

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики – процессов управления  
Кафедра математической теории экономических решений

**Буденкова Анастасия Алексеевна**

**Выпускная квалификационная работа**

**Построение планов производства по выпуску  
продукции в условиях случайных возмущений**

Научный  
руководитель,  
доктор физико-  
математических  
наук, профессор

**Колбин В.В.**

Рецензент,  
кандидат физико-  
математических  
наук, доцент

**Карелин В.В.**

Санкт-Петербург  
2020

# Оглавление

Введение .....	3
Глава 1. Двухэтапная задача стохастического программирования .....	4
1.1 Постановка задачи стохастического программирования .....	4
1.2 Двухэтапная задача Данцига – Маданского .....	4
Глава 2. Задача оптимизации.....	7
2.1 Линейное программирование .....	7
2.2 Симплекс метод.....	7
Глава 3. Второй этап задачи Данцига-Маданского .....	10
3.1 Основные определения из теории вероятностей .....	10
3.2 Рандомизация задачи линейного программирования .....	11
Глава 4. Применение .....	12
Выводы .....	17
Список литературы .....	18
Приложение .....	19

## Введение

У каждого предприятия есть план выпуска продукции или, другими словами, производственный план. То есть руководство заранее решает сколько и к какому сроку должно быть произведено продукции. При этом можно нацеливаться на достижение максимальной прибыли или на уменьшение затрат на производство.

Как правило, исходная информация для планирования в экономике не совсем точна, то есть, планирование происходит в условиях неполной информации о параметрах поставленной задачи. Для решения и рассмотрения таких задач существует раздел математического программирования – стохастическое программирование. Методы и математические модели стохастического программирования хорошо приспособлены к выбору решения в сложных ситуациях, где имеет место случайность, неточность, изменение спроса и предложения.

В работе рассматривается динамическая двухэтапная задача стохастического программирования Данцига-Маданского типа: «решение – наблюдение – решение». Где на первом этапе в условиях неполной информации определяется некоторый начальный план производства, затем после уточнения данных (наблюдения) начальный план корректируется.

# Глава 1. Двухэтапная задача стохастического программирования

## 1.1 Постановка задачи стохастического программирования

Многие задачи планирования и управления в условиях неполной информации рассматриваются и решаются как двухэтапные задачи стохастического программирования. Чаще всего встречаются стохастические задачи с компенсацией невязок в системах ограничений. Решением двухэтапной задачи являются случайный и детерминированный вектора. На первом этапе (до реализации случайных значений) находится начальный (предварительный) план, а после наблюдения реализации случайных параметров находится случайный вектор, компенсирующий появляющиеся невязки.

## 1.2 Двухэтапная задача Данцига – Маданского

Будем рассматривать линейную задачу стохастического программирования:

$$\begin{cases} \min c^T x, \\ Ax = b(\omega), \\ x \geq 0. \end{cases} \quad (1.2.1)$$

Любой фиксированный вектор  $x^0 \geq 0$ , являющийся допустимым решением задачи (1.2.1) при фиксированной реализации вектора  $b$ , не будет являться допустимым для других реализаций. В стохастической задаче линейного программирования такие отклонения допускаются и компенсируются вектором  $(y^+(\omega), y^-(\omega))$ , если  $y_i^+(\omega) = b_i(\omega) - (Ax^0)_i$ ;  $y_i^-(\omega) = 0$ , если  $b_i(\omega) \geq (Ax^0)_i$ ;  $y_i^+(\omega) = (Ax^0)_i - b_i(\omega)$ , если  $b_i(\omega) < (Ax^0)_i$ .

В связи с этим появляются дополнительные затраты, определяемые величиной:  $f^T |b(\omega) - Ax^0|$ .

Цель управляющего состоит в том, чтобы минимизировать эти затраты, т.е. решить задачу для каждой реализации  $\omega$ :

$$\begin{aligned} \min f^T \{y^+(\omega) + y^-(\omega)\}, \\ Jy^+(\omega) + Jy^-(\omega) = b(\omega) - Ax^0, \\ y^+(\omega), y^-(\omega) \geq 0, \end{aligned} \quad (1.2.2)$$

где  $J$  – единичная матрица.

Задача (1.2.2) является задачей второго этапа по терминологии Данцига и Маданского.

На первом этапе определяем фиксированный вектор  $x$ , на втором выбираем вектор  $(y^+(\omega), y^-(\omega))$ , который будет наиболее экономно компенсировать появляющиеся отклонения.

Задачи (1.2.1) – (1.2.2) можно объединить в одну:

$$\max_x \{c^T x + E \min_y f^T(y^+(\omega) + y^-(\omega))\}, \quad (1.2.3)$$

где  $x$  и  $(y^+(\omega), y^-(\omega))$  удовлетворяют ограничениям

$$\begin{aligned} Ax + Jy^+(\omega) - Jy^-(\omega) = b(\omega), \\ x, y^+(\omega), y^-(\omega) \geq 0. \end{aligned} \quad (1.2.4)$$

*Лемма 1.* [1] Функция  $c^T x + E \min_y f^T(y^+(\omega) + y^-(\omega))$  выпуклая по  $x$ .

Из леммы следует, что задача линейного программирования с случайным параметром  $b$  становится задачей выпуклого программирования с линейными ограничениями. И задача (1.2.2) имеет допустимое решение для каждого фиксированного  $\omega$ .

Поэтому (1.2.3) – (1.2.4) сводится к задаче:

$$\min_{x \geq 0} E \{c^T x + f^T |b(\omega) - Ax|\}. \quad (1.2.5)$$

*Лемма 2.* [1] Пусть  $x^*$  – решение задачи (1.2.3) – (1.2.4). Тогда  $f(x^*) \leq f(\bar{x}^*)$ , где  $f(x)$  – функционал в данной задаче.

Данная лемма даёт верхнюю границу для оптимального значения целевой функции задачи (1.2.3) – (1.2.4).

Рассмотрим задачу в более обобщённом виде:

Задача второго этапа:

$$\begin{aligned}
& \min f^T y(\omega), \\
& By(\omega) = b(\omega) - Ax^0, \\
& y(\omega) \geq 0,
\end{aligned}
\tag{1.2.6}$$

$B$  – матрица  $m \times p$ .

Объединяя задачи (1.2.1), (1.2.6), получаем задачу Данцига:

$$\begin{aligned}
& \min_x \left\{ c^T x + E \min_{y(\omega)} f^T y(\omega) \right\}, \\
& Ax + By(\omega) = b(\omega), \\
& x, y(\omega) \geq 0.
\end{aligned}
\tag{1.2.7}$$

## Глава 2. Задача оптимизации

### 2.1 Линейное программирование

На первом и втором этапе задачи Данцига-Маданского по сути нужно решить задачу линейного программирования. Линейное программирование – это совокупность методов поиска экстремального значения линейной функции, аргументы которой подвержены ограничениям в виде равенств (неравенств). Целевая функция – функция, значение которой нужно максимизировать (минимизировать).

**Общий вид ЗЛП** выглядит так:

Есть линейная целевая функция

$$z(x) = \sum_{j=1}^m c_j x_j$$

И задана система линейных равенств (неравенств)

$$\sum_{j=1}^m a_{ij} x_j \leq b_i, \quad x_j > 0, i = 1 \dots k,$$

Стандартная форма записи ЛП в матричном виде:

$$\max z = \max CX$$

$$AX \leq B$$

$$X \geq 0$$

### 2.2 Симплекс метод

Одним из способов решения ЗЛП является *симплекс-метод*, он легко применим к задачам большой размерности. Суть его заключается в поиске первоначального опорного плана и дальнейшем переходе к следующим опорным планам, пока не будет достигнуто оптимальное решение ЗЛП.

Чтобы воспользоваться симплекс-методом, необходимо привести ЗЛП к каноническому виду:

$$\max z(x) = \max \sum_{j=1}^m c_j x_j$$

$$\sum_{j=1}^m a_{ij} x_j + s_i = b_i$$

$$x_j > 0$$

$$i = 1 \dots n$$

Симплексная таблица в этом случае будет выглядеть так:

	$x_1$	...	$x_m$	$s_1$	...	$s_n$	
$s_1$	$a_{11}$	...	$a_{1m}$	1	...	0	$b_1$
...	...	...			...	...	...
$s_n$	$a_{n1}$	...	$a_{nm}$	0	...	1	$b_n$
z	$-c_1$	...	$-c_j$	0	...	0	$b_0$

Таблица 1

где  $b_0$  - это значение целевой функции.

*Опорный план* - это любое решение ЗЛП, которое удовлетворяет системе ограничений. Опорным планом являются значения  $b_1, \dots, b_n$ , а *базисными переменными*  $s_1, \dots, s_n$ .

#### Алгоритм симплекс-метода:

Вычисления по алгоритму состоят в выполнении следующих однотипных итераций.

#### Итерация:

- 1) Если все коэффициенты в z-уравнении неотрицательны, то текущее базисное решение является оптимальным.
- 2) Иначе, среди отрицательных  $c_j$  находим минимальное. Столбец в котором оно находится является *ведущим*.
- 3) Если в ведущем столбце симплексной таблицы нет положительных коэффициентов, то значение задачи ЛП неограниченно (нет оптимального решения).

- 4) Ведущая строка находится по правилу  $\min \left( \frac{b_i}{a_{ij}} \right)$ ,  $a_{ij} > 0$ ,  $j$  - номер ведущего столбца.
- 5) Меняются местами переменные ведущей строки и ведущего столбца ( $x_j$  и  $s_i$ ).
- 6) Преобразование симплексной таблицы по методу Жордана-Гаусса.

# Глава 3. Второй этап задачи Данцига-Маданского

## 3.1 Основные определения из теории вероятностей

**Определение 3.1.1** [2] **Случайной величиной**, заданной на вероятностном пространстве  $\langle \Omega, \mathcal{F}, P \rangle$ , или  $\mathcal{F}$ -измеримой числовой функцией называется функция  $\xi$ , отображающая  $\Omega$  в множество  $\mathbb{R}$  вещественных чисел такая, что для любого  $B \in \mathcal{B}(\mathbb{R})$ :  $\xi^{-1}(B) = \{\omega \mid \xi(\omega) \in B\} \in \mathcal{F}$  (полный прообраз любого борелевского множества содержится в  $\sigma$ -алгебре  $\mathcal{F}$ ).

В дальнейшем случайный объект  $b$ , будем обозначать  $b(\omega)$ .

Определим функцию  $P_\xi(B)$  как функцию аргумента  $B \in \mathcal{B}(\mathbb{R})$  следующим образом:

$$P_\xi(B) = P(\xi^{-1}(B)) = P\{\xi \in B\}. \quad (3.1.1)$$

**Определение 3.1.2** [2] Функция  $P_\xi$ , определённая равенством (3.1.1) на  $\sigma$ -алгебре борелевских множеств, называется **вероятностным распределением** случайной величины  $\xi$ .

**Определение 3.1.3** [2] **Функцией распределения** случайной величины  $\xi$  будем называть функцию

$$F_\xi(x) = P\{\xi \leq x\}.$$

**Плотностью распределения** называют производную от функции распределения.

Если плотность распределения и функция распределения имеют следующий вид, то случайная величина распределена по **равномерному закону**:

$$f(x) = \begin{cases} \frac{1}{b-a}, & a < x < b, \\ 0, & x < a, x > b, \end{cases} \quad F(x) = \begin{cases} 0, & x < a, \\ \frac{x-a}{b-a}, & a < x < b, \\ 1, & x > b. \end{cases}$$

Или для интервала  $[-\Delta, \Delta]$ :

$$f(x) = \begin{cases} \frac{1}{2\Delta}, & -\Delta < x < \Delta, \\ 0, & x < -\Delta, x > \Delta, \end{cases} \quad F(x) = \begin{cases} 0, & x < -\Delta, \\ \frac{x+\Delta}{2\Delta}, & -\Delta < x < \Delta, \\ 1, & x > \Delta. \end{cases}$$

### 3.2 Рандомизация задачи линейного программирования

Так как на втором этапе задачи появляется случайная величина  $b(\omega)$ , то задача линейного программирования в этом случае будет выглядеть так:

$$z(x; \omega) = \sum_{j=1}^m c_j x_j \quad (3.2.1)$$

$$\sum_{j=1}^m a_{ij} x_j + s_i = b_i(\omega), x_j > 0, i = 1 \dots k. \quad (3.2.2)$$

Коэффициенты  $b_i(\omega)$  - случайные независимые величины.

Данную ЗЛП также можно решить симплекс-методом, так как в каждой реализации после разыгрывания случайной величины, случайные коэффициенты будут принимать конкретное значение.

Коэффициенты  $b_i$  - это случайные величины распределённые на некотором интервале  $[b_i - \Delta b_i, b_i + \Delta b_i]$ . Границы данного интервала можно условно задать как:

$$\Delta = \frac{\min_i b_i}{k} \quad \text{или} \quad \Delta = \frac{\max_i x_i}{k}$$

Где  $k$  – некоторая константа.

## Глава 4. Применение

### Рассмотрим двухэтапную задачу на примере пекарни:

Для производства хлебобулочных изделий применяется основное сырье (мука, вода, соль, дрожжи) и вспомогательное сырье (сахар, масло, яйца и др.). Сырье используется по определённой рецептуре, которая устанавливается в расчёте на 100 кг муки. В плане должны быть указаны все наименования производимых продуктов с указанием затрат сырья на единицу продукции. Известен расход сырья на единицу изделия. Рассматриваемые изделия: круассан, берлинер, сдобное изделие. Известны цены за единицу продукции, а также пределы количества производства каждого изделия (условные, среднестатистические).

Представим данные в виде таблиц:

	Круассан	Берлинер	Сдобное изделие	Ограничения сырья
Мука(г)	175	125	50	2 000 000
Дрожжи(г)	6	6	3	100 000
Сахар(г)	8	8	5	150 000
Соль(г)	4	1	2	85 000
Вода(мл)	125	100	50	1 500 000
Яйца(шт.)	1	1	0,5	15 000
Жир(г)	5	5	2	108 000
Цена продажи	75	50	60	
Затраты на производство	25	15	17	

Таблица 2

Ограничения на производство		
<b>круассан</b>	Не более	3 000
<b>берлинер</b>	Не более	5 000
<b>сдобное изделие</b>	Не более	2 500

Таблица 3

При помощи программы на первом этапе было найдено начальное решение:  $x_1=3000$  ,  $x_2 =3900$  ,  $x_3 =2500$  , целевая функция  $Z = 570\,000$ .

После случайной реализации на втором этапе вектора  $b$  (при коэффициенте  $k = 3$  и  $\Delta = \frac{\max x_i}{k}$ ) таблица выглядит следующим образом (без учёта того, что какое-то количество продукции уже произведено):

	Круассан	Берлинер	Сдобное изделие	Ограничения сырья
Мука(г)	175	125	50	2 001 423
Дрожжи(г)	6	6	3	100 771
Сахар(г)	8	8	5	151 937
Соль(г)	4	1	2	85 492
Вода(мл)	125	100	50	1 501 786
Яйца(шт.)	1	1	0,5	8 628
Жир(г)	5	5	2	108 959
Цена продажи	75	50	60	
Затраты на производство	25	15	17	

Таблица 4

А ограничения на производство в связи со случайным спросом поменялись так:

Ограничения на производство		
<b>круассан</b>	Не более	3 327
<b>берлинер</b>	Не более	5 729
<b>сдобное изделие</b>	Не более	1 411

Таблица 5

Количество сырья могло измениться за счёт остатков ресурсов, хранящихся на складе, или закупке дополнительного сырья.

Так как на первом этапе уже было произведено некоторое количество продукции, нужно вычислить количество ресурсов, которое остались для производства продукции на втором этапе. Программа произвела все необходимые вычисления, и, с учётом уже произведённой продукции, план производства выглядит следующим образом:

	Круассан	Берлинер	Сдобное изделие	Ограничения сырья
Мука(г)	175	125	50	863 923
Дрожжи(г)	6	6	3	51 871
Сахар(г)	8	8	5	84 237
Соль(г)	4	1	2	64 592
Вода(мл)	125	100	50	611 786
Яйца(шт.)	1	1	0,5	478
Жир(г)	5	5	2	69 459
Цена продажи	75	50	60	
Затраты на производство	25	15	17	

Таблица 6

Ограничения на производство		
<b>круассан</b>	Не более	327
<b>берлинер</b>	Не более	1829
<b>сдобное изделие</b>	Не более	-1089

Таблица 7

По таблице видно, что сдобных изделий было произведено на 1089 больше, чем требует спрос, следовательно, этот вид продукции производить больше не требуется, и, к тому же, это влечёт потери от невозможности реализации скоропортящейся продукции.

С учётом выше сказанного, ограничения на производство изменятся так:

Ограничения на производство		
<b>круассан</b>	Не более	327
<b>берлинер</b>	Не более	1829
<b>сдобное изделие</b>	Не производим	0

Таблица 8

Решая задачу второго этапа получаем вектор компенсаций  $y$ :

$$y_1 = 327$$

$$y_2 = 151$$

$$y_3 = 0.$$

**Суммарные затраты на производство равны: 186 440**

**Доход равен: 536 735**

**Прибыль составляет: 350 295**

Как видим, спрос на «круассаны» был полностью удовлетворён, продукции «берлинер» было произведено недостаточно, а сдобных изделий произведено слишком много, что повлекло за собой уменьшение возможной прибыли.

При другой реализации случайного спроса и коэффициенте  $k = 5$ , получили следующие результаты:

$$y_1 = 0$$

$$y_2 = 593$$

$$y_3 = 762.$$

**Суммарные затраты на производство равны: 197 849**

**Доход равен: 633 220**

**Прибыль составляет: 435 371**

## **Выводы**

В работе была рассмотрена и решена двухэтапная задача стохастического программирования на примере пекарни. Также была сделана её программная реализация в среде MATLAB. Результаты исследования можно использовать для прогнозирования прибыли в ситуациях случайных срывов производственного плана и в его планировании. Программу можно использовать для моделирования различных ситуаций в условиях случайного спроса, задавая необходимые отклонения параметров.

## Список литературы

1. Колбин В. В. Специальные методы оптимизации. Издательство “Лань”, 2014.
2. Буре В. М., Парилина Е. М. Теория вероятностей и математическая статистика. Издательство “Лань”, 2013.
3. Колбин В. В. Вероятностное программирование. Издательство “Лань”, 2016.
4. Юдин Д. Б. Математические методы управления в условиях неполной информации. Издательство “Советское радио”, 1974.
5. Чинаев П. И., Минин Н. А., Перевозников А. Ю., Черенков А. А. Высшая математика, специальные главы. Киев, издательство «Вища школа», 1981.
6. [Электронный ресурс]: - [Производственный план пекарни](#), дата посещения 12.05.2020

# Приложение

## Приложение 1. Листинг программы

```
clc
clear
SE = [25 15 17]; %себестоимость изготовления ед. продукции
A1 = [175 125 50;6 6 3;8 8 5;
      4 1 2;125 100 50;1 1 0.5;
      5 5 2;1 0 0;0 1 0;0 0 1]; %технологическая матрица
%вектор ограничений b
BB = [2000000 100000 150000 85000 1500000 8150 108000 3000
      5000 2500];
C1 = [75 50 60]; %цена продажи ед. продукции
n = length(BB);
mm = length(C1);
AA = [A1,eye(n)];
CC=[-1*C1,zeros(1,n)];
m = length(CC);

[XX,Z]=simplex(AA,BB,CC,mm);
disp('значение целевой функции:');
disp(Z);
disp('значение вектора x:');
disp(XX(1:mm));

%выбор отклонений
max_x = max(XX(1:mm));
k = 3;
delta = max_x / k;
Delta = 2 * delta;
A=AA;
B=BB;
C=CC;

%рандомизация параметров
for i = 1:n-mm
B(i) = BB(i) + rand() * delta;
B(i) = ceil(B(i));
end
for i = n-mm+1:n
B(i) = BB(i) - delta + rand() * Delta;
B(i) = ceil(B(i));
end
disp('новая реализация вектора B:');
disp(B);

X1 = XX(1:mm);
B = B - (A1*X1)';
```

```

disp('остаток ресурсов:');
disp(B);

X_1 = zeros(1,3);
for i = n-mm+1:n
    if B(i) < 0
        X_1(i-(n-mm))=abs(B(i));
        B(i)=0;
    end
end

[YY,Z1]=simplex(A,B,C,mm);
disp('значение вектора y:');
disp(YY(1:mm));

Y1=YY(1:mm);
expenses = (X1+Y1)*SE';
revenue = (X1+Y1-X_1)*C1';
profit = revenue - expenses;

disp('Затраты на производство = ');
disp(expenses);
disp('Доход = ');
disp(revenue);
disp('Прибыль= ');
disp(profit);

```

## Функция симплекс-метод

```
function [XX,Z] = simplex(A,B,C,mm)

n = length(B);
m = length(C);
kk=zeros(1,n);

for p = 1:n
    kk(p) = mm+p;
end

Z = 0;
schet = 0;
schet1 = 0;
mins = 0;
minr = 10000000;
r = 0;
s = 0;

for j = 1:100
for i = 1:m
if C(i) < 0
schet = schet + 1;
end
end

if schet == 0
break;
end
schet = 0;

for i = 1:m
if C(i) < mins
mins = C(i);
s = i;
end
end
mins = 0;

for i = 1:n
if A(i, s) > 0
schet1=schet1+1;
end
end

if schet1 == 0
disp('решение не ограничено');
break;
```

```

end
schet1 = 0;
for i = 1:n
if A(i, s) > 0
if (B(i) / A(i, s)) < minr
minr = B(i) / A(i, s);
r = i;
end
end
end

kk(r) = s;
L = A(r, s);
minr = 10000000;

for i=1:n
for jj = 1:m
if (i ~= r) && (jj ~= s)
A(i, jj) = A(i, jj) - A(r, jj) * A(i, s) / L;
end
end
if i ~= r
    B(i) = B(i) - B(r) * A(i, s) / L;
end
end

for jj = 1:m
if jj ~= s
C(jj) = C(jj) - A(r, jj) * C(s) / L;
end
end

for i = 1:n
A(i, s) = 0;
end
for jj = 1:m
A(r, jj) = A(r, jj) / L;
end

Z = Z - B(r) * C(s) / L;
A(r, s) = 1;
B(r) = B(r) / L;
C(s) = 0;
end
XX = zeros(1,m);

for p = 1:n
XX(kk(p)) = B(p);
end

```