

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

ГЛАВНОЕ УПРАВЛЕНИЕ ОБРАЗОВАНИЯ, НАУКИ И КАДРОВ

Учреждение образования
«БЕЛОРУССКАЯ ГОСУДАРСТВЕННАЯ
СЕЛЬСКОХОЗЯЙСТВЕННАЯ АКАДЕМИЯ»

В. Н. Кулаков, Д. В. Редько, М. Н. Марченко

ПАКЕТ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ LРХ88

*Практикум
для студентов, обучающихся по экономическим
специальностям*

Горки
БГСХА
2017

УДК 519.85(076.5)

*Рекомендовано методической комиссией
экономического факультета.
Протокол № 6 от 23 февраля 2016 г.*

Авторы:

кандидат экономических наук, доцент *В. Н. Кулаков*;
магистр экономических наук *Д. В. Редько*;
заведующий лабораторией *М. Н. Марченко*

Рецензенты:

кандидат экономических наук, доцент *Ю. В. Чеплянский*;
кандидат экономических наук, доцент *И. В. Горбатенко*

Пакет линейного программирования LPX88 : практикум /
В. Н. Кулаков, Д. В. Редько, М. Н. Марченко. – Горки : БГСХА, 2017. –
44 с.

ISBN 978-985-467-681-4.

Изложены практические аспекты применения пакета линейного программирования
LPX88 с реализацией решения типовых задач оптимизации на компьютере.

Для студентов, обучающихся по экономическим специальностям.

ISBN 978-985-467-681-4

© УО «Белорусская государственная
сельскохозяйственная академия», 2017

ВВЕДЕНИЕ

Линейное программирование — математическая дисциплина, изучающая теорию и методы решения экстремальных задач на множествах n -мерного векторного пространства, задаваемых системами линейных уравнений и неравенств.

Линейное программирование является частным случаем выпуклого программирования, которое, в свою очередь, является частным случаем математического программирования. Одновременно оно составляет основу нескольких методов решения задач целочисленного и нелинейного программирования. Одним из обобщений линейного программирования является дробно-линейное программирование.

Многие свойства задач линейного программирования можно интерпретировать так же, как свойства многогранников, и таким образом геометрически формулировать и доказывать их.

Математические исследования отдельных экономических проблем, математическая формализация числового материала проводились еще в XIX веке. При математическом анализе процесса расширенного производства использовались алгебраические соотношения, анализ их проводился с помощью дифференциального исчисления. Это давало возможность получить общее представление о проблеме.

Развитие экономики в XX веке требовало новых подходов к изучению количественных показателей, и в 1920-е годы была разработана теория межотраслевого баланса (МОБ). Данная теория послужила толчком для создания и исследования математических моделей. Разработка МОБ в 1924–1925 годах в СССР повлияла на работы экономиста и статистика В. В. Леонтьева и послужила основой для создания межотраслевой модели производства и распределения продукции.

В 1939 году советский математик Л. В. Канторович опубликовал работу «Математические методы организации и планирования производства», в которой сформулировал новый класс экстремальных задач с ограничениями и разработал эффективный метод их решения. Таким образом были заложены основы линейного программирования.

Изучение подобных задач привело к созданию новой научной дисциплины — линейного программирования и открыло новый этап в развитии экономико-математических методов.

В 1949 году американский математик Джордж Бернард Данциг разработал эффективный метод решения задач линейного программирования (ЗЛП) – симплекс-метод [1].

Термин «программирование» в данном контексте следует понимать как «планирование» (один из переводов англ. programming). Он был предложен в середине 1940-х годов Дж. Данцигом, одним из основателей линейного программирования, еще до того, как компьютеры были использованы для решения линейных задач оптимизации.

Развитие компьютерных технологий во второй половине XX века позволило ученым-исследователям создавать специализированные программные пакеты, реализующие алгоритм симплекс-метода. Одним из наиболее распространенных пакетов, разработанных с данной целью, является LPX88.

В данном практикуме изложены основные практические аспекты применения пакета линейного программирования LPX88 с реализацией решения типовых задач оптимизации на компьютере.

1. ФУНКЦИОНАЛЬНЫЕ ОСОБЕННОСТИ ПРОГРАММНОГО ПАКЕТА LPX88

LPX88 – это программа, работающая на персональном компьютере под управлением операционной системы Windows.

В случае ошибки запуска из-за версии Windows возможен запуск в эмуляторе MS DOS.

Пакет программ позволяет решать задачи линейного программирования, содержащие до 510 ограничений и 2510 переменных. В перечень задач, решаемых LPX88, входят: задачи производства, составления расписания, управления запасами, наличием, транспортные задачи, сетевые задачи.

В LPX88 применяется модифицированный симплекс-алгоритм для эффективного решения задач линейного программирования. Кроме того, LPX88 включает мощную систему команд для установки и хранения в памяти задач, управления ходом вычислений и получения распечатанных выходных данных.

Выходные данные программы могут быть напечатаны или сохранены в файле.

Можно решать как задачи максимизации, так и минимизации. Ограничения могут включать любую комбинацию соотношений: $\geq$, $=$ или $\leq$. Переменным присваиваются имена по умолчанию, которые пользователь может заменить. Аналогичным образом имена по умолчанию присваиваются ограничениям, их также можно заменить. Эти имена используют для того, чтобы ясно обозначить все выходные данные на экране дисплея и в распечатке.

Размерность задач можно увеличить или уменьшить во время их ввода или редактирования.

Коэффициенты задачи линейного программирования могут вводиться с клавиатуры с помощью экранного редактора, или с предварительно записанного последовательного файла, или с командного файла в пакетном режиме.

Задачи записываются в компактном упорядоченном виде, обеспечивающем их быстрый поиск и использование. Базисы решений также можно сохранять, отыскивать и затем использовать для эффективного повторного решения задачи линейного программирования. Существует несколько вариантов запуска симплекс-алгоритма. Алгоритм можно запустить с исходного базиса, включающего единичные элементы и искусственные переменные. Если сохраненный базис доступен с

предыдущего решения, то его также можно использовать в качестве отправной точки. Наконец, алгоритм можно повторно запустить с завершающей точки предыдущего этапа вычислений.

В ходе решения задачи текущая запись статуса вычислений сохраняется, выводится на экран, включая, если позволяет пространство, текущий базис.

Пакет программ снабжен системой генерирования отчетов. Список имеющихся отчетных таблиц включает:

- Полный массив задачи;
- Результат прямого/двойственного решения;
- Решение прямой задачи;
- Решение двойственной задачи;
- Анализ чувствительности элемента стоимости/полезности;
- Анализ чувствительности элементов правой части;
- Базис обратной матрицы;
- Базис обратной матрицы * небазисные столбцы.

Контрольные вопросы

1. Основные требования, предъявляемые к системе для запуска LPX88.
2. Понятие задачи в LPX88.
3. Алгоритм решения задачи в LPX88.
4. Возможности решения задач в LPX88.
5. Основные отчеты в LPX88.

1.1. Пример выполнения задачи линейного программирования в LPX88

Общей (стандартной) задачей линейного программирования называется задача нахождения минимума линейной целевой функции (линейной формы) вида [3]

$$f(x) = \sum_{j=1}^n c_j x_j = c_1 x_1 + c_2 x_2 + \dots + c_n x_n.$$

Задача, в которой фигурируют ограничения в форме неравенств, называется основной задачей линейного программирования (ОЗЛП):

$$\sum_{j=1}^n a_{ij}x_j \geq b_i (i = 1, 2, \dots, m);$$

$$x_j \geq 0 (j = 1, 2, \dots, n).$$

Задача линейного программирования будет иметь канонический вид, если в основной задаче вместо первой системы неравенств имеет место система уравнений с ограничениями в форме равенства [8]:

$$\sum_{j=1}^n a_{ij}x_j = b_i (i = 1, 2, \dots, m).$$

Основную задачу можно свести к канонической путем введения дополнительных переменных.

Задачи линейного программирования наиболее общего вида (задачи со смешанными ограничениями: равенствами и неравенствами, наличием переменных, свободных от ограничений) могут быть приведены к эквивалентным (имеющим то же множество решений) заменами переменных и заменой равенств на пару неравенств [4].

Легко заметить, что задачу нахождения максимума можно заменить задачей нахождения минимума, взяв коэффициенты с обратным знаком.

Симплекс-метод имеет среднюю полиномиальную сходимости при широком выборе распределения значений в случайных матрицах [4, 5].

Число ограничений больше влияет на вычислительную эффективность, чем число переменных, поэтому при формулировке задач линейного программирования нужно стремиться к уменьшению числа ограничений пусть даже путем роста числа переменных.

В качестве примера задачи линейного программирования рассмотрим задачу с четырьмя ограничениями и тремя переменными.

Исходная информация:

1. Ресурсы сельскохозяйственного предприятия: пашня – 2 000 га, труд годовой – 25 000 чел.-дн.

2. Культуры и отрасли: зерновые, многолетние травы на сено, кормовые.

3. Показатели расхода ресурсов и их окупаемости представлены в табл. 1.

4. Цель решения задачи: нахождение максимума стоимости товарной продукции при заданных условиях.

Таблица 1. Пример задачи линейного программирования

Показатель	Единица измерения	Сельскохозяйственные культуры и отрасли		
		Зерновые	Многолетние травы на сено	Коровы
Расход пашни	га/га	1	1	—
Расход трудовых ресурсов	чел.-дн/га (гол.)	10	5	25
Выход кормов	ц к. ед/га	15	30	—
Потребность в кормах	ц к. ед/гол.	—	—	40
Стоимость товарной продукции	у. д. е/га (гол.)	180	—	450

Ниже представлены ограничения задачи, составленные на основании исходной информации:

$$\begin{aligned}
 x_1 + x_2 &\leq 2\,000; \\
 10 \cdot x_1 + 5 \cdot x_2 + 25 \cdot x_3 &\leq 25\,000; \\
 15 \cdot x_1 + 30 \cdot x_2 - 40 \cdot x_3 &\geq 0; \\
 x_2 &\geq 400; \\
 180 \cdot x_1 + 450 \cdot x_3 &\rightarrow \max.
 \end{aligned}$$

Для начала работы с программой необходимо открыть проводник и загрузить файл LPX88.EXE, щелкнув по соответствующему ярлыку левой клавишей мыши 2 раза (рис. 1).

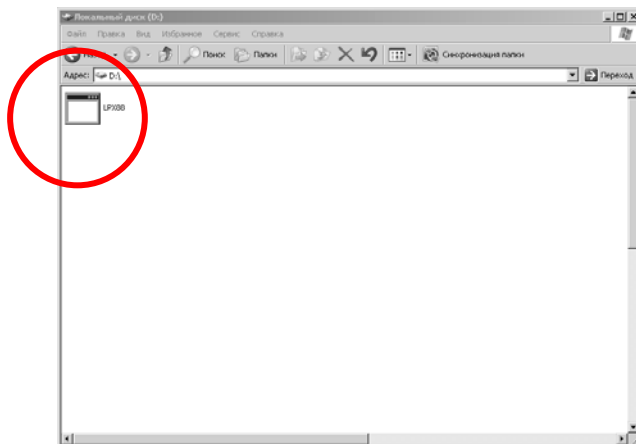


Рис. 1. Открытие программы на жестком диске

После запуска программы на экране появляется отображение вариантов меню, доступных для пользователя, под каждым из четырех меню программ. Управление процессом выполнения программы осуществляется с помощью четырех меню, управляемых функциональными клавишами в интерактивном режиме. Каждое меню управляет этапом в обработке задач линейного программирования.

После открытия программы на экране появится серия вопросов. Ответы на эти вопросы настроят LPX88 (рис. 2).

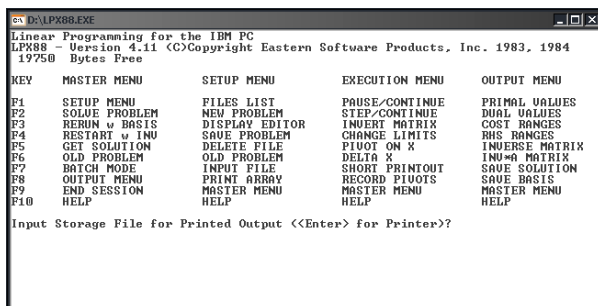


Рис. 2. Окно загруженной программы

Вы можете передать выходные данные с экрана дисплея, ответив на первый вопрос именем файла. В данном примере имя файла «example». Для ответа на второй вопрос необходимо нажать клавишу «Enter». На последующие вопросы – «a» и «b» (рис. 3).

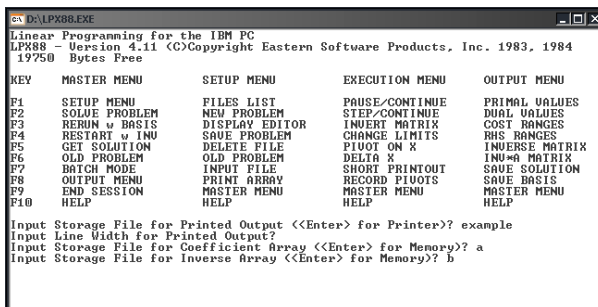


Рис. 3. Настройка программы перед работой

Выборка главного меню (MASTER MENU) появится в строке функциональной клавиши внизу экрана (рис. 4). LPX88 находится в интерактивном режиме и готова выполнять команды, как только вы нажмете функциональные клавиши F1 – F10, которые располагаются в верхней части клавиатуры (рис. 5).

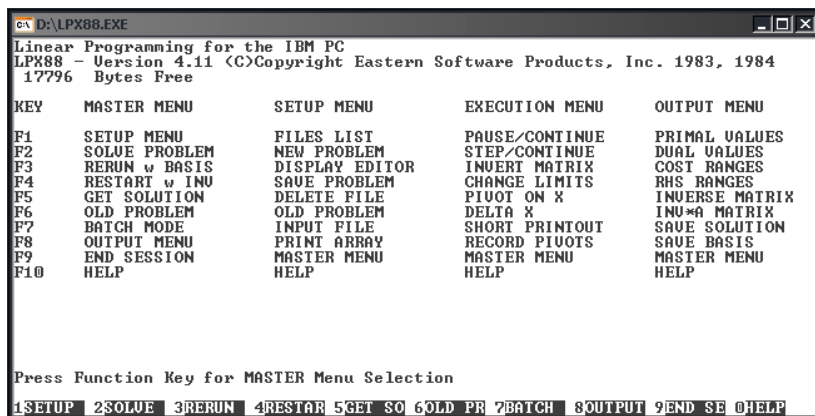


Рис. 4. Главное окно программы



Рис. 5. Клавиатура персонального компьютера

При проблемах несовместимости Windows с программой возможно использование LPX88 в эмуляторе. Эмулятор с графической оболочкой можно скачать по адресу: <http://dfendreloaded.sourceforge.net/>.

После скачивания файла установщика D-Fend-Reloaded-1.X.X-Setup.exe (X – обозначение версии программы) необходимо запустить файл.

При установке эмулятора у пользователя запрашивается язык. Необходимо нажать кнопку «ОК» (рис. 6).

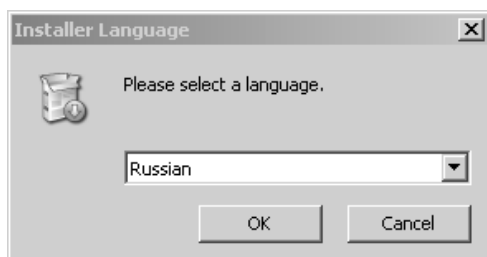


Рис. 6. Выбор языка при установке D-Fend

Затем пользователю необходимо нажать кнопку «Далее» (рис. 7).



Рис. 7. Приветствие при установке D-Fend

После этого пользователь выбирает необходимые настройки и нажимает кнопку «Далее» (рис. 8).

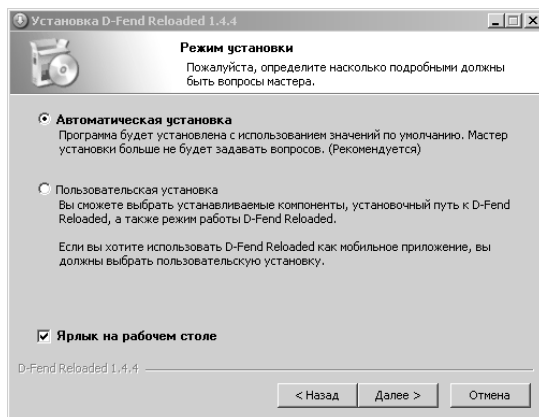


Рис. 8. Настройки при установке D-Fend

После завершения установки пользователь нажимает кнопку «Готово» (рис. 9).

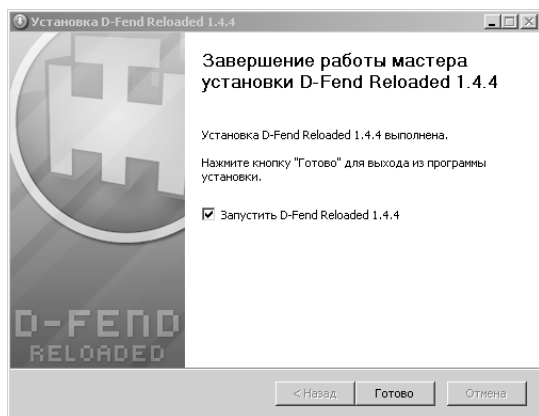


Рис. 9. Завершение установки D-Fend

После запуска программы в первый раз пользователю будет предложено выбрать параметры запуска. Необходимо нажать кнопку «ОК» (рис. 10).

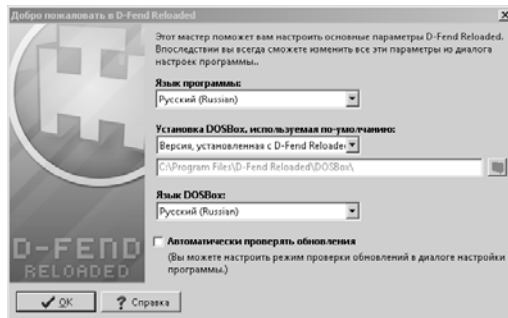


Рис. 10. Выбор параметров D-Fend

В дальнейшем для добавления LPX88 в программу нужно перетащить файл LPX88.EXE в поле программы D-Fend. После чего появится «Мастер добавления» (рис. 11).

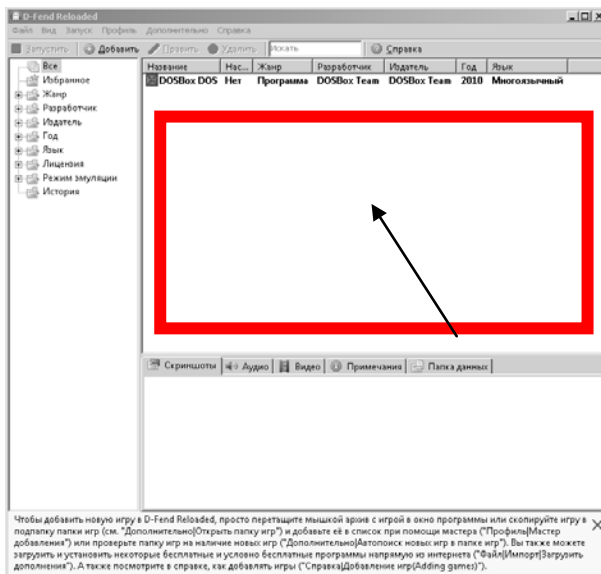


Рис. 11. Программа D-Fend

В дальнейшем на все настройки необходимо нажимать кнопку «Далее» (рис. 12).

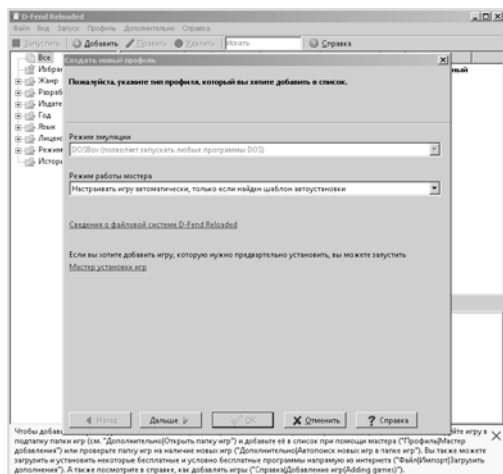


Рис. 12. Запуск «Мастера добавления» программы

В последнем диалоге необходимо нажать кнопку «OK» (рис. 13).

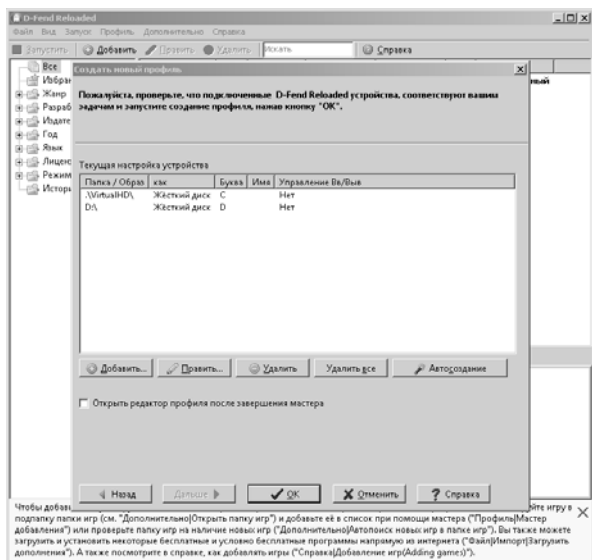


Рис. 13. Окончание добавления программы

После добавления LPX88 в программу двойным щелчком левой клавиши мыши по записи «Lpx88» можно запустить программу LPX88 (рис. 14).

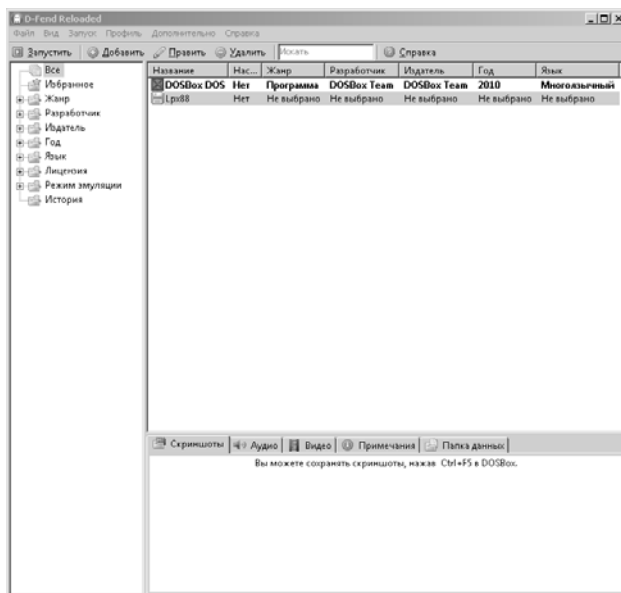


Рис. 14. Окончание добавления программы

Задания для самостоятельной работы

1. Запустите LPX88 с параметрами запуска: new1, d, f.
2. Запустите LPX88 с параметрами запуска: new2, b, n.
3. Запустите LPX88 с параметрами запуска: new3, h, j.

Контрольные вопросы

1. В чем заключается основная задача линейного программирования?
2. Как запустить LPX88?
3. Как для работы настроить LPX88?
4. Где располагаются функциональные клавиши LPX88?
5. Возможны ли альтернативные варианты запуска LPX88?

1.2. Ввод исходных данных

Задача линейного программирования состоит из множества элементов, которые либо задает пользователь, либо LPX88 назначает имена по умолчанию. Это следующие компоненты:

- Имя текущей задачи: должно быть задано пользователем;
- Базис текущего решения: может быть задан пользователем.

LPX88 создает исходный базис, когда запускают симплекс-алгоритм.

Имена переменных: LPX88 присваивает имена по умолчанию (X.1, X.2 и т. д.) нефиктивным переменным. Эти имена можно изменить с помощью редактора. Имена переменных не могут содержать более 6 знаков.

Имена ограничений: LPX88 присваивает ограничениям имена по умолчанию (Y.1, Y.2 и т. д.). Эти имена можно изменить с помощью экранного редактора. Имена ограничений не могут содержать более 6 знаков.

Имена фиктивных переменных: фиктивные переменные с именами S.1, S.2 и т. д. создаются для каждого ограничения. Имена фиктивных переменных изменить нельзя.

Целевая функция: должна быть задана пользователем: MAX (максимизировать) или MIN (минимизировать).

Ненулевые коэффициенты задачи: должны быть определены пользователем, кроме коэффициентов фиктивных переменных, которые LPX88 выводит из соотношений в ограничениях. Существует четыре типа коэффициентов задачи. Перечислим наиболее часто используемые из них:

- Элементы строки целевой функции;
- Элементы правой части;
- Ненулевые коэффициенты ограничений.

Соотношения в ограничениях: считается, что ограничения имеют вид равенства (=), пока пользователь не изменил их с помощью экранного редактора, ввода с последовательного файла. Пользователи могут задать ограничения вида как меньше или равно ($\leq$), так и больше или равно ($\geq$).

Для ввода новой задачи линейного программирования LPX88 должна находиться под управлением SETUP MENU. Это осуществляется нажатием функциональной клавиши F1. Экран очистится и вверху в двух строках появится заголовок задачи. Выбор-

ка MASTER MENU (главного меню) в строке функциональной клавиши будет заменена на выборку SETUP MENU. Если вы еще этого не сделали, то нажмите функциональную клавишу F1.

Теперь нажмите функциональную клавишу F2 и начинайте вводить задачу, отвечая на вопросы, появляющиеся в верхней части экрана (рис. 15).

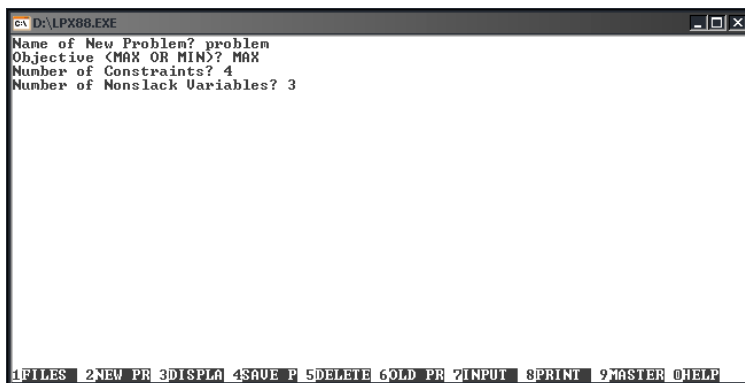


Рис. 15. Ввод параметров новой программы

Когда восстановится заголовок задачи, можно начинать ввод коэффициентов задачи линейного программирования. LPX88 считает «problem» текущей задачей. Все выборки всех меню применяются к «problem», пока она в качестве текущей не заменится на другую задачу линейного программирования.

Отметим, что имя текущей задачи появляется в реверсивном цвете в заголовке задачи. Это означает, что LPX88 ожидает нажатия одной из функциональных клавиш. LPX88 автоматически присваивает имена по умолчанию ограничениям и переменным новой задачи. Имена ограничений – Y.1, Y.2, Y.3 и Y.4, имена переменных – X.1, X.2, X.3. Для ограничений могут быть также созданы фиктивные переменные. В соответствии с ограничениями фиктивным переменным присваиваются имена S.1, S.2, S.3 и S.4.

Нажмите функциональную клавишу F3. Программа запросит имена строки и столбца в верхнем левом углу секции массива коэффициентов размерностью 15 строк на 10 столбцов. Эта секция массива вместе с соответствующими элементами строки целевой функции (RETURN) и столбца правой части (RHS) выводится на экран

для редактирования. Имя строки Y.1 и имя столбца X.1 указывают для экранного редактора верхний левый угол задачи «problem». Вы можете представить на экране всю задачу, отвечая на вопросы следующего вида (рис. 16).

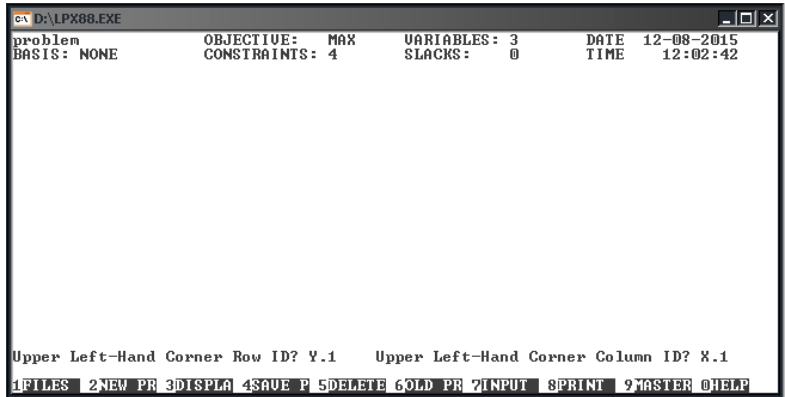


Рис. 16. Указание редактору имени строки и столбца

После введения имен строк и столбцов возникнет короткая пауза, пока LPX88 будет устанавливать изображение на экране. Когда на экране появится изображение, отметим, что число ограничений временно возросло до 15, а число переменных – до 10 (рис. 17).

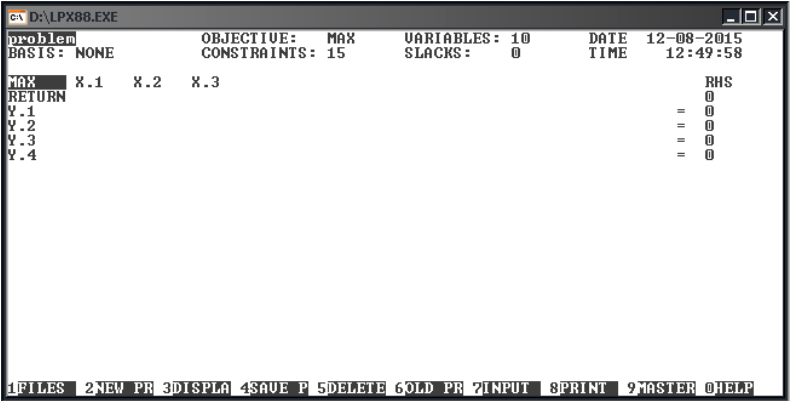


Рис. 17. Редактор LPX88

Эти дополнения позволят добавить в текущую задачу новые ограничения и целочисленные переменные (рис. 18).

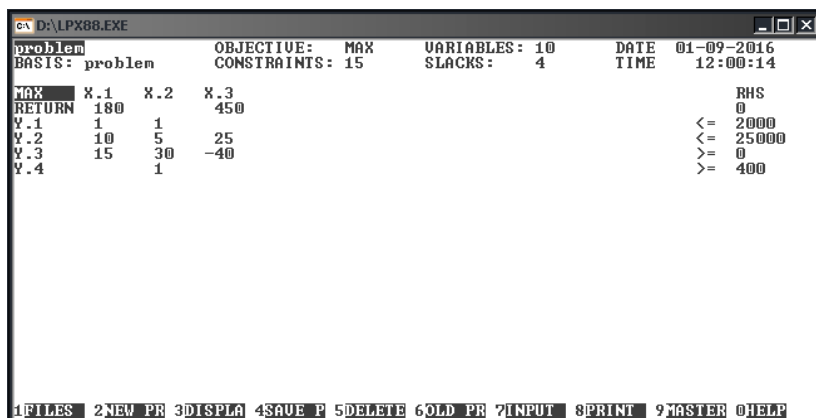


Рис. 18. Редактор LPX88 – введенная задача

Если вы когда-нибудь пользовались какой-либо популярной программой обработки электронных таблиц, то работа с экранным редактором не вызовет затруднений. Почти все, что вы видите на экране между заголовком и строкой функциональной клавиши, можно модифицировать с помощью экранного редактора, включая целевую функцию (MAX), имя параметров функции цели (RETURN), имена ограничений (Y.1 и т. д.), вид соотношения в ограничениях ($\geq$ и т. д.).

Все элементы массива коэффициентов, строки целевой функции и столбца правой части первоначально имеют значение 0. Соответствующие элементы не видны на экране, за исключением элементов правой части, где на экране появляется 0. Все ограничения представлены сначала в виде равенств.

Экранный редактор имеет курсор, включающий строку из шести символов. Вы можете найти курсор в начале 4-й строки экрана сразу за MAX (позиция «None»). Позиция курсора показывает компонент текущей задачи, который будет заменен на данные, вводимые с клавиатуры.

Для управления позицией курсора используется клавиша «Enter» и клавиши цифровой клавиатуры. Клавиши цифровой клавиатуры расположены в правой части клавиатуры персонального компьютера.

Клавиши, используемые экраным редактором, отмечены стрелками и словами «Home» и «End». Эти клавиши функционируют в соответствии с нанесенными на них символами передвижения курсора от элемента к элементу.

Введем коэффициенты задачи. Чтобы ввести коэффициент 1 для переменной X.1 нажмите клавишу «→», а затем «↓» два раза. Курсор теперь должен находиться под X.1 в строке с меткой Y.1. Отпечатайте «1» и передвиньте курсор на следующую позицию. Теперь коэффициент 1 появится на экране в нужном месте. Повторите процесс для остальных коэффициентов задачи. Не забывайте вводить «≤» и «≥» для ограничений Y.1, Y.2, Y.3 и Y.4 (см. рис. 18).

Задание для самостоятельной работы

Создайте новую задачу в LPX88 со следующими параметрами:

Название задачи: test2;

Целевая функция: MAX;

Количество переменных: 5;

Количество ограничений: 10.

Контрольные вопросы

1. Каким образом LPX88 присваивает имена переменным?
2. Какова размерность экрана LPX88?
3. Какие коэффициенты использует LPX88?
4. Какие клавиши использует LPX88 для управления редактором?
5. Под управлением какого меню можно редактировать переменные LPX88?

1.3. Сохранение и загрузка файла задачи

Для записи текущей задачи проверьте в первую очередь строку функциональной клавиши, чтобы убедиться, что система под управлением SETUP MENU, затем нажмите функциональную клавишу F4 SAVE PROBLEM (запись задачи) (рис. 19).

Для записи «problem» с именем файла «problem» ответьте на вопросы, появляющиеся под заголовком задачи.

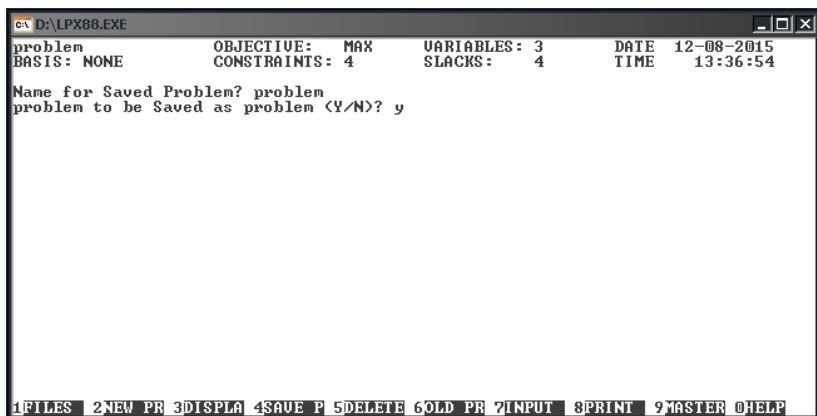


Рис. 19. Сохранение файла задачи

Файл «problem.LP» будет записан в каталоге программы. Теперь нажмите функциональную клавишу F6. Введите имя записанной задачи, ответив на вопрос (рис. 20). Не добавляйте к имени задачи расширения .LP.

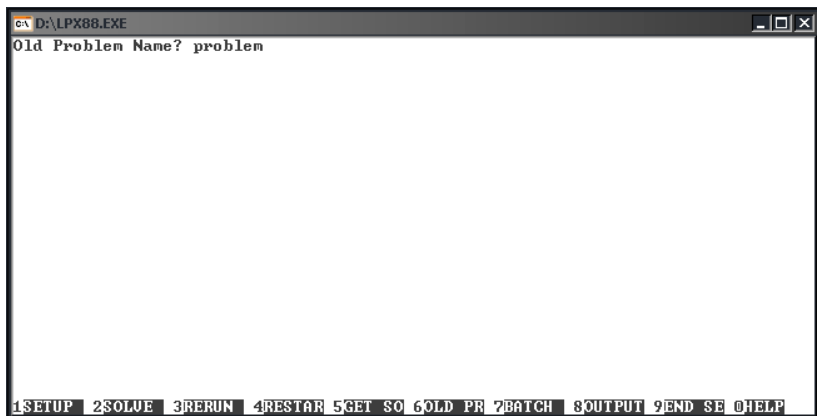


Рис. 20 Загрузка файла задачи

Задача линейного программирования найдена в том виде, в каком была записана, и теперь это текущая задача. Вы можете убедиться, что запись и поиск «problem» осуществлен правильно, выведя задачу на

экран с помощью экранного редактора. Просто нажмите функциональную клавишу F3 и введите, как и ранее, координаты левого верхнего угла.

Задания для самостоятельной работы

1. Сохраните задачу LPX88 с именем файла file1.
2. Сохраните задачу LPX88 с именем файла file2, затем откройте file2 в программе.

Контрольные вопросы

1. Какую функциональную клавишу для сохранения файла использует LPX88?
2. Где сохраняется файл задачи LPX88?
3. Какую функциональную клавишу для открытия файла использует LPX88?

1.4. Решение задачи

*В случае если на вашем экране в строке функциональной клавиши по-прежнему находится **SETUP MENU**, нажмите функциональную клавишу F9, чтобы вернуться к выборке главного меню. Когда на экране в строке функциональной клавиши появится **главное меню**, нажмите функциональную клавишу F2 (рис. 21).*

Пока LPX88 решает задачу линейного программирования, на экране сохраняется текущее описание статуса вычислений. Заголовок на экране разделен на две части. Теперь вы должны видеть исходное состояние вычислений, затем список значений по умолчанию для различных допусков и программных управлений.

Краткое изложение решения задачи линейного программирования посылается в файл, если это было предусмотрено при первоначальной установке.

Базис решения «problem» состоит из переменных, представленных на экране в строке BASIS. Значения их решений появляются ниже в строке PRIMAL. Все остальные переменные задачи равны 0, включая фиктивные переменные, вставленные LPX88.

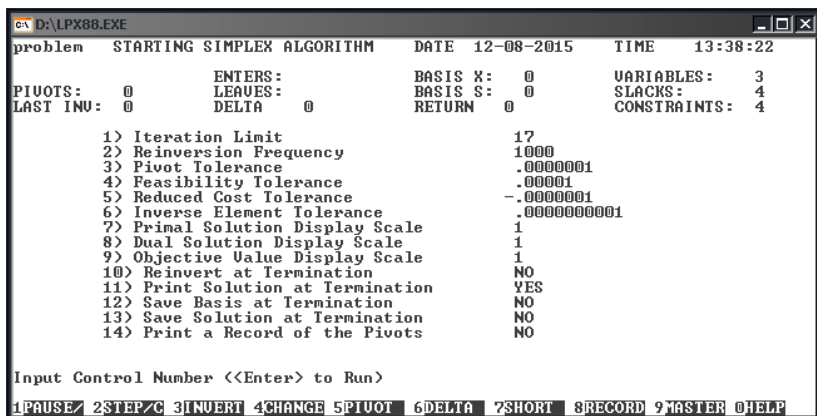


Рис. 21. Запуск задачи

Двойственные или теньевые значения ограничений появляются по порядку на экране в строке DUAL.

Вы также могли заметить имя **** среди списка базисных переменных. Этим способом LPX88 идентифицирует искусственную переменную в базе во время фазы I в симплекс-алгоритме.

LPX88 позволяет увидеть, как симплекс-алгоритм добавляет и удаляет переменные до тех пор, пока не найдено оптимальное решение или не установлено, что задача линейного программирования не может быть решена.

LPX88 печатает строку для каждой опорной точки, показывая входящие и выходящие переменные и другую информацию. Теперь нажмите клавишу «Enter», чтобы начать процесс решения «problem».

Вы можете провести LPX88 через симплекс-алгоритм с помощью функциональных клавиш, пока решается «problem». Для этого нажмите функциональную клавишу F2 сразу же после того, как LPX88 начнет решение «problem». LPX88 сделает паузу после каждой опорной точки. Во время паузы имя задачи появится в верхнем левом углу экрана в реверсивном цвете.

Для выполнения только одной опорной точки нажмите функциональную клавишу F1. После того как LPX88 выполнит опорную точку, она вновь сделает паузу. Вы должны нажимать функциональную клавишу F1 для каждой отправной точки до тех пор, пока вновь не нажмете F2. Когда вы нажмете F2 во второй раз, паузы прекратятся.

и возобновится непрерывное выполнение программы. Пока LPX88 решает задачу линейного программирования, на экране монитора сохраняется текущая запись состояния симплекс-алгоритма.

На рис. 22 показана следующая информация:

- DATE – дата;
- TIME – время;
- CONSTRAINTS – число ограничений;
- VARIABLES – число переменных;
- SLACKS – число положительных искусственных оценок или отрицательных переменных, остающихся в базисе во время фазы I;
- PIVOTS – число взятых опорных точек;
- COST или RETURN – результат текущего основного решения.

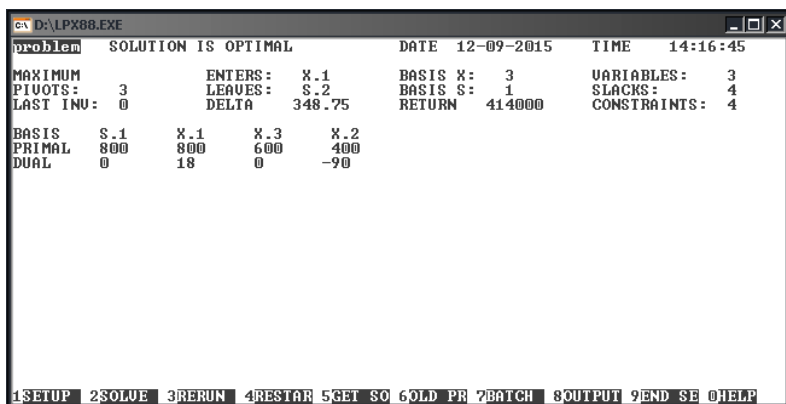


Рис. 22. Решение задачи

Задания для самостоятельной работы

1. Запустите задачу LPX88 с именем файла «problem».
2. Запустите задачу LPX88 с именем файла «problem» и выполните решение только одной опорной точки.

Контрольные вопросы

1. Какую функциональную клавишу для решения задачи использует LPX88?

2. Какая информация доступна на экране после решения задачи LPX88?

3. Можно ли просмотреть ход решения задачи?

1.5. Методы решения задачи

Задача линейного программирования состоит в том, что необходимо максимизировать или минимизировать некоторый линейный функционал на многомерном пространстве при заданных линейных ограничениях.

Заметим, что каждое из линейных неравенств на переменные ограничивает полупространство в соответствующем линейном пространстве. В результате все неравенства ограничивают некоторый многогранник (возможно, бесконечный), называемый также полиэдральным комплексом. Уравнение $W(x) = c$, где $W(x)$ – максимизируемый (или минимизируемый) линейный функционал, порождает гиперплоскость $L(c)$. Зависимость от c порождает семейство параллельных гиперплоскостей. Тогда экстремальная задача приобретает следующую формулировку: требуется найти такое наибольшее c , чтобы гиперплоскость $L(c)$ пересекала многогранник хотя бы в одной точке. Заметим, что пересечение оптимальной гиперплоскости и многогранника будет содержать хотя бы одну вершину, причем их будет более одной, если пересечение содержит ребро или k -мерную грань. Поэтому максимум функционала можно искать в вершинах многогранника. Принцип симплекс-метода состоит в том, что выбирается одна из вершин многогранника, после чего начинается движение по его ребрам от вершины к вершине в сторону увеличения значения функционала. Когда переход по ребру из текущей вершины в другую вершину с более высоким значением функционала невозможен, считается, что оптимальное значение c найдено.

Последовательность вычислений симплекс-методом можно разделить на две основные фазы:

1) нахождение исходной вершины множества допустимых решений;

2) последовательный переход от одной вершины к другой, ведущий к оптимизации значения целевой функции (рис. 23).

При этом в некоторых случаях исходное решение очевидно или его определение не требует сложных вычислений, например, когда все ограничения представлены неравенствами вида «меньше или равно» (тогда нулевой вектор совершенно точно является допустимым реше-

нием, хотя и, скорее всего, далеко не самым оптимальным). В таких задачах первую фазу симплекс-метода можно вообще не проводить. Симплекс-метод, соответственно, делится на однофазный и двухфазный.



Рис. 23. Переход от одной вершины к другой

Если в условии задачи линейного программирования не все ограничения представлены неравенствами типа $\leq$, то далеко не всегда нулевой вектор будет допустимым решением. Однако каждая итерация симплекс-метода является переходом от одной вершины к другой, и если неизвестно ни одной вершины, алгоритм вообще не может быть начат.

Процесс нахождения исходной вершины не сильно отличается от однофазного симплекс-метода, однако может в итоге оказаться сложнее, чем дальнейшая оптимизация.

Все ограничения задачи модифицируются согласно следующим правилам:

- ограничения типа $\leq$ переводятся на равенства путем создания дополнительной переменной с коэффициентом $+1$. Эта модификация проводится и в однофазном симплекс-методе, дополнительные переменные в дальнейшем используются как исходный базис;
- ограничения типа $\geq$ дополняются одной переменной с коэффициентом -1 . Поскольку такая переменная из-за отрицательного коэффициента не может быть использована в исходном базисе, необходимо создать еще одну, вспомогательную, переменную. Вспомогательные переменные всегда создаются с коэффициентом $+1$;
- ограничения типа $=$ дополняются одной вспомогательной переменной.

Соответственно, будет создано некоторое количество дополнительных и вспомогательных переменных. В исходный базис выбира-

ются дополнительные переменные с коэффициентом +1 и все вспомогательные. Осторожно: решение, которому соответствует этот базис, не является допустимым.

Несмотря на то, что и дополнительные, и вспомогательные переменные создаются искусственно и используются для создания исходного базиса, их значения в решении сильно отличаются:

- дополнительные переменные сообщают, насколько соответствующее им ограничение недоиспользовано. Значение дополнительной переменной, равное нулю, соответствует равенству значений правых и левых частей ограничения;
- вспомогательные переменные сообщают, насколько данное условие далеко от допустимого (относительно конкретного ограничения). Если значение вспомогательной переменной больше нуля, то данное решение не выполняет определенное ограничение, а значит, не является допустимым.

То есть ненулевое значение дополнительной переменной может (но не должно) сигнализировать о неоптимальности решения. Ненулевое значение вспомогательной переменной сигнализирует о недопустимости решения.

После того, как было модифицировано условие, создается вспомогательная целевая функция. Если вспомогательные переменные были обозначены как y_i , $i \in \{1, \dots, k\}$, то вспомогательную функцию определим следующим образом:

$$z' = \sum_{i=1}^k y_i \rightarrow \min.$$

После этого проводится обыкновенный симплекс-метод относительно вспомогательной целевой функции. Поскольку все вспомогательные переменные увеличивают значение z' , в ходе алгоритма они будут поочередно выводиться из базиса, при этом после каждого перехода новое решение будет все ближе к множеству допустимых решений.

Когда будет найдено оптимальное значение вспомогательной целевой функции, могут возникнуть две ситуации:

1) оптимальное значение z' больше нуля. Это значит, что как минимум одна из вспомогательных переменных осталась в базисе. В таком случае можно сделать вывод, что допустимых решений данной задачи линейного программирования не существует;

2) оптимальное значение z' равно нулю. Это означает, что все вспомогательные переменные были выведены из базиса и текущее решение является допустимым.

Во втором случае мы имеем допустимый базис, или, иначе говоря, исходное допустимое решение. Можно проводить дальнейшую оптимизацию с учетом исходной целевой функции, при этом уже не обращая внимания на вспомогательные переменные. Это и является второй фазой решения.

Задачи линейного программирования решаются посредством LPX88 с использованием версии явного обращения усовершенствованного симплекс-метода. Матрица, обратная к матрице базисных столбцов, вычисляется и сохраняется после решения линейной задачи. LPX88 использует версию «обратной матрицы в форме произведений» усовершенствованного симплекс-метода. Обратная матрица вычисляется как произведение серии элементарных матриц.

Алгоритм зависит от выбора базиса. Это тот же список переменных, что разрешает использовать ненулевые значения. В нормальной ситуации базис включает как многие переменные, так и ограничения линейной задачи. Однако, если в базисе находится меньше переменных, оставшиеся позиции занимают «искусственными» переменными.

LPX88 может начинать и прогонять симплексный алгоритм следующими четырьмя различными путями:

1. Решение задачи: алгоритм начинает работу с нахождения некоторого начального базиса. Матрица базисных столбцов обращается для получения начальной обратной базисной матрицы.

2. Перепропуск с базиса: алгоритм начинает работу с базиса, вычисленного предварительно. Если базис существует для линейной задачи, которая охватывает текущую задачу, то этот базис обычно дает лучшую исходную точку, чем базис, получаемый по предыдущему методу. Матрица базисных столбцов обращается.

3. Рестарт с обращением: алгоритм начинается с текущей точки последнего прогона. Старая базисная обратная матрица используется для перевычисления прямого и двойственного решения.

4. Получение решения: так же, как и при перепропуске с базиса, но без дальнейших преобразований текущих значений. В действительности LPX88 вычисляет решение для некоторого данного базиса без попыток максимизации или минимизации целевой функции.

Поиск опорных точек осуществляется в двух фазах. В фазе I для выбора переменных, входящих в базис, применяются искусственные оценки. Фаза I завершается либо определением допустимого базиса, либо доказательством того, что задача линейного программирования не имеет допустимых решений. Допустимый базис – это базис со всеми неотрицательными переменными и нулевыми искусственными оценками.

В фазе II используются двойственные оценки для выбора входящих в базис переменных. Поиск опорных точек в фазе II завершается получением оптимального решения или обнаружением, что задача линейного программирования является неограниченной (двойственно недопустимой).

Аномальное завершение симплекс-алгоритма случается, если число опорных точек превышает заранее определенный предел итераций или при нажатии функциональной клавиши F9.

Вычисления в симплекс-алгоритме выполняются с помощью набора заранее заданных допусков и управлений. Этим допускам присваиваются значения по умолчанию, которые пользователь может заменить во время выполнения процедуры:

1. Предел итераций: симплекс-алгоритм завершится, когда число опорных точек достигнет предела для итераций. По умолчанию это число переменных плюс два раза число строк.

2. Частота повторного инвертирования: обратная матрица вычисляется вновь после того, как интегральное число опорных точек задано как частота повторного инвертирования. По умолчанию это дважды число строк.

3. Допуск по опорным точкам: элемент опорных точек должен быть больше по величине, чем допуск по опорным точкам. Значение по умолчанию .000001.

4. Допуск по допустимости: базисным переменным присваивают нулевое значение, если они меньше по абсолютному значению, чем допуск по допустимости. Значение по умолчанию .000001.

5. Допуск по приведенной стоимости: переменные, входящие в базис, должны иметь приведенную стоимость меньше, чем допуск приведенной стоимости, если имеем задачу минимизации. В задаче максимизации предельная полезность должна быть больше, чем отрицательная величина допуска приведенной стоимости. Значение по умолчанию .0000001.

6. Допуск по элементам обратной матрицы: в опорных точках элементы массива обратной матрицы считаются нулевыми, если они меньше по величине, чем допуск по элементам обратной матрицы. Значение по умолчанию .0000000001.

7. Экранный масштаб решения прямой задачи: значение переменных прямой задачи умножается на экранный масштаб решения прямой задачи до вывода на экран. Значение по умолчанию 1,0 (нет масштабирования).

8. Экранный масштаб решения двойственной задачи: значения двойственных переменных умножаются на экранный масштаб решения двойственной задачи до вывода на экран. Значение по умолчанию 1.0 (нет масштабирования).

9. Экранный масштаб значения целевой функции: значение целевой функции умножается на экранный масштаб значения целевой функции до вывода на экран. Значение по умолчанию 1,0 (нет масштабирования).

10. Повторное инвертирование при завершении: матрица базисных столбцов повторно инвертируется до завершения симплекс-алгоритма. По умолчанию NO (нет).

11. Распечатка всех решений при завершении: краткое итоговое решение отпечатают для прямой и двойственной задач. По умолчанию YES (да).

12. Запись базиса при завершении: при завершении базис автоматически записывается под текущим именем базиса. По умолчанию NO (нет).

13. Запись решения при завершении: при завершении процедуры решение автоматически записывается на последовательный файл с использованием имени текущей задачи с модификатором .LPS. По умолчанию NO (нет).

14. Распечатка записи опорных точек: для каждой опорной точки записывается строка, показывающая переменные входящие в базис и выходящие из базиса, приведенные затраты или граничный возврат, текущие COST или RETURN и статус вычислений. По умолчанию NO (нет).

Задания для самостоятельной работы

1. Запустите задачу LPX88 с именем файла «problem» и измените экранный масштаб на значение «2».

2. Запустите задачу LPX88 с именем файла «problem» и измените экранный предел итераций на значение «5».

3. Запустите задачу LPX88 с именем файла «problem» и измените запись базиса при завершении на значение «да».

Контрольные вопросы

1. Какова последовательность действий при решении симплекс-метода?

2. Какой метод решения задачи использует LPX88?
3. Можно ли менять настройки и допуски решения задачи?
4. Сколько фаз симплекс-метода существует?

1.6. Получение отчетов

OUTPUT MENU управляет генерированием отчетов и анализом чувствительности. Нажмите функциональную клавишу F8 OUTPUT MENU. В строке функциональной клавиши теперь появится выборка OUTPUT MENU (меню вывода). Нажмите функциональную клавишу F1 (Прямые значения). LPX88 образует и напечатает таблицу, представляющую основное решение задачи.

Листинг задачи аналогичен по внешнему виду формату, используемому для вывода задач на экран дисплея. Может быть получен с помощью функциональной клавиши F8 под управлением SETUP MENU. LPX88 образует таблицу и включает в нее максимально возможное число переменных по горизонтали.

Генерирование отчетов в виде набора таблиц осуществляется пользователем под управлением OUTPUT MENU. Нажав соответствующую функциональную клавишу, при завершении симплекс-алгоритма можно выбрать и отпечатать до шести различных таблиц, содержащих информацию о задаче и результаты ее решения. Необязательно, чтобы алгоритм завершился оптимальным решением (рис. 24).

problem	SOLUTION IS OPTIMAL				DATE	XX-XX-XXXX	TIME	XX:XX:XX
MAXIMUM					BASIS X:	3	VARIABLES:	3
PIVOTS:	3				BASIS S:	1	SLACKS:	4
LAST INV:	0				RETURN	414000	CONSTRAINTS:	4
		DELTA	0					
BASIS	S.1	X.1	X.3	X.2				
PRIMAL	800	800	600	400				
DUAL	0	18	0	-90				

Рис. 24. Листинг задачи

Каждой таблице предшествует краткий заголовок, идентифицирующий задачу и ее решение.

Переменные прямой задачи перечислены в порядке следования столбцов (рис. 25):

- VARIABLE – имя переменной;
- STATUS – базисная (BASIS) или небазисная (NONBASIS) переменная. Базисные переменные могут иметь положительные значения, небазисные – всегда нулевые;
- VALUE – значение переменной в решении RETURN/UNIT – элемент строки целевой функции для переменных;
- VALUE/UNIT – сумма произведений элементов столбцов и двойственных (теневых) значений, т. е. условно начисленная стоимость ресурсов, использованных на единицу продукции;
- NET RETURN – RETURN/UNIT минус VALUE/UNIT – также известны как приведенная стоимость переменной.

problem	SOLUTION IS MAXIMUM		RETURN	414000	DATE XX-XX-XXXX
	PRIMAL PROBLEM SOLUTION				TIME XX:XX:XX
VARIABLE	STATUS	VALUE	RETURN/UNIT	VALUE/UNIT	NET RETURN
X.1	BASIS	800	180	180	0
X.2	BASIS	400	0	0	0
X.3	BASIS	600	450	450	0
S.1	BASIS	800	0	0	0
S.2	NONBASIS	0	0	18	-18
S.3	NONBASIS	0	0	0	0
S.4	NONBASIS	0	0	90	-90
S.4	NONBASIS	0	0	90	-90

Рис. 25. Клавиша F1 – Прямые значения

Двойственные (теневые) переменные перечислены в порядке следования строк (рис. 26):

- ROW ID – имя строки, соответствующей двойственной переменной;
- STATUS – связанное (BINDING) или несвязанное (NON-BINDING) ограничение. Связанные ограничения могут иметь ненулевые двойственные значения переменных, несвязанные ограничения всегда имеют нулевые значения переменных;
- DUAL VALUE – значение двойственной переменной в решении;
- RHS VALUE – элемент правой части ограничений USAGE – сумма произведений строчных элементов ограничений и значений переменных основной задачи, т. е. использование ресурсов для ограничений;

• SLACK – значение RHS минус USAGE, т. е. значение фиктивной переменной ограничения.

problem	SOLUTION IS MAXIMUM		RETURN	414000	DATE	XX-XX-XXXX
	DUAL PROBLEM SOLUTION				TIME	XX:XX:XX
ROW ID	STATUS	DUAL VALUE	RHS VALUE	USAGE	SLACK	
Y.1	NONBINDING	0	2000	1200	800	
Y.2	BINDING	18	25000	25000	0	
Y.3	NONBINDING	0	0	0	0	
Y.4	BINDING	-90	400	400	0	

Рис. 26. Клавиша F2 – Двойственные значения

Переменные прямой задачи перечислены в порядке следования столбцов. Фиктивные переменные не включены (рис. 27):

- VARIABLE – имя переменной;
- STATUS – базисная (BASIS) или небазисная (NONBASIS) переменная. Базисные переменные могут иметь положительные значения, небазисные – всегда нулевые;
- VALUE – значение переменной в решении;
- RETURN/UNIT – элемент строки целевой функции для переменной;
- MINIMUM – базис решения изменится, если RETURN/UNIT переменной станет ниже минимума;
- MAXIMUM – базис решения изменится, если RETURN/UNIT переменной поднимется выше максимума.

problem	SOLUTION IS MAXIMUM		RETURN	414000	DATE	XX-XX-XXXX
	OBJECTIVE ROW RANGES				TIME	XX:XX:XX
VARIABLE	STATUS	VALUE	RETURN/UNIT	MINIMUM	MAXIMUM	
X.1	BASIS	800	180	180	NONE	
X.2	BASIS	400	0	NONE	90	
X.3	BASIS	600	450	-480	450	
X.3	BASIS	600	0	NONE	90	

Рис. 27. Клавиша F3 – Область значений COST

Двойственные (теневые) переменные перечислены в порядке следования строк (рис. 28):

- ROW ID – имя строки, соответствующей двойственной переменной;

- STATUS – связанное (BINDING) или несвязанное (NON-BINDING) ограничение. Связанные ограничения могут иметь ненулевые двойственные значения переменных, несвязанные ограничения всегда имеют нулевые значения переменных;

- DUAL VALUE – значение двойственной переменной в решении;

- RHS VALUE – элемент правой части ограничения;

- MINIMUM – базис решения изменится, если значение правой части ограничения станет ниже минимума;

- MAXIMUM – базис решения изменится, если значение правой части ограничения станет выше максимума.

problem	SOLUTION IS MAXIMUM		RETURN	414000	DATE XX-XX-XXXX
	RIGHT HAND SIDE RANGES				TIME XX:XX:XX
ROW ID	STATUS	DUAL VALUE	RHS VALUE	MINIMUM	MAXIMUM
Y.1	NONBINDING	0	2000	1200	NONE
Y.2	BINDING	18	25000	9500	40500
Y.3	NONBINDING	0	0	-24800	24800
Y.4	BINDING	-90	400	0	1052.632
Y.4	BINDING	-90	0	-24800	24800

Рис. 28. Клавиша F4 – Область значений RHS

В таблице, приведенной на рис. 29, базисные переменные перечислены по вертикали в порядке следования строк и по горизонтали в позиции их опорной точки.

RETURN – это чистый доход или приведенные затраты. RETURN имеет ненулевое значение только для субоптимальных решений. Обратная матрица базисных столбцов распечатана в транспонированной форме, т. е. отпечатанные строки – это столбцы базиса обратной матрицы.

problem	SOLUTION IS MAXIMUM		RETURN	414000	DATE XX-XX-XXXX
	INVERSE COEFFICIENTS				TIME XX:XX:XX
	RETURN	S.1	X.1	X.3	X.2
X.1	0	0	.05161	.03226	-1.2258
X.2	0	0	0	0	1
X.3	0	0	.01935	-.0129	.29032
S.1	0	1	-.05161	-.03226	.22581

Рис. 29. Клавиша F5 – Обратная матрица

В таблице, приведенной на рис. 30, небазисные переменные перечислены по вертикали в порядке следования столбцов, базисные переменные – по горизонтали в позиции их опорных точек.

RETURN – это чистая полезность или приведенная стоимость. Матричное произведение обратной матрицы базисных столбцов и небазисных столбцов массива коэффициентов печатается в транспонированной форме, т. е. каждая напечатанная строка – это произведение базиса обратной матрицы и небазисных столбцов массива коэффициентов.

problem	SOLUTION IS MAXIMUM	RETURN	414000	DATE	XX-XX-XXXX
	INVERSE * NONBASIS COLUMNS			TIME	XX:XX:XX
RETURN	S.1	X.1	X.3	X.2	

Рис. 30. Клавиша F6 – Обратная матрица * небазисные столбцы

Задание для самостоятельной работы

Получите все отчеты после решения задачи «problem» в LPX88.

Контрольные вопросы

1. Как получить область определения COST?
2. Как получить двойственные значения?
3. Как получить листинг решения?

1.7. Функциональные клавиши

Использование функциональных клавиш описывается в табл. 2.

Таблица 2. Функциональные клавиши программы LPX88

Клавиша	Функция
Главное меню (MASTER MENU)	
F1 (SETUP MENU)	Передаёт управление SETUP MENU (меню установки)
F2 (SOLVE PROBLEM)	Решает текущую задачу с исходной точки. Запускается с исходным базисом, содержащим единичные элементы и искусственные оценки
F3 (RERUN w BASIS)	Решает текущую задачу с заданного базиса. Повторный запуск симплекс-алгоритма с текущим базисом или предварительно записанным базисом. Массив обратной матрицы вычисляется повторно

Клавиша	Функция
F4 (RESTART w INV)	Перезапуск текущей задачи с текущим базисом и обратной матрицей. Вновь вычисляются решения прямой и двойственной задач, но НЕ массив обратной матрицы. Если изменен какой-либо элемент базисных столбцов массива коэффициентов, то перезапуск использовать не следует
F5 (GET SOLUTION)	Повторно вычисляются решения прямой и двойственной задач с использованием текущего базиса и предварительно записанного базиса. Не берутся опорные точки
F6 (OLD PROBLEM)	Осуществляет поиск в памяти записанной ранее линейной задачи и делает ее текущей задачей
F7 (BATCH MODE)	Начинает автоматическую обработку команд
F8 (OUTPUT MENU)	Передаёт управление меню вывода (OUTPUT MENU) после выполнения симплекс-алгоритма. Меню вывода появляется в строке функциональной клавиши
F9 (END SESSION)	У пользователя запрашивают верификацию конца сеанса. В конце сеанса все файлы закрыты и управление возвращается к операционной системе DOS
F10 (HELP)	На экране появляется сообщение об авторских правах на LPX88 и все четыре меню программ
Меню установки (SETUP MENU)	
F1 (FILES LIST)	Запрашивает имя диска, и на экране появляется листинг всех файлов на диске с модификаторами имени .LP, .LPB, .LPS
F2 (NEW PROBLEM)	Текущая задача уничтожается. У пользователя запрашивается имя и размерность новой задачи. Первоначально все коэффициенты равны нулю по умолчанию, а все соотношения в ограничениях являются равенствами (=). Переменным и ограничениям присваиваются имена по умолчанию. Затем должно следовать имя задачи, MAX или MIN, число ограничений и количество нефиктивных переменных
F3 (DISPLAY EDITOR)	Включает редактор. На экране для редактирования появляется сегмент текущей задачи размерностью 15 строк на 10 столбцов. Если экранный редактор уже используется, то появится новый сегмент. У пользователя запрашивается строка и столбец ID элемента в верхнем левом углу
F4 (SAVE PROBLEM)	В названный пользователем файл записывается текущая задача. К имени файла автоматически добавляется модификатор .LP. Затем должно следовать имя текущей задачи, которое будет использоваться как имя файла
F5 (DELETE FILE)	У пользователя запрашивается имя. Удаляются все файлы, имеющие имена с модификаторами .LP, .LPB, .LPS
F6 (OLD PROBLEM)	Осуществляет поиск записанной ранее задачи и делает ее текущей. Имя задачи вводится пользователем
F7 (INPUT FILE)	Создается новая задача, как и при нажатии F2, однако ненулевые элементы массива коэффициентов, строки целевой функции, столбца правой части считываются с последовательного файла, названного пользователем. Имена и соотношения в ограничениях могут также вводиться из файла. Затем должно следовать имя задачи, MAX или MIN, число ограничений, количество нефиктивных переменных

Клавиша	Функция
F8 (PRINT ARRAY)	Печатает или посылает в выходной файл листинг всей текущей задачи
F9 (MASTER MENU)	Возвращает управление главному меню. В строке функциональной клавиши появляется выборка главного меню
F10 (HELP)	На экране появляется сообщение об авторских правах и все четыре меню программ
Меню выполнения (EXECUTION MENU)	
F1 (PAUSE/CONTINUE)	Переключатель паузы. LPX88 остановится до опорной точки. Делая паузу, LPX88 продолжает выполнение программы. Во время паузы имя задачи показано на экране в реверсивном цвете
F2 (STEP/CONTINUE)	Переключатель шага. LPX88 сделает паузу после каждой опорной точки во время поиска опорных точек. Во время переключения шага LPX88 возобновит непрерывное выполнение программы
F3 (INVERT MATRIX)	Базис обратной матрицы должен быть повторно вычислен для текущего базиса
F4 (CHANGE LIMITS)	На экран выводятся текущие значения всех вычислительных пределов, допусков и управлений. Теперь пользователь может их изменить. В пакетном режиме затем должен следовать ряд управляющих индексов и новые значения
F5 (PIVOT ON X)	Пользователь обозначает небазисную переменную для ввода в базис в следующей опорной точке. LPX88 выбирает переменную, оставляющую базис, в соответствии с общими правилами
F6 (DELTA X)	Вычисляется приведенная стоимость (или предельная полезность) и выводится на экран для небазисной переменной, выбранной пользователем. LPX88 идентифицирует также переменную, покидающую базис, в соответствии с общими правилами
F7 (SHORT PRINTOUT)	Печатает или посылает в выходной файл краткий итог решения, соответствующий текущему базису
F8 (RECORD)	Переключатель записи опорной точки. Включает PIVOTS и включает печать строки информации для каждой опорной точки
F9 (MASTER MENU)	Возвращает управление главному меню. В строке функциональной клавиши вновь появляется выборка главного меню
F10 (HELP)	На экране появляется сообщение об авторских правах и все четыре меню программ
Меню вывода (OUTPUT MENU)	
F1 (PRIMAL VALUES)	Сохраняет в выходной файл таблицу с информацией о каждой переменной (включая фиктивные)
F2 (DUAL VALUES)	Сохраняет в выходной файл таблицу с информацией о каждом ограничении
F3 (COST RANGES)	Сохраняет в выходной файл таблицу с информацией о каждом значении переменной
F4 (RHS RANGES)	Сохраняет в выходной файл таблицу с информацией о каждом значении ограничения
F5 (INVERSE MATRIX)	Печатает или посылает в выходной файл листинг всего базиса решения массива обратной матрицы

Клавиша	Функция
F6 (INV*A MATRIX)	Печатает или посылает в выходной файл листинг матричного произведения базиса решения обратной матрицы и матрицы коэффициентов небазисных столбцов
F7 (SAVE SOLUTION)	Записывает решение задачи линейного программирования и двойственной к ней в файл, названном пользователем
F8 (SAVE BASIS)	Записывает текущий базис с именем файла, указанным пользователем. К имени файла, названному пользователем, автоматически добавляется модификатор .LP

Задания для самостоятельной работы

1. Откройте LPX88 и создайте новую задачу с любым именем.
2. Закройте LPX88.
3. Откройте LPX88, создайте новую задачу, измените количество переменных на 5.

Контрольные вопросы

1. Какие функциональные клавиши необходимы для решения задачи с помощью LPX88?
2. Как получить справочную информацию о программе LPX88?

2. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Задача 1

Получить оптимальный рацион кормления среднегодовой коровы, минимальный по себестоимости.

Исходная информация:

1. Оптимальный рацион рассчитать на базе следующих кормов: концентраты, сено, сенаж, картофель, зеленый корм и солома.
2. Планируемый удой равен 35 ц. Для получения 1 ц молока необходимо израсходовать 1,15 ц к. ед.
3. На 1 ц к. ед. должно приходиться не менее 10,3 кг переваримого протеина (ПП).
4. В соответствии с зоотехническими требованиями питательность грубых кормов не должна превышать 30 %, а сочных – 40 % от питательности рациона.

5. В группе грубых кормов (сено, солома, сенаж) на долю соломы приходится 35 % от питательности кормов этой группы.

6. Ограничения по скормливанию отдельных видов кормов: концентраты – от 10 до 15 ц, зеленый корм – от 40 до 80 ц.

7. Характеристика кормов приведена в табл. 3.

Таблица 3. Характеристика кормов

Корма	Содержится в 1 ц корма		Себестоимость 1 ц корма, у. д. е.
	ц к. ед.	кг ПП	
Концентраты	1,05	10,9	65
Сено	0,50	5,3	36
Сенаж	0,28	3,5	27
Картофель	0,30	1,0	17
Зеленый корм	0,19	2,1	19
Солома	0,25	3,3	12

Задача 2

Рассчитать программу размещения культур по полям, которая обеспечит получение максимума прибыли.

Исходная информация:

1. Поля и их площадь: 1-е поле – 40 га, 2-е поле – 30, 3-е поле – 50 га.

2. Площадь посева культур, прибыль от реализации продукции, полученной с 1 га, представлены в табл. 4.

Таблица 4. Характеристика культур

Наименование культуры	Прибыль с 1 га, у. д. е.			Площадь посева, га
	1-е поле	2-е поле	3-е поле	
Яровая пшеница	620	490	520	65
Ячмень	450	460	420	10
Озимая рожь	580	560	700	45

Задача 3

Исходная информация. Три агроторговых предприятия (АТП) области, занимаясь выращиванием овощей, получили возможность их реализовать в четырех специализированных магазинах. Наличие овощей в хранилищах первого АТП – 600 т, второго – 900, третьего – 800 т. Потребности магазинов, исходя из мощности реализации, со-

ставляют соответственно 720, 570, 310, 700 т. Расстояния от АТП до магазинов приведены в табл. 5.

Таблица 5. **Расстояния от АТП до магазинов**

АТП	Магазин			
	1	2	3	4
	Расстояние от АТП до магазина, км			
1	77	52	84	64
2	68	55	60	75
3	49	50	46	48

Составить оптимальный план перевозки овощей из агроторговых предприятий в овощные магазины, чтобы затраты на перевозки оказались минимальными.

Задача 4

Составить оптимальный план трансформации земельных угодий с целью получения максимума дополнительной прибыли.

Исходная информация:

1. Ресурсы: труд – 2 000 чел.-дн., материально-денежные средства – 1 500 руб.

2. Возможные виды трансформации: сенокосы в пашню, сенокосы в пастбища и прочие угодья в пастбища.

3. Показатели расхода ресурсов и их окупаемость представлены в табл. 6.

Таблица 6. **Расход ресурсов и их окупаемость**

Показатели	Единицы измерения	Виды трансформации		
		Сенокосы в пашню	Сенокосы в пастбища	Прочие угодья в пастбища
Труд	чел.-дн.	20	50	40
Материально-денежные средства	у. д. е.	20	50	40
Дополнительная прибыль на 1 га	у. д. е.	40	100	90

4. Максимальная площадь трансформации прочих угодий в пастбища 20 га.

5. Минимальная площадь трансформации сенокосов в пастбища 10 га.

6. Минимальная площадь трансформации сенокосов в пашню 20 га.

Задача 5

Исходная информация. Определить оптимальный объем производства различных видов молочной продукции, обеспечивающий получение максимальной прибыли перерабатывающему предприятию при следующих условиях:

1. Предполагаемый объем поставки молока на переработку 20 тыс. т за год.

2. Предприятие может выпускать следующие виды продукции: масло животное, сыр жирный, цельномолочную продукцию (ЦМП). Расход молока на производство 1 т соответственно составляет 21; 9 и 0,91 т.

3. Максимальный объем производства исходя из мощности имеющихся линий, т: масло животное – 700; сыр жирный – 420; ЦМП – 8 200.

4. Для удовлетворения потребности региона минимальные поставки продукции на рынок составляют, т: масло животное – 390; сыр жирный – 210; ЦМП – 5 900.

5. Постоянные производственные затраты 1 300 у. д. е. Кроме того, для производства 1 т масла необходимо дополнительно затратить 6,5 у. д. е; сыра – 12,5; ЦМП – 0,59 у. д. е.

6. Цены реализации, у. д. е/т: масло животное – 9,5; сыр жирный – 13,7; ЦМП – 0,81.

Задача 6

Исходная информация:

1. Арендное подразделение сельскохозяйственного производственного кооператива (СПК) имеет следующие марки тракторов: МТЗ-80, МТЗ-82, МТЗ-952.

2. Согласно плановому хозрасчетному заданию на ремонты техники и технические уходы выделено 17 300 у. д. е. Минимальный объем денежных средств, который необходимо выделить для МТЗ-80, составляет 3 000 у. д. е. Исходя из особенностей состояния технического парка арендного подразделения, для тракторов МТЗ-82 предусмотрено направить не более 6 000 у. д. е.

3. Для обслуживания техники имеется ремонтная бригада, фонд рабочего времени которой составляет 652 чел.-ч. Трудоемкость ремонта и техуходов в расчете на 1 у. д. е. составляет по тракторам, чел.-ч: МТЗ-80 – 0,05; МТЗ-82 – 0,04; МТЗ-952 – 0,03.

4. Использование финансовых средств в данном арендном подразделении обеспечивает увеличение производительности тракторов, усл. эт. га на 1 у. д. е. затрат: МТЗ-80 – 0,3; МТЗ-82 – 0,32; МТЗ-952 – 0,24.

Составить условия экономико-математической задачи оптимизации использования.

Задача 7

Исходная информация:

1. Организация рекламирует свою продукцию с использованием следующих средств массовой информации: телевидения, радио, газет, журналов, афиш.

2. Из проведенных за прошлые периоды маркетинговых исследований известно, что эти средства приводят к увеличению прибыли соответственно на 7, 9, 4, 12, 3 у. д. е. в расчете на 1 у. д. е. затраченных средств.

3. Рекламный бюджет не должен быть более выделенной суммы, равной 37 тыс. у. д. е.

4. На афиши следует расходовать не менее 10 и не более 15 % рекламного бюджета.

5. На рекламу на телевидении должно быть выделено не менее 30 и не более 40 % рекламного бюджета.

6. На рекламу на радио целесообразно расходовать не менее $\frac{3}{4}$ той суммы, которую планируется использовать на телевидении.

7. Затраты на рекламу в газетах должны составить $\frac{1}{2}$ от средств, выделяемых на рекламу в газетах и журналах.

Используя приведенную информацию, необходимо:

1) ввести неизвестные величины задачи;

2) составить условия экономико-математической задачи, записать целевую функцию.

ЛИТЕРАТУРА

1. Канторович, Л. В. Математические методы организации планирования производства / Л. В. Канторович. – Л.: ЛГУ, 1939.
2. Гасс, С. Линейное программирование (методы и приложения) / С. Гасс. – М.: Гос. изд-во физ.-мат. лит., 1961. – (Физико-математическая библиотека инженера).
3. Муравьев, В. И. Метод последовательного улучшения с базисом переменного размера для задач линейного программирования / В. И. Муравьев // Исследование операций и методы статистического моделирования: сб. – Л.: ЛГУ, 1972.
4. Таха, Хемди А. Введение в исследование операций = Operations Research: An Introduction / Хемди А. Таха. – 7-е изд. – М.: Вильямс, 2007. – Гл. 3: Симплекс-метод. – С. 95–141.
5. Акулич, И. Л. Математическое программирование в примерах и задачах / И. Л. Акулич. – М.: Высш. шк., 1986. – Гл. 1: Задачи линейного программирования. – 319 с.
6. Алгоритмы: построение и анализ = Introduction to Algorithms / Томас Х. Кормен [и др.]. – 2-е изд. – М.: Вильямс, 2006. – Гл. 29: Линейное программирование. – С. 1296.
7. Шевченко, В. Н. Линейное и целочисленное линейное программирование / В. Н. Шевченко, Н. Ю. Золотых. – Н. Новгород: Изд-во Нижегород. гос. ун-та им. Н. И. Лобачевского, 2004. – С. 63–66.
8. How good is the simplex algorithm? // Inequalities III (Proceedings of the Third Symposium on Inequalities held at the University of California, Los Angeles, Calif., September 1–9, 1969, dedicated to the memory of Theodore S. Motzkin). – New York – London: Academic Press, 1972. – P. 159–175.

СОДЕРЖАНИЕ

Введение.....	3
1. Функциональные особенности программного пакета LPX88.....	5
1.1. Пример выполнения задачи линейного программирования в LPX88.....	6
1.2. Ввод исходных данных.....	16
1.3. Сохранение и загрузка файла задачи.....	20
1.4. Решение задачи.....	22
1.5. Методы решения задачи.....	25
1.6. Получение отчетов.....	31
1.7. Функциональные клавиши.....	35
2. Задачи линейного программирования.....	38
Литература.....	43