Предисловие для преподавателя

Наверное, все преподаватели современной высшей школы сталкиваются в своей работе с одной и той же проблемой: уменьшение количества аудиторных часов, отведенных на изучение той или иной дисциплины, из-за чего всё большее и большее количество вопросов приходится оставлять для самостоятельной работы студента. Это зачастую сочетается с полным неумением и нежеланием студентов самим искать информацию, по крайней мере, когда речь идет о математических дисциплинах. Способность заниматься самостоятельно, которую должна была воспитать в учениках еще средняя школа, у многих современных студентов развита очень слабо. Одновременно наблюдаются серьезные пробелы в их школьном образовании. Автору этой книги приходилось иметь дело с бакалаврами, которые свободно владеют несколькими языками программирования, но при этом не могут изобразить прямую на плоскости, тем более — графически решить систему линейных неравенств. Или с учащимися, которые неплохо знают экономику, но не умеют выполнять четыре действия арифметики в простых дробях.

В результате практические занятия по линейному программированию очень часто сводятся к работе с какими-нибудь пакетами прикладных программ. Само по себе это, конечно, неплохо, но в результате многие выпускники способны только на то, чтобы ввести в компьютер условие задачи, а потом записать ответ, механически и бездумно, без понимания сути происходящего.

Исправить сложившуюся ситуацию призвана домашняя контрольная работа, описанная в данной книге. В сущности, это система связанных между собой заданий, подобранных с таким расчетом, чтобы продемонстрировать обучающемуся основные идеи линейного программирования. В результате в процессе выполнения своего варианта контрольной работы студент сможет ознакомиться с простейшими алгоритмами, лежащими в основе линейного программирования, и на примере своих задач убедиться в справедливости основных его теорем. Так, решая одну и ту же задачу сперва графически, а затем симплекс-методом, легко увидеть, что каждому допустимому базисному плану соответствует одна из вершин области допустимых планов; что каждому псевдоплану соответствует одна из псевдовершин. Сравнивая друг с другом решение прямой и двойственной задач, нетрудно проверить, выполняются ли для них теоремы двойственности. Впрочем, сразу оговоримся, что в данной книге теоретический материал рассмотрен только в той мере, в какой это необходимо для решения задач, и для более глубокого знакомства с теорией учащимся следует обратиться к учебникам по линейному программированию или по исследованию операций. Иногда одни и те же сведения повторяются по нескольку раз при решении разных задач. Это позволяет обучающемуся не прорабатывать книгу последовательно, страница за страницей, а использовать ее в качестве справочника.

Заметим, что, снабдив студента очень четким и подробным образцом решения предложенных ему задач, преподаватель сможет не расходовать те немногие часы, которые отводятся ему для практических или лабораторных занятий, на отработку элементарных навыков графического решения неравенств или пересчета симплекс-таблиц. Вместо этого аудиторные часы можно будет потратить на обсуждение тех проблем, для изучения которых действительно необходимо личное общение студента и преподавателя. Например, можно выяснить, как происходит переход от словесного и несовершенного описания одной конкретной ситуации к обобщенной абстрактной модели, как выглядит переход от словесной модели к математической. Можно вырабатывать у обучающихся умение правильно классифицировать модель, можно провести сравнительный анализ достоинств и недостатков аналитических и численных методов решения задач оптимизации, обсудить проблемы зацикливания и устойчивости решений. Примеры более сложных теоретических вопросов приведены в Приложении 2; часть из них можно предложить студентам для написания рефератов и курсовых работ.

Еще раз отметим, что задания контрольной работы подобраны с таким расчетом, чтобы обучающийся имел возможность сравнить друг с другом первую и вторую геометрическую интерпретации задач линейного программирования, особенности графического решения и решения симплекс-методом, свойства прямой и двойственной задач. С одной стороны, мы даем обучающемуся возможность увидеть их взаимосвязь, понять, например, как изменяется решение задачи, когда целевая функция изменяет знак или когда в задачу добавляется условие целочисленности. С другой стороны, это облегчает преподавателю процесс проверки заданий. Большая часть заданий зависит от параметров, что дает возможность без труда найти общую формулу, по которой записывается ответ. Кроме того, во многих упражнениях студенту предлагается решить другим методом уже решенные ранее задачи и сравнить результат.

Вообще говоря, предполагается, что студенты будут выполнять свою контрольную работу письменно, в специальной тетради. С другой стороны, умение программировать всегда приветствуется и поощряется. По крайней мере, в задачах, требующих относительно сложных расчетов (например, в заданиях 2 и 3а из работы 2 или в работе 4) всю вычислительную часть можно выполнять с помощью прикладных программ, а еще лучше — с помощью программ, написанных самостоятельно на любом языке программирования. Таким образом, мы позволяем студентам приобретать дополнительные навыки программирования и одновременно — привить себе культуру вычислений и умение работать с большими объемами числовой информации, что, например, очень важно для будущих экономистов.

Примерные правила выполнения работы приведены в Приложении 3. Различные темы линейного программирования распределены по задачам контрольной работы следующим образом (римские цифры обозначают номер работы, арабские — номера задач).

- Графическое решение ЗЛП **I** 1–3
  - на плоскости **I** 1–2
    - когда область допустимых планов D ограничена **I** 1,2г
    - когда область D неограничена **I** 2 а,б
    - когда область D пуста **I** 2 в
  - в пространстве **I** 3
- ЗЛП в векторной форме, 1-я и 2-я геометрические интерпретации, их взаимосвязь **I** 4–5
- Построение КЗЛП в общем случае **I** 6
- Симплекс-метод **II**
  - прямой **II** 1–3
    - построение первой симплекс-таблицы методом элементарных преобразований **II** 1 б,в
    - построение первой симплекс-таблицы матричным методом **II** 2

- - - случай неограниченной целевой функции **II** 1 б
- - - случай, когда область допустимых планов пуста **II** 3 б
- - - поиск первоначального допустимого базиса **II** 1,3
    - - - - путем подбора и элементарных преобразований **II** 1
    - - - - решением вспомогательной задачи **II** 3
- – двойственный **II** 4,5 а; **III** 3; **IV** 1 в, 2 в
- – исследование на устойчивость, изменение условий задачи **II** 5

- Двойственность в линейном программировании **III**
    - построение двойственной задачи **III** 1, 2
    - теоремы двойственности **III** 2
    - решение прямой и двойственной задач симплекс-методом **III** 3

- Целочисленное линейное программирование **IV**
    - метод ветвей и границ **IV** 1б, 2б
    - метод Гомори **IV** 1в, 2в

Напоследок сделаем такое полезное замечание. В данной книге всегда предполагается, что в канонической задаче линейного программирования целевая функция стремится к *максимуму*, и при заполнении симплекс-таблицы ее коэффициенты *меняют знак*. Строка оценок записывается в верхней части таблицы и называется «нулевой». Поскольку в других источниках предпочтение может отдаваться другой терминологии и другой системе записи, имеет смысл объяснить студенту, что строку оценок можно записывать снизу, а не сверху; что в канонической задаче целевую функцию можно устремлять к минимуму, а не к максимуму; что в симплекс-таблицу можно записывать целевую функцию, **не изменяя** знаки ее коэффициентов, но тогда условием оптимальности будет **неположительность** всех чисел в строке оценок. Важно каждый раз понимать, какой системой записи мы пользуемся при решении конкретной задачи и как можно без труда переходить от одной системы записи к другой.

Наконец, все симплекс-таблицы записываются в развернутой форме, что, на взгляд автора, целесообразно при первоначальном знакомстве с симплекс-алгоритмом. Возможные сокращенные формы записи симплекс-таблиц, как и модифицированный симплекс-метод, обсуждаются в Приложении 2.

Предисловие для студента

В данном учебном пособии рассказано о том, какими способами можно решать *задачи линейной оптимизации*, то есть о том, как находить наибольшее и наименьшее значение *линейной функции* f — целевой функции — в области *допустимых планов* D, если область D можно описать как множество решений системы *линейных уравнений и неравенств*.

Напомним, что *линейная функция* n переменных имеет вид

$$f(x_1, x_2, \ldots, x_n) = a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n + b,$$

где a_1, a_2, $\ldots$, a_n и b — некоторые (числовые) коэффициенты. Соответственно, *линейное уравнение*

$$a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_n \cdot x_n + b = 0,$$

задает в пространстве переменных $(x_1, x_2, \ldots, x_n)$ некоторое множество точек, которое называется *линейным многообразием.* Примерами линейных многообразий служат прямые на плоскости и в пространстве, плоскости в пространстве, отдельные точки. Задавая систему линейных уравнений и неравенств, мы можем описать отдельную точку, отрезок, многоугольник, многогранник в трехмерном пространстве, а в случае $n > 3$ — так называемые *многомерные многогранники*, которые уже не допускают наглядной геометрической интерпретации. Иначе говоря, линейные многообразия и полученные с их помощью многогранники — это объекты с наиболее простыми геометрическими свойствами. Точно так же линейная функция — наиболее простая из всех числовых функций, задаваемых аналитически. Если нам известно, что какие-то две величины связаны друг с другом *линейным соотношением*

$$Y = kX + b,$$

то достаточно будет провести всего два эксперимента, чтобы по двум точкам (X_1, Y_1) и (X_2, Y_2) восстановить значения параметров k и b. После чего мы легко сможем изобразить зависимость Y от X графически и предсказать, как будет меняться Y при изменении X.

Почему же линейные модели настолько важны? Почему в высшей математике существует особый раздел, который называется «линейная алгебра», а вот отдельной квадратичной алгебры, тригонометрической алгебры или логарифмической алгебры не существует? Почему все модели оптимизации принято делить на линейные и нелинейные? Почему предмет «Исследование операций» традиционно включает в себя особый и очень важный раздел под названием «линейное программирование»?

То, что в математике именно линейные функции, линейные задачи и линейные модели занимают особое положение и в какой-то мере противопоставляются всем остальным, можно объяснить двумя причинами.

О первой причине мы уже упоминали: линейные модели **всегда** очень просты. Если мы рассмотрим *систему линейных уравнений*

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2 \\ \dotfill \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n = b_m, \end{cases}$$

или *задачу линейной оптимизации*

$$\begin{cases} c_1x_1 + c_2x_2 + \ldots + c_nx_n \to \max(\min) \\ a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n \leqslant b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n \leqslant b_2 \\ \dotfill \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n \leqslant b_m, \end{cases}$$

то можно указать достаточно простые алгоритмы (метод Гаусса в первом случае и симплекс-метод во втором), которые *всегда* позволяют ответить на вопрос, имеют ли данные задачи решение, и полностью описать множество решений, даже если оно бесконечно велико. Для систем нелинейных уравнений, как и для задач нелинейной оптимизации, это удается сделать только в исключительных, наиболее простых случаях. То же самое можно, например, сказать о системах линейных и нелинейных дифференциальных уравнений. Отсюда, в частности, следует, что любая линейная модель *всегда предсказуема*: зная начальное состояние системы, мы сразу можем определить, как она поведет себя в дальнейшем.

Кроме относительной простоты, у линейных функций есть второе, не менее важное качество: их можно использовать для приближенной записи других, более сложных функций. Действительно, любая дифференцируемая сколь угодно много раз функция (то есть, в частности, любая элементарная функция) удовлетворяет следующей *формуле Тейлора*:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \ldots,$$

причем *при* x, *близких к* x_0, все слагаемые, начиная с третьего, пренебрежимо малы по сравнению с первыми двумя. При фиксированном x_0 значения $f'(x_0) = a$ и $f(x_0) = b$ — две константы, так что справедливо приближенное равенство $f(x) \approx b + a(x - x_0)$, которое выполняется тем точнее, чем меньше модуль разности $|x - x_0|$. Как видим, *локально* (то есть вблизи некоторой точки) произвольная дифференцируемая функция $f(x)$ ведет себя *почти так же*, как линейная функция $y = b + a(x - x_0)$.

Аналогичными свойствами обладают и функции нескольких переменных. Для них справедливо приближенное равенство

$$f(x_1, x_2, \ldots, x_n) \approx b + a_1(x_1 - x_1^0) + a_2(x_2 - x_2^0) + \ldots + a_n(x_n - x_n^0),$$

где $b = f(x_1^0, x_2^0, \ldots, x_n^0)$, $a_i = \dfrac{\partial f(x_1^0, x_2^0, \ldots, x_n^0)}{\partial x_i}$ — константы, зависящие от выбора исходной точки $x^0(x_1^0, x_2^0, \ldots, x_n^0)$. Отсюда, в частности, следует, что, изучая систему *нелинейных* дифференциальных уравнений

$$\begin{cases} x_1' = f_1(x_1, x_2, \ldots, x_n) \\ x_2' = f_2(x_1, x_2, \ldots, x_n) \\ \dots\dots\dots\dots\dots\dots\dots\dots \\ x_m' = f_m(x_1, x_2, \ldots, x_n), \end{cases}$$

где $x_1, x_2, \ldots, x_n$ — неизвестные функции, мы можем выбрать какую-нибудь начальную точку $x_1^0, x_2^0, \ldots, x_n^0$ и заменить исходную систему на систему *линейных* дифференциальных уравнений

$$\begin{cases} x_1' = b_1 + a_{11}(x_1 - x_1^0) + a_{12}(x_2 - x_2^0) + \ldots + a_{1n}(x_n - x_n^0) \\ x_2' = b_2 + a_{21}(x_1 - x_1^0) + a_{22}(x_2 - x_2^0) + \ldots + a_{2n}(x_n - x_n^0) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ x_m' = b_m + a_{m1}(x_1 - x_1^0) + a_{m2}(x_2 - x_2^0) + \ldots + a_{mn}(x_n - x_n^0). \end{cases}$$

Можно доказать, что в большинстве случаев так называемые *качественные* характеристики системы (существование и единственность решений, поведение решений на бесконечности, зависимость решений от некоторого параметра) после такой *линеаризации* не изменяются. Конечно, подобная замена допустима только *локально*, то есть вблизи точки $x^0(x_1^0, x_2^0, \ldots, x_n^0)$, и погрешность решения будет тем больше, чем больше разница между точками x и x^0. Однако, выбирая различные x^0, можно получить достаточно полное представление о поведении исходной нелинейной системы в целом.

Благодаря своим достоинствам линейные функции широко используются в различных математических моделях. Поскольку любая модель предполагает идеализацию и упрощение изучаемой системы, имеет смысл выбирать модель с таким расчетом, чтобы ей соответствовали линейные функции, линейные уравнения и неравенства. Предположение о том, что в той или иной ситуации для описания системы допустимо использовать линейные функции, называют *гипотезой линейности*. Разумеется, гипотезу линейности каждый раз необходимо обосновывать, доказывая, что даже если в результате модель получается несколько искаженной, это не повлияет на её основные качественные характеристики. Например, в задачах механики часто пренебрегают трением и другими силами сопротивления (если они достаточно малы и не оказывают существенного влияния на происходящие процессы). В моделях биологии и экономики пренебрегают обратной связью и эффектом масштаба.

Однако надо отметить, что в некоторых случаях переход к линейной модели невозможен, так как возникающие искажения будут слишком велики. Принято считать, что линейные модели не имеет смысла использовать в метеорологии, в теории вихрей, при изучении некоторых биологических и социальных процессов, при определении цен на денежных и товарных рынках.

Впрочем, вопрос о допустимости и недопустимости перехода к линейной модели выходит за рамки курса линейного программирования и данного учебного пособия. Цель данной книги — научить вас решать задачи линейной оптимизации, что даст вам возможность исследовать линейную модель в тех случаях, когда линеаризация допустима.

Для первоначального знакомства с линейным программированием достаточно знать алгебру и геометрию в объеме средней школы. Желательно, хотя во многих случаях и необязательно, иметь представление об алгебре матриц и о базисах векторного пространства.