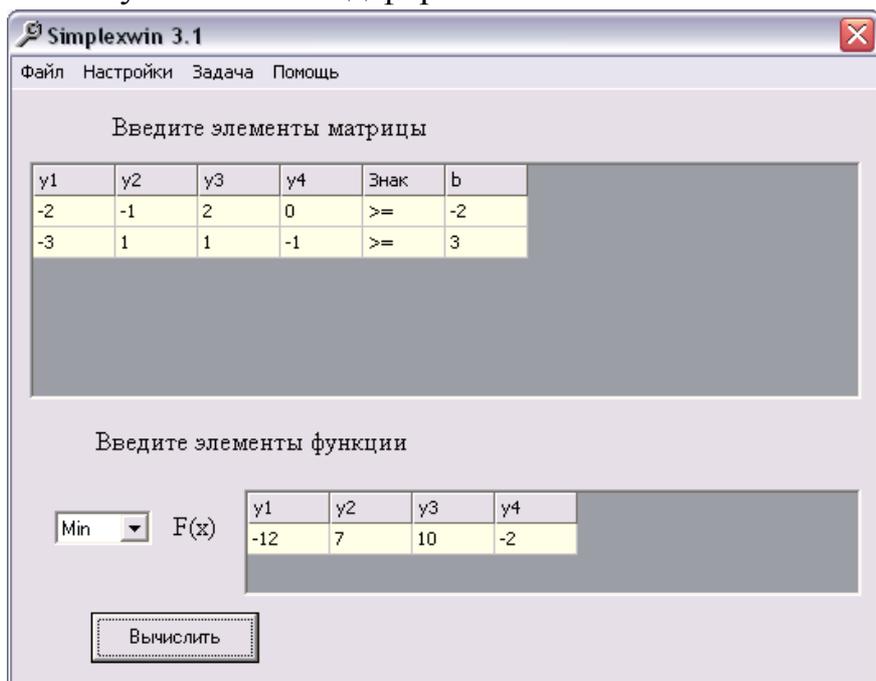


**Внимание!** Все коэффициенты необходимо вводить через пробелы.

Загружаем полученный файл: надо выбрать из меню **Файл** пункт **Открыть**. И составляем двойственную задачу: выбираем пункт **Составить двойственную** из меню **Задача**.

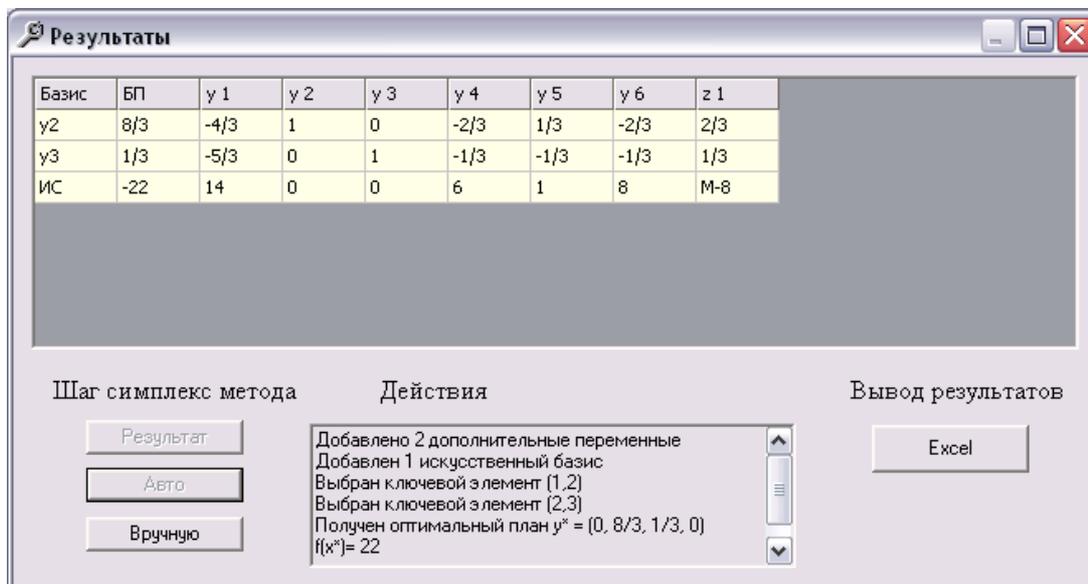


Получим такой вид формы:



Для решения задачи надо нажать кнопку **Вычислить**.

Далее нажимаем несколько раз кнопку **Авто**, чтобы получить подробный план решения нашей задачи. В конце мы получим оптимальный план решения:



Если у Вас возникнут какие-либо вопросы, можете посмотреть файлы «ReadMe.txt» или «Руководство.txt».

Если Вы испытываете трудности с введением задачи из файла, можете загрузить пробные файлы.

## Задачи для самостоятельной работы

### Задача об использовании ресурсов.

Для изготовления двух видов продукции  $p_1$  и  $p_2$  используют четыре вида ресурса  $s_1, s_2, s_3, s_4$ . Запасы ресурсов и число единиц ресурсов, затраченных на изготовление одного вида продукции, приведено в таблице.

Вид ресурса	Запас ресурса	Число единиц ресурсов, затраченных на изготовление одной единицы продукции	
		$p_1$	$p_2$
$s_1$	18	1	3
$s_2$	16	2	1
$s_3$	5	-	1
$s_4$	21	3	-

Прибыль, полученная от единицы продукции, соответственно равна: от  $p_1$  - 2 рубля, от  $p_2$  - 3 рубля. Необходимо составить такой план производства продукции, при котором прибыль от реализации была максимальной.

Матрица задачи должна выглядеть таким образом:

The screenshot shows the Simplexwin 3.1 application window. The title bar reads "Simplexwin 3.1". The menu bar includes "Файл", "Настройки", "Задача", and "Помощь". The main area is titled "Введите элементы матрицы" (Enter matrix elements). It contains a table with the following data:

x1	x2	Знак	b
1	3	<=	18
2	1	<=	16
0	1	<=	5
3	0	<=	21

Below this table is the section "Введите элементы функции" (Enter function elements). It features a dropdown menu set to "Max" and a label "F(x)". To the right is a small table with the following data:

x1	x2
2	3

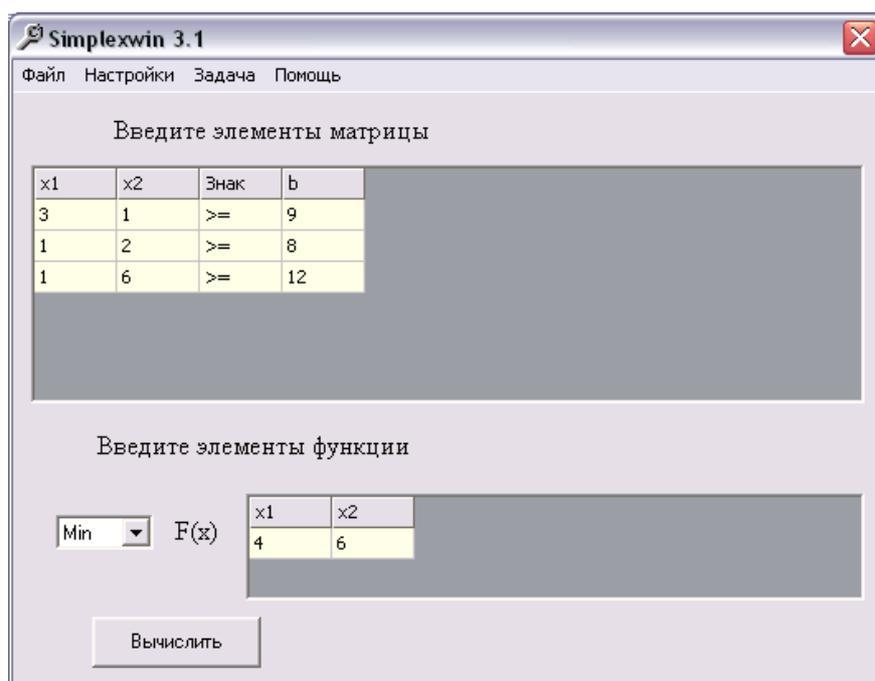
At the bottom of the window is a button labeled "Вычислить" (Calculate).



### Задача о составлении рациона (задача о диете, задача о смесях).

Имеются два вида корма, содержащие питательные вещества: витамины  $s_1, s_2, s_3$ . Стоимость одного килограмма первого вида корма – 4 рубля, второго – 6 рублей. Необходимо составить дневной рацион, имеющий минимальную стоимость, в котором содержание каждого вида питательных веществ было бы не менее установленного предела.

Питательное вещество	Необходимый минимум	Число ед-ц питательных веществ в 1 кг. корма	
		I	II
$s_1$	9	3	1
$s_2$	8	1	2
$s_3$	12	1	6



Результаты

Базис	БП	x 1	x 2	x 3	x 4	x 5	z 1	z 2	z 3
x1	2	1	0	-2/5	1/5	0	2/5	-1/5	0
x5	8	0	0	4/5	-17/5	1	-4/5	17/5	-1
x2	3	0	1	1/5	-3/5	0	-1/5	3/5	0
ИС	-26	0	0	2/5	14/5	0	M-2/5	M-14/5	M

Шаг симплекс метода      Действия      Вывод результатов

Добавлено 3 искусственных базиса  
 Выбран ключевой элемент (3,2)  
 Выбран ключевой элемент (1,1)  
 Выбран ключевой элемент (2,5)  
 Получен оптимальный план  $x^* = (2, 3)$   
 $f(x^*) = 26$

## 6. Разработка своей программы Simplex

После просмотра приведенных выше программ для решения задач линейного программирования, мы решили попробовать создать свою программу, полагаясь на выявленные недостатки и достоинства.

Для реализации программы нами был выбран язык программирования Турбо Паскаль.

Простой интерфейс, так как программа адресована на подготовленных пользователей, понимающих смысл данного метода и для упрощения кодирования программы.

В ходе анализа уже имеющих программ были выявлены недостатки, которые нужно устранить в своей программе:

1. увеличить количество начальных переменных и уравнений (неравенств) до 10;
2. программа должно работать с системой уравнений и неравенств, а не только с системой уравнений, или же неравенств со знаком «<=»;
3. Ответ должен быть выведен в файл с таблицами каждой итерации.

### Постановка задачи

Пусть задача линейного программирования дана системой линейных уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \geq b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \geq b_2 \\ \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \geq b_m \end{cases} \quad (1)$$

и целевая функция имеет вид:

$$F(x) = \alpha_1x_1 + \alpha_2x_2 + \dots + \alpha_nx_n \rightarrow \max(\min) \quad (2)$$

Пусть  $kell$  – количество переменных,  $ktr$  – количество уравнений. Эти данные введем в основной программе.

Тогда, сначала нужно внести коэффициенты  $a_{ij}$  системы (1) в массив **Xnew**(1..kstr, 1..kell). Это делается в процедуре **Simplex**.

Знаки системы будем хранить в массиве **Znac**(1..kstr).

А для свободных членов введем массив **B**(1..kstr).

Далее нужно указать в каком направлении следует искать оптимальное решение: переменная **Fm**=1, если на максимум, и **Fm**=2, в случае минимума.

Учтем в программе целочисленность решения в переменной - **PrGomory**='Y', в случае целочисленного решения.

Ответ должен быть выведен в файл **Simplex.dat**, и содержать таблицу каждой итерации хода решения задачи.

Будем использовать следующие процедуры:

**SimvB** – для того, чтобы иметь строковое представление переменных, это удобно при выводе результатов, так при вычислении симплекс-методом первоначальный набор переменных может изменяться.

**Save** – вывод шагов итерации и конечного результата в файл **Simplex.dat**.

**Dop-Per** – для определения дополнительных переменных.

**Sokr** – процедура сокращения Y.

**Gomory** – выполнение метода Гомори для поиска целочисленного решения задачи.

**Simplex** – основная процедура программы. В ней происходит ввод коэффициентов системы (1), знаков, целевой функции и собственно итерации симплекс-метода, использует предыдущие вспомогательные процедуры.

## Ход решения задачи

Рассмотрим более подробно некоторые процедуры программы.

### Процедура **Dop\_Per**

Нужна для определения дополнительных переменных после ввода коэффициентов системы (1) и целевой функции. Работает с отдельным

уравнением (или неравенством). Порядковый номер уравнения (неравенства) передается в процедуру с помощью переменной **П1**.

Процедуру можно разделить на три основные части в зависимости от знака:

1. Знак уравнения (неравенства) с порядковым номером **П1** « $\Rightarrow$ ».

Количество переменных увеличивается на 1, дается имя этой переменной **У** (со следующим индексом **DPy**). Эта переменная добавляется в **П1**-ое уравнение с коэффициентом 1, а в остальные уравнения с коэффициентом 0. Переменная добавляется к целевой функции с коэффициентом  $-1$  (случай максимизации) или 1 (случай минимизации).

2. Знак уравнения (неравенства) с порядковым номером **П1** « $\geq$ »

Количество переменных увеличивается на 1, дается имя этой переменной **X** (со следующим индексом **DPx**). Эта переменная добавляется в **П1**-ое неравенство с коэффициентом  $-1$ , а в остальные неравенства с коэффициентом 0. Переменная добавляется к целевой функции с коэффициентом 0.

Далее добавляется переменная **У** с условиями как в пункте 1.

3. Знак уравнения (неравенства) с порядковым номером **П1** « $\leq$ »

Количество переменных увеличивается на 1, дается имя этой переменной **X** (со следующим индексом **DPx**). Эта переменная добавляется в **П1**-ое неравенство с коэффициентом 1, а в остальные неравенства с коэффициентом 0. Переменная добавляется к целевой функции с коэффициентом 0.

### **Процедура Simplex.**

Здесь вводятся сама система уравнений (неравенств) (1), коэффициенты целевой функции.

Далее идет индексация переменных **X**, т.е. копирование их в дополнительный строковой массив с индексами. Это удобно при выводе результатов. Таким образом мы имеем готовый набор строковые данные с названиями переменных с индексами.

Затем запускается процедура **Dop\_Per**, для объявления дополнительных переменных.

После чего идет замена оптимальной функции с максимизации на минимизацию при наличии в базисе  $Y$ -ков.

Далее нужно отсортировать дополнительные переменные по индексу.

Нам нужны будут дополнительные столбцы в симплекс таблице:

С – для хранения значений свободных членов; (**Cnew**)

Б – для хранения списка базисных переменных; (**BS**)

Н – для хранения коэффициентов перед базисными переменными в целевой функции. (**CPNew**)

На этом этапе все начальные данные подготовлены для начала основной итерации симплекс-таблицы. Итерация объявлена как цикл с постусловием (**REPEAT..UNTIL**) с постоянно не выполняющимся условием. То есть бесконечный цикл. Для завершения программы внутри цикла имеются операторы **HALT** при нахождении оптимального решения или же при нахождении факта, что оно не существует.

Первым делом надо вывести симплекс-таблицу в том виде, в каком оно находится на данный момент, для чего вызывается процедура **Save**. Затем вычисляются значения в индексной строке, для подсчета и проверки значения целевой функции на оптимальность. И в случае выполнения данного условия, вывести результаты о найденном оптимальном решении задачи или же о его отсутствии.

В случае не выполнения условия оптимальности, идет поиск ключевого столбца и запись его порядкового номера в переменную **KLst**, и ключевой строки **KLstr**. Если нельзя найти ключевую строку, происходит завершение программы, т.к. не может быть найдено оптимальное решение.

Затем происходит построение следующей симплекс-таблицы и переход к следующей итерации.

# ДИНАМИЧЕСКОЕ ПРОГРАММИРОВАНИЕ

## Многошаговые процессы в динамических задачах

*Динамическое программирование (планирование)* представляет собой математический метод для нахождения оптимальных решений многошаговых (многоэтапных) задач. Некоторые из таких задач естественным образом распадаются на отдельные шаги (этапы), но имеются задачи, в которых разбиение приходится вводить искусственно, для того чтобы их можно было решить методом динамического программирования.

Пусть на некоторый период времени  $T$ , состоящий из  $t$  лет, планируется деятельность группы промышленных предприятий. В начале планируемого периода на развитие предприятий выделяются основные средства  $Q_0$ , которые необходимо распределить между предприятиями. В процессе функционирования предприятий выделенные им средства расходуются. Однако каждое из этих предприятий за определенный период времени (хозяйственный год) получает прибыль, зависящую от объема вложенных средств. В начале каждого года имеющиеся средства могут перераспределяться между предприятиями. Требуется определить, сколько средств надо выделить каждому предприятию в начале каждого года, чтобы суммарный доход от всей группы предприятий за весь период времени  $T$  был максимальным.

Процесс решения такой задачи является многошаговым. Шаг управления (планирования) здесь - хозяйственный год. Управление процессом состоит в распределении (перераспределении) средств в начале каждого хозяйственного года.

Рассмотрим другую задачу. Пусть имеется груз, состоящий из неделимых предметов различных типов, который нужно погрузить в самолет грузоподъемностью  $P$ . Стоимость и вес каждого предмета  $j$ -го типа известны и составляют соответственно  $c_j$  и  $p_j$  ( $j=\overline{1,n}$ ) единиц. Требуется определить, сколько предметов каждого типа надо загрузить в самолет, чтобы суммарная

стоимость груза была наибольшей, а вес не превышал грузоподъемности самолета.

Математически задача записывается следующим образом: найти такие целые неотрицательные значения  $x_j$  ( $j=1, n$ ), которые бы максимизировали функцию

$$f(x) = \sum_{j=1}^n c_j x_j$$

при ограничении

$$\sum_{j=1}^n p_j x_j \leq P$$

где  $x_j$  - количество груза  $j$ -го типа, позволяющее достичь  $\max f(x)$ .

Процесс решения сформулированной задачи разобьем на этапы: на первом этапе рассмотрим всевозможные варианты загрузки самолета предметами первого типа и среди них найдем оптимальный. На втором этапе определим оптимальный вариант загрузки самолета предметами первого и второго типов и т.д. Процесс решения задачи продолжается до тех пор, пока не будет найден оптимальный вариант загрузки самолета предметами  $n$  типов.

### **Принцип оптимальности и рекуррентные соотношения**

Метод динамического программирования позволяет одну задачу со многими переменными заменить рядом последовательно решаемых задач с меньшим числом переменных. Процесс решения задачи разбивается на шаги. При этом нумерация шагов, как правило, осуществляется от конца к началу.

Основным принципом, на котором базируется оптимизация многошагового процесса, является принцип оптимальности Р. Беллмана. Этот принцип выделяет класс задач, для которых применим вычислительный метод динамического программирования.

Принцип оптимальности. Оптимальное поведение в задаче должно обладать тем свойством, что каковы бы ни были начальное состояние и начальное решение, последующие решения должны составлять оптимальное

поведение относительно состояния, полученного в результате первоначального решения.

Принцип оптимальности непосредственно указывает процедуру нахождения оптимального решения. Математически он записывается выражением вида:

$$f_{n-1}(S_l) = \underset{U_{l+1}}{\text{optimum}} [R_{l+1}(S_l, U_{l+1}) + f_{n-(l+1)}(S_{l+1})], \quad l = \overline{0, n-1} \quad (1)$$

где  $U_l = (u_l^{(1)}, \dots, u_l^{(m)})$  - решение (управление), выбранное на  $l$ -м шаге;

$S_l = (s_l^{(1)}, \dots, s_l^{(m)})$  - состояние системы на  $l$ -м шаге;

$R_l$  - непосредственный эффект, достигаемый на  $l$ -м шаге;

$f_{n-1}$  - оптимальное значение эффекта, достигаемого за  $n-1$  шагов;

$n$  – количество шагов (этапов).

«Optimum» в выражении (1) означает максимум или минимум в зависимости от условия задачи.

Все вычисления, дающие возможность найти оптимальное значение эффекта, достигаемого за  $n$  шагов,  $f_n(S_0)$ , проводятся по формуле (1), которая носит название *основного уравнения Беллмана или рекуррентного соотношения*. Действительно, при вычислении очередного значения функции  $f_{n-1}$  используются значение функции  $f_{n-(l+1)}$ , полученное на предыдущем шаге, и непосредственное значение эффекта  $R_{l+1}(S_l, U_{l+1})$ , достигаемого в результате выбора решения  $U_{l+1}$  при заданном состоянии системы  $S_l$ . Процесс вычисления значений функции  $f_{n-1}(l=0, n-1)$  осуществляется при естественном начальном условии  $f_0(S_n) = 0$ , которое означает, что за пределами конечного состояния системы эффект равен нулю.

Рассмотрим применение рекуррентного соотношения Беллмана на приведенном примере загрузки самолета.

Как уже было отмечено, процесс решения задачи начинаем с загрузки самолета предметами первого типа ( $n=1$ ). Обозначив максимальную стоимость груза  $f_1(P)$ , нетрудно определить, что  $f_1(P) = \max\{c_1, x_1\}$  при условиях  $p_1 x_1 \leq P$ ,  $x_1 = 0, 1, 2, \dots, [P/p_1]$ , где  $[P/p_1]$  – наибольшее целое число, не превосходящее  $P/p_1$ .

Тогда максимальная стоимость груза

$$f_1(P) = [P/p_1] \cdot c_1$$

Определим теперь максимальную стоимость груза (обозначим ее  $f_2(P)$ ) при загрузке самолета предметами первого и второго типов, т.е. для  $n=2$ .

Если загрузить самолет предметами второго типа в количестве  $x_2$ , то по весу предметов первого типа можно взять не больше чем  $P-p_2x_2$ , а максимальная стоимость их будет равна  $f_1(P-p_2x_2)$ . Тогда максимальная стоимость груза из предметов первого и второго типов определится как сумма стоимости предметов второго типа для всех возможных вариантов значений  $x_2$  и стоимости предметов первого типа  $f_1(P-p_2x_2)$ , т.е.

$$f_2(P) = \max_{0 \leq x_2 \leq [P/p_2]} \{c_2x_2 + f_1(P - p_2x_2)\}$$

Рекуррентное соотношение для загрузки самолета предметами трех типов ( $n=3$ ) запишется как сумма стоимости предметов третьего типа и стоимости предметов первых двух типов  $f_2(P-p_3x_3)$ . Таким образом,

$$f_3(P) = \max_{0 \leq x_3 \leq [P/p_3]} \{c_3x_3 + f_2(P - p_3x_3)\}$$

Общее рекуррентное соотношение для любого конечного значения  $n$  запишется так:

$$f_n(P) = \max_{0 \leq x_n \leq [P/p_n]} \{c_nx_n + f_{n-1}(P - p_nx_n)\}$$

## Вычислительная схема

Оптимальное решение задачи методом динамического программирования находится на основе уравнения (1). Чтобы определить его, необходимо:

1) записать уравнение для последнего состояния процесса (ему соответствует  $l=n-1$ ):

$$f_1(S_{n-1}) = \underset{U_n}{\text{optimum}}[R_n(S_{n-1}, U_n) + f_0(S_n)]$$

2) найти  $R_n(S_{n-1}, U_n)$  из дискретного набора его значений при некоторых фиксированных  $S_{n-1}$  и  $U_n$  из соответствующих допустимых областей (так как  $f_0(S_0)=0$ , то  $f_1(S_{n-1}) = \underset{U_n}{\text{optimum}}[R_n(S_{n-1}, U_n)]$ ). В результате после первого шага известно решение  $U_n$  и соответствующее значение функции  $f_1(S_{n-1})$ ;

3) уменьшить значение  $l$  на единицу записать соответствующее функциональное уравнение. При  $l=n-k$  ( $k=2, n$ ) оно имеет вид

$$f_k(S_{n-k}) = \underset{U_{n-k+1}}{\text{optimum}}[R_{n-k+1}(S_{n-k}, U_{n-k+1}) + f_{k-1}(S_{n-k+1})] \quad (2)$$

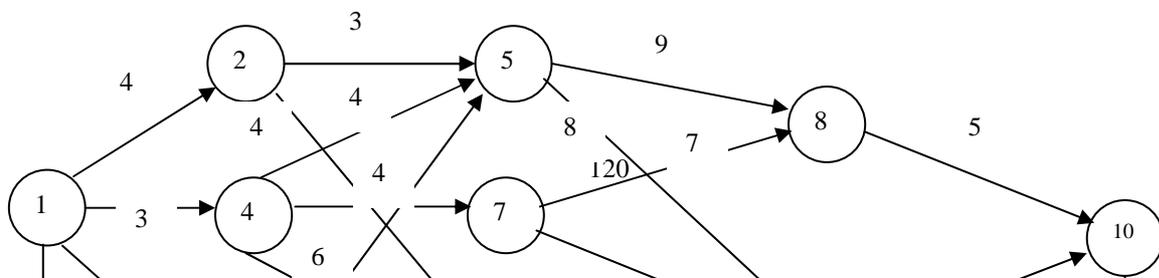
4) найти условно-оптимальное решение на основе выражения (2);

5) проверить, чему равно значение  $l$ . Если  $l=0$ , расчет условно-оптимальных решений закончен, при этом найдено оптимальное решение задачи для первого состояния процесса. Если  $l \neq 0$ , перейти к выполнению п.3;

б) вычислить оптимальное решение задачи для каждого последующего шага процесса, двигаясь от конца расчетов к началу.

**Пример.** Требуется перевезти груз из города **A** в город **B**. Сеть дорог, связывающих эти города, изображена на рис. Стоимость перевозки груза из города  $s$  ( $s=1, \dots, 9$ ) в город  $j$  ( $j=2, \dots, 10$ ) проставлена над соответствующими дугами сети. Необходимо найти маршрут, связывающий города **A** и **B**, для которого суммарные затраты на перевозку груза были бы наименьшими.

На рис. городам поставлены в соответствие вершины сети, обозначенные кружками с номером, а транспортным магистралям — дуги (линии со стрелками), соединяющие вершины.



Разобьем все множество вершин (городов) на подмножества. В первое подмножество включим исходную вершину 1. Во второе - вершины, в которые входят дуги, выходящие из вершины 1. В третье - вершины, в которые входят дуги, выходящие из вершин второго подмножества. Таким образом, продолжая разбиение дальше, получим пять подмножеств: {1}, {2,3,4}, {5,6,7}, {8,9}, {10}. Очевидно, что любой маршрут из города 1 в город 10 содержит ровно четыре дуги, каждая из которых связывает вершины, принадлежащие соответствующим подмножествам. Следовательно, процесс решения задачи (нахождения оптимального маршрута) разбивается на четыре этапа. На первом этапе принимается решение, через какой город, принадлежащий второму подмножеству, везти груз из города 1. На втором этапе необходимо определить, через какой город третьего подмножества везти груз из некоторого города, принадлежащего второму подмножеству, и т.д.

### **Применение информационных технологий**

Применение информационных технологий Excel рассмотрим на примере, для чего занесем исходные данные в память ЭВМ в следующем виде:

	A	B	C	D	E
1	<b>Исходные данные</b>				
2	1-2	4	2-7	1000	
3	1-3	11	3-7	1000	
4	1-4	3	4-7	4	
5	2-5	3	5-8	9	
6	3-5	1	6-8	1000	
7	4-5	4	7-8	7	
8	2-6	4	5-9	8	
9	3-6	6	6-9	5	
10	4-6	6	7-9	12	
11			8-10	5	
12			9-10	3	
13					

Исходные данные сформированы так, что все дуги, входящие в любую из вершин, записаны одним блоком, например: 2-5; 3-5; 4-5. Такое представление позволяет автоматизировать процесс занесения расчетных формул. Так, в табл. В ячейку В31 занесена формула =СУММ(D5;C\$22). Аналогичные формулы занесены в ячейки В32 и В33 перемещением курсора мыши. Большие расстояния (1000) приписаны в несуществующие связи между городами 2-7, 3-7, 6-8.

	A	B	C	D
13				
14		n=0	$F_0(10)=0$	
15				
16		n=1	$F_1(S) = \min_{j} [C_{sj} + F_0(J)]$	
17				
18				Таблица 6.11
19	<b>Из города</b>	<b>Расчет расстояний</b>	<b>Минимальное расстояние</b>	<b>В город</b>
20	<b>S</b>	<b>10</b>	<b>F<sub>1</sub>(s)</b>	<b>J<sub>1</sub>(s)</b>
21				
22	8	5	5	10
23	9	3	3	10
24				
25		n=2	$F_2(S) = \min_{j} [C_{sj} + F_1(J)]$	
26				
27				

	A	B	C	D
13				
14		n=0	$F_0(10)=0$	
15				
16		n=1	$F_1(S) = \min_{s,j} [C_{sj} + F_0(J)]$	
17				
18				Таблица 6.11
19	<b>Из города S</b>	<b>Расчет расстояний</b>	<b>Минимальное расстояние <math>F_1(s)</math></b>	<b>В город <math>J_1(s)</math></b>
20				
21		10		
22	8	=СУММ(D11;C\$14)	5	10
23	9	=СУММ(D12;C\$14)	3	10
24				
25		n=2	$F_2(S) = \min_{s,j} [C_{sj} + F_1(J)]$	
26				

	A	B	C	D	E	F
28	<b>Из города S</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное расстояние <math>F_2(s)</math></b>	<b>В город <math>J_2(s)</math></b>	
29						
30		8	9			
31	5	=СУММ(D5;C\$22)	=СУММ(D8;C\$23)	=МИН(B31;C31)	=ЕСЛИ(B31<C31;\$B\$30;\$C\$30)	
32	6	=СУММ(D6;C\$22)	=СУММ(D9;C\$23)	=МИН(B32;C32)	=ЕСЛИ(B32<C32;\$B\$30;\$C\$30)	
33	7	=СУММ(D7;C\$22)	=СУММ(D10;C\$23)	=МИН(B33;C33)	=ЕСЛИ(B33<C33;\$B\$30;\$C\$30)	
34						
35		n=3	$F_3(S) = \min_{s,j} [C_{sj} + F_2(J)]$			
36						
37						
38	<b>Из города S</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное расстояние <math>F_3(s)</math></b>	<b>В город <math>J_3(s)</math></b>	<b>В город <math>J_3(s)</math></b>
39						
40		8	9			
41	2	=СУММ(B5;D\$31)	=СУММ(B8;D\$32)	=СУММ(D2;D\$33)	=МИН(B41;D41)	=ЕСЛИ(И(B41<C41;B41<D41);\$B\$40;ЕСЛИ(И(C41<D41;C41<B41);C\$40;D\$41))
42	3	=СУММ(B6;D\$31)	=СУММ(B9;D\$32)	=СУММ(D3;D\$33)	=МИН(B42;D42)	=ЕСЛИ(И(B42<C42;B42<D42);\$B\$40;ЕСЛИ(И(C42<D42;C42<B42);C\$40;D\$41))
43	4	=СУММ(B7;D\$31)	=СУММ(B10;D\$32)	=СУММ(D4;D\$33)	=МИН(B43;D43)	=ЕСЛИ(И(B43<C43;B43<D43);\$B\$40;ЕСЛИ(И(C43<D43;C43<B43);C\$40;D\$41))
44						
45		n=4	$F_4(S) = \min_{s,j} [C_{sj} + F_3(J)]$			
46						
47						
48	<b>Из города S</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное расстояние <math>F_4(s)</math></b>	<b>В город <math>J_4(s)</math></b>	<b>В город <math>J_4(s)</math></b>
49						
50		8	9			
51	1	=СУММ(B2;E41)	=СУММ(B3;E42)	=СУММ(B4;E43)	=МИН(B51;C51;D51)	=ЕСЛИ(И(B51<C51;B51<D51);\$B\$50;ЕСЛИ(И(C51<B51;C51<D51);C\$50;D\$51))

25	A	B	C	D	E	F
26		n=2	$F_2(S) = \min_{s,j} [C_{sj} + F_1(J)]$			
27						
28	<b>Из</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное</b>	<b>В</b>	
29	<b>города</b>			<b>расстояние</b>	<b>город</b>	
30	<b>S</b>	<b>8</b>	<b>9</b>	<b>F<sub>2</sub>(s)</b>	<b>J<sub>2</sub>(s)</b>	
31	5	14	11	11	9	
32	6	1005	8	8	9	
33	7	12	15	12	8	
34						
35		n=3	$F_3(S) = \min_{s,j} [C_{sj} + F_2(J)]$			
36						
37						
38	<b>Из</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное</b>	<b>В</b>	<b>В</b>
39	<b>города</b>			<b>расстояние</b>	<b>город</b>	<b>город</b>
40	<b>S</b>	<b>8</b>	<b>9</b>	<b>F<sub>2</sub>(s)</b>	<b>J<sub>2</sub>(s)</b>	<b>J<sub>3</sub>(s)</b>
41	2	14	12	1012	14	9
42	3	12	14	1012	12	8
43	4	15	14	16	15	9
44						
45		n=4	$F_4(S) = \min_{s,j} [C_{sj} + F_3(J)]$			
46						
47						
48	<b>Из</b>	<b>Расчет расстояний в город J</b>		<b>Минимальное</b>	<b>В</b>	<b>В</b>
49	<b>города</b>			<b>расстояние</b>	<b>город</b>	<b>город</b>
50	<b>S</b>	<b>8</b>	<b>9</b>	<b>F<sub>2</sub>(s)</b>	<b>J<sub>2</sub>(s)</b>	<b>J<sub>4</sub>(s)</b>
51	1	18	23	18	18	18
52						
53						

Схема решения рассматриваемого примера средствами Excel достаточно проста и в тоже время в некоторой степени трудоемка (недостаточно автоматизирован процесс вычислений). Однако рассмотренный подход оправдывает себя, если нужно решить несколько задач с аналогичной топологией сети и другими исходными данными.

Из результатов видно, что минимальное расстояние из города 1 в город 10 найдено при  $n=4$ ,  $F_4(s)=F_4(1)=18$ , а оптимальный маршрут 1-2-6-9-10.

# Применение Mathcad при решении задач распределения средств между предприятиями и замены оборудования.

## 1. Задача распределения средств между предприятиями

ORIGIN:= 1

Матрица распределяемых ресурсов:

$$Q := \begin{pmatrix} 0 \\ 80 \\ 160 \\ 240 \\ 320 \\ 400 \end{pmatrix}$$

Матрица доходности (прибыльности) предприятий в зависимости от выделенных ресурсов:

$$F := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 30 & 28 & 35 & 27 \\ 57 & 62 & 67 & 73 \\ 120 & 122 & 130 & 125 \\ 150 & 146 & 144 & 152 \\ 180 & 175 & 180 & 178 \end{pmatrix}$$

```
Result := for j ∈ cols(F)..1
  for i ∈ 1..rows(F)
    Zi,j ← ( Fi,j Qi ) if j = cols(F)
    otherwise
      Zmax ← 0
      Xz ← -1
      if i ≠ 1
        for s ∈ 1..i
          if Zmax < Fs,j + max[(Zi,j+1) <1> ] if s = 1
            Zmax ← Fs,j + max[(Zi,j+1) <1> ]
            Xz ← Qs
          if Zmax < Fi,j + max[(Z1,j+1) <1> ] if s = i
            Zmax ← Fi,j + max[(Z1,j+1) <1> ]
            Xz ← Qi
          if Zmax < Fs,j + max[(Zs,j+1) <1> ] if s = (i+1)/2
            Zmax ← Fs,j + max[(Zs,j+1) <1> ]
            Xz ← Qs
          if Zmax < Fs,j + max[(Zi-s+1,j+1) <1> ] otherwise
            Zmax ← Fs,j + max[(Zi-s+1,j+1) <1> ]
            Xz ← Qs
      Zi,j ← ( Zmax Xz )
    Zi,j ← ( 0 0 ) otherwise
```

```

Xz ← Qrows(Q)
for j ∈ 1..cols(Z)
  Zmax ← -1
  for i ∈ 1..rows(Z)
    if Zmax < max[(Zi,j)(1)] if Qi ≤ Xz
      Zmax ← max[(Zi,j)(1)]
      s ← i
  Fmax ← Zmax if j = 1
  Xz ← Xz - max[(Zs,j)(2)]
  Xj ← max[(Zs,j)(2)]
R ←  $\begin{pmatrix} Fmax \\ X \\ Z \end{pmatrix}$ 

```

### Матрица условных максимумов:

$(Result_{3,1})_{1,1} = (0 \ 0)$	$(Result_{3,1})_{1,2} = (0 \ 0)$	$(Result_{3,1})_{1,3} = (0 \ 0)$	$(Result_{3,1})_{1,4} = (0 \ 0)$
$(Result_{3,1})_{2,1} = (35 \ 0)$	$(Result_{3,1})_{2,2} = (35 \ 0)$	$(Result_{3,1})_{2,3} = (35 \ 80)$	$(Result_{3,1})_{2,4} = (27 \ 80)$
$(Result_{3,1})_{3,1} = (73 \ 0)$	$(Result_{3,1})_{3,2} = (73 \ 0)$	$(Result_{3,1})_{3,3} = (73 \ 0)$	$(Result_{3,1})_{3,4} = (73 \ 160)$
$(Result_{3,1})_{4,1} = (130 \ 0)$	$(Result_{3,1})_{4,2} = (130 \ 0)$	$(Result_{3,1})_{4,3} = (130 \ 240)$	$(Result_{3,1})_{4,4} = (125 \ 240)$
$(Result_{3,1})_{5,1} = (160 \ 0)$	$(Result_{3,1})_{5,2} = (160 \ 0)$	$(Result_{3,1})_{5,3} = (160 \ 80)$	$(Result_{3,1})_{5,4} = (152 \ 320)$
$(Result_{3,1})_{6,1} = (203 \ 0)$	$(Result_{3,1})_{6,2} = (203 \ 0)$	$(Result_{3,1})_{6,3} = (203 \ 240)$	$(Result_{3,1})_{6,4} = (178 \ 400)$

**Ответ:**  $Result_{1,1} = 203$

**Максимальная прибыль.**

$Result_{2,1}^T = (0 \ 0 \ 240 \ 160)$

**Оптимальное распределение ресурсов: 203 усл. ед. выделить третьему хозяйствующему субъекту 240 ед., 160 ед. - четвертому.**

## 2. Задача замены оборудования

ORIGIN:= 1

Исходные данные по задаче динамического программирования замены оборудования.

Metod := 1

T:= 5

Горизонт планирования.

$F := ( 6000 \ 5000 \ 3000 \ 1000 \ 500 \ 0 \ 0 )^T$  Вектор ликвидной стоимости оборудования. Актуален при Metod=1.

p := 8000

Первоначальная стоимость оборудования.

$r := ( 600 \ 800 \ 1100 \ 1500 \ 2000 \ 0 \ 0 )^T$  Эксплуатационные затраты в зависимости от возраста оборудования.

```

D := Kni ← 1
P1 ← p
for i ∈ 1.. T
    Kni+1 ← Kni · ( 1 +  $\frac{R_i}{100}$  )
    Pi+1 ← p · Kni+1
    oti ← Fi if Metod = 1
    otherwise
        oti ← ( p - i · p ·  $\frac{1}{T_a}$  ) if Am = 1
        otherwise
            oti ← ( p - p ·  $\frac{1}{T_a}$  ) if i = 1
            oti ← ( oti-1 - oti-1 ·  $\frac{1}{T_a}$  ) otherwise
for i ∈ 1.. T
    for t ∈ 1.. T
        Zi,t ← ( 0 "O" )
k ← 0
for i ∈ T.. 1
    for t ∈ 1.. i - 1
        if i = T
            if rt+1 · Kni - ott+1 · Kni+1 ≠ p · Kni + r1 · Kni - ott · Kni - ot1 · Kni+1
                Zt,i ← ( rt+1 · Kni - ott+1 · Kni+1 "S" ) if rt+1 · Kni - ott+1 · Kni+1 < p · Kni + r1 · Kni - ott · Kni - ot1 · Kni+1
                Zt,i ← ( p · Kni + r1 · Kni - ott · Kni - ot1 · Kni+1 "Z" ) otherwise
            otherwise
                Zt,i ← ( rt+1 · Kni - ott+1 · Kni+1 "SZ" )
                k ← k + 1
        Zt,i ← [ p + rt + [(Zt,i+1)<1> ]1 "P" ] if t = 1 if i = 1
        otherwise
            if rt+1 · Kni + [(Zt+1,i+1)<1> ]1 ≠ p · Kni + r1 · Kni - ott · Kni + [(Z1,i+1)<1> ]1
                Zt,i ← [ rt+1 · Kni + [(Zt+1,i+1)<1> ]1 "S" ] if rt+1 · Kni + [(Zt+1,i+1)<1> ]1 < p · Kni + r1 · Kni - ott · Kni + [(Z1,i+1)<1> ]1
                Zt,i ← [ p · Kni + r1 · Kni - ott · Kni + [(Z1,i+1)<1> ]1 "Z" ] otherwise
            otherwise
                Zt,i ← [ rt+1 · Kni + [(Zt+1,i+1)<1> ]1 "SZ" ]
                k ← k + 1

```

Программный модуль, реализующий задачу динамического программирования замены оборудования (обратная схема Р.Э.Беллмана).

```

for s ∈ 1..k+1
  for i ∈ 1..T
    Xs,i ← ( "P" 0 ) if i = 1
    Xs,i ← ( "O" 0 ) otherwise
n ← 1
for s ∈ 1..k+1
  for i ∈ 1..T-1
    if i = 1 ∧ s = 1 ∧ [Xs,i+1(1)]1 = "O"
      Xs,i+1 ← ( "S" 1 ) if [Zi,i+1(2)]1 = "S"
      Xs,i+1 ← ( "Z" 1 ) if [Zi,i+1(2)]1 = "Z"
      if [Zi,i+1(2)]1 = "SZ"
        Xs,i+1 ← ( "S" 1 )
        Xn+1,i+1 ← ( "Z" 1 )
        n ← n + 1
    otherwise
      if [Xs,i(1)]1 = "S" ∧ [Xs,i+1(1)]1 = "O"
        Xs,i+1 ← [ [Z[Xs,i(2)]1+1,i+1(2)]1 [Xs,i(2)]1 + 1 ] if [ [Z[Xs,i(2)]1+1,i+1(2)]1 ]1 = "S"
        Xs,i+1 ← [ [Z[Xs,i(2)]1+1,i+1(2)]1 [Xs,i(2)]1 + 1 ] if [ [Z[Xs,i(2)]1+1,i+1(2)]1 ]1 = "Z"
        if [ [Z[Xs,i(2)]1+1,i+1(2)]1 ]1 = "SZ"
          Xs,i+1 ← [ "S" [Xs,i(2)]1 + 1 ]
          for f ∈ 1..i
            Xn+1,f+1 ← [ [Xs,f+1(1)]1 [Xs,f+1(2)]1 ] if f ≠ i
            Xn+1,f+1 ← [ "Z" [Xs,i(2)]1 + 1 ] if f = i
          n ← n + 1
        if [Xs,i(1)]1 = "Z" ∧ [Xs,i+1(1)]1 = "O"
          Xs,i+1 ← [ [Zi,i+1(2)]1 1 ] if [Zi,i+1(2)]1 = "S"
          Xs,i+1 ← [ [Zi,i+1(2)]1 1 ] if [Zi,i+1(2)]1 = "Z"
          if [Zi,i+1(2)]1 = "SZ"
            Xs,i+1 ← ( "S" 1 )
            for f ∈ 1..i
              Xn+1,f+1 ← [ [Xs,f+1(1)]1 [Xs,f+1(2)]1 ] if f ≠ i
              Xn+1,f+1 ← ( "Z" 1 ) if f = i
            n ← n + 1
  for s ∈ 1..n
    for i ∈ 1..T
      Xopts,i ← Xs,i

```

$R \leftarrow ( Z \text{ Xopt } P \text{ r ot } Kn )^T$



## Варианты задач для самостоятельной работы

### 1. Задача распределения средств между предприятиями.

Имеются четыре предприятия, между которыми необходимо распределить 100 тыс. усл. ед. средств. Значения прироста выпуска продукции на предприятиях в зависимости от выделенных средств  $x$  представлены в таблице. Составьте оптимальный план распределения средств, позволяющий максимизировать общий прирост выпуска продукции.

#### Вариант 1

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	16	14	15	15
40	30	32	36	25
60	49	50	45	22
80	51	48	57	36
100	72	60	70	51

#### Вариант 16

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	9	14	10	15
40	26	22	23	18
60	35	28	27	16
80	32	38	32	20
100	41	46	48	34

#### Вариант 2

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	19	14	20	25
40	36	32	36	53
60	51	52	47	66
80	72	61	72	70
100	81	79	80	84

#### Вариант 17

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	2	6	8	5
40	8	3	6	9
60	12	5	10	14
80	11	8	7	13
100	20	10	17	15

#### Вариант 3

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	10	14	14	19
40	16	14	15	15
60	30	32	36	25
80	45	43	47	36
100	60	50	55	53

#### Вариант 18

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	26	31	32
40	49	30	46	45
60	45	59	64	56
80	82	56	82	87
100	86	89	90	84

#### Вариант 4

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	14	17	22	20
40	26	20	21	33
60	35	32	37	46
80	52	61	67	30
100	61	72	58	42

#### Вариант 19

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	10	16	19
40	30	32	36	25
60	44	54	34	22
80	51	48	47	36
100	62	56	57	48

**Вариант 5**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	42	40	25	24
40	34	52	36	45
60	47	50	46	32
80	51	48	57	36
100	62	60	67	54

**Вариант 6**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	19	48	42	45
40	36	32	56	53
60	54	62	67	66
80	72	81	82	70
100	88	95	98	84

**Вариант 7**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	22	17	18	35
40	43	39	33	42
60	49	51	45	55
80	61	75	57	68
100	82	79	67	81

**Вариант 8**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	15	11	10
40	23	27	21	19
60	30	29	34	36
80	42	46	45	47
100	58	61	58	54

**Вариант 9**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	9	11	14	8
40	19	14	20	15
60	30	32	16	25
80	36	30	38	33
100	48	44	52	36

**Вариант 10**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	19	33	29	35
40	26	43	36	45
60	35	52	49	56
80	47	60	62	72
100	68	79	82	94

**Вариант 20**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	11	24	12	35
40	26	22	28	33
60	31	32	37	36
80	42	41	47	40
100	58	59	53	54

**Вариант 21**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	16	12	15	24
40	30	36	36	22
60	49	34	45	32
80	51	47	57	41
100	72	57	70	59

**Вариант 22**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	10	16	19
40	26	22	28	33
60	31	52	48	56
80	42	41	47	40
100	68	71	58	54

**Вариант 23**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	14	15	13
40	36	32	36	33
60	34	50	45	35
80	47	48	57	34
100	57	60	70	47

**Вариант 24**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	14	20	29
40	36	32	36	33
60	34	42	27	46
80	49	56	32	50
100	55	59	36	44

**Вариант 25**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	11	14	12	15
40	24	32	39	25
60	34	50	40	22
80	27	48	37	36
100	37	60	47	57

**Вариант 11**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	24	10	20
40	21	17	16	25
60	20	21	25	22
80	30	38	22	23
100	42	35	18	41

**Вариант 12**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	22	24	28	25
40	38	32	46	33
60	45	44	57	46
80	52	56	67	58
100	51	69	70	68

**Вариант 13**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	12	24	25	18
40	23	32	36	22
60	33	40	44	32
80	45	48	47	36
100	52	60	57	35

**Вариант 14**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	29	24	22	25
40	36	33	36	35
60	48	22	44	46
80	52	46	53	49
100	58	39	68	32

**Вариант 15**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	6	4	5	8
40	10	12	16	15
60	24	25	24	22
80	21	24	27	31
100	32	30	37	45

**Вариант 26**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	29	24	28	25
40	36	23	39	23
60	45	32	47	36
80	47	46	37	40
100	41	27	50	34

**Вариант 27**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	11	14	12	15
40	24	32	39	25
60	34	50	40	22
80	27	48	37	36
100	37	60	47	57

**Вариант 28**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	19	14	33	25
40	45	32	45	53
60	38	52	33	66
80	48	61	67	70
100	77	79	61	84

**Вариант 29**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	10	12	14	19
40	14	37	48	45
60	34	27	37	38
80	42	40	48	48
100	66	56	64	77

**Вариант 30**

$x$	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$
20	14	19	33	41
40	48	45	59	81
60	37	38	33	52
80	48	58	77	73
100	64	67	61	92

## 2. Задача замены оборудования

Оборудование эксплуатируется в течение 5 лет, после чего продается. В начале каждого года принимается решение сохранить оборудование или заменить его новым. Известны первоначальная стоимость нового оборудования  $p(t)=p_0=const$ , затраты на содержание оборудования  $r(t)$  и ликвидная стоимость оборудования  $\varphi(t)$ . Необходимо определить оптимальную стратегию эксплуатации оборудования, обеспечивающую минимальные суммарные затраты на эксплуатацию в течение 5 лет.

Исходные данные для решения задачи замены оборудования (по вариантам):

### Вариант 1.

t	0	1	2	3	4	5
r(t)	450	500	560	630	700	-
$\varphi(t)$	-	7000	6700	6600	6000	5400
p(t)	8600	8600	8600	8600	8600	8600

### Вариант 2.

t	0	1	2	3	4	5
r(t)	250	270	310	360	400	-
$\varphi(t)$	-	8900	7700	5300	4000	500
p(t)	11000	11000	11000	11000	11000	11000

### Вариант 3.

t	0	1	2	3	4	5
r(t)	2500	2750	2800	3100	3500	-
$\varphi(t)$	-	7500	7000	6800	6300	5000
p(t)	12000	12000	12000	12000	12000	12000

### Вариант 4.

t	0	1	2	3	4	5
r(t)	600	800	1100	1500	2000	-
$\varphi(t)$	-	4000	3600	3200	3000	2500
p(t)	5000	5000	5000	5000	5000	5000

### Вариант 5.

t	0	1	2	3	4	5
r(t)	500	900	1200	2000	3000	-
$\varphi(t)$	-	10000	8500	6500	4000	2000
p(t)	13000	13000	13000	13000	13000	13000

**Вариант 6.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	500	700	900	1100	1500	-
<b>φ(t)</b>	-	5000	4000	3000	1000	500
<b>p(t)</b>	7500	7500	7500	7500	7500	7500

**Вариант 7.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	1000	1100	1300	1500	1600	-
<b>φ(t)</b>	-	13000	12000	11000	9000	7000
<b>p(t)</b>	15000	15000	15000	15000	15000	15000

**Вариант 8.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	600	800	1200	1500	2000	-
<b>φ(t)</b>	-	7000	6000	4000	2000	1000
<b>p(t)</b>	10000	10000	10000	10000	10000	10000

**Вариант 9.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	700	900	1200	1600	2000	-
<b>φ(t)</b>	-	7000	6000	4000	2000	1000
<b>p(t)</b>	9000	9000	9000	9000	9000	9000

**Вариант 10.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	500	900	1000	1300	1600	-
<b>φ(t)</b>	-	7000	5000	4000	1000	500
<b>p(t)</b>	10000	10000	10000	10000	10000	10000

**Вариант 11.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	1200	1500	1900	2400	3000	-
<b>φ(t)</b>	-	10000	8800	6000	3000	1000
<b>p(t)</b>	12000	12000	12000	12000	12000	12000

**Вариант 12.**

<b>T</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	700	910	1200	1500	2000	-
<b>φ(t)</b>	-	7000	6000	4500	2400	1200
<b>p(t)</b>	9500	9500	9500	9500	9500	9500

**Вариант 13.**

<b>T</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	700	900	1200	1600	3000	-
<b>φ(t)</b>	-	8500	8100	7600	6500	4100
<b>p(t)</b>	10000	10000	10000	10000	10000	10000

**Вариант 14.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	400	600	1000	1450	2000	-
<b>φ(t)</b>	-	4100	3200	2500	1400	1000
<b>p(t)</b>	6000	6000	6000	6000	6000	6000

**Вариант 15.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	500	600	900	1000	1200	-
<b>φ(t)</b>	-	6500	6000	5400	5100	4500
<b>p(t)</b>	8100	8100	8100	8100	8100	8100

**Вариант 16.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	600	800	1200	1500	1900	-
<b>φ(t)</b>	-	8000	7000	5000	3000	1000
<b>p(t)</b>	10000	10000	10000	10000	10000	10000

**Вариант 17.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	600	950	1300	1900	2500	-
<b>φ(t)</b>	-	10000	8000	5000	2000	550
<b>p(t)</b>	12000	12000	12000	12000	12000	12000

**Вариант 18.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	800	1100	1500	1900	2500	-
<b>φ(t)</b>	-	7000	6000	4500	2300	700
<b>p(t)</b>	9500	9500	9500	9500	9500	9500

**Вариант 19.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	2500	3600	7500	8500	10000	-
<b>φ(t)</b>	-	25000	20000	15500	14000	10000
<b>p(t)</b>	30000	30000	30000	30000	30000	30000

**Вариант 20.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	2100	2500	3000	3250	3600	-
<b>φ(t)</b>	-	12000	10000	9000	7000	5600
<b>p(t)</b>	15500	15500	15500	15500	15500	15500

**Вариант 21.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	210	250	300	325	360	-
<b>φ(t)</b>	-	5000	4800	4500	4000	3500
<b>p(t)</b>	5500	5500	5500	5500	5500	5500

**Вариант 22.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	210	250	300	325	360	-
<b>φ(t)</b>	-	1200	1000	900	700	560
<b>p(t)</b>	1550	1550	1550	1550	1550	1550

**Вариант 23.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	1100	1500	2000	2250	2600	-
<b>φ(t)</b>	-	9000	8200	7500	7000	5600
<b>p(t)</b>	9800	9800	9800	9800	9800	9800

**Вариант 24.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	5000	5250	6000	6250	6350	-
<b>φ(t)</b>	-	19000	18200	17500	17000	15600
<b>p(t)</b>	19800	19800	19800	19800	19800	19800

**Вариант 25.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	6000	6250	7000	7250	7350	-
<b>φ(t)</b>	-	20000	19200	18000	17000	15600
<b>p(t)</b>	20800	20800	20800	20800	20800	20800

**Вариант 26.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	800	900	1100	1500	2000	-
<b>φ(t)</b>	-	7000	6000	4500	2300	700
<b>p(t)</b>	10500	10500	10500	10500	10500	10500

**Вариант 27.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	200	500	900	1400	2000	-
<b>φ(t)</b>	-	7000	6500	5800	4500	3700
<b>p(t)</b>	10000	10000	10000	10000	10000	10000

**Вариант 28.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	600	800	1100	1500	2000	-
<b>φ(t)</b>	-	7000	6600	6200	6000	5500
<b>p(t)</b>	9000	9000	9000	9000	9000	9000

**Вариант 29.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	1250	1270	1310	1360	1400	-
<b>φ(t)</b>	-	12000	11300	10000	9500	9200
<b>p(t)</b>	21000	21000	21000	21000	21000	21000

**Вариант 30.**

<b>t</b>	<b>0</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>r(t)</b>	750	800	860	930	1000	-
<b>φ(t)</b>	-	7000	6700	6600	6000	5400
<b>p(t)</b>	9000	9000	9000	9000	9000	9000

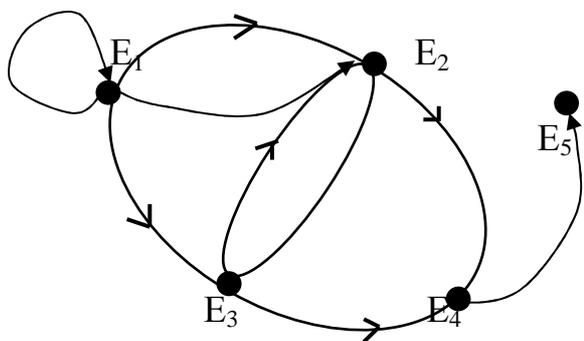
# ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭЛЕМЕНТАХ ТЕОРИИ ГРАФОВ И СЕТЕВОМ УПРАВЛЕНИИ

## Способы задания графа смежности

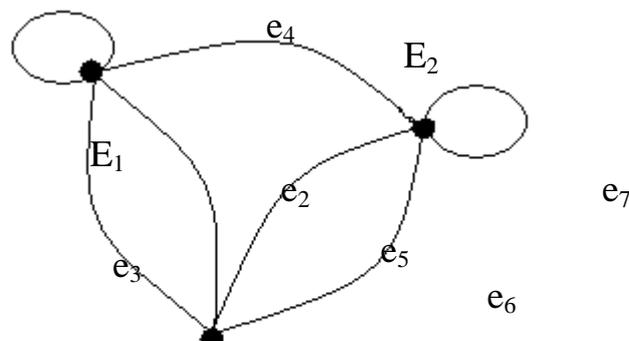
### Матрицы смежности и инцидентности

Имеется несколько способов задания графа. Во многих случаях граф удобно задавать в виде матрицы смежности вершин, матрицы смежности дуг или матрицы инцидентности.

Матрица смежности вершин орграфа называется квадратная матрица  $A$ , каждый  $ij$ -й элемент которой численно равен количеству дуг, идущих из вершины  $E_i$  в вершину  $E_j$ . Если  $G=(E, e)$  – неориентированный граф, то ему соответствует симметричная матрица смежности, так как дуги  $(E_i, E_j)$  и  $(E_j, E_i)$  существуют одновременно. Если  $G=(E, e)$  ортограф, то соответствующая ему матрица смежности может не являться симметрической. Матрицы смежности вершин графов можно представить в таблицах.



$E_i \backslash E_j$	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$
$E_1$	1	1	1	0	0
$E_2$	1	0	1	1	0
$E_3$	0	1	0	1	0
$E_4$	0	0	0	0	1



$E_i \backslash E_j$	$E_1$	$E_2$	$E_3$
$E_1$	1	1	2
$E_2$	1	1	2
$E_3$	2	2	0

Любая симметрическая матрица смежности с целыми неотрицательными элементами может быть интерпретирована как граф.

Введем понятие матрицы смежности дуг (ребер) графа. Для простоты предположим, что граф не имеет петель и кратных ребер.

Матрицей смежности дуг (ребер) орграфа (графа) называется квадратная матрица  $A$ , каждый  $ij$ -й элемент которой равен единице, если конечная вершина дуги  $e_i$  является начальной вершиной дуги  $e_j$ -м(если ребра имеют общую вершину), и нулю во всех остальных случаях.

Матрицей инцидентности орграфа называется прямоугольная матрица  $A$ , строки которой соответствуют вершинам, столбцы – дугам, а элементы равны 1, -1, или 0. При этом на пересечении вершины  $E$  и дуги  $e$  ставится значение  $\varepsilon(E,e)=1$ , если  $E$  – начальная вершина дуги  $e$ ,  $\varepsilon(E,e)=-1$ , если  $E$  – конечная вершина, и  $\varepsilon(E,e)=0$ , если  $E$  не инцидентна  $e$ .

$e_i \backslash e_j$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$	$e_6$	$e_7$	$e_8$	$e_9$
$e_1$		1	1	1	1				1
$e_2$	1				1	1		1	1
$e_3$	1			1	1		1	1	
$e_4$	1		1			1	1		1
$e_5$	1	1	1				1	1	1
$e_6$		1		1			1	1	1
$e_7$			1	1	1	1		1	1
$e_8$		1	1		1	1	1		
$e_9$	1	1		1	1	1	1		

$e_i \backslash e_j$	$e_1$	$e_2$	$e_3$	$e_4$	$e_5$
$e_1$		1			1
$e_2$	1		1		
$e_3$				1	
$e_4$		1			1
$e_5$				1	

Если  $G$  – неориентированный граф, то можно использовать значения  $\varepsilon=0$ ,  $\varepsilon=1$ .

## Разбиение элементов орграфа по рангам

### Отношение старого порядка в орграфе

Некоторые операции, подлежащие исследованию, включают значительное количество разнообразных компонентов. Орграфы, описывающие такие операции, содержат большое количество элементов (вершин и дуг). Выработка управленческих решений по осуществлению операций проводится, как правило, на ЭВМ и требует упорядоченной нумерации элементов орграфа.

Рассмотрим связный орграф без контуров. Положим  $E_i < E_j$  – отношение строгого порядка, если существует путь ненулевой длины от  $E_i$  к  $E_j$ . Это отношение обладает свойствами:

1. антисимметричности: если  $E_i < E_j$ , то не существует  $E_j < E_i$ , так как в противном случае орграф содержал бы контур;
2. транзитивности: если  $E_i < E_j$  и  $E_j < E_s$ , то  $E_i < E_s$  (справедливость этого свойства очевидна из определения пути).

Теперь остановимся на разбиении элементов орграфа по рангам (слоям).

Вершина  $E_i$  называется *вершиной 1-го ранга*, если ее полустепень захода равна нулю ( $p(E_i)=0$ ). Вершина называется *вершиной 2-го ранга*, если в нее входят одна или несколько дуг из вершин 1-го ранга, и не входит ни одна дуга из вершин выше 1-го ранга. Вершина  $E_i$  называется *вершиной  $k$ -го ранга*, если в нее входят дуги из вершин не выше  $(k-1)$ -го ранга, и при этом имеется хотя бы одна дуга, исходящая из вершин  $(k-1)$ -го ранга. Дуга  $e_i$  называется *дугой  $k$ -го ранга*, если она опирается на дугу (дуги) не выше  $(k-1)$ -го ранга, среди которых есть хотя бы одна дуга  $(k-1)$ -го ранга.

После разбиения в случае необходимости элементам орграфа придается новая, более удобная нумерация, такая, чтобы:

1. все номера элементов некоторого ранга были меньше номеров элементов следующего ранга; номера элементов 1-го ранга были наименьшими, а номера элементов последнего ранга – наибольшими;

2. порядок номеров элементов внутри одного и того же ранга был безразличен (так как вершины, принадлежащие одному рангу, не соединены между собой дугами, а дуги друг на друга не опираются).

Рассмотрим методы, используемые для нахождения рангов вершин и дуг в связном орграфе без контуров.

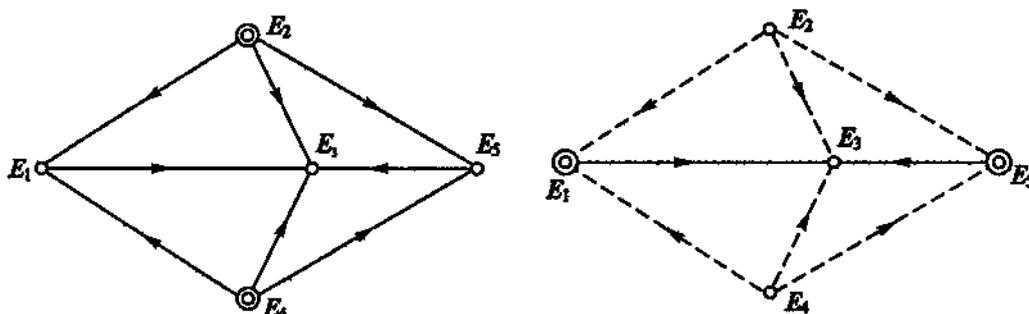
### Нахождение рангов вершин на чертеже орграфа

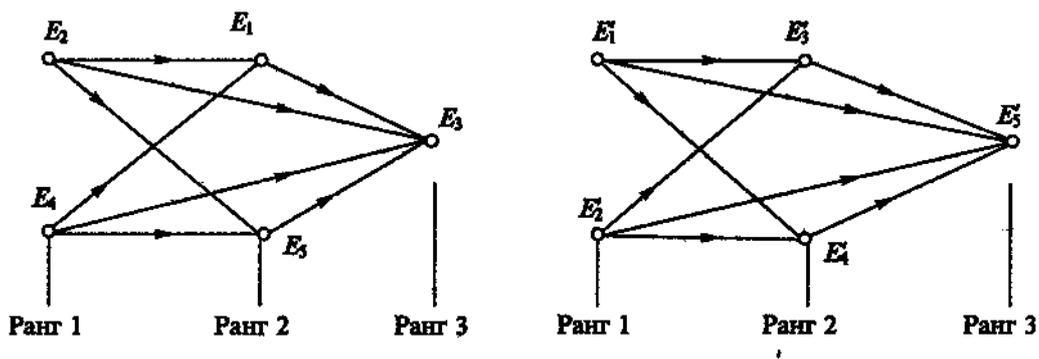
Идея метода основана на последовательном нахождении вершин орграфа, полустепень захода которых  $p^*(E_i)=0$ , и отбрасывании этих вершин и выходящих из них дуг.

Рассмотрим нахождение рангов вершин связного орграфа без контуров на примере. Найдем вершины 1-го ранга, т.е. локальной степени  $p^*(E_i)=0$ . Таких вершин две –  $E_2$  и  $E_4$ .

Отбросим вершины  $E_2$  и  $E_4$  и дуги, выходящие из них, и на полученном орграфе (на рис. отброшенные дуги показаны пунктиром) снова найдем вершины локальной степени  $p^*(E_i)=0$ . Ими являются вершины 2-го ранга  $E_1$  и  $E_5$ . Продолжая процесс решения, находим, что единственная вершина  $E_3$  3-го ранга.

Орграф с вершинами, разбитыми по рангам, приведен на рис. На рис. показан тот же орграф с новой нумерацией вершин ( $E_1'=E_2$ ,  $E_2'=E_4$ ,  $E_3'=E_1$ ,  $E_4'=E_5$ ,  $E_5'=E_3$ ). Каждый ранг содержит вершины, не соединенные между собой дугами. С вершин меньшего ранга дуги входят только в вершины большего ранга. Нумерация вершин возрастает от 1-го ранга к последнему.





### Метод Демукрона нахождения рангов вершин орграфа

Метод Демукрона основан на использовании матрицы смежности вершин орграфа. Пусть орграф представлен матрицей смежности  $A_{n \times n}$ .

Обозначим через  $v_{E_i} = (a_{i1}, a_{i2}, \dots, a_{in})$ ,  $i=1, 2, \dots, n$ , векторы, являющиеся строками матрицы смежности. Вычислим

$$v_1 = \sum_{i=1}^n v_{E_i}$$

и поместим результат в  $(n+1)$ -ю строку табл. Среди неотрицательных компонентов вектора  $v_1 = (a_1^{(1)}, a_2^{(1)}, \dots, a_n^{(1)})$  находим компоненты, равные нулю.

номер строки	$E_j$		...		...		...		Ранг	Новая нумерация вершин
	$E_i$	$E_1$		$E_k$		$E_s$		$E_n$		
$1$	$E_1$	$a_{11}$	...	$a_{1k}$	...	$a_{1s}$	...	$a_{1n}$	$r$	$E_n'$
...	...	...	...	...	...	...	...	...	...	...
$k$	$E_k$	$a_{k1}$	...	$a_{kk}$	...	$a_{ks}$	...	$a_{kn}$	$1$	$E_1'$
...	...	...	...	...	...	...	...	...	...	...
$s$	$E_s$	$a_{s1}$	...	$a_{sk}$	...	$a_{ss}$	...	$a_{sn}$	$1$	$E_2'$
...	...	...	...	...	...	...	...	...	...	...
$n$	$E_n$	$a_{n1}$	...	$a_{nk}$	...	$a_{ns}$	...	$a_{nn}$	$2$	$E_3'$
$n+1$	$v_1$	$a_1^{(1)}$	...	$a_k^{(1)}$	...	$a_s^{(1)}$	...	$a_n^{(1)}$		

$n+2$	$v_2$	$a_1^{(2)}$	...	$a_k^{(2)}$	...	$a_s^{(2)}$	...	$a_n^{(2)}$		
...	...	...	...	...	...	...	...	...	...	...
$n+r$	$v_r$	$0$	...	\	...	\	...	\		

Пусть  $a_k^{(1)}$  и  $a_s^{(1)}$  равны нулю. Это значит, что в вершины  $E_k$  и  $E_s$  не входит ни одна дуга, и они относятся к 1-му рангу. В столбце, соответствующем рангу, в строках  $k$  и  $s$  запишем ранг этих вершин.

Из вектора  $v_1$  вычтем векторы-строки, соответствующие вершинам 1-го ранга  $E_k$  и  $E_s$ . В результате получим вектор  $v_2 = v_1 - v_{E_k} - v_{E_s}$ , в котором появятся новые компоненты, равные нулю.

Пусть  $a_n^{(2)} = 0$ . Следовательно, вершина  $E_n$  относится ко 2-му рангу. Ранг этой вершины, равный 2, заносим в таблицу.

Далее вычисляем вектор  $v_3 = v_2 - v_{E_n}$ , в котором появятся новые нули. Следовательно, тем самым будут определены вершины 3-го ранга.

Процесс вычислений продолжается аналогично ранее изложенному до тех пор, пока не будет получен вектор, все компоненты которого равны нулю. Пусть  $v_r$  – нулевой вектор, новый нулевой компонент которого  $a_1^{(r)}$ . Тогда  $E_1$  – вершина  $r$ -го ранга.

Перенумеруем вершины орграфа согласно их рангам и запишем новые номера в правый столбец матрицы, как показано в таблице.

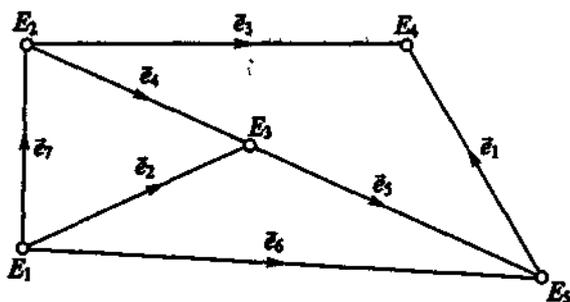
Результаты вычислений используются для изображения орграфа с новой нумерацией вершин.

### Нахождение рангов дуг орграфа

Для нахождения рангов дуг орграфа можно воспользоваться ранее изложенными методами, несколько модифицировав их. Рассмотрим один из подходов к определению рангов дуг орграфа, заданного списковым способом. После разбиения дуг орграфа по рангам им придается более удобная нумерация, такая, что каждая дуга может опираться на дуги с меньшими порядковыми номерами. Практическую реализацию этого подхода проиллюстрируем на примере.

Требуется разбить по рангам дуги орграфа, заданного списковым способом, и при необходимости перенумеровать их. Орграф, представленный в таблице, изображен на рисунке.

Дуга	Опирается на дуги	Ранг	Новая нумерация дуг
$e_1$	$e_5, e_6$	4	$b_7$
$e_1$	—	1	$b_1$
$e_1$	$e_7$	2	$b_4$
$e_1$	$e_7$	2	$b_5$
$e_1$	$e_2, e_4$	3	$b_6$
$e_1$	—	1	$b_2$
$e_1$	—	1	$b_3$



К 1-му рангу будут относиться те дуги, которые не опираются ни на какие другие дуги.

В табл. дуги  $e_2, e_6, e_7$  не опираются ни на какие другие дуги, следовательно, они относятся к 1-му рангу. Ранг этих дуг запишем в графе «Ранг» таблицы.

Дуги  $e_3$  и  $e_4$  опираются на дугу  $e_7$ , следовательно, эти дуги 2-го ранга. Дуга  $e_3$  относится к 3-му рангу, так как она опирается на дугу  $e_2$  1-го ранга и дугу  $e_4$  2-го ранга. Дуга  $e_1$  опирается на дуги  $e_5, e_6$ , первая из которых 3-го ранга, а вторая - 1-го, следовательно, дуга  $e_1$  4-го ранга.

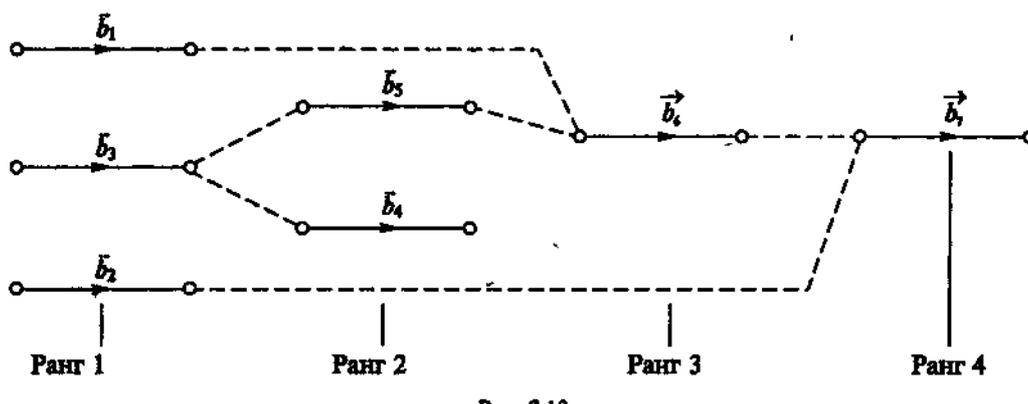
Перенумеруем дуги орграфа в соответствии с их рангом. Новая нумерация дуг записана в правом столбце таблицы. Чтобы записать орграф с новой нумерацией дуг в структурную таблицу, поступим следующим образом.

В левом столбце запишем по порядку все дуги орграфа в новой нумерации, а в правом — дуги, на которые они опираются.

Дуга	Опирается на дуги	Дуга	Опирается на дуги
$b_1$	—	$b_5$	$b_3$
$b_2$	—	$b_6$	$b_1, b_5$
$b_3$	—	$b_7$	$b_2, b_6$
$b_4$	$b_3$		

Так как дуги  $b_1, b_2, b_3$  1-го ранга, то в правом столбце в первых трех строках запишем прочерки. Следующая по списку дуга  $b_4$  в старой нумерации имела номер  $e_3$  и опиралась на дугу  $e_7$ , которая в новой нумерации имеет номер  $b_3$ . Следовательно, дуга  $b_4$  опирается на дугу  $b_3$ . Продолжая аналогичным образом, определим, что дуга  $b_5$  опирается на дугу  $b_3$ ,  $b_6$  — на  $b_1$  и  $b_5$ ,  $b_7$  — на  $b_2$  и  $b_6$ .

Орграф с новой нумерацией дуг, заданный списковым способом, представлен в табл. и геометрически на рис. Нетрудно заметить, что после нумерации каждая дуга опирается на дуги с меньшими порядковыми номерами. После разбиения элементов орграфа по рангам и их перенумерации возможно осуществление других важных исследований.



## ПРИМЕНЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ EXCEL

Рассмотрим применение Excel для нахождения рангов вершин графа на примере. Запишем граф в виде матрицы смежности вершин.

	А	В	С	Д	Е	Ф	Г	Н	І
1	Номер строки	Вершины							Новая нумерация вершин
2		Вершины	$E_1$	$E_2$	$E_3$	$E_4$	$E_5$	Ранг	
3	1	$E_1$			1			2	
4	2	$E_2$	1		1		1	1	
5	3	$E_3$						3	
6	4	$E_4$	1		1		1	1	
7	5	$E_5$			1			2	
8	6	$V_1$	2	0	4	0	2		
9	7	$V_2$	0	0	2	0	0		
10	8	$V_3$	0	0	0	0	0		
11									

Найдем суммы элементов в столбцах  $E_j$  ( $j=1, \dots, 5$ ), для чего в ячейку С8 занесем функцию СУММ С3:С7, далее выделим эту ячейку и, установив курсор на малый квадратик в правом нижнем углу (он имеет вид черного крестика), протащим его таким образом при помощи мыши до ячейки G8. В результате функция суммирования элементов будет занесена в столбцы остальных ячеек 8-й строки электронной таблицы.

В 6-й строке (вектор  $V_1$ ) находятся два нулевых элемента. Следовательно, вершины  $E_2$  и  $E_4$  1-го ранга. Занесем их ранг, равный единице, в графу Ранг. В ячейку С9 занесем формулу =С8-С4-С6 и, протащив курсор, как и ранее, занесем соответствующие формулы в ячейки D9:G9. В 7-й строке (вектор  $V_2$ ) появятся два новых нулевых элемента, находящиеся в столбцах, соответствующих вершинам  $E_1$  и  $E_5$ . Следовательно, эти вершины 2-го ранга. В ячейку С10 занесем формулу: =С9-С3-С7, а также занесем аналогичные формулы в ячейки D10:G10. В результате в 8-й строке (вектор  $V_3$ ) получим еще один нуль в столбце с вершиной  $E_5$ . Следовательно, она 3-го ранга. Так как  $V_3$  — нуль-вектор, на этом вычисления заканчиваются. Новая нумерация вершин занесена в последнюю графу табл.

Орграф, построенный с учетом ранга вершин и их новой нумерации, будет изоморфен орграфу. В табл. показаны формулы, занесенные в ячейки

диапазона C8:G10. Если снять флажок **Формулы** в диалоговом окне **Параметры**, вызываемом командой **Параметры** из меню **Сервис**, таблица снова примет первоначальный вид.

	A	B	C	D	E	F	G	H	I
1	Номер строки	Вершины							Новая нумерация вершин
2		Вершины	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	Ранг	
3	1	E <sub>1</sub>			1			2	
4	2	E <sub>2</sub>	1		1		1	1	
5	3	E <sub>3</sub>						3	
6	4	E <sub>4</sub>	1		1		1	1	
7	5	E <sub>5</sub>			1			2	
8	6	V <sub>1</sub>	=СУММ(C3:C7)	=СУММ(D3:D7)	=СУММ(E3:E7)	=СУММ(F3:F7)	=СУММ(G3:G7)		
9	7	V <sub>2</sub>	=C8-C4-C6	=D8-D4-D6	=E8-E4-E6	=F8-F4-F6	=G8-G4-G6		
10	8	V <sub>3</sub>	=C9-C3-C7	=D9-D3-D7	=E9-E3-E7	=F9-F3-F7	=G9-G3-G7		
11									

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ MS EXCEL РЕШЕНИЯ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Решение задач нелинейного программирования (НЛП) в Excel осуществляется командой **Поиск решения** из меню **Сервис**, в которой реализованы два метода: метод Ньютона и метод сопряженных градиентов. В методе Ньютона для определения направления и шага перемещения в новую точку используются вторые производные. Использование вторых производных ведет к большому объему вычислений на каждом шаге итерационного процесса, но сокращает число шагов для достижения оптимального решения.

Метод сопряженных градиентов использует только первые производные, что приводит к сокращению объема вычислений на каждом шаге итерационного процесса, но и к возможному увеличению числа шагов.

Решение нелинейных задач командой **Поиск решения** осуществляется так же, как и линейных. Однако при решении нелинейных задач необходимо, чтобы функция  $f(X)$  в начальной точке  $X^{(0)}$  была отлична от нуля. Дело в том, что на каждом шаге проверяется достижение оптимального решения по

формуле  $\Delta f = \frac{f_{k+1} - f_k}{f_k} \leq \varepsilon$  ( $\varepsilon$  – заданная величина точности решения), а на нуль

делить нельзя. Кроме того, для ускорения нахождения оптимального решения начальную точку  $X^{(0)}$  следует, по возможности, выбрать ближе к оптимальной, т.е. установить в изменяемых ячейках, значения близкие к оптимальным.

Чтобы решить задачу НЛП, в диалоговом окне **Параметры поиска решения** активизируется одна из программ «**Метод Ньютона**» или «**Метод сопряженных градиентов**», что обеспечивается щелчком по названию программы. Так как решается нелинейная задача, то не следует устанавливать флажок «**Линейная модель**».

Более полную процедуру решения задач НЛП средствами Excel рассмотрим на примере. Так, пусть предприятие может выпускать два вида продукции. На ее изготовление расходуются ресурсы  $i$ -го вида ( $i=1,2,3$ ). С

учетом брака расходов ресурсов на единицу производимой продукции  $j$ -го вида ( $j=1,2$ ) определяется выражением  $a_{ij}+k_{ij}x_j$ , а прибыль в зависимости от объемов производства равна  $p_j+l_jx_j$ , где

$x_j$  – искомый объем производства продукции  $j$ -го вида;

$a_{ij}$  – норма расхода  $i$ -го ресурса на производство единицы продукции  $j$ -го вида;

$k_{ij}$  – коэффициент изменения расхода соответствующего ресурса с учетом выпуска бракованных изделий;

$p_j$  – прибыль от единицы продукции  $j$ -го вида;

$l_j$  – коэффициент изменения прибыли, влияющей на объемы производства продукции  $j$ -го вида.

Требуется найти такие объемы производства продукции, при которых прибыль была бы максимальной. Составим математическую модель задачи с учетом значений параметров, приведенных в таблице, по критерию максимума прибыли.

Ресурс (i)	Запас ресурса	Норма расхода ресурсов ( $a_{ij}$ ) на продукцию вида (j)		Коэффициент изменения норм расхода ресурсов ( $k_{ij}$ ) на продукцию вида (j)	
		1	2	1	2
1	1350	15	18	0,1	0,05
2	1400	12	16	0,2	0,2
3	1580	17	14	0,1	0,15
Прибыль (ден.ед)		100	120		
Коэффициент изменения прибыли		-0,08	-0,1		

Целевая функция имеет вид:

$$f(x_1, x_2) = (100 - 0.08x_1)x_1 + (120 - 0.1x_2)x_2 \rightarrow \max$$

или

$$f(x_1, x_2) = 100x_1 - 0.08x_1^2 + 120x_2 - 0.1x_2^2 \rightarrow \max$$

Максимум этой функции находится при ограничениях:

$$\begin{cases} (15 + 0.1x_1)x_1 + (18 + 0.05x_2)x_2 \leq 1350 \\ (12 + 0.2x_1)x_1 + (16 + 0.2x_2)x_2 \leq 1400 \\ (17 + 0.1x_1)x_1 + (14 + 0.15x_2)x_2 \leq 1580 \end{cases}$$

или

$$\begin{cases} 15x_1 + 0.1x_1^2 + 18x_2 + 0.05x_2^2 \leq 1350 \\ 12x_1 + 0.2x_1^2 + 16x_2 + 0.2x_2^2 \leq 1400 \\ 17x_1 + 0.1x_1^2 + 14x_2 + 0.15x_2^2 \leq 1580 \end{cases} \quad x_1 \geq 0, \quad x_2 \geq 0$$

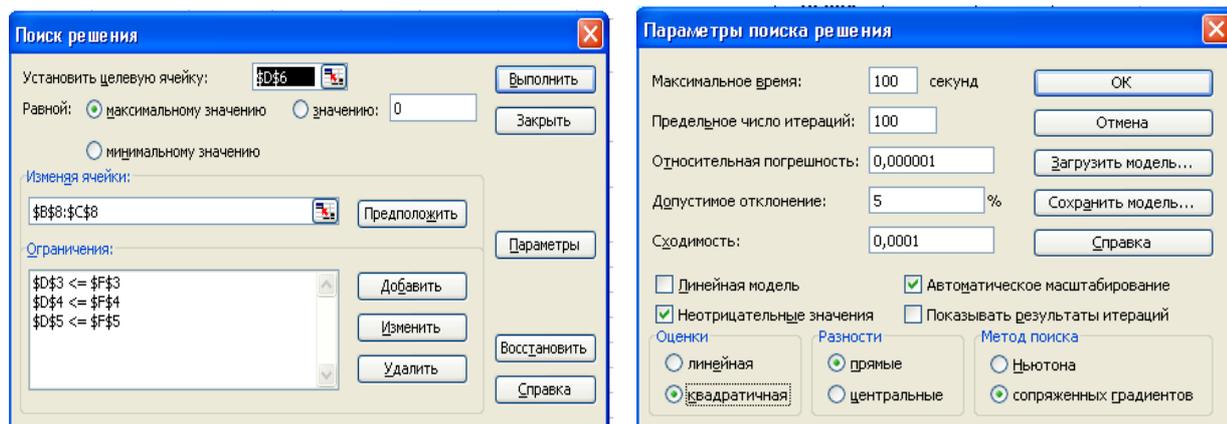
В рассмотренном примере и функция, и ограничения являются нелинейными, так как содержат неизвестные во второй степени.

	A	B	C	D	E	F
1	ограничения и др. Показатели	Переменные		Левая часть ограничений	Вид ограничений	Правая часть огр.
2		X <sub>1</sub>	X <sub>2</sub>			
3		ресурс 1	=15*B8+0,1*B8^2			
4	ресурс 2	=12*B8+0,2*B8^2	=16*C8+0,2*C8^2	=СУММ(B4:C4)	<=	1400
5	ресурс 3	=17*B8+0,1*B8^2	=14*C8+0,15*C8^2	=СУММ(B5:C5)	<=	1580
6	Прибыль	=100*B8-0,08*B8^2	=120*C8-0,1*C8^2	=СУММ(B6:C6)	max	
7						
8	Измен. Перемен. (объем производства)					
9						
10	Нижняя граница изм. Переменных					
11		2	2			
12						

В ячейки B3:B6 и C3:C6 занесены соответственно формулы, отражающие элементы функции и левых частей. Поясним суть выражения в ячейке B3. В первом ограничении два первых слагаемых имеют вид  $15x_1 + 0,1x_1^2$ . Под значение  $x_1$  (изменяемой переменной) отведена ячейка B8, поэтому в ячейке B3 занесено выражение  $15*B8 + 0,1*B8^2$ . Аналогично занесены выражения в другие ячейки. В ячейках D3:D5 представлены формулы для подсчета расхода ресурсов на производство продукции в объемах  $x_1$  и  $x_2$ . Так как на производство продукции первого вида в объеме  $x_1$  расходуется первого ресурса  $15*B8 + 0,1*B8^2$ , а на производство продукции второго вида в объеме  $x_2$  расходуется того же ресурса  $18*C8 + 0,5*C8^2$  и эти величины находятся в ячейках B3 и C3, то суммарный расход первого ресурса занесен в ячейку D3, что отражено формулой =СУММ(B3:C3). Аналогично формулы занесены в

ячейки D4 и D5. В ячейку D6 занесена суммарная прибыль от производства продукции.

Далее в меню **Сервис** командой **Поиск решения** откроем диалоговое окно **Поиск решения** и занесем в него необходимые данные: адрес ячейки целевой функции – D6: направление оптимизации – максимизировать; адреса изменяемых переменных B8 и C8; ограничения задачи и требование целочисленности неизвестных величин.



В диалоговом окне **Параметры поиска решения**, вызванном командой **Параметры** диалогового окна **Поиск решения**, установим флажки **Неотрицательные значения**, **Автоматическое масштабирование**, **«Сопряженных градиентов»**. По команде **Выполнить**, получим оптимальное решение задачи.

	A	B	C	D	E	F	G
1	ограничения и др.	Переменные		Левая часть ограничений	Вид ограничений	Правая часть огр.	
2	Показатели	$x_1$	$x_2$				
3	ресурс 1	595,64	684,66	1280,31	<=	1350	
4	ресурс 2	604,18	795,82	1400,00	<=	1400	
5	ресурс 3	660,88	666,25	1327,13	<=	1580	
6	Прибыль	3176,60	4042,84	7219,44	max		
7							
8	Измен. Перемен. (объем производства)	33	35				
9							
10	Нижняя граница изм. Переменны	2	2				
11							
12							

В ячейках B8 и C8 представлены искомые объемы производства продукции  $x_1=33$  и  $x_2=35$ . Суммарная максимальная прибыль 7 219,44 ден.ед. представлена в ячейке D6. В ячейках B3:B5 и C3:C5 находится информация о

расходе ресурсов на производственную продукцию, а ячейках D3:D5 – о суммарном расходе ресурсов.

Результат поиска решения можно представить в виде отчета. Отчет по результатам решения задачи НЛП не отличается от отчета для задач линейной оптимизации, а отчеты по устойчивости и пределам в задачах с требованием целочисленности не выдаются. Учитывая, что отчет по устойчивости для нелинейных задач отличается от аналогичного отчета для линейных задач, решим рассматриваемую задачу без требования целочисленности. В отчете приведены числовые значения изменяемых переменных (значения неизвестных), нормированный градиент, а ниже – данные об ограничениях (расходов ресурсов и множители Лагранжа). Множители Лагранжа – величины, аналогичные двойственным оценкам в задачах линейной оптимизации. Они показывают, насколько изменится значение целевой функции в оптимальном решении при изменении правой части системы ограничений на единицу.

В нашем примере множители Лагранжа  $\lambda_1=\lambda_3=0$ ,  $\lambda_2=3,78$ . Это говорит о том, что ресурс 1 и ресурс 3 являются не дефицитными (они расходованы не полностью), ресурс 2 дефицитный (расходован полностью). Если увеличить ресурс 2 на единицу, то значение функции увеличится на  $\lambda_2=3,78$  ден.ед. Если этот ресурс увеличить на 2 единицы, то прибыль увеличится на 7,56 ден.ед. и т.д., пока ресурс будет оставаться дефицитным.

	A	B	C	D	E	I
1	Microsoft Excel 11.0 Отчет по устойчивости					
2	Рабочий лист: [задача нелинейного программирования.xls]Лист1					
3	Отчет создан: 07.11.2005 14:37:56					
4						
5						
6	Изменяемые ячейки					
7						
8		<b>Ячейка</b>	<b>Имя</b>	<b>Результ. значение</b>	<b>Нормир. градиент</b>	
9		\$B\$8	Измен. Перемен. (объем производства) X1	33	0	
10		\$C\$8	Измен. Перемен. (объем производства) X2	35	0	
11						
12	Ограничения					
13						
14		<b>Ячейка</b>	<b>Имя</b>	<b>Результ. значение</b>	<b>Лагранжа Множитель</b>	
15		\$D\$3	ресурс 1 Левая часть ограничений	1280,31	0,00	
16		\$D\$4	ресурс 2 Левая часть ограничений	1400,00	3,78	
17		\$D\$5	ресурс 3 Левая часть ограничений	1327,13	0,00	
18						

## Упражнения

1. Предприятие может производить продукцию по двум технологическим способам производства. Для производства продукции используется сырье двух видов. Требуется определить, сколько продукции производить по каждому из технологических способов, чтобы получить максимальную прибыль, если известно: объемы ресурсов, которыми располагает предприятие,  $b_1=186$ ,  $b_2=210$ , оптовая цена единицы продукции,  $P_1=52$ ,  $P_2=68$ , себестоимость  $c_j$ , выраженная функцией  $c_j = c'_j + c''_j x_j$ ,  $j=1,2$  ( $c'_1=1.0$ ,  $c''_1=0.1$ ,  $c'_2=2.0$ ,  $c''_2=0.1$ ), и нормы расхода ресурсов на производство единицы продукции по каждому технологическому способу  $a_{ij}$  ( $a_{11}=6$ ;  $a_{12}=4$ ;  $a_{21}=5$ ;  $a_{22}=10$ ). Решить задачу на ПЭВМ методом сопряженных градиентов.

2. Найти минимум функции  $f = x_1^2 - 10x_1 + x_2^2 - 4x_2$  при ограничениях:

$$\begin{cases} 2x_1 + 5x_2 \leq 30 \\ 8x_1 + 4x_2 \leq 48 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

3. Найти методом множителей Лагранжа решение задачи. Предприятие может изготовить 200 изделий по двум технологическим способам производства. При производстве одного изделия первым технологическим способом себестоимость его равна  $6+x_1$ , а вторым способом –  $2+x_2$ ,  $x_1$  и  $x_2$  – объемы производства продукции по соответствующему технологическому способу. Требуется найти, сколько изделий изготовить по каждому из способов производства, чтобы себестоимость произведенной продукции была минимальной.

С учетом обозначений минимальная суммарная себестоимость продукции выразится как

$f(x) = (6 + x_1)x_1 + (2 + x_2)x_2 = 6x_1 + x_1^2 + 2x_2 + x_2^2 \rightarrow \min$ , при ограничении-равенстве на объем производимых изделий

$$\begin{cases} x_1 + x_2 = 200 \\ x_1 \geq 0, x_2 \geq 0 \end{cases}$$

Составим функцию Лагранжа и найдем минимальное значение функции  $f(x)$  без учета условия неотрицательности неизвестных.

$$L(x_1, x_2, \lambda) = 6x_1 + x_1^2 + 2x_2 + x_2^2 + \lambda(200 - x_1 - x_2)$$

$$\begin{cases} \frac{\partial L}{\partial x_1} = 6 - 2x_1 - \lambda = 0 \\ \frac{\partial L}{\partial x_2} = 2 + 2x_2 - \lambda = 0 \\ \frac{\partial L}{\partial \lambda} = 200 - x_1 - x_2 = 0 \end{cases}$$

4. Найти множителем Лагранжа максимальное значение функции  $f=x_1x_2$  при условиях

$$\begin{cases} x_1^2 + x_2^2 = 10 \\ x_1 - x_2 = 2 \end{cases}$$

## Варианты задач для самостоятельной работы

а) Решить задачу без ограничений.

б) Решить задачу с линейными ограничениями.

### Вариант 1

$$z = 4x_1^2 - 3x_1x_2 + 5x_2^2 - x_1 + 15$$

$$\begin{cases} 2x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 3

$$z = 5x_1^2 - 2x_1x_2 + 3x_2^2 + x_1 + x_2 - 15$$

$$\begin{cases} 2x_1 + 3x_2 \geq 15, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 5

$$z = 3x_1^2 - 2x_1x_2 + 4x_2^2 - x_1 - x_2 + 10$$

$$\begin{cases} 2x_1 + 4x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 7

$$z = x_1^2 - 2x_1x_2 + 5x_2^2 - x_1 - x_2 + 7$$

$$\begin{cases} 3x_1 + 5x_2 \geq 4, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 9

$$z = x_1^2 - 2x_1x_2 + 10x_2^2 - 3x_1 + 8$$

$$\begin{cases} 2x_1 + x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 11

$$z = 2x_1^2 - 8x_1x_2 + 10x_2^2 - x_2 + 8$$

$$\begin{cases} x_1 + x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 2

$$z = 3x_1^2 - x_1x_2 + x_2^2 - x_1 - x_2 + 3$$

$$\begin{cases} x_1 + 3x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 4

$$z = x_1^2 - x_1x_2 + 5x_2^2 - x_2 - 7$$

$$\begin{cases} x_1 + 2x_2 \geq 3, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 6

$$z = 2x_1^2 - x_1x_2 + x_2^2 - x_2 + 1$$

$$\begin{cases} x_1 + 4x_2 \geq 7, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 8

$$z = 3x_1^2 - x_1x_2 + 4x_2^2 - x_1 - x_2 + 4$$

$$\begin{cases} 3x_1 + x_2 \geq 20, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 10

$$z = 3x_1^2 - 4x_1x_2 + 5x_2^2 - x_1 - 4x_2 + 6$$

$$\begin{cases} x_1 + 4x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

### Вариант 12

$$z = 3x_1^2 - 2x_1x_2 + 3x_2^2 - x_1 + 10$$

$$\begin{cases} 2x_1 + 3x_2 \geq 20, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 13**

$$z = 4x_1^2 - x_1x_2 + 5x_2^2 - x_2 + 25$$

$$\begin{cases} x_1 + 4x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 15**

$$z = 7x_1^2 - 3x_1x_2 + x_2^2 - x_1 + 17$$

$$\begin{cases} 2x_1 + 3x_2 \geq 7, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 17**

$$z = x_1^2 - x_1x_2 + 4x_2^2 - x_1 + 3x_2 + 19$$

$$\begin{cases} 3x_1 + 5x_2 \geq 15, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 19**

$$z = 3x_1^2 - 2x_1x_2 + 4x_2^2 - x_1 + x_2 + 20$$

$$\begin{cases} x_1 + 4x_2 \geq 15, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 21**

$$z = 2x_1^2 - 2x_1x_2 + 4x_2^2 - x_1 + 2$$

$$\begin{cases} x_1 + 10x_2 \geq 25, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 23**

$$z = 4x_1^2 - 3x_1x_2 + 8x_2^2 - x_1 + x_2 - 10$$

$$\begin{cases} 2x_1 + 4x_2 \geq 9, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 25**

$$z = 2x_1^2 - 3x_1x_2 + 4x_2^2 - x_1 + x_2 + 10$$

$$\begin{cases} 3x_1 + 2x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 14**

$$z = 5x_1^2 - 2x_1x_2 + 6x_2^2 - 3x_1 + 2x_2 - 10$$

$$\begin{cases} x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 16**

$$z = 8x_1^2 - 4x_1x_2 + 3x_2^2 - x_2 + 18$$

$$\begin{cases} x_1 + 3x_2 \geq 3, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 18**

$$z = 2x_1^2 - x_1x_2 + x_2^2 - x_1 + 10x_2 - 8$$

$$\begin{cases} 2x_1 + x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 20**

$$z = 4x_1^2 - x_1x_2 + 8x_2^2 - x_1 + 2x_2 + 30$$

$$\begin{cases} 2x_1 + 5x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 22**

$$z = 3x_1^2 - x_1x_2 + x_2^2 - x_1 + x_2 + 10$$

$$\begin{cases} 2x_1 + 3x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 24**

$$z = x_1^2 - x_1x_2 + 3x_2^2 - 3x_1 - 2x_2 + 15$$

$$\begin{cases} 2x_1 + 3x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 26**

$$z = 3x_1^2 - x_1x_2 + x_2^2 + x_1 - x_2 + 10$$

$$\begin{cases} 2x_1 + 3x_2 \geq 15, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 27**

$$z = 4x_1^2 - 2x_1x_2 + x_2^2 - x_1 + x_2 + 15$$

$$\begin{cases} x_1 + 4x_2 \geq 10, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 28**

$$z = -0,5x_1^2 - 0,5x_2^2 + x_1 + 2x_2 - 5$$

$$\begin{cases} 2x_1 + 3x_2 \leq 6, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 29**

$$z = x_1^2 - 2x_1x_2 + 2x_2^2 - 3x_1 - 7x_2 + 17$$

$$\begin{cases} 2x_1 + 3x_2 \geq 7, \\ x_1, x_2 \geq 0 \end{cases}$$

**Вариант 30**

$$z = 2x_1^2 - 4x_1x_2 + x_2^2 - x_2 + 15$$

$$\begin{cases} x_1 + 4x_2 \geq 5, \\ x_1, x_2 \geq 0 \end{cases}$$

Найти экстремумы функций методом множителей Лагранжа.

**1**  $z = 6 - 4x_1 - 3x_2$ , при  $x_1^2 + x_2^2 = 1$

**2**  $z = x_1x_2$ , при  $x_1 + x_2 = 1$

**3**  $z = x_1^2 + x_2^2$ , при  $x_1 + x_2 = 1$

**4**  $z = x_1^2 - x_2^2$ , при  $x_1 - x_2 = 4$

**5**  $z = \frac{x_1}{2} + \frac{x_2}{3}$ , при  $x_1^2 + x_2^2 = 1$

**6**  $z = x_1^2 + x_2^2$ , при  $\frac{x_1}{2} + \frac{x_2}{3} = 1$

**7**  $z = 4x_1^2 + 10x_2^2 + 5x_1$ , при  $x_1 + x_2 = 8$ ,  $x_1, x_2 \geq 0$

**8**  $z = (x_1 - 2)^2 + (x_2 - 3)^2$ , при  $2x_2 - 2x_1 = 6$

**9**  $z = x_1 - 2x_2 + 2x_3$ , при  $x_1^2 + x_2^2 + x_3^2 = 9$

**10**  $z = x_1x_2^2x_3^3$ , при  $x_1 + x_2 + x_3 = 12$

**11**  $z = x_1x_2 + x_2x_3$ , при  $x_1^2 + x_2^2 = 2$ ,  $x_2 + x_3 = 2$ ,  $x_1, x_2, x_3 \geq 0$

**12**  $z = x_1^2 + 2x_2^2 + 2x_3^2$ , при  $x_1^2 + x_2^2 + x_3^2 = 1$ ,  $x_1 + 2x_2 + 3x_3 = 0$

**13**  $z = x_1x_2x_3$ , при  $x_1 + x_2 + x_3 = 5$ ,  $x_1x_2 + x_2x_3 + x_1x_3 = 8$

**14**  $z = x_1^2 + x_2^2 + x_3$ , *npu*  $x_1 + x_2 + x_3 = 4$ ,  $2x_1 + 3x_2 = 12$

**15**  $z = x_1 x_2 x_3$ , *npu*  $2x_1 - x_2 = 8$ ,  $2x_1 x_2 + x_2 x_3 = 12$

**16**  $z = x_1 x_2 + x_2 x_3$ , *npu*  $x_1 + x_2 = 4$ ,  $x_2 + x_3 = 4$

**17**  $z = 3x_1^2 + 2x_2^2 + 2x_3^2 + 2x_1 + 4x_2 x_3$ , *npu*  $x_1^2 + 2x_2^2 = 19$ ,  $x_1 + x_2 x_3 = 11$

**18**  $z = x_1^2 x_2^3 x_3^4$ , *npu*  $x_1 + x_2 + x_3 = 18$

**19**  $z = 4x_1 + x_1^2 + 8x_2 + x_2^2$ , *npu*  $x_1 + x_2 = 180$ ,  $x_1, x_2 \geq 0$

## ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ MS EXCEL В ИГРАХ С ПРИРОДОЙ

Рассмотрим применение информационных технологий Excel для нахождения оптимальных стратегий в играх с природой на следующем примере.

**Пример.** Торговое предприятие планирует продажу сезонных товаров с учетом возможных вариантов поведения покупательского спроса ( $\Pi_1, \Pi_2, \Pi_3, \Pi_4$ ). Предприятием разработаны три хозяйственные стратегии продажи товаров ( $A_1, A_2, A_3$ ). Требуется найти оптимальное поведение торгового предприятия, пользуясь критериями Сэвиджа и Гурвица при  $\lambda=0,6$ , если товарооборот, зависящий от стратегий предприятия и покупательского спроса, представлен в виде платежной матрицы:

$A_i \backslash \Pi_j$	$\Pi_1$	$\Pi_2$	$\Pi_3$	$\Pi_4$
$A_1$	40	20	30	35
$A_2$	60	80	20	35
$A_3$	35	45	50	70

Начнем решение с критерия Гурвица, для чего разместим исходные данные примера в диапазоне ячеек A2:E6. Параметр  $\lambda=0,6$  занесем в ячейку A8, а  $(1-\lambda)=1-0,6=0,4$  – в ячейку B8. Справа от исходных данных в ячейку F4 занесем формулу  $\lambda \cdot \min_j(a_{ij})$ , которая с учетом размещения исходных данных в электронной таблице имеет вид: =МИН(B4:E4)\*A\$8. Занесем аналогичные формулы в ячейки F5 и F6. В ячейку G4 занесем формулу  $(\lambda-1) \cdot \min_j(a_{ij})$ , которая в Excel имеет вид: =МАКС(B4:E4)\*B\$8, и так же занесем соответствующие формулы в ячейки G5 и G6.

В ячейках H4:H6 найдем суммы  $S_i$ .  $S_i = \lambda \min_j a_{ij} + (1-\lambda) \cdot \max_j a_{ij}$ , а в ячейке H7 – наибольшую из этих сумм ( $\max(H4:H6)$ ), которая определяет выигрыш торгового предприятия и его оптимальную стратегию. Оптимальная стратегия  $A_i$  определяется логическим выражением, занесенным в ячейку H8.

Стратегии торгового предприятия	Стратегии покупательского спроса				$\alpha \cdot \min_j a_{ij}$	$(1 - \alpha) \cdot \min_j a_{ij}$	$S_j$
	$P_1$	$P_2$	$P_3$	$P_4$			
$A_1$	40	20	30	35	=МИН(В4:Е4)*А\$8	=МАКС(В4:Е4)*В\$8	=СУММ(Ф4:Г4)
$A_2$	60	80	20	40	=МИН(В5:Е5)*А\$8	=МАКС(В5:Е5)*В\$8	=СУММ(Ф5:Г5)
$A_3$	35	45	50	70	=МИН(В6:Е6)*А\$8	=МАКС(В6:Е6)*В\$8	=СУММ(Ф6:Г6)
Выигрыш торгового предприятия -					=МАКС(Н4:Н6)		
Оптимальная стратегия -					=ЕСЛИ(И(Н4>Н5;Н4>Н6);А4;ЕСЛИ(И(Н5>Н4;Н5>Н6);А5;А6))		

Чтобы найти оптимальную стратегию по критерию Сэвиджа, рассчитаем элементы матрицы рисков, для чего найдем сначала максимальный элемент  $\beta_j$  ( $j=1,2,3,4$ ) в каждом столбце матрицы выигрышей, а потом вычтем из этого элемента элементы соответствующего столбца той же матрицы рисков.

Чтобы занести формулы во все ячейки, достаточно занести формулу **=B\$16-B13** в ячейки B22:E22, а потом до ячейки E24. В ячейки F22:F24 занесем формулы для нахождения  $r_i = \max_j r_{ij}$ , а в ячейку F25 – формулу, определяющую минимальный риск  $r = \min_i r_i$ . В ячейку F26 помещено логическое выражение для определения оптимальной стратегии  $A_i$ .

Стратегии торгового предприятия		Стратегии покупательского спроса			
		$P_1$	$P_2$	$P_3$	$P_4$
$A_1$		40	20	30	35
$A_2$		60	80	20	40
$A_3$		35	45	50	70
		$=\text{МАКС}(B13;E13)$	$=\text{МАКС}(C13;E13)$	$=\text{МАКС}(D13;E13)$	$=\text{МАКС}(E13;E13)$
торгового предприятия		Стратегии покупательского спроса			
		$P_1$	$P_2$	$P_3$	$P_4$
$A_1$	$r_1$	$=B\$16-B13$	$=C\$16-C13$	$=D\$16-D13$	$=E\$16-E13$
$A_2$		$=B\$16-B14$	$=C\$16-C14$	$=D\$16-D14$	$=E\$16-E14$
$A_3$		$=B\$16-B15$	$=C\$16-C15$	$=D\$16-D15$	$=E\$16-E15$
					$r = \text{МИН}(F22:F24)$
					Оптимальная стратегия - $=\text{ЕСЛИ}(\text{И}(F22 < F23; F22 < F24); A22; \text{ЕСЛИ}(\text{И}(F23 < F22; F23 < F24); A23; A24))$

Получим числовые значения искомых параметров:

**по критерию Гурвица** – максимальный выигрыш торгового предприятия равный 49 и оптимальную стратегию  $A_3$ ;

**по критерию Сэвиджа** – минимальный риск, равный 30, и оптимальную стратегию  $A_2$ .

Предлагается найти оптимальную стратегию торгового предприятия по критерию Вальда.

## Упражнения

1. Объединение «Станкоэкспорт» ведет переговоры с одной из зарубежных фирм о продаже станков. Объединение в ходе переговоров стремится повысить эффективность заключаемого соглашения и для достижения цели может воспользоваться четырьмя возможными стратегиями ( $A_1, A_2, A_3, A_4$ ). Фирма имеет возможность применить три стратегии ( $P_1, P_2, P_3$ ), позволяющие снизить эффективность соглашения для объединения.

Прибыль объединения (тыс. ден. ед.) при возможных его стратегиях поведения  $A_i$  и стратегиях фирмы  $P_j$  представлена в таблице. Найти оптимальную стратегию объединения, пользуясь критериями Вальда и Сэвиджа.

I \ II	II		
	$P_1$	$P_2$	$P_3$
$A_1$	100	120	150
$A_2$	140	110	120
$A_3$	90	100	120
$A_4$	160	100	130

2. Сельскохозяйственное предприятие планирует выращивать капусту на одном из участков земли. На урожайность капусты влияют погодные условия и количество внесенных удобрений. Лето может сухое –  $P_1$ , нормальное (со средним количеством осадков) –  $P_2$  и влажное –  $P_3$ . Сельскохозяйственное предприятие может внести на 1 га количество удобрений, соответствующее норме  $A_1$ , больше нормы –  $A_2$  и меньше нормы –  $A_3$ . Требуется определить, какое количество удобрений внести на 1 га посева, чтобы получить наибольшую прибыль при самых неблагоприятных погодных условиях. Предприятие рассчитало прибыль в зависимости от погодных условий и возможных своих стратегий по внесению удобрений. Все данные расчетов (тыс. ден.ед.) приведены в таблице.

I \ II	II		
	$P_1$	$P_2$	$P_3$
$A_1$	45	50	30
$A_2$	25	52	43
$A_3$	32	40	27

3. Необходимо определить, какую электростанцию построить в одном из районов страны, чтобы эффективность использования капиталовложений была наибольшей при самых неблагоприятных условиях. Планирующий орган имеет три стратегии использования капиталовложений:  $A_1$  – вложить средства в гидроэлектростанцию,  $A_2$  – в тепловую,  $A_3$  – в атомную. Случайные факторы, влияющие на экономическую эффективность применяемых плановым органом стратегий, можно рассматривать как состояния природы ( $P_1, P_2$ ),  $P_1$  –

благоприятное состояние, а  $\Pi_2$  – неблагоприятное. Экономический эффект (млн. ден.ед.) трех стратегий планового органа оценен с учетом затрат на строительство и издержек в процессе эксплуатации, зависящих от состояний природы, и представлен в таблице.

I \ II	$\Pi_1$	$\Pi_2$
$A_1$	7,0	3,5
$A_2$	5,0	6,0
$A_3$	4,3	4,0

4. Оперирующей стороной принято решение о строительстве крупного завода по производству автомобилей по лицензии одной из зарубежных фирм. Предварительный анализ проектов показал, что наиболее предпочтительными для закупки лицензий являются три фирмы, поэтому стратегиями оперирующей стороны являются:  $A_1$  – закупить лицензию первой фирмы,  $A_2$  – второй и  $A_3$  – третьей фирмы. технико-экономические параметры автомобилей (грузоподъемность, комфортабельность, экономичность двигателя, скорость и др.) этих трех фирм существенно между собой не различаются, однако параметры соответствия выпускаемого автомобиля перспективному техническому уровню каждой из фирм значительно отличаются, что влияет на экономическую эффективность приобретения лицензии. Можно выделить два состояния:  $\Pi_1$  – выпуск автомобиля по купленной лицензии будет рентабельным и соответствовать техническому уровню через 10 лет,  $\Pi_2$  – через 15 лет. Значение экономического эффекта закупки лицензии в зависимости от состояний  $\Pi_1$  и  $\Pi_2$  приведено в таблице.

I \ II	$\Pi_1$	$\Pi_2$
$A_1$	45	54
$A_2$	39	50
$A_3$	34	42

Требуется выбрать такую лицензию, чтобы экономический эффект (млн. ден.ед) в худших условиях был максимальным. Решить задачу по критерию Вальда и Сэвиджа.

## СПИСОК ЛИТЕРАТУРЫ

1. Афанасьев М.Ю., Суворов Б.П. Исследование операций в экономике. Учебное пособие. – М.: Изд. Инфра-М, 2003,-443 с.
2. Балдин К. В. Информационные системы в экономике [Текст]: учебник / К. В. Балдин, В. Б. Уткин. - Москва: Дашков и К', 2009. - 395 с.
3. Бережная Е.В. Математические методы моделирования экономических систем / Е.В. Бережная, В.И. Бережной. – М.: Финансы и статистика, 2002.
4. Гареева Г.А., Еремина И.И., Файзуллина А.Г. Линейное программирование. Элементы теории, алгоритмы и примеры: учебное пособие – Набережные Челны: Издательско-полиграфический центр НЧИ КФУ, 2015. – 148с.: ил.
5. Гвоздева В. А. Основы построения автоматизированных информационных систем [Текст]: учебник / В. А. Гвоздева, И. Ю. Лаврентьева. - Москва: ФОРУМ, 2009. - 320 с.
6. Гетманчук, А.В. Экономико-математические методы и модели: Учебное пособие для бакалавров / А.В. Гетманчук, М. М. Ермилов. — М.: Издательско-торговая корпорация «Дашков и Ко», 2013. - 188 с.
7. Голицына О.Л. Информационные системы [Электронный ресурс]: учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - 2-е изд. - Москва: Форум: НИЦ ИНФРА-М, 2014. – 448 с.: ил. - (Высшее образование). ISBN 978-5-91134-833-5. – Режим доступа: <http://znanium.com/bookread.php?book=435900> .
8. Информационные системы и технологии управления [Текст]: учебник для вузов / [И. А. Коноплева и др.]; под ред. Г. А. Титоренко. - Москва: ЮНИТИ-ДАНА, 2010. - 591 с.
9. Информационные технологии в экономике: Учеб. пособие / Н.Н. Карабутов. – М.: Экономика, 2012. – 208 с.: ил.

10. Информационные технологии: учебник для студ. Высш. Учеб. Заведений / В.П. Мельников. – 2-е изд., стер. – М.: Издательский центр «Академия», 2009. – 432 с.
11. Колемаев, В.А. Математические методы и модели исследования операций: учебник для студентов вузов, обучающихся по специальности 080116 «Математические методы в экономике» и другим экономическим специальностям / В. А. Колемаев; под ред. В. А. Колемаева. - М.: ЮНИТИ-ДАНА, 2012. - 592 с.
12. Компьютерное моделирование в системе Mathcad: учеб. пособие. – М.: Финансы и статистика, 2006. – 144 с.: ил.
13. Компьютерный практикум для менеджеров: информационные технологии и системы: учеб. пособие / Л.Г. Гагарина, Е.М. Портнов, И.С. Холод; под ред. Л.Г. Гагариной. – М.: Финансы и статистика, 2006. – 352 с.: ил.
14. Красс М.С. Математика для экономических специальностей. – 4-е изд. – М.: Дело, 2003.
15. Красс М.С. Математические методы и модели для магистрантов экономики / М.С. Красс, Б.П. Чупрынов. – М.-СПб: Питер, 2004.
16. Кремер, Н.Ш. Исследование операций в экономике: учеб. пособие для студ. вузов по экон. спец. и направл. / под ред. Н.Ш. Кремера. - 2-е изд., доп. и перераб. - М.: Юрайт, 2011. - 430 с.
17. Кузнецов А.В. и др. Высшая математика: Мат. программир.: Учеб./А.В. Кузнецов, В.А. Сакович, Н.И. Холод; Под общ. ред. А.В. Кузнецова. – Мн.: Высш. шк., 1994. – 286 с.: ил.
18. Максимов Н. В. Современные информационные технологии [Текст]: учебник / Н. В. Максимов, Т. Л. Партыка, И. И. Попов. - Москва: ФОРУМ, 2012. - 512 с.
19. Математическое программирование: Информационные технологии оптимальных решений: Учеб. пособие / Л.С. Костевич. – Мн.: Новое знание, 2013. – 424 с.: ил.

20. Орлова, И.В. Экономико-математическое моделирование: Практическое пособие по решению задач / И.В. Орлова. - 2-е изд., испр. и доп. - М.: Вузовский учебник: НИЦ ИНФРА-М, 2013. - 140 с.
21. Пикуза В., Геращенко А. Экономические и финансовые расчеты в Excel: Самоучитель. СПб.: Питер, 2012.
22. Решение математических задач средствами Excel: Практикум / В.Я. Гельман. – СПб.: Питер 2010. – 240 с.: ил.
23. Румянцева Е.Л. Информационные технологии [Электронный ресурс]: учебное пособие / Е.Л. Румянцева, В.В. Слюсарь; под ред. Л.Г. Гагариной. - Москва: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 256 с.: ил. - ISBN 978-5-8199-0305-6. – Режим доступа: <http://znanium.com/bookread.php?book=392410> .
24. Федотова Е.Л. Информационные технологии и системы [Электронный ресурс]: учебное пособие / Е.Л. Федотова. - Москва: ИД ФОРУМ: НИЦ ИНФРА-М, 2014. - 352 с.: ил. - (Высшее образование). ISBN 978-5-8199-0376-6. – Режим доступа: <http://znanium.com/bookread.php?book=429113> .

# ПРИЛОЖЕНИЯ

## Приложение 1. Текст программы на языке Турбо Паскаль

```
PROGRAM SIMPLEX_METOD;
USES CRT;
LABEL ZN,ST,ELL,_END;
TYPE MAS=ARRAY[1..30] OF REAL;
MASB=ARRAY[1..30] OF STRING[3];
MASX=ARRAY[1..30,1..30] OF REAL;
VAR Fo,FunctPr,B,H,Hnew,C,Cnew,CPr,CPrnew,FX:MAS;
X,Xnew:MASX;
BS,Bvsp,ZNAC:MASB;
MIN,I1,I,J,Kx,Ky,Kit,NachKell,NachY,K_st:INTEGER;
PriznacY,KLstr,KLst,ErrCode,Dop_X:INTEGER;
P,P1,Mo,F0,Epsilon,Z:REAL;
VSP,S,PrGomory:STRING;
F:TEXT;
DPx,DPy,Fm,Kell,Kstr:INTEGER;

FUNCTION SIMVB(V:INTEGER;S:CHAR):STRING; { Функция создания индексов }
VAR M,Z:STRING;
BEGIN
STR(V,M);
Z:=S+M;
SIMVB:=Z;
END;

PROCEDURE SAVE(X1:REAL;K:STRING;Mstr:INTEGER);{ Процедура записи данных в } VAR
V:STRING; { файл }
BEGIN
ASSIGN(F,'SIMPLEX.DAT');
APPEND(F);
CASE Mstr OF
0:WRITELN(F,'');
1:BEGIN IF K=' ' THEN STR(X1:1:0,V) ELSE STR(X1:10:4,V);
WRITE(F,V); WRITE(F,' ');
END;
2:WRITE(F,K);
3:WRITELN(F,K);
END;
CLOSE(F);
END;

PROCEDURE DOP_PER; { Определение дополнительных переменных }
BEGIN
IF ZNAC[I1]='=' THEN
BEGIN
Kell:=Kell+1;Bvsp[Kell]:=SIMVB(DPy,'Y');
DPy:=DPy+1;
Xnew[I1,Kell]:=1;
IF Fm=1 THEN FX[Kell]:=-1 ELSE FX[Kell]:=1;
FunctPr[Kell]:=1;
FOR I:=1 TO Kstr DO IF I<>I1 THEN Xnew[I,Kell]:=0;
END;
IF ZNAC[I1]='>=' THEN
BEGIN
Kell:=Kell+1;Bvsp[Kell]:=SIMVB(DPx,'X');
DPx:=DPx+1;Dop_X:=Dop_X+1;
Xnew[I1,Kell]:=-1;FX[Kell]:=0;

FOR I:=1 TO Kstr DO IF I<>I1 THEN Xnew[I,Kell]:=0;
```

```

    Kell:=Kell+1;Bvsp[Kell]:=SIMVB(DPy, 'Y');
    DPy:=DPy+1;
    Xnew[I1,Kell]:=1;
    IF Fm=1 THEN FX[Kell]:=-1 ELSE FX[Kell]:=1;
    FunctPr[Kell]:=1;
    FOR I:=1 TO Kstr DO IF I<>I1 THEN Xnew[I,Kell]:=0;
END;

IF ZNAC[I1]='<=' THEN
BEGIN
    Kell:=Kell+1;Bvsp[Kell]:=SIMVB(DPx, 'X');
    DPx:=DPx+1;Dop_X:=Dop_X+1;
    Xnew[I1,Kell]:=1;FX[Kell]:=0;
    FOR I:=1 TO Kstr DO IF I<>I1 THEN Xnew[I,Kell]:=0;

END;
END;

PROCEDURE SOKR;          { Процедура сокращения Y }
VAR P:INTEGER;
BEGIN
    Kell:=Kell-1;
    FOR P:=NachKell+DOP_X TO Kell DO
        IF Bvsp[P]=BS[KLstr] THEN BEGIN
            FOR J:=P TO Kell DO
                Bvsp[J]:=Bvsp[J+1];
                FunctPr[J]:=FunctPr[J+1];
                Fx[J]:=Fx[J+1];
                FOR I:=1 TO Kstr DO
                    Xnew[I,J]:=Xnew[I,J+1]
                END;
        END;
    END;

PROCEDURE GOMORY; { Процедура, выполняющая метод Гомори }
VAR MAX,Z:REAL;
BEGIN
    KLstr:=1;
    MAX:=H[1]-INT(H[1]);
    FOR I1:=2 TO Kstr DO
        IF (H[I1]-INT(H[I1]))>=MAX THEN BEGIN MAX:=H[I1]; KLstr:=I1;END;
    Kstr:=Kstr+1;
    Hnew[Kstr]:=H[KLstr]-INT(H[KLstr]);
    FOR I1:=1 TO Kell DO
        BEGIN
            Z:=INT(X[KLstr,I1]);
            IF X[KLstr,I1]<0 THEN Z:=Z-1;
            Xnew[Kstr,I1]:=X[KLstr,I1]-Z;
        END;
    ZNAC[Kstr]:='>=';
END;

PROCEDURE SIMPLEX; { Процедура, выполняющая Симплекс метод }
    LABEL POVZNAC,NACH;
BEGIN
    { Подготовка к вводу данных }
    NachKell:=Kell;
    DPx:=Kell+1;DPy:=1;
    Kx:=1;Ky:=4;
    Epsilon:=0.00001;
    CLRSCR;
    WRITELN('Введите систему уравнений:');
    WRITELN('(коэффициенты при всех X, знак и свободные члены)');

    { Ввод данных }

```

```

FOR I:=1 TO Kstr DO
  BEGIN
POVZNAC:
  WRITELN('Введите ',I,'-е уравнение:');

      { Ввод коэффициентов при X в I-том уравнении }

FOR J:=1 TO Kell DO
  BEGIN
    GOTOXY(Kx,Ky);Kx:=Kx+6;
    READLN(Xnew[I,J]);
  END;

      { Ввод знака в I-том уравнении }

Kx:=Kx+6;GOTOXY(Kx,Ky);READLN(ZNAC[I]);

      {Проверка введенного знака на правильность}

IF (ZNAC[I]<>'>=') AND (ZNAC[I]<>'=') AND (ZNAC[I]<>'<=')
THEN BEGIN
  WRITELN('Неправильно задан знак');
  Ky:=Ky+3;Kx:=1;
  GOTO POVZNAC;
  END;
IF (ZNAC[I]='=') OR (ZNAC[I]='>=') THEN PriznacY:=1;

      { Ввод свободного члена в I-том уравнении }

Kx:=Kx+6;GOTOXY(Kx,Ky);READ(B[I]);
Kx:=1;
Ky:=Ky+2;
END;
WRITELN('Введите коэффициенты при X в целевой функции:');

      { Ввод коэффициентов при X в целевой функции }
FOR J:=1 TO Kell DO
  BEGIN
    GOTOXY(Kx,Ky);Kx:=Kx+6;
    READ(FX[J]);
  END;

      { Подготовка индексации X }
FOR J:=1 TO Kell DO
  Bvsp[J]:=SIMVB(J,'X');

      { Определение дополнительных переменных }
FOR I1:=1 TO Kstr DO
  DOP_PER;

      { Замена оптимальной функции с MAX на MIN при наличии
        в базисе Y-ков если идет исследование на минимум }
MIN:=0;
IF (Fm=1) AND (PriznacY=1) THEN
  BEGIN
    MIN:=Fm;Fm:=2;
    FOR J:=1 TO Kell DO
      FX[J]:=-FX[J];
    END;

      { Сортировка дополнительных переменных по индексу }
FOR I1:=NachKell+1 TO Kell DO
  FOR J:=I1+1 TO Kell DO
    IF Bvsp[J]<Bvsp[I1] THEN
      BEGIN

```

```

VSP:=Bvsp[J];Bvsp[J]:=Bvsp[I1];Bvsp[I1]:=VSP;
P:=FX[J];FX[J]:=FX[I1];FX[I1]:=P;
P:=FunctPr[J];FunctPr[J]:=FunctPr[I1];FunctPr[I1]:=P;
FOR I:=1 TO Kstr DO
BEGIN
P:=Xnew[I,I1];Xnew[I,I1]:=Xnew[I,J];Xnew[I,J]:=P;
END;
END;
Kit:=1;
CLRSCR;

      { Подготовка столбцов С,В,Н }
FOR I:=1 TO Kstr DO
BEGIN
Hnew[I]:=B[I];
FOR J:=NachKell+1 TO Kell DO
IF Xnew[I,J]=1 THEN
BEGIN
BS[I]:=Bvsp[J];
Cnew[I]:=FX[J];
CPrnew[I]:=FunctPr[J];
END;
END;
NACH:;

REPEAT
PriznacY:=0;
      { Передача данных в исходные переменные с обнулением чисел,
по модулю меньших чем 0.00001 }
FOR I:=1 TO Kstr DO
BEGIN
IF INT(10000*Hnew[I])=0 THEN H[I]:=+0 ELSE H[I]:=Hnew[I];
C[I]:=Cnew[I];
CPr[I]:=CPrnew[I];
IF BS[I][1]='Y' THEN PriznacY:=1;
FOR J:=1 TO Kell DO
IF INT(10000*Xnew[I,J])=0 THEN X[I,J]:=+0 ELSE X[I,J]:=Xnew[I,J];
END;

      { Обнуление и вывод индексации элементов индексной строки }
SAVE(0,' С Б Н ',2);
FOR J:=1 TO Kell DO
BEGIN
SAVE(0,Bvsp[J],2);
P1:=LENGTH(Bvsp[J]);
IF P1=2 THEN SAVE(0,' ',2);
SAVE(0,' ',2);
Fo[J]:=0;
END;
SAVE(0,' ',0);

      { Вывод Симплекс-таблицы }
P1:=0;
FOR I:=1 TO Kstr DO
BEGIN
IF CPr[I]=1 THEN
IF C[I]<0 THEN SAVE(0,'-M ',2)
ELSE SAVE(0,'+M ',2)
ELSE SAVE(C[I],' ',1);
SAVE(0,BS[I],2);
P1:=LENGTH(BS[I]); IF P1=2 THEN SAVE(0,' ',2);
SAVE(0,' ',2);SAVE(H[I],' ',1);
FOR J:=1 TO Kell DO
SAVE(X[I,J],' ',1);

```

```

SAVE(0, '', 0);
END;
      { Вычисление значений в индексной строке }
F0:=0;
FOR J:=1 TO Kell DO
  Fo[J]:=0;
FOR I1:=1 TO Kstr DO
  BEGIN
  IF PriznacY=1 THEN
    IF BS[I1][1]='Y' THEN
      BEGIN
        F0:=F0+H[I1];
        FOR J:=1 TO Kell DO
          Fo[J]:=Fo[J]+X[I1,J];
        END;
      IF PriznacY=0 THEN
        BEGIN
          F0:=F0+H[I1]*C[I1];
          FOR J:=1 TO Kell DO
            Fo[J]:=Fo[J]+C[I1]*X[I1,J];
          END;
        FOR J:=1 TO Kell DO
          IF Bvsp[J][1]='Y' THEN Fo[J]:=+0
            ELSE IF ABS(Fo[J])<Epsilon THEN Fo[J]:=+0;
        END;

          { Вывод значений целевой функции }
SAVE(0, '          ', 2);SAVE(F0, '', 1);
FOR J:=1 TO Kell DO
  BEGIN
    IF PriznacY<>1 THEN Fo[J]:=Fo[J]-FX[J];
    SAVE(Fo[J], '', 1);
  END;
SAVE(0, '', 0);

      { Проверка условия оптимальности }
P:=0;
FOR J:=1 TO Kell DO
  IF Fm=1 THEN IF Fo[J]<-Epsilon THEN
    BEGIN
      P:=1;
      CONTINUE;
    END ELSE
    ELSE IF Fo[J]>Epsilon THEN
      BEGIN
        P:=1;
        CONTINUE;
      END;
IF P<>1 THEN
  BEGIN
    SAVE(0, 'В ', 2);SAVE(Kit, ' ', 1);
    SAVE(0, '-й итерации было получено оптимальное решение', 3);
    SAVE(0, 'т.к. при исследовании на ', 2);
    IF Fm=1 THEN
      SAVE(0, 'МАКСИМУМ индексная строка не содержит отрицательных элементов.', 3)
    ELSE
      SAVE(0, 'МИНИМУМ индексная строка не содержит положительных элементов.', 3);
    FOR I1:=1 TO Kstr DO
      IF BS[I1][1]='Y' THEN
        BEGIN
          SAVE(0, 'Но т.к. из базиса не выведены все Y, то ', 3);
          SAVE(0, 'можно сделать вывод, что РЕШЕНИЙ НЕТ', 3);
          HALT;
        END;

```

```

        { Округление значений массива X до целого числа,
          если разность округленного и обычного значений
          по модулю меньше чем 0.00001 }
FOR I:=1 TO Kstr DO
  BEGIN
    Z:=ROUND(H[I]);
    IF ABS(Z-H[I])<Epsilon THEN H[I]:=ROUND(H[I]);
    FOR J:=1 TO Kell DO
      BEGIN
        IF X[I,J]<0 THEN Z:=ROUND(X[I,J]);
        IF ABS(Z-X[I,J])<Epsilon THEN X[I,J]:=ROUND(X[I,J]);
      END;
    END;

    { Проверка целочисленности решения }

P1:=0;
FOR I:=1 TO Kstr DO
  BEGIN
    IF INT(10000*FRAC(H[I]))<>0 THEN BEGIN P1:=1;CONTINUE; END;

    FOR J:=1 TO Kell DO
      IF BS[I]=Bvsp[J] THEN
        FOR I1:=1 TO Kstr DO
          IF ABS(FRAC(X[I1,J]))>=Epsilon THEN BEGIN P1:=1;CONTINUE; END;
        END;

        { Составление новой базисной строки для целочисленного решения }

    IF (PrGomory='Y') AND (P1=1) THEN
      BEGIN
        GOMORY;
        NachKell:=Kell;
        I1:=Kstr;DPY:=1;
        DOP_PER;
        BS[Kstr]:=Bvsp[Kell];
        CPrnew[Kstr]:=FunctPr[Kell];
        Cnew[Kstr]:=FX[Kell];
        GOTO NACH;
      END;

    IF P1=0 THEN SAVE(0,'Данное решение является целочисленным.',3);

    SAVE(0,'При этом:',3);
    IF MIN=1 THEN BEGIN F0:=-F0;Fm:=MIN; END;
    IF Fm=1 THEN
      SAVE(0,'Fmax=',2)
      ELSE
      SAVE(0,'Fmin=',2);

      SAVE(F0,' ',1);
      SAVE(0,' ',0);

    FOR I1:=1 TO Kstr DO
      BEGIN
        SAVE(0,' ',2);
        SAVE(0,BS[I1],2);SAVE(0,'=',2);
        SAVE(H[I1],' ',1);
        SAVE(0,' ',0);
      END;
    HALT;

```

```

END;

      { Нахождение ключевого столбца }

KLst:=1;Mo:=0;
FOR J:=1 TO Kell DO
  IF Fm=1 THEN
    IF Fo[J]<Mo THEN Mo:=Fo[J];

FOR J:=1 TO Kell DO
  BEGIN
    IF Bvsp[J][1]<>'Y' THEN
      IF Fm=1 THEN
        BEGIN
          IF Fo[J]<0 THEN
            IF Fo[J]>=Mo THEN
              BEGIN
                Mo:=Fo[J]; KLst:=J;
              END;
            END
          ELSE
            BEGIN
              IF Fo[J]>0 THEN
                IF Fo[J]>=Mo THEN
                  BEGIN
                    Mo:=Fo[J]; KLst:=J;
                  END;
                END;
              END;
            END;
          END;
        END;

SAVE(0,'Ключевой столбец: ',2);SAVE(KLst,' ',1);

      { Нахождение ключевой строки }

Pl:=0;K_st:=0;
FOR J:=1 TO Kell DO
  IF ABS(Mo-Fo[J])<Epsilon THEN
    BEGIN
      K_st:=K_st+1;
      FOR I:=1 TO Kstr DO
        IF X[I,KLst]>0 THEN BEGIN B[I]:=H[I]/X[I,KLst]; P:=B[I];KLstr:=I; END
          ELSE BEGIN B[I]:=-1; Pl:=Pl+1; END;
        END;
      END;

IF Pl=Kstr*K_st THEN
  BEGIN
    SAVE(0,' ',0);
    SAVE(0,'РЕШЕНИЙ НЕТ т.к. невозможно определить ключевую строку',3);
    HALT;
  END;

Pl:=0;
FOR J:=1 TO Kell DO
  IF ABS(Mo-Fo[J])<Epsilon THEN
    FOR I:=1 TO Kstr DO
      IF B[I]>=0 THEN BEGIN
        IF B[I]<P THEN IF Bvsp[KLst]<>BS[I] THEN BEGIN P:=B[I]; KLstr:=I; END;

        IF INT(10000*B[I])=INT(10000*P) THEN
          IF (BS[I][1]='Y') AND (BS[KLstr][1]='X') THEN
            IF Bvsp[KLst]<>BS[I] THEN BEGIN P:=B[I]; KLstr:=I; END;
          END;
        END;
      END;
    END;
  SAVE(0,'Ключевая строка: ',2);SAVE(KLstr,' ',1);
  SAVE(0,' ',0);

```

```

FOR I:=1 TO Kstr DO
  IF Bvsp[KLst]=BS[I] THEN
    BEGIN
      SAVE(0,'РЕШЕНИЙ НЕТ т.к. в базисном столбце уже есть ',3);
      SAVE(0,'такая переменная.',3);
      HALT;
    END;

IF CPr[KLstr]=1 THEN SOKR; { Вызов процедуры сокращения Y }

      { Построение следующей Симплекс-таблицы }
BS[KLstr]:=Bvsp[KLst];
Cnew[KLstr]:=FX[KLst];
CPrnew[KLstr]:=FunctPr[KLst];
FOR I:=1 TO Kstr DO
  BEGIN
    IF I=KLstr THEN Hnew[I]:=H[I]/X[KLstr,KLst]
      ELSE Hnew[I]:=H[I]-(H[KLstr]*X[I,KLst]/X[KLstr,KLst]);
    FOR J:=1 TO Kell DO
      BEGIN
        IF (I=KLstr) AND (J=KLst) THEN Xnew[I,J]:=1;
        IF (I=KLstr) AND (J<>KLst) THEN Xnew[I,J]:=X[I,J]/X[KLstr,KLst];
        IF (I<>KLstr) AND (J=KLst) THEN Xnew[I,J]:=0;
        IF (I<>KLstr) AND (J<>KLst) THEN
          Xnew[I,J]:=X[I,J]-(X[KLstr,J]*X[I,KLst]/X[KLstr,KLst]);
        END;
      END;
    END;
  END;
  KLst:=0;KLstr:=0;
  Kit:=Kit+1;
  UNTIL (Kit=0);
END; {SIMPLEX}

BEGIN          { Основная программа }
CLRSCR;
Kit:=0;Dop_X:=0;
ASSIGN(F,'SIMPLEX.DAT');
REWRITE(F);
CLOSE(F);
ST;;
WRITE('Введите кол-во строк:');READLN(Kstr);
IF Kstr>10 THEN
  BEGIN
    WRITELN('Программа не рассчитана на введенное кол-во строк!');
    GOTO ST;
  END;

ELL: WRITE('Введите кол-во элементов:');READLN(Kell);
IF Kell>10 THEN
  BEGIN
    WRITELN('Программа не рассчитана на введенное кол-во элементов!');
    GOTO ELL;
  END;
ZN: WRITE('Исследуем на МАКСИМУМ(1) или МИНИМУМ(2):');READLN(Fm);
IF (Fm<>1) AND (Fm<>2) THEN
  BEGIN
    WRITELN('Введите снова');GOTO ZN;
  END;

WRITE('Целочисленное решение(Y/N): ');READLN(PrGomory);
IF (PrGomory='Y') OR (PrGomory='y') THEN PrGomory:='Y' ELSE PrGomory:='N';
SIMPLEX; { Вызов процедуры SIMPLEX}
END.

```

## Приложение 2. Результаты работы программы на примере задачи из 2 раздела

С	Б	Н	X1	X2	X3	X4	X5	X6
0.0000	X3	18.0000	1.0000	3.0000	1.0000	0.0000	0.0000	0.0000
0.0000	X4	16.0000	2.0000	1.0000	0.0000	1.0000	0.0000	0.0000
0.0000	X5	5.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000
0.0000	X6	21.0000	3.0000	0.0000	0.0000	0.0000	0.0000	1.0000
		0.0000	-2.0000	-3.0000	0.0000	0.0000	0.0000	0.0000

Ключевой столбец: 1    Ключевая строка: 4

С	Б	Н	X1	X2	X3	X4	X5	X6
0.0000	X3	11.0000	0.0000	3.0000	1.0000	0.0000	0.0000	-0.3333
0.0000	X4	2.0000	0.0000	1.0000	0.0000	1.0000	0.0000	-0.6667
0.0000	X5	5.0000	0.0000	1.0000	0.0000	0.0000	1.0000	0.0000
2.0000	X1	7.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.3333
		14.0000	0.0000	-3.0000	0.0000	0.0000	0.0000	0.6667

Ключевой столбец: 2    Ключевая строка: 2

С	Б	Н	X1	X2	X3	X4	X5	X6
0.0000	X3	5.0000	0.0000	0.0000	1.0000	-3.0000	0.0000	1.6667
3.0000	X2	2.0000	0.0000	1.0000	0.0000	1.0000	0.0000	-0.6667
0.0000	X5	3.0000	0.0000	0.0000	0.0000	-1.0000	1.0000	0.6667
2.0000	X1	7.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.3333
		20.0000	0.0000	0.0000	0.0000	3.0000	0.0000	-1.3333

Ключевой столбец: 6    Ключевая строка: 1

С	Б	Н	X1	X2	X3	X4	X5	X6
0.0000	X6	3.0000	0.0000	0.0000	0.6000	-1.8000	0.0000	1.0000
3.0000	X2	4.0000	0.0000	1.0000	0.4000	-0.2000	0.0000	0.0000
0.0000	X5	1.0000	0.0000	0.0000	-0.4000	0.2000	1.0000	0.0000
2.0000	X1	6.0000	1.0000	0.0000	-0.2000	0.6000	0.0000	0.0000
		24.0000	0.0000	0.0000	0.8000	0.6000	0.0000	0.0000

В 4 -й итерации было получено оптимальное решение

т.к. при исследовании на МАКСИМУМ индексная строка не содержит отрицательных элементов.

Данное решение является целочисленным.

При этом:

Fmax= 24.0000  
X6= 3.0000  
X2= 4.0000  
X5= 1.0000  
X1= 6.0000



# РОССИЙСКАЯ АКАДЕМИЯ ЕСТЕСТВОЗНАНИЯ

105037 г. Москва, а/я 47,  
Академия Естествознания  
Тел. (499) 7041341,  
7 (499) 709-8104  
Тел/Факс: (8452) 477677  
E-mail: [stukova@rae.ru](mailto:stukova@rae.ru)

№ \_\_\_\_\_ от \_\_\_\_\_ 19.05.2015 г. \_\_\_\_\_

Гареева Г.А., Еремина И.И., Файзуллина А.Г.

на № \_\_\_\_\_ от \_\_\_\_\_

## Решение о присвоении грифа Учебно-методического объединения по классическому университетскому и техническому образованию Российской Академии естествознания

Учитывая положительное заключение экспертизы, УМО РАЕ по классическому университетскому и техническому образованию приняло решение (Протокол № 513 от «19» мая 2015 г.) о присвоении учебному пособию «Информационные технологии в экономико-математическом моделировании» Гареева Г.А., Еремина И.И., Файзуллина А.Г. грифа УМО РАЕ:

«Рекомендовано УМО РАЕ по классическому университетскому и техническому образованию в качестве учебного пособия для студентов высших учебных заведений, обучающихся по направлениям подготовки: 01.03.02 (010400.62) – «Прикладная математика и информатика», 09.03.02 – «Информационные системы и технологии», 38.03.01 (080100.62) – «Экономика», 080200.62 (38.03.02) – «Менеджмент», 080500.62 (38.03.05) – «Бизнес-информатика», 230100.62 – «Информатика и вычислительная техника», 09.03.03 (230700.62) – «Прикладная информатика (по отраслям)».

Текст грифа УМО размещается на левой стороне титульного листа подзаголовочных данных.

Председатель УМО РАЕ  
Д.м.н., профессор, академик РАЕ

Ученый секретарь  
К.м.н.



М.Ю. Ледванов

М.Н. Бизенкова

# РОССИЙСКАЯ АКАДЕМИЯ ЕСТЕСТВОЗНАНИЯ

**ЗОЛОТОЙ ФОНД ОТЕЧЕСТВЕННОЙ НАУКИ**

## **СЕРТИФИКАТ УЧАСТНИКА МЕЖДУНАРОДНОЙ ВЫСТАВКИ-ПРЕЗЕНТАЦИИ УЧЕБНО-МЕТОДИЧЕСКИХ ИЗДАНИЙ**

**(Москва, 2015)**

**Гареева Г.А., Еремина И.И., Файзуллина А.Г.**

Учебное пособие

**«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКО-  
МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ»**

Набережные Челны: Издательско-полиграфический центр НЧИ КФУ, 2015.

**ПРЕЗИДЕНТ  
РОССИЙСКОЙ АКАДЕМИИ ЕСТЕСТВОЗНАНИЯ**

**академик РАН М.Ю.Ледванов**



**WWW.RAE.RU**

РОССИЙСКАЯ АКАДЕМИЯ ЕСТЕСТВОЗНАНИЯ

**ЗОЛОТОЙ ФОНД  
ОТЕЧЕСТВЕННОЙ  
НАУКИ**

ЛУЧШЕЕ УЧЕБНО-МЕТОДИЧЕСКОЕ ИЗДАНИЕ В ОТРАСЛИ

**ДИПЛОМ**

ЛАУРЕАТА  
МЕЖДУНАРОДНОЙ ВЫСТАВКИ

Москва, 2015

Гареева Г.А., Еремина И.И., Файзуллина А.Г.

Учебное пособие

«ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ЭКОНОМИКО-  
МАТЕМАТИЧЕСКОМ МОДЕЛИРОВАНИИ»

Набережные Челны: Издательско-полиграфический центр НЧИ КФУ, 2015

ПРЕЗИДЕНТ

М.Ю.ЛЕДВАНОВ

<http://www.rae.ru/ru/DIPLOM/>



**Гульнара Альбертовна Гареева**

**Ирина Ильинична Еремина**

**Айгуль Гинатуловна Файзуллина**

**Информационные технологии в экономико-  
математическом моделировании**

**Учебное пособие**

**Кафедра Бизнес-информатики и математических  
методов в экономике**

---

**Формат 60x84 1/16. Печать ризографическая.  
Бумага офсетная №1 Гарнитура «Timis New Roman»  
Усл. печ. л. 11,0. Уч.-изд.л. 11,0 Тираж 300 экз. Заказ №462.**

**Издательско-полиграфический центр  
Набережночелнинского института КФУ  
423810, г.Набережные Челны, Новый город, пр.Мира, 68/19**