МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»
Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.04.02 Прикладная математика и информатика
(код и наименование направления подготовки)

Математическое моделирование
(направленность (профиль))

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему: Математическое моделирование задачи максимизации прибыли предприятия на основе стохастического программирования.

Студент А. Базаева
(И.О. Фамилия) (личная подпись)

Научный руководитель к.ф-м. н., доцент Г.А. Тырыгина
(ученая степень, звание, И.О. Фамилия)

Тольятти 2020

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 3
Глава 1 ПОСТАНОВКА ЗАДАЧИ СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ 5
1.1 Проблема выбора решения в условиях риска и неопределенности 6
1.2 Одноэтапные задачи стохастического программирования 6
1.3. Двухэтапные задачи стохастического программирования 21
Глава 2 СПЕЦИАЛЬНЫЕ МЕТОДЫ В ЗАДАЧАХ СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ 43
2.1 Метод проектирования стохастических квазиградиентов 43
2.2 Применение метода стохастических квазиградиентов к задачам стохастического программирования 45
2.3 Метод стохастической декомпозиции 50
2.4 Метод возмущений решения задач стохастического программирования 59
Глава 3 ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ МАКСИМИЗАЦИИ СРЕДНЕЙ ПРИБЫЛИ ПРОИЗВОДСТВА 59
3.1 Анализ предметной области 66
3.2 Математическая модель задачи максимизации прибыли производства 69
Глава 4 РЕАЛИЗАЦИЯ АЛГОРИТМА СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ 75
4.1 Выбор программного обеспечения 75
4.2 Обзор и обоснование выбора среды разработки Matlab 77
4.3 Реализация алгоритма 78
ЗАКЛЮЧЕНИЕ 86
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 87

ВВЕДЕНИЕ

На данный момент наиболее изученной областью математического программирования является задача линейного программирования. Линейное программирование широко используется для решения задач в таких областях как экономика, индустрия, сельское хозяйство, военное дело, а также в социальных науках. Однако, как показывает практика большинство оптимизационных задач содержат случайные или неопределенные исходные параметры. Примером может служить выпуск нового товара на некотором производстве, в силу того, что неизвестен спрос, параметры задаются случайно. Итак, под влиянием некоторых причин, по которым нет возможности определить значение параметров исследуемой проблемы, мы эту проблему относим к проблемам стохастического характера, а модели и методы, которые применимы в случае решения задач со случайными и неопределенными факторами, называются моделями и методами стохастического программирования. Стохастическим программированием называется подход в математическом программировании, позволяющий учитывать неопределенность в оптимизационных моделях. Отметим основные особенности стохастического программирования:

1. Область применения. Используется в задачах, где все или отдельные параметры являются случайными величинами.

2. Сущность. Решение задачи заключается в оптимизации некоторой вторичной функции, которая представляет собой стохастическую характеристику исходной функции. Исходя из математической модели (аналитической, вероятностной или статистической) выбираются стохастические характеристики такие как, математическое ожидание, дисперсия, вероятности либо их оценки. Когда же стохастические характеристики неслучайны задача сводится к детерминированной.

3. Применение. Когда известно выражение для вторичной функции стохастическую задачу можно преобразовать в ее детерминированный

эквивалент. Иначе, необходимо решать задачу оптимизации численными методами, для чего потребуется вычислять функцию в различных точках при N реализациях, причем число N должно обеспечить достаточную точность и надежность получаемых характеристик.

Этим определяется актуальность работы, в которой объектом исследования является математическая модель максимизации прибыли предприятия, предметом исследования –математическая модель максимизации прибыли предприятия на основе стохастического программирования.

Цель: Построение и реализация математической модели максимизации прибыли на основе стохастического программирования.

Задачами являются:

- Проведение анализа различные модели стохастического программирования;
- проведение анализа предметной области деятельности предприятия с точки зрения максимизации прибыли;
- построение математическое модели максимизации прибыли с использованием стохастического программирования;
- осуществление программной реализации математической модели максимизации прибыли предприятия;
- тестирование программы.

Новизна:

Разработана математическая модель максимизации прибыли предприятия на основе стохастического программирования:

1. Разработан программный комплекс, который реализует вычисления максимальной прибыли предприятия, путем перехода от системы уравнений со случайными переменными к его детерминированному эквиваленту.

2. Получены результаты и рекомендации для использования математической модели максимизации прибыли предприятия.

Глава 1 ПОСТАНОВКА ЗАДАЧИ СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

Джордж Данциг вдохновил целое сообщество исследователей на изучение «проблемы неопределенности», назвав его стохастическим программированием. В настоящее время эта область является одной из самой главной частью программирования, охватывающей исследования, начиная от статистических подходов и заканчивая надежной оптимизацией, мерами риска и, конечно, стохастическим программированием. Среди его многолетнего наследия Джордж Данциг сыграл важную роль в продвижении использования выборки в стохастическом программировании (Dantzig and Infanger 1991). Его интуитивное объяснение, что «не нужно знать рост каждого американца, чтобы оценить средний рост американцев», было призвано убедить сообщество математического программирования в том, что для разработки масштабируемых методов стохастического программирования необходимо отойти от детерминированных алгоритмов [2].

Стохастическое программирование - область математического программирования, рассматривающая оптимизацию при наличии неопределенности. За последние четыре десятилетия появилось огромное количество литературы. Исследователи в этой области никогда не переставали подчеркивать присущие им трудности в решении задач стохастического программирования, и не давая теоретического обоснования. Последние разработки в теории вычислительной сложности позволяют установить сложность большинства моделей стохастического программирования.

Итак, задачи стохастического программирования разделяют на два вида: задачи оперативного стохастического программирования, задачи перспективного стохастического программирования. Решение в задачах оперативного стохастического программирования принимается после эксперимента над состоянием природы ω, а решение в задачах перспективного стохастического программирования совершенно противоположен принятию решений оперативного стохастического программирования и следовательно решение детерминированно.

В большинстве случаев задачи стохастического программирования задаются следующим образом:

$$M_\omega\{f(X,\omega)\} = F(X) \to min \tag{1.1}$$

при

$$M_\omega\{g_i(X,\omega)\} = G_i(X) \le 0, i = \overline{1,m}, X \in Y, \tag{1.2}$$

где M_ω - операция математического ожидания;

$$P\{f(X,\omega) \ge a\} \to min \tag{1.3}$$

при следующих ограничениях

$$P\{g_i(X,\omega) \le 0\} \ge P_i, i = \overline{1,m}, \tag{1.4}$$

где a, P_i - некоторые числа; p- вероятность.

В постановках задач (1.1), (1.2) и (1.3), (1.4) существуют различия, но это не мешает свести их к некой общей постановке. Таким образом, вводим следующие характеристические функции [6].

$$h_0(X,\omega) = \begin{cases} 1, \text{если } f(X,\omega) \ge a; \\ 0, \text{в противном случае;} \end{cases}$$

$$h_0(X,\omega) = \begin{cases} 1, \text{если } g_i(X,\omega) \le 0; \\ 0, \text{в противном случае;} \end{cases}$$

для которых

$$M_\omega\{h_0(X,\omega)\} = P\{f(X,\omega) \ge a\}; M_w\{h_i(X,\omega)\} = P\{g_i(X,w) \le 0\}$$

И тогда задачи (1.3), (1.4) выглядят следующим образом:

$$M_\omega\{h_0(X,\omega)\} \to min$$

при ограничении

$$M_\omega\{h_i(X,\omega)\} \ge P_i, i = \overline{1,m}$$

1.1 Проблема выбора решения в условиях риска и неопределенности

Джордж Бернард Данциг (8 ноября 1914 г. - 13 мая 2005 г.) рассматривается многими как один из великих математиков двадцатого века и был иконой в области исследований операций и наук управления. Он был изобретателем симплекс-метода, алгоритма для решения задачи линейного программирования, для которого эффективная методология решения привела к его широкому применению в армии, правительстве и промышленности. Сегодня, несмотря на развитие современных внутренних методов, симплекс-метод остается быстрой и надежной рабочей лошадкой для решения больших задач линейного программирования на регулярной основе для многих приложений во множестве отраслей промышленности.

За его вклад Джордж Б. Данциг был удостоен значительной чести при жизни. В 1975 году он был награжден президентом Фордом Национальной медалью науки. Он был первым лауреатом премии им. Джона фон Неймана в 1975 году. Эта награда была учреждена Институтом исследований операций и наук управления, который ежегодно присуждается человеку, который внес фундаментальный и устойчивый вклад в теорию исследований операций и наук управления. Общество математического программирования и Общество промышленной и прикладной математики учредили в 1979 году премию Джорджа Б. Данцига, присуждаемую за оригинальные исследования, оказавшие значительное влияние на область математического программирования. Он был членом Национальной академии наук, Национальной академии наук инженерии. и Американская академия искусств и наук. Кроме того, он получил многочисленные почетные докторские степени от университетов по всему миру.

В широком спектре его работ, одна тема, казалось, сохранялась на протяжении всей его научной карьеры. Его стремление «автоматизировать (или механизировать) процесс планирования» выросло из задания в Пентагоне (где во время Второй мировой войны он отвечал за планирование военных поставок) и продолжалось в течение его лет в корпорации RAND (1952–1960).), в качестве профессора на кафедре промышленной инженерии в Калифорнийском университете в Беркли (1960–1966), где он основал и руководил Центром исследования операций, а также профессором CA Criley по транспортным наукам и профессором исследований операций и компьютеров Наука в Стэнфордском университете (1966–1996), где он основал и направил Лабораторию системной оптимизации в Департаменте исследований операций и занялся принудительным выходом на пенсию. Его «основной» интерес был к теории, решению и применению линейных временных шкал системы, где обработка неопределенности в данных становилась все более и более его любимым делом.

После того, как он предложил симплекс-метод в 1948 году, Джордж Данциг осознал, что «настоящая проблема» касается программирования в условиях неопределенности. В 1955 году он написал «Линейное программирование в условиях неопределенности» перепечатанный в этой книге в качестве первой главы, оригинальной статьи, в которой он ввел неопределенные параметры в качестве коэффициентов и правых частей задачи линейного программирования. В качестве примеров он изложил примеры использования ожидаемых значений для достаточных распределений (отметив, что «может быть нежелательно минимизировать ожидаемое значение затрат, если в решении слишком велико изменение фактических общих затрат»), и сослался на работу Гарри Марковица. анализ среднего отклонения, который позволяет инвестору пожертвовать частью своих ожиданий, чтобы контролировать свои риски), модель простого регресса, двухэтапная линейная программа с регрессом и многоступенчатая линейная программа с регрессом, теперь все это важно архетипы в области стохастического программирования.

Новый подход к решению крупномасштабных стохастических программ, который объединяет декомпозицию, выборку Монте-Карло и параллельные процессоры, был начат в 1989 году Данцигом и Глинном (1990) и Инфангером (1992). В этом подходе использовалась выборка значений для оценки ожидаемых затрат на втором этапе, а также коэффициентов и правых сторон сокращений в Бендерах. Успех этого подхода привел к тесному сотрудничеству в течение нескольких лет между

Dantzig и Infanger в области стохастического программирования и его приложений. Dantzig и Infanger (1992) приводят численные результаты для практического крупномасштабного планирования электроэнергетики и финансовых проблем, показывая, что значительное уменьшение дисперсии в оценке сокращений привело к небольшим доверительным интервалам вокруг оптимальной цели задач. Такие проблемы до сих пор считались неразрешимыми из-за их размера.

Задачи оптимизации являются детерминированными в случае, когда в задачах присутствует однозначная информация. Но в реальности, в практических задачах, точные параметры определить не предоставляется возможным. Все это объяснимо неточностью данных, на основе которой формируется модель.

Присутствие случайных переменных в задаче способствует переходу из задачи в которой требуется нахождение оптимального решения в задачу выбора решения в условии неопределенности. В некоторых случаях параметры носят вероятностный характер и представляют собой случайные величины, характеристики которых нам известны или в принципе могут быть получены. Таким образом, задачи в которых присутствуют случайные переменные, являются стохастическими.

«Неточные данные в задаче приводят к тому, чтобы прибегать к решению в условиях риска. Такие методы решения задач рассматриваются в стохастическом программировании.

С точки зрения исходных данных задачи стохастического программирования являются промежуточными между оптимизационными задачами, поставленными в условиях полной определенности и в условиях неопределенности. Построение модели в стохастическом программировании, которую предоставляется возможность решить, происходит преобразованием основной задачи в детерминированную при помощи постановок» [1].

Задачи в стохастическом программировании могут быть одношаговыми и многошаговыми. Одношаговыми задачами называются задачи, в которых решение принимается единожды, и в последующем не корректируется. В качестве целевой функции в таких задачах может быть:

- математическое ожидание;
- комбинированное сочетание математического ожидания и среднего квадратического отклонения.

Ограничения задачи в одних случаях могут выполняться при всех возможных значениях случайных параметров (жесткая постановка), а в других случаях требуется, чтобы вероятность попадания решения в допустимую область была не меньше заданной (модель с вероятностными ограничениями) [10].

В случае многошаговых задач, здесь решение может корректироваться неопределенное количество раз, единственное, нужно решить, что значит в конкретной задаче допустимым и оптимальным решением. Таким образом можно сделать вывод о том, что в практике лучше применять многошаговые задачи, так как предусмотрена корректировка решения.

1.2 Одноэтапные задачи стохастического программирования

Решения задачи одноэтапного стохастического программирования находят уже при известных стохастических характеристиках распределения случайных величин, т.е. до наблюдений за их реализациями. Постановка задачи осуществляется основными критериями такими как, характер решения, выбор показателей качества и способ декомпозиции ограничения задачи.

Переход со стохастической задачи на детерминированную осуществляется с помощью постановок задач.

В основном используются такие постановки как М-постановка (в некоторой литературе упоминается как Е-постановка), Р-постановка, а также V-постановка. Рассмотрим возможные постановки стохастических задач в зависимости от факторов со случайными элементами [5].

Постановка в которой целевая функция является математическое ожидание называется М-постановкой, она используется для нахождения значения x_j.

Рассмотрим случаи применения данной постановки.

1. Первый случай. Коэффициенты целевой функции являются случайными, а ограничения вероятностными.

Здесь происходит замена случайных величин их детерминированными эквивалентами, то есть средними значениями. «Такая замена осуществляется только в случае, когда случайные параметры немного отличаются от математических ожиданий, но, если влияние случайности на интересующий нас процесс существенно, то использование данного критерия является необоснованным. Оптимизация будет корректной, если процесс обладает свойством повторяемости и недостаток показателя эффективности в одних случаях компенсируется его избытком в других» [42].

Использование критерия среднего значения предполагает возможность многократного решения одной и той же задачи, пока не будут получены достаточно точные расчетные формулы. Математически это утверждение выражается следующим образом. «Пусть x – случайная величина с математическим ожиданием m и дисперсией σ 2. Если $(x1, x2, \ldots, xn)$ – случайная выборка объема n, то выборочное среднее имеет дисперсию σ 2 /n. Поэтому при $n \to \infty \sigma 2 x \to m / n \to 0$, и значит по вероятности» [27]. Другими словами, при достаточно большом объеме выборки разница между выборочным средним и математическим ожиданием стремится к нулю. Следовательно, использование критерия среднего значения допустимо лишь в случае, когда одно и то же решение приходится принимать большое число раз [15].

Напротив, если необходимость в принятии некоторого решения встречается редко, то среднее значение может значительно отличаться от математического ожидания, и ориентация на среднее значение будет приводить к неверным результатам для решений, которые приходится принимать небольшое число раз. Это означает целесообразность введения критерия, в котором учитывается не только среднее значение, но также разброс возможных значений относительно среднего, т. е. дисперсия или среднее квадратическое отклонение. Возможным критерием, отвечающим этой цели, является комбинация математического ожидания и среднего квадратического отклонения.

Математическая модель выглядит следующим образом:

$$M\{c^T x\} \to max \tag{1.5}$$

при

$$P\{\sum_{j=1}^{n} a_{ij}x_j \le b_i\} \ge a_i, i = \overline{1,m} \tag{1.6}$$

$$x_j \ge 0, j = \overline{1,n} \tag{1.7}$$

Детерминированный эквивалент для вероятностного ограничения (1.5)-(1.7) выглядит следующим образом.

$$c^{-T} x \to max \tag{1.8}$$

при ограничениях

$$\sum_{j=1}^{n} a_{ij}x_j \le \tilde{b}_i, i = \overline{1,m} \tag{1.9}$$

$$x_j \ge 0, \tag{1.10}$$

$$\bar{c} = M\{c\};$$

$\tilde{b}_i$ - корень уравнения $F(\tilde{b}_i) = 1 - a$, или $\tilde{b}_i = F^{-1}(1 - a_i), F_i(b_i)$, - функция распределения случайной величины b_i. Детерминированный эквивалент для вероятностного ограничения выглядит следующим образом:

$$\sum_{j=1}^{n} a_{ij}x_j \le B_i, i = \overline{m_1 + 1, m,}$$

где B_i является предельным значением случайных величин b_i, которое удовлетворяет условию

$$P[b_i \geq B_i] \geq \beta_i, i = \overline{m_1 + 1, m},$$

или эквивалентному ему $P[b_i \geq B_i] \geq 1 - \beta_i$, где B_i: $(1 - \beta_i)$ – квантиль.

а) Коэффициенты в системе ограничений случайны. $C_j, j = \overline{1, n}$ и $b_i, i = \overline{1, m}$ – детерминированы [8].

Постановка в которой целевая функция представляет собой наихудшее значение, и требует оптимизации называется Р-постановкой. Если же целевая функция стремится к максимуму, то тогда задается минимально допустимое значение F_{min} с обязательным условием: $F \geq F_{min}$. Однако, когда же функция стремится к минимуму происходит противоположный процесс задания значений, и необходимым условием является $F \leq F_{max}$.

Математическая модель стохастической задачи представлена в следующем виде:

$$P(f_{\max(\min)}) \to max,$$

$$P(\sum_{j=1}^{n} a_{ij}x_j \leq b_j) \geq \beta_i, i = \overline{1, m},$$

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

Целевая функция может быть представлена в виде $P(f) = P_{\text{зад}}$. $P_{\text{зад}}$ – заданное значение вероятности. Отметим то, что куда бы не стремилась целевая функция к минимуму или к максимуму, она всегда должна в итоге стремится к максимуму вероятности [13].

Необходимо знать, если целевая функция стремится к максимуму то наименьшее допустимое значение задается как F_{min}, а при минимизации F_{max}.

Детерминированный эквивалент задачи в Р-постановке:

$$M[f(x)] = \sum_{j=1}^{n} M[c_j]x_j,$$

$$\sigma[f(x)] = \sqrt{\sum_{j=1}^{n} \left(\sigma[c_j]x_j\right)^2},$$

или $\sigma[f(x)] = v[f(x)] \cdot M[f(x)], v[f(x)]$ – коэффициент вариабильности, который показывает относительную величину разброса случайных величин.

Таким образом, в Р-постановке находятся значения x_j, при которых максимизируется вероятность и тогда целевая функция будет не хуже предельно допустимого значения.

Задачи стохастического программирования в большинстве случаев решаются в один или в два этапа. И таким образом задачи стохастического программирования бывают одноэтапными, двухэтапными, и редко многоэтапными.

Если решение принимается на основе уже известных стохастических параметров до наблюдения за их реализациями, то такие задачи являются одноэтапными [19]. Так как задачи стохастического программирования со случайными переменными не предоставляет возможности решить в чистом виде, требуется перейти к ее детерминированному эквиваленту. Для того чтобы совершить такой переход, необходимо использовать постановки задач рассмотренные выше. Постановка задачи зависит от типа задач. Далее, после того как была выбрана постановка задачи, происходит переход к детерминированному эквиваленту стохастической задачи. После того, как произошел данный переход, задача решается симплекс-методом.

Рассмотрим модель применения задач стохастического программирования со случайными переменами в целевой функции.

«В процессе эксплуатации станка могут возникать неисправности, и станок останавливается для их устранения. Время устранения неисправности постоянно и равно T_{b1}. После ремонта станок вновь включается и продолжает свою работу. Предположим, что в течение времени t станок n(t) раз выходит из строя, и пусть эта величина имеет порядок $n(t) = at^{\alpha}$, где α – случайная величина с математическим ожиданием m = M(a) и средним квадратическим отклонением σ = σ(a). Показатель α > 1, так как с течением времени станок вырабатывает свой ресурс, и его надежность уменьшается (значение α = 1 означает, что отказы станка растут пропорционально времени и «старение» не происходит). Через определенное время проводится профилактика станка, продолжающаяся время T_{b2}, и он восстанавливает все свои характеристики надежности до первоначального состояния. Требуется определить оптимальное время между соседними профилактиками, при котором коэффициент простоя $К_п$ станка был бы минимальным. Под коэффициентом простоя будем понимать долю времени, в течение которого станок не работает» [12].

В течение периода длительности t станок простаивает время n(t)·T_{b1} +T_{b2}, и, следовательно, коэффициент простоя станка равен

$$К_п = \frac{n(t)T_{b1} + T_{b2}}{t} = \frac{aT_{b1}t^{\alpha} + T_{b2}}{t} \quad (1.11)$$

Задача минимизации функции (1.11) является задачей стохастического программирования, поскольку параметр α случаен. Для решения задачи применимы два критерия, позволяющие свести задачу к детерминированной: критерий среднего значения и комбинированный критерий среднего значения и среднего квадратического отклонения. Согласно первому критерию в качестве целевой функции берется математическое ожидание коэффициента простоя, т. е.

$$z_1 = M(К_п) = \frac{mT_{b1}t^{\alpha} + T_{b2}}{t} = mT_{b1}t^{\alpha-1} + \frac{T_{b2}}{t}$$

Оптимальное решение t_1^* находится из уравнения $(\alpha - 1)\mathrm{mT_{b1}}t^{\alpha-2} - \frac{T_{b2}}{t} = 0$ или $z_1' = 0$, откуда

$$t_1^* = \left(\frac{T_{b2}}{(\alpha - 1)\mathrm{mT_{b1}}}\right)^{\frac{1}{\alpha}}.$$

Согласно комбинированному критерию в качестве целевой функции берется$z_2 = \mathrm{M}(К_п) + \mathrm{C}\sigma(К_п)$, где С – заданная постоянная, определяющая «степень важности» среднего квадратического отклонения по отношению к математическому ожиданию. Так как $\sigma(К_п) = \sigma T_{b1} \cdot t^{\alpha-1}$, то

$$\mathrm{z}_2 = (m + C\sigma)T_{b1}t^{\alpha-1} + \frac{T_{b2}}{t}.$$

Как и в случае первого критерия находим оптимальное время между профилактиками

$$t_2^* = \left(\frac{T_{b2}}{(\mathrm{\alpha}-1)(\mathrm{m}+\mathrm{C\sigma})\mathrm{T_{b1}}}\right)^{\frac{1}{\mathrm{\alpha}}}.$$

Задачи стохастического программирования со случайными параметрами в системе ограничений

Преобразование задачи в детерминированную. Ограничимся изучением только задач линейной оптимизации с вероятностными ограничениями. Модель с вероятностными ограничениями состоит в определении неотрицательных значений $x_1, x_2, \dots, x_n$, для которых

$$z = \sum_{j=1}^{n} c_j x_j \to max \tag{1.12}$$

при ограничениях

$$P(\sum_{j=1}^{n} a_{ij} x_j \leq b_i) \geq a_i\ , \quad i = \overline{1,m} \tag{1.13}$$

«Название «вероятностное ограничение» обусловлено тем обстоятельством, что каждое ограничение $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$может выполняться не всегда, а с некоторой вероятностью, не меньшей α_i $(0 < \alpha_i < 1)$. Предполагается, что параметры a_{ij} и b_i являются случайными величинами с известными законами распределения. Для простоты будем считать, что эти параметры независимы и обладают нормальными распределениями с математическими ожиданиями $M(a_{ij})$, $\mathrm{M}(b_i)$ и дисперсиями $D(a_{ij})$, $D(b_i)$ соответственно» [22]. Ограничения (1.13) можно записать в виде

$$P(h_i \leq 0) \geq a_i, \quad i = \overline{1,m} \tag{1.14}$$

где функция h_i определяется равенством $h_i = \sum_{j=1}^{n} a_{ij} x_j - b_i$.

Известно, что сумма независимых нормально распределенных случайных величин также имеет нормальное распределение. Следовательно, случайная величина h_i имеет нормальное распределение с математическим ожиданием

$$M(h_i) = \sum_{j=1}^{n} M(a_{ij})x_j - M(b_i) \tag{1.15}$$

и дисперсией

$$D(h_i) = \sum_{j=1}^{n} D(a_{ij})x_j^2 + D(b_i) \tag{1.16}$$

Так как для случайной величины h_i, имеющей нормальное распределение, справедливо соотношение

$$P(h_i \leq 0) = \frac{1}{2} + \Phi_0\left(-\frac{M(h_i)}{\sqrt{D(h_i)}}\right),$$

Где Φ_0 – функция Лапласа, то ограничения (1.14) принимают вид

$$\Phi_0\left(-\frac{M(h_i)}{\sqrt{D(h_i)}}\right) \geq a_i - \frac{1}{2}, \quad i = \overline{1,m}$$

По таблицам функции Лапласа можно найти квантиль $\gamma_{a_i-\frac{1}{2}}$, соответствующий значению $a_i - \frac{1}{2}$, и тогда $-\frac{M(h_i)}{\sqrt{D(h_i)}}$. Используя (1.15) и (1.16), получим

$$\sum_{j=1}^{n} M(a_{ij})x_j + \gamma_{a_i-\frac{1}{2}}\sqrt{\sum_{j=1}^{n} D(a_{ij})x_j^2 + D(b_i)} \leq M(b_i), \quad i = \overline{1,m} \tag{1.17}$$

Тем самым линейные ограничения превратились в ограничения, содержащие квадраты неизвестных величин. Ограничения (1.17) могут быть .записаны в несколько иной форме, если ввести новые переменные

$$y_i = \sqrt{\sum_{j=1}^{n} D(a_{ij})x_j^2 + D(b_i)}\,, i = \overline{1,m}$$

Тогда при сделанных допущениях вероятностные ограничения (1.13) будут эквивалентны неравенствам:

$$\sum_{j=1}^{n} M(a_{ij})x_j + \gamma_{a_i - \frac{1}{2}} y_i \leq M(b_i)$$

и уравнениям

$$\sum_{j=1}^{n} D(a_{ij})x_j^2 + D(b_i) - y_i = 0,$$

где

$$y_i \geq 0, \ \ i = \overline{1, m}.$$

При формализации стохастической задачи можно привести в соответствие всей области определения целевой функции одно или несколько вероятностных ограничений [4]. Условия задачи могут быть следующими:

а) $P\{\sum a_{ij}x_j \leq b_i\} \geq \alpha_i, i = 1,2, \dots, m; 0 \leq \alpha \leq 1;$

б) $P\{Ax \leq B\} \geq \alpha, 0 \leq \alpha < 1;$

в) $P\{\sum_{j=1}^{n} a_{ikj}x_j Ю \geq b_{ik}, i_k \in I\} \geq \alpha_k, 0 \leq a_k \leq 1; \ \ k = \overline{1, K}$

Ниже представлены некоторые модели задач.

1) Задача с вероятностными ограничениями типа (а):

$$M\{c^T x\} \ \rightarrow max \tag{1.18}$$

$$P\{\sum_{j=1}^{n} a_{ij}x_j \leq b_i\} \geq a_i, i = \overline{1, m} \tag{1.19}$$

$$x_j \geq 0, j = \overline{1, n} \tag{1.20}$$

При детерминированной матрице $A = \|a_{ij}\|$ и случайном векторе $b = |b_i|$ задача (1.18-1.20) сводится к эквивалентной детерминированной задаче линейного программирования.

Пусть $P(b_1, b_2, \ldots, b_m)$ - совместная плотность распределения составляющих b_i случайного вектора b. Находим плотность распределения b_i:

$$P(b_i) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} p(b_1, b_2, .. b_m) db_1 db_2 \ldots db_{i-1} db_{i+1} \ldots db_m.$$

Находим $\tilde{b}_i$ из уравнения

$$\int_{b_i}^{\infty} P(b_i) db_i = \alpha_i, i = \overline{1, m} \qquad (1.21)$$

Если решение уравнения (1.21) является не единственным, то в качестве $\tilde{b}_i$ выберем наибольший корень.

Видно, что условия (1.19) эквивалентны неравенствам

$$\sum_{j=1}^{n} a_{ij} x_j \le \tilde{b}_i \ ,$$

где $\tilde{b}_i$ удовлетворяет соотношениям (1.21). Таким образом задача стохастического программирования (1.18)-(1.20) имеет следующий детерминированный эквивалент:

$$c^{-T} x \to max \qquad (1.22)$$

при условиях

$$\sum_{j=1}^{n} a_{ij} x_j \le \tilde{b}_i, i = \overline{1, m} \qquad (1.23)$$

$$x_j \ge 0 \qquad (1.24)$$

где $\bar{c} = M\{c\}$;

$\tilde{b}_i$ - корень уравнения $F(\tilde{b}_i) = 1 - a$, или $\tilde{b}_i = F^{-1}(1 - a_i), F_i(b_i),$ - функция распределения случайной величины b_i.

Двойственная задача для задачи (1.18)-(1.21) с детерминированной матрицей A выглядит таким образом:

рассмотрим задачу

$$\tilde{b}^T y \to min \qquad (1.25)$$

при условиях

$$P(A^T y \geq c\} \geq \beta \quad (1.26)$$

$$y \geq 0 \quad (1.27)$$

Решением задачи (1.25)-(1.27) является детерминированный вектор. Допустим, что $G_j(\xi)$ является функцией распределения случайного коэффициента G_j функции (1), т.е.$G_j(\xi) = P\{C_j \leq \xi\}$. При $G_j(\xi) = \beta_j$ запись $\xi = G_j^{-1}(\beta_j)$ эквивалентна записи $\xi = \min\{y|G_j(y) \geq \beta_j\}$. Итак, задача (1.18)-(1.20) будет представлена в следующем виде

$$\tilde{b}^T y \to min \quad (1.28)$$

при условии

$$A^T y \geq G^{-1}(\beta), y \geq 0. \quad (1.29)$$

Сравнивая задачу (1.28) - (1.29) с исходной (1.18)-(1.19), получаем, что при $\beta = G(\tilde{C})$ следующие две одноэтапные задачи представлены в виде двойственной пары:

$$G^{-1}(\beta)x \to max, P\{Ax \leq b\} \geq \alpha,\ x \geq 0;$$

$$F^{-1}(1-\alpha)y \to min,\ \ P\{A^T y \geq c\} \geq \beta,\ Y \geq 0.$$

Если задача стохастического программирования задана Р-постановкой, то нужно привести к минимуму порог k при следующем условии

$$P\{\sum_{j=1}^{n} c_j x_j \leq k\} = a_0\ .$$

Будем считать, что случайные коэффициентами $c_j, j = \overline{1,n}$, являются нормально распределенными с математическим ожиданием $\bar{c}_j$ и корреляционной матрицей $C = \|c_{ij}\|$, где $c_{tj} = M\{(c_t - \bar{c}_t)(c_j - \bar{c}_j)\}$. При таких допущениях форма $c^T x = \sum_{j=1}^{n} c_j x_j$ распределена с математическим ожиданием $\bar{c}^T x$ и дисперсией $\sum_{t=1}^{n}\sum_{j=1}^{n} c_{tj} x_t x_j$. Исходя из выше изложенного соотношение (19) будет выглядеть так:

$$\Phi\left(\frac{k - \sum_{j=1}^{n} \bar{c}_j x_j}{\sqrt{\sum_{t=1}^{n}\sum_{j=1}^{n} c_{tj} x_t x_j}}\right) = a_0\ .$$

Следовательно, можно увидеть , что k$\to min$ при следующем условии $P\{\sum_{j=1}^{n} c_j x_j \leq k\} = a_0$ будет равнозначным минимизации

$$k(x) = \sum_{j=1}^{n} \bar{c}_j x_j + \Phi^{-1}(a_0)\sqrt{\sum_{t=1}^{n}\sum_{j=1}^{n} c_{tj} x_t x_j}\ .$$

При $a \geq 0{,}5\ k$ представляет собой выпуклую вниз функцию по переменным x_j. Таким образом, при сделанных допущениях задаче стохастического программирования:

минимизировать k

при условиях

$$P\{c^T x \leq k\} = a_0\ ,$$

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

соответствует детерминированный эквивалент:

$$k(x) = \sum_{j=1}^{n} \overline{c_j} x_j + \Phi^{-1}(a_0)\sqrt{\sum_{t=1}^{n}\sum_{j=1}^{n} c_{tj} x_t x_j} \to min$$

при условии

$$\Phi^{-1}(a_i)(\sum_{t}\sum_{j} v_{itj} x_t x_j - 2\sum_{j} v_{ij} x_j + \theta_i^2)^{\frac{1}{2}} \leq \bar{b}_i - \sum_{j=1}^{n} \overline{a_{ij}} x_j, \qquad i = \overline{1,m}$$

Задача (1.25), (1.27) представляет собой задачу выпуклого программирования [18].

1.3. Двухэтапные задачи стохастического программирования

Двухэтапные задачи - это такие задачи, в которых необходимо принять решение в условиях неопределенности и риска, когда не все условия проведения операции известны, тем самым заставляя процесс управления этой операцией разбивать на два этапа. На первом этапе определяется предварительный план, который позволяет принять оперативные и оперативные работы. На втором этапе после наблюдения значений ранее неизвестных параметров операций производится корректирующий план. Этот план корректирует предварительное решение. Данная корректировка снижает ожидаемую эффективность экономической операции, проводимой по одноэтапному плану. Это является платой за отсутствие необходимой информации. Оба планы должны быть согласованы, то есть выбор предварительного плана должен гарантировать существование корректирующего плана, а также снижение эффективности операции, связанное с двухэтапным управлением, должно быть минимальным [22].

Давно признано, что традиционные детерминированные модели не отражают истинное динамическое поведение реальных приложений, потому что они не учитывают неопределенность. Поскольку Данциг (1955) и Бил (1955) независимо друг от друга предложили стохастическую формулировку модели, эти модели интенсивно изучались. В литературе был предложен ряд различных алгоритмических подходов, которые мы можем широко классифицировать как детерминированные методы, аппроксимационные схемы, алгоритмы на основе выборки и другие.

Детерминированные методы пытаются решить детерминированную эквивалентную проблему, либо напрямую, либо используя структуру. Видными среди них являются L-образный метод Ван Слайка и Уэтса (1969), его многорезанный вариант Биржа и Луво (1988), прогрессивный алгоритм хеджирования Рокафеллара и Уетса (1989), использование методов внутренних точек Люстиг и соавт. (1991), метод регуляризованной декомпозиции Ruszzzynski (1986) и другие крупномасштабные методы, реализованные на последовательных и параллельных компьютерах. Очевидно, что даже самые сложные детерминированные методы могут решить проблемы только с ограниченным числом сценариев. До настоящего времени проблемы с несколькими миллионами сценариев были решены с использованием детерминированных методов.

Аппроксимационные схемы вычисляют детерминированные нижние и верхние оценки для оптимальной цели задачи через неравенства Дженсена (1906) (нижняя граница) и Эдмундсона (1956) и Маданского (1959) (верхняя граница) и последовательно улучшают эти оценки. Уточнения этих границ были предложены многими авторами, например, Kall (1974), Huang et al. (1977), Frauendorfer and Kall (1988), Frauendorfer (1988, 1992), BirgeandWallace (1988), BirgeandWets (1987, 1989), Prékopa (1990) и др. Методы аппроксимации работают очень хорошо для проблем с небольшим числом стохастических параметров, но, похоже, встречаются трудности, когда число стохастических параметров велико.

Алгоритмы, основанные на выборке, могут быть далее разделены на методы, которые предварительно выбирают несколько сценариев, чтобы создать поддающуюся обработке детерминистическую эквивалентную проблему, которая затем решается подходящим детерминированным методом, и методы, которые используют выборку в алгоритме. Для первой категории доверительные интервалы по оптимальному объективному классу процедур на основе выборки на основе «приближения выборки по среднему значению» были предложены Shapiro и Homem-de-Mello (1998) и Mak et al. (1999), которые асимптотически верны в качестве размера выборки, используемой для генерации приближенных задач, приближающихся к бесконечности. В последнюю категорию попадают стохастические квазиградиентные методы (Ермолиев (1983) и Гайворонский (1988)), которые выбирают последовательно случайные направления поиска на основе ограниченного числа наблюдений случайной функции в каждой итерации. Другие основаны на модификациях методов детерминированной декомпозиции, чтобы учесть выборку. Метод стохастического разложения Higle and Sen (1991) основан на том, что на одну итерацию берется только одно наблюдение или очень небольшое количество наблюдений, в то время как Pereira et al. (1989) использовали контрольные переменные в качестве метода уменьшения дисперсии в выборке Монте-Карло в модифицированной структуре разложения Бендерса.

Выборка представляется наилучшим методом для практических задач с большим количеством стохастических параметров. Подход Dantzig and Glynn (1990) и Infanger (1992) объединяет разложение Бендерса и выборку важности Монте-Карло для решения стохастических линейных программ. Выборка важности служит методом уменьшения дисперсии и на практике часто приводит к получению точных оценок только с небольшими размерами выборки. Infanger (1992) и Dantzig and Infanger (1992) сообщают о решении на персональных компьютерах крупных проблем, которые в прошлом казались неразрешимыми даже на крупных мэйнфреймах.

В этой главе мы представляем теорию для получения вероятностной нижней границы для истинного оптимального объективного значения при использовании разложения Бендерса (1962) и выборки Монте-Карло для оценки коэффициентов и правых частей разрезов для решения двухэтапных стохастических линейных программ.

Рассмотрим следующую двухэтапную стохастическую линейную программу:

$$\min z = cx + E^w f^w y^w$$
$$\frac{s}{t} Ax = b$$
$$-B^w x + D^w y^w = d^W, \qquad w \in \Omega$$
$$x, y^w \geq 0$$

Матрица коэффициентов ограничения A, правый вектор b и коэффициенты целевой функции с первого этапа предполагаются известными с уверенностью. На втором этапе матрица B перехода, матрица D технологии (или регресса), правосторонний вектор d и коэффициенты объективной функции f точно неизвестны - предполагается, что известно только их совместное распределение значений.

Обозначим конкретные результаты через $B = B^w, D = D^w, d = d^w, f = f^w, w = 1, \dots, W$ и известную вероятность реализации ω через $p\,(\omega)$. Множество всевозможных реализаций ω обозначается через $\Omega = \{1, \dots, W\}$ или $\omega \in \Omega$. Поскольку во многих практических приложениях D и f не зависят от ω, мы опускаем их ω верхний индекс в остальной части главы упрощают представление; анализ будет таким же, если D и f заменить на D^ω и f^ω.

Эквивалентная детерминированная форма:

$$minz = cx + \theta$$
$$-1\colon -\theta + p^1 f y^1 + \dots + p^w f y^w + \dots + p^W f y^W = 0$$

$$
\begin{array}{llll}
p\colon Ax & & & = b \\
p^1\pi^1\colon -B^1x + Dy^1 & & & = d^1 \\
\quad\vdots & \ddots & & \vdots \\
p^w\pi^w\colon -B^wx + & Dy^w & & = d^w \\
\quad\vdots & \ddots & & \vdots \\
p^W\pi^W\colon -B^Wx & & +Dy^W & = d^W \\
x, y^1, \dots, y^w, \dots, y^W \geq 0, & & &
\end{array}
$$

где θ обозначает ожидаемые затраты второго этапа.

Двойственные переменные, соответствующие основным ограничениям, отображаются в столбце слева от уравнений. В частности, ρ - вектор двойственных переменных, связанных с ограничениями первого этапа $Ax = b$, а $p^\omega\pi^\omega$ - вектор двойственных переменных, связанных с ограничениями второго этапа $-B^\omega x + Dy^\omega = d^\omega$, для каждого $\omega \in \Omega$.

В практических применениях число возможных реализаций W второй стадии может быть очень большим (например, 1020 или даже 10100), и невозможно явно выразить систему. Различные реализации неявно генерируются по мере необходимости посредством комбинации небольшого набора из h независимых случайных параметров $\varphi_1, \varphi_2, \dots, \varphi_h$, где h может быть 100 или около того. Отбор проб необходим для решения таких проблем. Структура φ, лежащая в основе набора, позволяет использовать методы уменьшения дисперсии, такие как выборка важности, для уменьшения размера требуемой выборки. Поскольку теория оценки нижней границы такая же, как и при уменьшении дисперсии, мы опускаем обсуждение последней, чтобы упростить представление.

Стохастический алгоритм следует тем же шагам, что и исходный алгоритм Бендерса, за исключением того, что необходимые условия (истинные срезы) аппроксимируются псевдосрезами, полученными суммированием по случайной выборке из ω вместо всех ω. После назначения заданного числа повторений алгоритм K определяет окончательное решение первой ступени $x = \xi 1$, которое дает ожидаемую ожидаемую первую ступень.

Приведем к рассмотрению задачу, где требуется

$$\text{минимизировать } c^T x \tag{1.30}$$

при условиях

$$Ax = B \tag{1.31}$$

$$A^1 x = B^1 \tag{1.32}$$

$$x \geq 0 \tag{1.33}$$

где $C^t = [c_j], j = \overline{1,n}$;

$A = \|a_{ij}\|, i = \overline{1,m}, j = \overline{1,n}$;

$A^1 = \|a_{kj}^1\|, k =$1,m.

Решение задачи (1.31)-(1.33) представим следующим образом. Так как, задача должна решаться в два этапа, на первом этапе выберем вектор x, удовлетворяющий условиям (1.32), (1.33). Затем зафиксируем реализацию случайного события и значения $A(\overline{\omega}), B(\overline{\omega})$ которые ему соответствует. Оценивается невязка $B(\overline{\omega}) - A(\omega)$ в условиях (1.31) и вычисляется вектор $y \geq 0$, который производит компенсацию невязки в соответствии с соотношением

$$Dy = B(\overline{\omega}) - A(\omega)x, ,$$

где соответственно $D = \|d_{il}\|, i = \overline{1,m}, l = 1, n$ является матрицей компенсации, элементы матрицы случайны.

Если задачу (1.30)-(1.33) рассматривать в качестве планирования производства и A - матрица основных технологических способов производства, то тогда D будет считаться матрицей аварийных технологических способов, которая компенсирует обнаруженные невязки. Если же нарушаются условия задачи, то тогда налагается штраф, который зависит от величины составляющих вектора y, компенсирующего невязку. Штраф характеризуется величиной $q^T y$, в общем случае q - случайный n_1-мерный вектор, $q \geq 0$. Существуют и другие способы выбора вектора y.

Выбор вектора **y** происходит таким образом, для того чтобы был обеспечен минимальный ожидаемый штраф за компенсацию условий задачи, определяемых предварительным планом x. Другими словами, на втором этапе решается задача, где требуется

минимизировать $q^T y$

при условиях

$$Dy = B - Ax, \tag{1.34}$$

$$y \geq 0 \tag{1.35}$$

Эти два этапа могут быть сведены в один:

$$\min_x M_\omega \{c^T(\omega)x + \min_y [q^T(\omega)y]\} \tag{1.36}$$

при условиях

$$A^1 x = B^1 \tag{1.37}$$

$$D(\omega)y = B(\omega) - A(\omega)x \tag{1.38}$$

$$x \geq 0, y \geq 0 \tag{1.39}$$

Сложность решения двухэтапных задач при случайных моделях по параметрам. Обсуждаются некоторые первые шаги в изучении сложности решения многоэтапных задач. В частности, мы показываем, что при некоторой конкретной случайной задачи неопределенные параметры многоступенчатых задач решения являются PSPACE-трудными [20].

Затраты на план-компенсацию должны быть минимальными. Поэтому целесообразно для перспективного планирования (двухэтапной схемы) использовать малочувствительные к вариации параметров условия задачи.
Рассмотрим область определения планов первого этапа. Изучим предварительные планы x двухэтапной задачи.

Допустим x это вектор, который удовлетворяет ограничениям задачи (8)-(10). В общем случае, когда матрица компенсации B задана произвольным образом, нет гарантии, что не существует такой реализации $\widetilde{\omega}$, при которой система $D(\widetilde{\omega})y = B(\omega) - A(\omega)x,\ y \geq 0$,не будет иметь решения. В этом случае невозможно будет удалить появившуюся невязку с помощью вектора компенсации y. Для того, чтобы вектор x был допустимым планом двухэтапной задачи, необходимо, чтобы для всех $\omega \in \Omega$ существовал вектор y такой, что $D(\omega)y = B(\omega) - A(\omega)x$.

Такие неявно заданные ограничения на выбор вектора y называют индуцированными. Условия (1.37), (1.39) называются фиксированными.

Обозначим множество векторов x, определяемое фиксированными ограничениями, через R_1, а индуцированными - через R_2, т.е.

$$R_1 = \{x : A^1 x = B^1; x \geq 0\};$$

$$R_2 = \{x : \forall(\omega \in \Omega) \exists y \geq 0, D(\omega)y = B(\omega) - A(\omega)x\}..$$

Тогда область определения вектора X задается множеством $R = R_1 \cap R_2$. Рассмотрим случай, когда матрица B детерминированная. Установим достаточные условия для того, что $R_2 = E^n$, где E^n - n-мерное эвклидово пространство, т.е. достаточные условия того, чтобы множество планов задачи второго этапа было не пусто [4].

В качестве начала многих методов решения задач стохастического программирования формулируется так называемая детерминированная эквивалентная задача. Основное предположение для постановки этой задачи состоит в том, что реализации случайных параметров задаются в виде вариантов. Каждый вариант содержит полное описание (q, T, h), так как значения этих объектов имеют в одной реализации.

Результатом детерминированного эквивалента является предварительный план X. По составляющим оптимального предварительного плана X и реализации параметров условий задачи ω строится задача второго этапа - задача ЛП, решение которой определяет необходимую компенсацию плана.

Детерминированный эквивалент:

$$\min_{x \epsilon R} G(x) .$$

Рассмотрим задачу второго этапа (1.38) - (1.39) и двойственную к ней задачу для каждого A, B, x .

Допустим, что задача второго этапа и двойственная к ней разрешимы. По теореме двойственности для задач линейного программирования имеем

$$P(x, A, B) = Q(x, A, B) = z^{*T}(A, x, B)(B - Ax),$$

где соответственно $z^{*T}(A, x, B)$ оптимальное решение задачи (1.38), зависящее от параметров A, B, x .

Учитывая эти обозначения, двухэтапную задачу можно переписать так:

$$minG(x) = \min\{\overline{C^T}x + MQ(x, A, B) \qquad (1.40)$$

или

$$\min\{\overline{c^T}x + M(z^{*T}(B - Ax)\}\} .$$

Итак, рассмотрим двухэтапную задачу, в которой случайным является только вектор ограничений $b(\omega)$, а матрица компенсации D представлена в виде $D = [E, -E]$, где E является единичной матрицей.

Векторы y и q разбиваются на две части: $[y_1, y_2]$ и $[q_1, q_2]$, соответствующие матрицам E и -E. Тогда задача (3.7)-(3.10) будет иметь вид

$$Q(x) = c^T x + MP(x, b) \to min \qquad (1.41)$$

при условиях

$$A^1 x = B^1 \qquad (1.42)$$

$$x \geq 0 \qquad (1.43)$$

где

$$P(x, b) = \min(q_1^T y_1 + q_2^T y_2), \qquad (1.44)$$

при условиях

$$y_1 - y_2 = b(\omega) - Ax \qquad (1.45)$$

$$y_1 \geq 0, y_2 \geq 0 \qquad (1.46)$$

Здесь y_1, y_2, q_1, q_2 - n-мерные векторы.

Задача (1.46) второго этапа имеет планы при произвольной реализации ω и при любом предварительном плане x, т.е.$R_2 = R^n$ и $R = R_1 = \{x: A^1 x = b^1, x \geq 0\}$.

Необходимое и достаточное условие существования конечного решения задачи второго этапа в данном случае приобретает следующий вид. Как известно, в общем случае это условие записывается в форме
$D^T x \leq q$.

В рассматриваемом случае $\{z: [E, -E]^T z \leq q\} = \{z: -q_2 \leq z \leq q_1\}$. Так как обычно $q_1 \geq 0, q_2 \geq 0$, то это условие всегда выполняется.
Задача, двойственная к (1.44)-(1.46), записывается так:

$$Q(x, b) = max z^T (b - Ax) \tag{1.47}$$

при условии

$$-q_2 \leq z \leq q_1 \tag{1.48}$$

Задача (1.47) легко решается и ее решение имеет вид

$$Q(x, b) = \sum_{i=1}^{m} Q_i(x, b_i), \tag{1.49}$$

где

$$Q(x, b_i) = \begin{cases} (b_i - \sum_{j=1}^{n} a_{ij} x_j) q_{1i}, \text{если } b_i - \sum_{j=1}^{n} a_{ij} x_j \geq 0 \\ -(b_i - \sum_{j=1}^{n} a_{ij} x_j) q_{2i}, \quad \text{в противном случае.} \end{cases} \tag{1.50}$$

Поэтому эквивалентная выпуклая задача для двухэтапной стохастической задачи (1.36)-(1.39) записывается следующим образом:

$$(\mathrm{c}^T x + M(\sum_{i=1}^{m} Q_i(x, b_i))) \rightarrow min \tag{1.51}$$

при условиях

$$A^1 x = B^1, \tag{1.52}$$

$$x \geq 0, \tag{1.53}$$

где $Q(x, b_i)$ определяются из соотношений (1.50).

Приведем еще одну форму записи этой же задачи. Введем обозначения

$$Ax = W, \tag{1.54}$$

$$(Ax)_i = \sum_{j=1}^{n} a_{ij}x_j = W_i, \tag{1.55}$$

$$MQ(x,b) = M(\sum_{i=1}^{m} Q_i(x,b_i)) = M\sum_{i=1}^{m} Q_i(W_i,b_i) = \sum_{i=1}^{m} Q_i(W_i), \tag{1.56}$$

Заметим, что функции $Q_i(W_i)$ так же, как и $Q_i(W_i, b_i)$, выпуклы по W_i. Двухэтапная стохастическая задача в этой простейшей постановке сводится, таким образом, к задаче выпуклого программирования:

$$c^T x + \sum_{l}^{m} Q_i(W_i) \to min \tag{1.57}$$

при условиях

$$A^1 x = B^1, \tag{1.58}$$

$$Ax = W, \tag{1.59}$$

$$x \geq 0 \tag{1.60}$$

Исследуем функции $Q_i(W_i)$. Для этого обозначим через δ_i, γ_i соответственно точные верхнюю и нижнюю грани случайной величины $b_i(\omega)$, через $\overline{\bar{b} = [\overline{b_1}, \overline{b_2}, \dots, \overline{b_m}] = Mb}$ - математическое ожидание вектора ограничений, а через $F_i(b_i)$ - функцию распределения компонент $b_i(\omega)$. Разобьем множество значений ω_i на три области: $(-\infty; \gamma_i), [\gamma_i, \delta_i], (\delta_i, \infty)$.

Если $b_i(\omega)$ не имеет нижней грани (соответственно верхней), то положим $\gamma_i = -\infty (\delta_i = \infty)$. Вычислим значения $Q_i(W_i)$ в каждой из этих областей.

Получим

$$Q_i(W_i) = M\{Q_i(W_i, b_i)\}$$

$$= -q_i^{(2)} \int\limits_{b_i \le W_i} (b_i - W_i) dF_i(b_i)$$

$$+ q_i^{(1)} \int\limits_{b_i > W_i} (b_i - W_i) dF_i(b_i)$$

$$= q_i^{(1)} \int\limits_{\Omega} (b_i - W_i) dF_i(b_i) - (q_i^{(1)} + q_i^{(2)}) \int\limits_{b_i \le W_i} (b_i - W_i) dF_i(b_i)$$

Отсюда

$$Q_i(W_i) = q_i^{(1)}\left(\overline{b_\imath} - W_i\right) - (q_i^{(1)} + q_2^{(2)}) \int\limits_{b_i \le W_i} (b_i - W_i) dF_i(b_i) \tag{1.61}$$

Рассмотрим три случая.

1. Пусть $W_i < \gamma_i$. Тогда $\{b_i : b_i \le W_i\} = 0$ и получим

$$Q_i(W_i) = q_i^{(1)}(\bar{b}_i - W_i)\ .$$

2. Пусть $W_i \in [\gamma_i, \delta_i]$. Тогда функция $Q_i(W_i)$ на этом отрезке выражается соотношением

$$Q_i(W_i) = q_i^{(1)}\left(\overline{b_\imath} - W_i\right) - (q_i^{(1)} + q_i^{(2)}) \int_i^{W_i} (b_i - W_i) dF_i(b_i)\ .$$

3. $W_i > \delta_i$. В этом случае множество $\{b_i : b_i \le W_i\}$ совпадает со всей областью изменения $b_i(\omega)$, т.е. Ω, и поэтому

$$\int\limits_{b_i \le W_i} (b_i(\omega) - W_i) dF_i(b_i) = \int\limits_{\Omega} (b_i(\omega) - W_i) dF_i(b_i) = \bar{b}_i - W_i$$

Исходя из выше изложенного, функция $Q_i(W_i)$ на интервале (δ_i, ∞) имеет вид $Q_i(W_i) = q_i^{(2)}\left(W_i - \bar{b}_i\right) = -q_i^{(2)}(\bar{b}_i - W_i)$.Функция $Q_i(W_i)$ непрерывна и выпукла на всей числовой оси.

При некоторых частных распределениях составляющих вектора $b(\omega)$ двухэтапная задача может быть сведена к стандартным задачам ЛП или выпуклого программирования [24].

Пример использования двухэтапной задачи.

В проблеме Фергюсона и во многих проблемах снабжения общие затраты могут быть разделены на две части: во-первых, затраты на распределение различных ресурсов по нескольким направлениям j, а во-вторых, затраты (или потерянные доходы), понесенные из-за сбоя общих сумм u1, u2, ..., un назначены для удовлетворения потребностей в различных пунктах назначения в неизвестных количествах d1, d2, ..., dn соответственно.

Специальные задачи двухэтапного программирования, которые мы рассматриваем, имеют следующую структуру. Для первого этапа

$$\sum_{j=1}^{n} x_{ij} = a_i \left(x_{ij} \geq 0\right),$$

$$\sum_{i=1}^{m} b_{ij} x_{ij} = u_j,$$

где x_{ij} представляет количество i-го ресурса, назначенного j-му пункту назначения, а b_{ij} представляет количество единиц спроса в пункте назначения j, которое может быть удовлетворено одной единицей ресурса i. Для второго этапа

$$d_j = u_j + v_j - s_j \ \ (j = 1,2, \dots, n),$$

где v_j - дефицит предложения, а s_j - избыток предложения. Предполагается, что функция общей стоимости имеет вид

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{j=1}^{n} a_j v_j,$$

то есть линейно зависит от выбора x_{ij} и недостатков x_j (которые зависят от заданий u_j и требований d_j. Нашей целью будет минимизировать общие ожидаемые затраты. Пусть $\varphi_j\ (u_j\ |d_j)$ будет минимальными затратами в пункте назначения, если предложение равно u_j, а спрос - d_j. Ясно, что

$$\varphi_j\ (u_j\ |d_j) = \begin{cases} a_j(d_j - u_j) \text{ если } d_j \geq u_j \\ 0, \text{если } d_j < u_j, \end{cases}$$

где α_j - коэффициент пропорциональности.

Теорема. Ожидаемое значение $\varphi_j\ (u_j\ |d_j)$, обозначаемое $\varphi_j\ (u_j)$, является выпуклой функцией u_j. Доказательство. Пусть $p\ (d_j)$ - плотность вероятности d_j, тогда

$$\varphi_j\ (u_j) = a_j \int_{x=u_j}^{\infty} (x - u_j)p(x)dx = a_j \int_{x=u_j}^{\infty} xp(x)dx - a_j u_j \int_{x=u_j}^{\infty} p(x)dx,$$

откуда дифференцируем $\varphi\ (u_j)$

$$\varphi_j'\ (u_j) = -a \int_{x=u_j}^{\infty} p(x)dx.$$

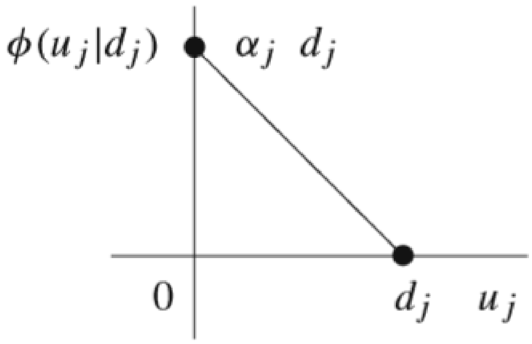
Ясно, что $\varphi_j'\ (u_j)$является неубывающей функцией u_j с $\varphi_j''(u_j) \geq 0$ и что $\varphi_j(u_j)$является выпуклым. Альтернативное доказательство (также из-за Скарфа) получается путем применения леммы, которую мы будем использовать позже.

Лемма. Если $\varphi(x_1, x_2, \dots, x_n|\theta)$- выпуклая функция над фиксированной областью Ω для любого значения θ, то любая положительная линейная комбинация таких функций также выпукла в Ω .

В частности, если θ - случайная величина с плотностью вероятности p (θ), то ожидаемое значение φ

$$\varphi(x_1, x_2, \dots, x_n) = \int_{-\infty}^{+\infty} \varphi(x_1, x_2, \dots, x_n)|\theta)p(\theta)d\theta$$

выпуклое. Например, из $\varphi_j\left(u_j\,|d_j\right)=\begin{cases}a_j\left(d_j-u_j\right) \text{ если } d_j \geq u_j \\ 0, \text{если } d_j<u_j,\end{cases}$ $\varphi_j\left(u_j\,|d_j\right)$, изображенный ниже, является выпуклым

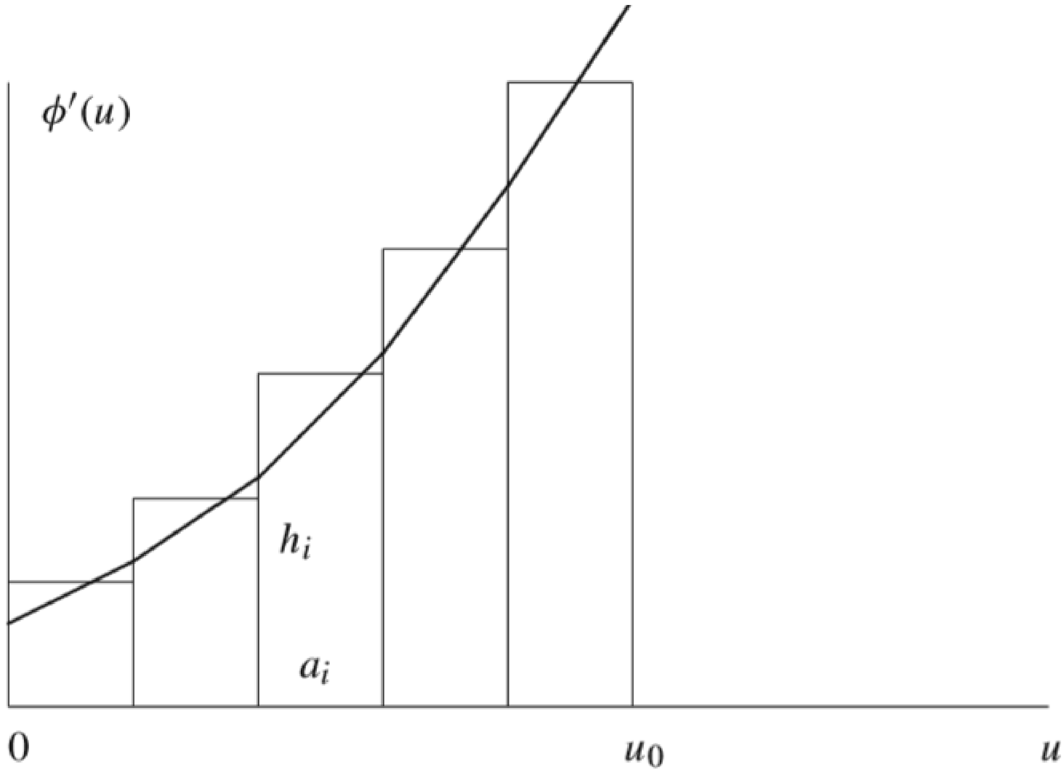


Из леммы легко следует, что $\varphi_j\left(u_j\right)$, выпукло. Из основной теоремы ожидаемое значение целевой функции

$$Exp\ C=\sum c_{ij}x_{ij}+\sum_{j=1}^{n}a_j\varphi_j\left(u_j\right),$$

где $\varphi_j\left(u_j\right)$ - выпуклые функции. Таким образом, исходная задача была сведена к минимизации.

Это позволяет применять хорошо известное устройство для аппроксимации такой задачи стандартной задачей линейного программирования в случае, когда целевая функция может быть представлена суммой выпуклых функций. Для этого аппроксимируем производную функции $\varphi(w)$в некотором достаточно большом диапазоне $0\leq u\ \leq u_0$ ступенчатой функцией



включает k шагов, где размер i-го основания a_i, а его высота h_i где $h_1 \leq h_2 \leq \cdots \leq h_k$, потому что φ выпуклый. Приближение для $\varphi\ (u)$ дается

$$\varphi\ (u) = \varphi(0) + Min \sum_{1}^{k} h_i \Delta_i$$

при условии

$$u = \sum_{i=1}^{k} \Delta_i, \quad 0 \leq \Delta_i \leq a_i.$$

В самом деле, это справедливо в отношении того, что приближение приводит к минимальному выбору $\Delta_1 = a_1, \Delta_2 = a_2, \ldots$ до тех пор, пока кумулятивная сумма Δ_i не превысит u для некоторого $\Delta_{r+i} = 0$. В качестве значения остатка выбирается r, а все остальные $r + i = 0$. Другими словами, мы аппроксимировали интеграл суммой прямоугольных областей под кривой до u, то есть

$$\varphi\ (u) = \varphi(0) + \int_{0}^{u} \varphi'(x)dx = \sum_{i=1}^{r} h_i a_i + h_r \Delta_r$$

Следующим шагом является замена $\varphi(u)$ на $\sum_1^k h_i\Delta_i$ и u на $\sum_1^k \Delta_i$в задаче программирования и добавление ограничений $0 \le i \le a_i$. Если целью является минимизация от общих затрат, по необходимости, при любом значении $u = \sum_1^n \Delta_i$ и $0 \le i \le a_i$, минимизируется $\sum_1^k h_i\Delta_i$. Таким образом, этот класс двухэтапных задач линейного программирования, связанных с неопределенностью, может быть сведен к стандартной задаче линейного программирования.

1.2.1. Некоторые частные постановки двухэтапной задачи

Решение в детерминированном эквиваленте двухэтапной задаче находят приближенно, т.к. найти точное решение весьма затруднительно. В ряде случаев задача упрощается и ее решение можно найти классическими методами линейного и выпуклого программирования. Перейдем к рассмотрению некоторых случаев.

Рассмотрим двухэтапную задачу, в которой случайным является только вектор ограничений $b(\omega)$, а матрица компенсации D после соответствующей перестановки строк и столбцов может быть представлена в виде $D = [E, -E]$, где E - единичная матрица размером .

Аналогичным образом разобьем векторы y и q на две части: $[y_1, y_2]$ и $[q_1, q_2]$, соответствующие матрицам E и -E. Тогда задача (1.19)-(1.22) будет иметь вид

$$Q(x) = c^T x + MP(x, b) \to min \qquad (1.62)$$

при ограничениях

$$A^1 x = B^1, \qquad (1.63)$$

$$x \ge 0 \qquad (1.64)$$

где

$$P(x, b) = \min(q_1^T y_1 + q_2^T y_2), \qquad (1.65)$$

при ограничениях

$$y_1 - y_2 = b(\omega) - Ax \qquad (1.66)$$

$$y_1 \ge 0, y_2 \ge 0 \qquad (1.67)$$

Здесь y_1, y_2, q_1, q_2 - n-мерные векторы. Очевидно, что задача (1.65) второго этапа имеет планы при произвольной реализации ω и при любом предварительном плане x, т.е.$R_2 = R^n$ и $R = R_1 = \{x: A^1x = b^1, x \geq 0\}$.

Необходимое и достаточное условие существования конечного решения задачи второго этапа в данном случае приобретает следующий простой вид. Как известно, в общем случае это условие записывается в форме

$D^T x \leq q..$

В рассматриваемом случае $\{z: [E, -E]^T z \leq q\} = \{z: -q_2 \leq z \leq q_1\}$. Так как обычно$q_1 \geq 0, q_2 \geq 0$, то это условие всегда выполняется .

Задача, двойственная к (1.65)-(1.67), записывается так:

$$Q(x, b) = max z^T(b - Ax) \tag{1.68}$$

при ограничении

$$-q_2 \leq z \leq q_1 \tag{1.69}$$

Задача (1.68) легко решается и ее решение имеет вид

$$Q(x, b) = \sum_{i=1}^{m} Q_i(x, b_i), \tag{1.70}$$

где

$$Q(x, b_i) = \begin{cases} (b_i - \sum_{j=1}^{n} a_{ij}x_j)q_{1i}, \text{если } b_i - \sum_{j=1}^{n} a_{ij}x_j \geq 0 \\ -(b_i - \sum_{j=1}^{n} a_{ij}x_j)q_{2i}, \quad \text{в противном случае.} \end{cases} \tag{1.71}$$

Поэтому эквивалентная выпуклая задача для двухэтапной стохастической задачи (1.65)-(1.67) записывается следующим образом:

$$(c^T x + M(\sum_{i=1}^{m} Q_i(x, b_i))) \to min \tag{1.72}$$

при условиях

$$A^1x = B^1, \tag{1.73}$$

$$x \geq 0, \tag{1.74}$$

где $Q(x, b_i)$ определяются из соотношений (1.71).

Приведем еще одну форму записи этой же задачи. Введем обозначения

$$Ax = W, \tag{1.74}$$

$$(Ax)_i = \sum_{j=1}^{n} a_{ij}x_j = W_i, \tag{1.75}$$

$$MQ(x,b) = M(\sum_{i=1}^{m} Q_i(x,b_i)) = M\sum_{i=1}^{m} Q_i(W_i,b_i) = \sum\nolimits_{i=1}^{m} Q_i(W_i) \tag{1.76}$$

Заметим, что функции $Q_i(W_i)$ так же, как и $Q_i(W_i, b_i)$, выпуклы по W_i. Двухэтапная стохастическая задача в этой простейшей постановке сводится, таким образом, к задаче выпуклого программирования:

$$c^T x + \sum_{l}^{m} Q_i(W_i) \to min \tag{1.77}$$

при условиях

$$A^1 x = B^1 \tag{1.78}$$

$$Ax = W, \tag{1.79}$$

$$x \geq 0 \tag{1.80}$$

Исследуем функции $Q_i(W_i)$. Для этого обозначим через δ_i, γ_i соответственно точные верхнюю и нижнюю грани случайной величины $b_i(\omega)$, через $\overline{\bar{b} = [\overline{b_1}, \overline{b_2}, \dots, \overline{b_m}] = Mb}$ - математическое ожидание вектора ограничений, а через $F_i(b_i)$ - функцию распределения компонент $b_i(\omega)$. Разобьем множество значений ω_i на три области: $(-\infty; \gamma_i), [\gamma_i, \delta_i], (\delta_i, \infty)$.

Если $b_i(\omega)$ не имеет нижней грани (соответственно верхней), то положим $\gamma_i = -\infty (\delta_i = \infty)$.

Вычислим значения $Q_i(W_i)$ в каждой из этих областей. Используя (1.75), получим

$$Q_i(W_i) = M\{Q_i(W_i, b_i)\}$$

$$= -q_i^{(2)} \int\limits_{b_i \le W_i} (b_i - W_i) dF_i(b_i)$$

$$+ q_i^{(1)} \int\limits_{b_i > W_i} (b_i - W_i) dF_i(b_i)$$

$$= q_i^{(1)} \int\limits_{\Omega} (b_i - W_i) dF_i(b_i) - (q_i^{(1)} + q_i^{(2)}) \int\limits_{b_i \le W_i} (b_i - W_i) dF_i(b_i)$$

Отсюда

$$Q_i(W_i) = q_i^{(1)}\left(\overline{b_\imath} - W_i\right) - (q_i^{(1)} + q_2^{(2)}) \int_{b_i \le W_i} (b_i - W_i) dF_i(b_i).$$

Рассмотрим три случая.

1. Пусть $W_i < \gamma_i$. Тогда $\{b_i : b_i \le W_i\} = 0$ и с (1.75) получим

$$Q_i(W_i) = q_i^{(1)}(\bar{b}_i - W_i) \,.$$

2. Пусть $W_i \in [\gamma_i, \delta_i]$. Тогда функция $Q_i(W_i)$ на этом отрезке выражается соотношением

$$Q_i(W_i) = q_i^{(1)}\left(\overline{b_\imath} - W_i\right) - (q_i^{(1)} + q_i^{(2)}) \int_i^{W_i} (b_i - W_i) dF_i(b_i) \,.$$

3. $W_i > \delta_i$. В этом случае множество $\{b_i : b_i \le W_i\}$ совпадает со всей областью изменения $b_i(\omega)$, т.е. Ω, и поэтому

$$\int\limits_{b_i \le W_i} (b_i(\omega) - W_i) dF_i(b_i) = \int\limits_{\Omega} (b_i(\omega) - W_i) dF_i(b_i) = \bar{b}_i - W_i$$

Следовательно, функция $Q_i(W_i)$ на интервале (δ_i, ∞) имеет вид

$$Q_i(W_i) = q_i^{(2)}\left(W_i - \bar{b}_i\right) = -q_i^{(2)}(\bar{b}_i - W_i) \,.$$

Как видим, функция $Q_i(W_i)$ непрерывна и выпукла на всей числовой оси.

При некоторых частных распределениях составляющих вектора $b(\omega)$ двухэтапная задача может быть сведена к стандартным задачам ЛП или выпуклого программирования [26].

Итак, рассмотрим задачу

Существует несколько предприятий, которые производят некоторый однородный продукт. Сами предприятия располагаются в пунктах $A_1, A_2, \dots, A_m$.

Объемы производства продукта обозначим $a_i, i = \overline{1,m}$.

Итак, введем некоторые обозначения:
c_i- производственные затраты в пункте A_i;
$c_{ij}(i = \overline{1,m}, j = \overline{1,n})$- транспортные затраты из пункта A_i в B_j;
b_j-величина спроса ;
$F_j(b_j)$ - равномерное распределение на отрезке $[\gamma_i, \delta_j]$;
s_j –затраты по хранению избыточного продукта в пункте B_j;
$q_j = \min(c_i + c_{ij})$- удельные затраты по производству и ввозу в пункт B_j;
Что нам необходимо найти? Необходимо минимизировать сумму ожидаемых затрат по производству и перевозке продукции, соответственно определив объемы перевозок x_{ij} из пункта A_i в пункт B_j, а также план выпуска продукции.

Р е ш е н и е. Составим математическую модель задачи. На первом этапе необходимо найти такой план перевозок $x = \|x_{ij}\|$, для которого

$$M\{c^T x\} = M_c\{\sum_i \sum_j c_{ij} x_{ij}\} \to min \tag{1.81}$$

при условиях

$$A^{(1)} x = \sum_{j=1}^{n} x_{ij} \leq a_i, i = \overline{1,m} \tag{1.82}$$

Обозначим объем производства продукта, ввезенного в пункт B_j, через W_j. Очевидно,

$$A^{(1)} x = \sum_{i=1}^{m} x_{ij} = W_j, j = \overline{1,n}\ .$$

Введем вектор компенсации $Y = \|y_i\|$, где $y_j = b_j - W_j, j = \overline{1,n}$ и матрицу компенсации $Q(x,b) = \sum_{j=1}^{n} Q\left(x, b_j\right) = \sum_{j=1}^{n} Q_j(W_j, b_j)$,

где

$$Q_j(W_j, b_j) = \begin{cases} [b_j - W_j]q_j, \text{если } b_j \geq W_j, \\ -[b_j - W_j]s_j, \text{если } b_j < W_j \end{cases}$$

Используя соотношения (1.77)-(1.80), запишем детерминированный эквивалент для данной задачи в следующем виде:

$$M\{\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} Q_j(W_j, b_j)\} \rightarrow min \quad (1.83)$$

при ограничениях

$$\sum_{j=1}^{n} x_{ij} \leq a_i, i = \overline{1, m} \quad (1.84)$$

$$\sum_{i=1}^{m} x_{ij} = W_j, j = \overline{1, n} \quad (1.85)$$

$$x_{ij} \geq 0 \quad (1.86)$$

Для того, чтобы привезти задачу к следующему виду, используем выражение (1.82)

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \overline{c_{\iota J}}x_{ij} + \sum_{j=1}^{n} q_j(\bar{b}_j - W_j) + \sum_{j=1}^{n} \frac{s_j + q_j}{2(\delta_j - \gamma_j)}(W_i - \gamma_j)^2 \rightarrow min \quad (1.87)$$

при условиях

$$\sum_{j=1}^{n} x_{ij} \leq a_i, i = \overline{1, m}; \quad (1.88)$$

$$\sum_{i=1}^{m} x_{ij} = W_j, j = \overline{1, n} \quad (1.89)$$

где

$$x_{ij} \geq 0 \quad (1.90)$$

Задача (1.87)-(1.90) является задачей квадратичного программирования с ограничениями-равенствами и ограничениями-неравенствами. Для ее решения, применим теорему Куна-Таккера.

Составим функцию Лагранжа для данной задачи

$$L(X,\Lambda) = \sum_{i=L}^{m} \bar{c}_{ij} x_{ij}$$

$$+ \sum_{j=L}^{n} q_j \left(b_j - \sum_i x_{ij} \right)$$

$$+ \sum_{j=L}^{n} \frac{s_j + q_j}{2(\delta_j - \gamma_j)} \times (\sum_{i=L}^{m} x_{ij} - \gamma_j)^2 + \sum_{i=L}^{m} \lambda_i (\sum x_{ij} - a_i)$$

Запишем необходимые и достаточные условия оптимальности:

$$\frac{\partial L(X,\Lambda)}{\partial x_{ij}} = \bar{c}_{ij} - q_i + \frac{s_j + q_j}{\delta_j - \gamma_j} (\textstyle\sum_{i=L}^{m} x_{ij} - \gamma_j) + \lambda_i \geq 0, i = \overline{1,m}, j = \overline{1,n};$$

$$\frac{\partial L(X,\Lambda)}{\partial \lambda_i} = \textstyle\sum_{j=1}^{n} x_{ij} - a_i \leq 0,\ i = \overline{1,m};$$

$$\frac{\partial L(X,\Lambda)}{\partial \lambda_{ij}} x_{ij} = \left[\delta_{ij} - q_j + \frac{s_j + q_j}{\delta_j - \gamma_j} (\sum_{i=L}^{m} x_{ij} - \gamma_j) + \lambda_i \right] \cdot x_{ij} = 0,$$

$$i = \overline{1,m}, j = \overline{1,n}$$

$$\frac{\partial L(X,\Lambda)}{\partial \lambda_i} \cdot \lambda_i = (\textstyle\sum_{j=1}^{n} x_{ij} - a_i) \cdot \lambda_i = 0, i = \overline{1,m}\ .$$

Для дальнейшего решения этой задачи применим стандартные методы линейного программирования или метод перебора вариантов [16].

Глава 2 СПЕЦИАЛЬНЫЕ МЕТОДЫ В ЗАДАЧАХ СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

2.1Метод проектирования стохастических квазиградиентов

Перейдем к рассмотрению класса оперативных задач, где требуется произвести минимизацию целевой функции $f(x,w)$.

«Точные значения функции $f(x)$ и ее производных $f_x(x) = \nabla f(x)$ вычислить не предоставляется возможным, но есть возможность многократного наблюдения за состоянием природы w, а также можно произвести оценку реализации целевой функции $f(x_1, w_1), f_2(x_2, w_2), \dots, f(x_s, w_s)$. Таким образом здесь можно воспользоваться прямыми методами стохастического программирования, к примеру, стохастический квазиградиентный метод, разработанный Ю. М. Ермольевым» [15].

Итак, необходимо

$$f(x,w) = f(x_1, x_2, \dots, x_n, w) \to min \quad (2.1)$$

при условии

$$x \epsilon R(x)$$

где $R(x)$ - выпуклое замкнутое множество;

$f(x)$ - выпуклая негладкая функция.

Существует некое вероятностное пространство с множеством элементарных событий w, а также последовательность случайных точек $x_s(w), s = 0,1, \dots$, которые можно определить по следующей формуле

$$x_{s+1} = \pi_x(x_s - p_s y_s \xi_s), \quad (2.2)$$

где π_x - оператор проектирования на множество $R(x)$;

x_0- произвольная начальная точка пространства;

p_s - величина шага; y - нормируюий множитель;

$\xi_s(w)$ - случайный вектор, условное математическое ожидание которого равно

$$M\{\xi_s | x_0, x_1, \dots, x_s\} = a_s f_x^n(x_s) + B_s, \quad (2.3)$$

где скаляр $a_s > 0$;

$B_s = [b_{1s}, b_{2s}, \dots, b_{ns}]$ - случайный вектор, не зависящий от последовательности $(x_0, x_1, \dots, x_s)$;

величины p_s, y_s измеримы по Борелю.

Обобщенный градиент функции $f(x_s)$ в точке x_s - вектор $f_s\hat{}(x_s)$ - удовлетворяет при любом векторе z неравенству

$$f(z) - f(x_s) \geq f_x^T(x_s)(z - x_s) \qquad (2.4)$$

Вектор ξ_s, удовлетворяющий (2.3), с точностью до параметров a_s, B_s совпадает в среднем с вектором обобщенного градиента. Если положить $a_s = 1, B_s = 0$ то ξ_s вполне естественно назвать стохастическим обобщенным градиентом (или квазиградиентом), а процедуру (2.2) - методом проектирования стохастических квазиградиентов.

«При анализе процедуры (2.3) возникает вопрос о сходимости последовательности $\{x_s\}$. Существуют различные варианты сходимости случайных последовательностей: по вероятности, в среднем квадратическом и с вероятностью 1» [3].

2.2 Применение метода стохастических квазиградиентов к задачам стохастического программирования

Рассмотрим задачу управления запасами.

Итак, есть склад некой вместимости a, в нем необходимо создать запас некого продукта (однородного). Спрос будет случайным с функцией распределения $H(z) = P\{w \leq z\}$., обозначим его w, а планируемый объем обозначим через x единиц.

Теперь, рассмотрим функцию затрат.

$$f(x, w) = \begin{cases} \alpha(x - w), \text{если } x \geq w - \text{затраты по хранению;} \\ \beta(w - x), \text{если } x < w - \text{штраф вследствие дефицита;} \end{cases} \qquad (2.5)$$

здесь α это удельные затраты по хранению единицы запасов;

β это удельный штраф вследствие дефицита. Нам необходимо определить уровень запасов x, при котором

$$F(x) = M\{f(x,w)\} = \alpha\int_0^x (x-z)dH(z) + \beta\int_x^\infty (z-x)dH(z) \to min \qquad (2.6)$$

Как нам известно функция $f(x,w)$ - недифференцируема при $x = w$, то в общем случае $F(x)$ - тоже недифференцируема. Очевидно,

$$F(x) = Mmax(\alpha(x-w), \beta(w-x)), \qquad (2.7)$$

то мы пришли к частному случай минимаксной задачи. Теперь определяем стохастический квазиградиент

$$\xi_s = f_x(x_s) = \begin{cases} \alpha, \text{если } x_s \geq w_s; \\ -\beta, \text{если } x_s < w_s \end{cases} \qquad (2.8)$$

и рекуррентный процесс определения оптимального объема запасов x^* и выглядит это так:

$$x_{s+1} = x_s - p_s\xi_s \text{ или } x_{s+1} = \pi_x(x_s - p_s\xi_s) \qquad (2.9)$$

2.3Двухэтапная задача стохастического программирования

Двухэтапная задача стохастического программирования:

$$F(x) = \min_{x,y} M_w\{c^T(w)x + q^T(w)y(x,w)\} \to min \qquad (2.10)$$

при условиях

$$A(w)x + B(w)y = D(w), \qquad (2.11)$$

$$x \in R(x), y \in R(y). \qquad (2.12)$$

Итак, $F(x)$ не имеет непрерывных производных. Предварительный план x и план-компенсация y при всех w удовлетворяет условию

$$g_i(x,y,w) \leq 0, i = \overline{1,m}, \qquad (2.13)$$

Все затраты, которые связаны с реализацией плана x и коррекцией этого плана пусть будут равны $f(x,y,w)$. Если мы обозначим план-компенсацию через $y(x,w)$ который минимизирует целевую функцию при ограничениях Обозначим через план-компенсацию, который минимизирует $f(x,y,w)$ при (2.12), (2.13) , то затраты на план и коррекцию плана будут выглядеть так

$$F(x) = M_\infty(f(x,y,w)\}, \qquad (2.14)$$

и необходимо найти такой x, для которого

$$F(x) \to min$$

при ограничениях (2.12), (2.13).

Будем предполагать, что имеется седловая точка $(x(w), y(w))$ функции Лагранжа, а также $g_i(x, y, w)$ и $f(x, y, w)$ при каждом w выпуклы вниз при всех w

$$\varphi(x, y, w) = f(x, y, w) + \sum_{i=1}^{m} \lambda_i g_i(x, y, w) \qquad (2.15)$$

при ограничениях $y \in R(y)$ и $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_m] \geq 0$. Допустим, что $f_x(x, y, w)$, $g_{ix}(x, y, w)$ являются обобщенными градиентаами функций $f(x, y, w)$, $g_i(x, y, w)$ по x при фиксированном w.

Тогда для решения задачи (2.10) методом СКГ при ограничениях (2.13), (2.12) на выпуклом и замкнутом множестве $R(x)$ следует положить

$$\xi_s = f_x(x_s, y\ (x_s, w_S), w_s) + \sum_{i=1}^{m} \lambda_i)(x_s, w_s) g_{ix}(x_s, w_s), w_s) \qquad (2.16)$$

$$x_{s+1} = x_s - p_s \xi_s \qquad (2.17)$$

или

$$x_{s+1} = \pi_x(x_s - p_s \xi_s) \qquad (2.18)$$

Можно показать , что для функции $F(x)$ и вектора ξ_s, задаваемого в (2.14), $M\big(\xi_s(x_s)\big) = F_x(x_s)$, то есть $M(\xi_s|x_S)$ - субградиент функции $F(x_s)$. Далее, найдем решение двухэтапной задачи стохастического программирования

$$F(x) = C^T x + M_w\{q^T(w) y(x, w)\} \to min \qquad (2.19)$$

при условиях

$$A(w)x + B(w)y = D(w), \qquad (2.20)$$

$$x \geq 0, y \geq 0, \qquad (2.21)$$

Вектор $Y(x, w)$ минимизирует

$$q^T(w)Y \qquad (2.22)$$

при условиях

$$B(w)Y = D(w) - A(w)x, \tag{2.23}$$

то есть является решением задачи линейного программирования (), (5.18).

Обозначим через $\Lambda(x,w) = \{\lambda_1(x,w), \lambda_2(x,w), \dots, \lambda_m(x,w)\}$ - двойственные переменные, отвечающие $y(x,w)$. «В этом случае вектор ξ_s в методе СКГ вычисляется так: после s-й итерации получаем x_s, наблюдаем w_s (например, моделируя некоторые сценарии на ЭВМ), находим $\Lambda(x_s, w_s)$ и вычисляем $\xi_s = C - A^T(w_s)\Lambda(x_s, w_s)$, где $A^T(w)$ - транспонированная матрица к $A(w_s)$» [30].

Запишем задачу, двойственную к (2.22), (2.23):

$$M_w\{\Lambda^T(D(w) - A(w)x\} \to max \tag{2.24}$$

при условии

$$B^T\Lambda \leq q \ .$$

Поскольку $\max_\lambda \Lambda(D(w) - A(w)x = \min_y q^T(w)y(x,w)$, то двухэтапной задаче (2.22)-(2.23) соответствует следующая эквивалентная задача

Минимизировать

$$F(x) = M_w\{c^T x + \Lambda^T(x,w)(D(w) - A(w)X)\}, \tag{2.25}$$

а так как $f_x(x_s) = C - A^T(w_s)\Lambda(x_s, w_s)$, то $\xi_s = C - A^T(w_s)\Lambda(x_s, w_s)$.

Соответствующая процедура для поиска оптимального решения записывается так:

$x_{s+1} = x_s - p_s\xi_s$ - для задачи без ограничений;

$x_{s+1} = \pi_x(x_s - p_s\xi_s\ )$ - для задачи с ограничениями.

2.1.1. Экстремальные задачи математической статистики

В математической статистике довольно распространенной оказывается задача оценки параметров распределения случайных величин и процессов, которая формулируется следующим образом [21].

Имеется случайная величина (или вектор) η, распределение которой известно с точностью до параметров $x = [x_1, x_2, \dots, x_N]$, и известны реализации $\eta_1, \eta_2, \dots, \eta_N$ случайной величины η при $x = x^*$. Требуется по этим наблюдениям определить x^*.

Обычно стремятся отыскать такую оценку $x^* = f(\eta_1, \eta_2, \dots, \eta_N)$, которая была бы несмещенной и эффективной. Покажем, что многие задачи оценки x^* сводятся к решению соответствующих задач стохастического программирования.

Оценка среднего значения. Пусть η - случайный вектор с неизвестным средним M_η, а $\eta_1, \eta_2, \dots, \eta\_N$ - независимые наблюдения вектора , по которым необходимо оценить M_η. Рассмотрим функцию $F(x) = M\left\|\eta - x\right\|^2$. Минимум этой функции достигается при $x^8 = M_\eta$. Кроме того, для $\xi_s = -2(\eta_{s+1} - x_s)$. имеем $M(\xi_s|x_s) = f_s(x_s)$. Получим следующую процедуру оценивания:

$$x_{s+1} = \pi_x\big(x_s + 2p_s(\eta_{n+1} - x_s)\big), s = 0,1, \dots \qquad (2.26)$$

Если $R(x) = E^{(n)}$, а $p_s = \frac{1}{2(s+1)}$, то процедура (2.26) примет вид

$$x_{s+1} = x_s + \frac{1}{s+1}(\eta_{s+1} - x_s) = \frac{1}{s+1}\sum_{k=1}^{s+1}\eta_k, \qquad (2.27)$$

то есть получим классическую формулу оценки среднего.

Оценка моментов. Пусть в постановке задачи предыдущего раздела (оценка среднего) требуется оценить моменты $M\eta^1, M|\eta|^l, M(\eta - M\eta)^l$. Предположим, что эти моменты принадлежат некоторому множеству $E^{(n)}$. Аналогично предыдущей задаче искомые моменты являются точками минимума функций

$$F_1(x) = M\left\|\eta^l - x\right\|\hat{}2 \qquad (2.28)$$

$F_2(x) = M\left\||\eta|^l - x\right\|^2, F_3(x) = M\left\{\left\|(\eta - M\eta)^l - x\right\|\right\}^2$, соответственно.

Применяя стохастический квазиградиентный метод для первых двух случаев в процедуре (2.26), положим $\xi_s = -2\big(\eta_{s+1}^l - x_s\big), \xi_s = -2(|\eta_{s+1}|^l - x_s)$, соответственно. Тогда приходим к процедуре

$$x_{s+1} = x_s + \frac{1}{s+1}(\eta_{s+1}^l - x_s) \tag{2.29}$$

При этом для соответствующей функции $F(x)$ имеем

$$M\{\xi_s|x_s) = F_x(x_s)$$

Для задачи минимизации функции

$$F(x) = M\{\left|\left|(\eta - M\eta)^l - x\right|\right|^2\}$$

в качестве вектора ξ_s можно взять :

$$\xi_s = -2[\prod_{k=1}^{l}(\eta_{s+1} - \eta_{s+1+k}) - x_s] \tag{2.30}$$

Тогда соответствующая рекуррентная процедура поиска оценки центрального момента -го порядка примет вид:

$$x_{s+1} = x_s + \frac{1}{s+1}\left[\prod_{k=1}^{l}(\eta_{s+1} - \eta_{s+1+k}) - x_s\right]. \tag{2.31}$$

Можно легко показать, что при этом $M\{\xi_s|x_s) = F_x(x_s)$. Действительно, в силу независимости η_{s+1} и η_{s+1+k} от x_s имеем

$$M(\xi_s|x_s) = -2\{M(\prod_{k=1}^{I}(\eta_{s+1} - \eta_{s+1+k})/x_s) - x\_s\,\}$$

$$= -2\{M(\prod_{k=1}^{I}(\eta_{s+1} - \eta_{s+1+k}) - x_k)\}$$

$$= -2M\{\prod_{k=1}^{I}(\eta - M\eta) - x_s\} = -2M\{(\eta - M\eta)^s - x_s = F_x(x_s)$$

2.3 Метод стохастической декомпозиции

Алгоритмы, называемые «стохастической декомпозицией» (SD), представляют собой класс методов декомпозиции, которые объединяют условно-детерминированную декомпозицию (например, декомпозицию Бендерса) с методами статистической выборки (Higle and Sen 1991, 1994, 1999). Можно сказать, что такие методы дают асимптотически оптимальные решения. Что еще более важно, алгоритмы SD были успешными в предоставлении решений с использованием компьютеров класса рабочих станций, где другие методы прибегали к высокопроизводительным вычислениям для решения тех же случаев (например, Sen et al. 1994). Тем не менее, есть вопросы, которые требуют дальнейшего изучения [40].

В общем случае двухэтапная стохастическая модель линейного программирования имеет следующий вид:

$$minf(x) := c^T x + E[h(x, \tilde{w})] \qquad (2.32)$$

$$Ax \leq b.$$

где A это $m_1 * n_1, b$ это $m_1 * 1, w$ случайная величина, определенная в пространстве вероятностей (A, F), и

$$h(x, w) = ming^T y$$

$$Wy = r(w) - T(w)x,$$

$$y \geq 0$$

В этой модели мы ограничили фиксированной версией регресса, в которой случайные величины не появляются в элементах данных g и W.

В сущности, SD - это стохастическая версия разложения Бендера (Benders1962), которая была признана как L-образный метод в литературе по SP (Van Slyke and Wets 1969). Основная идея этих методов разложения состоит в том, чтобы построить кусочно-линейную аппроксимацию функции регресса и обновить аппроксимацию во время каждой итерации алгоритма. Основное различие между SD и детерминированными методами разложения (разложение Бендера, L-образный метод) заключается в вычислительных усилиях по созданию нового сечения в приближении. В традиционных детерминированных приближениях функция регресса должна оцениваться для каждого результата случайной величины, что требует решения столько же секундных задач, сколько существует реализаций ω случайных величин ˜ ω. Следовательно, алгоритмы детерминированной декомпозиции в вычислительном отношении осуществимы только для тех случаев, в которых несколько результатов являются достаточными для моделирования случайности [39].

Более того, детерминированные методы разложения ограничены только теми случаями, в которых случайные величины являются дискретными. В отличие от этого, SD успешно преодолевает эти ограничения путем объединения выборки с последовательными аппроксимациями таким образом, чтобы уменьшить вычислительные усилия при создании нового среза (нового кусочно-линейного приближения) в каждой итерации.

В SD новый выборочный результат (ωk) получается независимо от всех предыдущих результатов, {ω1, ..., ωk − 1}, на итерации k. Фактически, алгоритм SD строит линейную аппроксимацию снизу средней функции выборки

$$H_k(x) = \frac{1}{k}\sum_{t=1}^{k} h(x, w^t)$$

Чтобы аппроксимировать вышеупомянутую функцию нижней границей, можно принять идеи, лежащие в основе разреза Бендерса. Однако такая стратегия потребовала бы от нас решения k линейных программ для получения субградиента для каждого результата h (x, ω t), t = 1, ..., k (в точке xk). На самом деле теория SD предполагает, что асимптотическая сходимость может быть достигнута даже без решения всех этих LP. Вместо этого достаточно решить только один второй этап LP (2.32) в любой итерации. Кроме того, мы включаем ранее полученные данные (об оптимальных двойственных решениях для (2.32)), чтобы определить нижнюю граничную аппроксимацию h (x, ω t), fort <k. Детали заключаются в следующем.

Для самого последнего результата, обозначенного ωk, мы оцениваем h (xk, ωk), который включает решение (2.32) с (xk, ωk) в качестве входных данных. Обозначим через πk k двойной оптимум этой задачи. Затем процесс сбора данных в алгоритме SD накапливает этот оптимальный двойной вектор в набор ранее обнаруженных оптимальных двойных векторов, обозначаемых как Vk-1. В результате обновленный набор оптимальных двойственных векторов на итерации k обозначается как Vk (то есть Vk = Vk − 1 ∪ πk k). Таким образом, множество Vk является набором оптимального двойственного вектора, обнаруженного при построении h (xt, t), t = 1, ..., k, в каждой итерации. Затем генерируется нижняя ограничительная функция для средней функции k-й выборки путем назначения двойного допустимого решения для каждого ранее наблюдаемого результата {ωt}, t <k. Чтобы увидеть это, отметим, что двойственность LP гарантирует, что для $\pi \in$ Vk

$$\pi^T(x)[r(w^t) - T(w^t)x] \leq h(x, w^t) \forall x$$

Таким образом, на итерации k двойной вектор в Vk, который обеспечивает наилучшее приближение нижней границы при {h (xk, ω t)}, for t <k, задается двойственным вектором, для которого

$$\pi_t^k(x) \epsilon argmax[\pi^T[r(w^t) - T(w^t)x] | \pi \in V_k]$$

Когда вышеуказанная операция выполняется с соответствующими структурами данных, она вычислительно быстрее, чем решение линейной программы с нуля. В любом случае из этого следует, что

$$H_k(x) \geq \frac{1}{k} \sum_{t=1}^{k} \left(\pi_t^k\right)^T [r(w^t) - T(w^t)x]$$

Используя нижнюю границу SD применяет процедуру, аналогичную разложению Бендерса, в которой правая часть добавляется как новый отрезок кусочно-линейного приближения E [h (x,w)] Наконец, отметим еще одно различие между детерминистическим и стохастическим разложением. Поскольку число результатов увеличивается с увеличением числа итераций, при создании приблизительных сокращений используется различное количество размеров выборки [28].

Таким образом, в итерации t SD используется разрез, который приближается к Ht (x), а не к Hk (x), для t <k. В результате, сначала необходимо откорректировать, чтобы обеспечить нижние границы для Hk

(x). , С этим предположением ясно, что Hk (x) ≥ t k Ht (x). Следовательно, путем умножения каждого ранее сгенерированного среза (t = 1, ..., k-1) на множитель t / k, все ранее сгенерированные срезы обеспечивают нижнюю границу для среднего приближения выборки Hk (x). В любом случае, аппроксимация целевой функции первого этапа на итерации k тогда дается

$$f_k(x) := c^T x + \max\{\frac{t}{k} * \frac{1}{t}\sum_{j=1}^{t}(\pi_j^t)^T[r(w^j) - T(w^j)x]\}$$

Самая базовая версия SD использует последовательность {xk} так, что

$$x^{k+1} \in argmin\,\{f_k(x)|x \in X\}$$

где X = {x | Ax ≤ b} обозначает выполнимую область первой ступени. Обозначая итерацию k как xk, рекомендуется следующее:

$$x^{k+1} \in argmin\{f_k(x) + \frac{1}{2}\left|\left|x - \tilde{x}^k\right|\right|^2 |Ax \leq b$$

В дальнейшем мы будем использовать ρk (x) = $\frac{1}{2}\left|\left|x - \tilde{x}^k\right|\right|^2$.Условие для обновления действующего оператора состоит в том, является ли точная оценка (выборочного среднего) объективного значения при xk + 1 лучше, чем точечная оценка объективного значения действующего агента xk. Если это так, то $\bar{x}k\ + 1\ =\ xk\ +\ 1$; иначе $\bar{x}k\ \ +\ 1$ = $\bar{x}k$. Эта процедура называется методом регуляризованного стохастического разложения (RSD). Отметим, что с изменением k меняется и Hk (x). Однако все наблюдения, кроме одного, используемые при определении Hk (x), используются в Hk − 1 (x). Следовательно, когда k становится большим, использование общих случайных чисел уменьшит дисперсию.

Higle и Sen (1996) доказали, что регуляризованный алгоритм SD сходится к оптимальному решению с вероятностью один. Эта теорема обобщена ниже.

Теорема 1. Пусть последовательности действующих и подходящих решений: {¯xk} ∞ k = 0 и {xk} ∞ k = 1, полученные с помощью регуляризованного алгоритма стохастического разложения. Тогда с вероятностью один существует подпоследовательность итераций, K $*$ такая, что limk∈K fk (xk + 1) + ρk (xk + 1) - fk ($\bar{x}k$) = 0 и каждая точка накопления {$\bar{x}k$ } k∈K оптимальна.

Higle и Sen (1999) показывают, как можно проверить сходимость подпоследовательности, комбинируя определение K $*$ с остановкой правил для нерегулярных SD. Однако эти результаты не гарантируют, что последовательность имеет уникальный предел. Следующая теорема дает следующие результаты:

Теорема 2. Обозначим через $\{\bar{x}k\}\quad K \rightarrow x *$ сходящуюся подпоследовательность, порожденную регуляризованным алгоритмом. Пусть f определено и предполагает, что f непрерывно дифференцируема в x $*$, а векторы строк соответствуют жестким ограничениям в $x *$ $(\text{т. е.} aT\ ix * = bi)$ линейно независимы. Тогда $\{\bar{x}k + 1\}\, K \rightarrow x *,$ подразумевая, что x $*$ является единственным пределом [27].

Вообще говоря, в стохастическом разложении есть два источника ошибок; одна - статистическая ошибка аппроксимации, а другая - результат линеаризации с использованием разрезов. Несмотря на то, что алгоритмизованный алгоритм алгоритма может достигать оптимальной оптимальности с конечными мастер-программами, асимптотическая сходимость (с вероятностью один) требует, чтобы число наблюдений в любом разрезе росло бесконечно. Эту проблему можно объяснить следующим образом: для генерации нового среза алгоритм стохастической декомпозиции требует все ранее сгенерированные выборки. По мере увеличения числа итераций вычислительные усилия приводят к генерации. Тем не менее, нет необходимости в полной мере использовать все сгенерированные образцы для создания нового среза. Кроме того, некоторые образцы могут быть весьма полезны для информирования о новых сокращениях, что побуждает нас применять концепцию передискретизации при стохастическом разложении.

Идея передискретизации проста: на итерации k вместо использования всех сгенерированных выборок (k) для создания нового среза мы используем только относительно небольшое количество выборок $(Lk$ и $Lk < k)$ для формирования нового среза. Когда k велико, мы можем значительно уменьшить вычислительные усилия, выбрав $Lk\, k$. С этой модификацией нужно только провести сравнение для элементов Lk. Более того, результаты сходимости SD переносятся в модифицированные SD с повторной выборкой. Прежде чем предоставить результаты сходимости SD с передискретизацией, кратко изложим метод начальной загрузки следующим образом [23].

Пусть {ω1, ..., ωk} - случайное значение выборки размера k с распределением F и буквой Fk будет эмпирическое распределение {ω1, ..., ωk}. Определите случайную величину $T\ (\omega 1, \ldots, \omega k;\ F)$, которая зависит от распределения F. Метод начальной загрузки состоит в аппроксимации распределения $T\ (\omega 1, \ldots, \omega k;\ F)$ по $T\ (\theta 1, \ldots, \theta k;\ Fk)$ в Fk, где $\{\theta 1, \ldots, \theta k\}$ обозначает случайную выборку размера k при распределении Fk. Далее мы суммируем важную теорему Сингха (1981).

Лемма 1. Пусть μ $= EF\ [\omega], \omega k\ = 1\ /\ k\ k\ i\ = 1\ \omega i, \theta k\ = 1\ /\ k$ k i = 1 θi и предположим, что $E\ [\omega 2]\ < \infty$. Пусть P и Pk обозначают вероятности при F и Fk соответственно. Затем

$$\lim |P\left\{k^{\frac{1}{2}}(w^k - \mu) \leq s\right\} - P_k\{k^{\frac{1}{2}}(\theta^k - w^k) \leq s$$

По существу, лемма 3 изучает сходимость нулевого несоответствия между распределением $k1\ /\ 2\ (\bar{}kk - \mu)$ и начальной загрузкой сопоставления. Однако, неохастическое разложение с повторной выборкой, асимптотические свойства аппроксимаций функции среднего значения выборки $Hk\ (xk)$ представляют интерес.

Одним из относительно простых способов включить повторную выборку в процесс генерации разреза является принятие / отклонение результата в выборке на основе случайной величины Бернулли. Таким образом, если p - это вероятность принятия (успеха), то мы генерируем равномерную случайную переменную для каждого ранее сгенерированного результата и используем только те результаты $(\omega t, t < k)$, для которых случайное число. Ясно, что по мере увеличения р сокращения обычно используются больше результатов в приближении [30].

Рассмотрим пример:
Мы демонстрируем эффективность повторной выборки, разрешая экземпляр, называемый 20Term, который возник при планировании перевозок (Infanger 1999, «личное общение»). Для этого экземпляра мы фиксируем максимальное число итераций на 800 для обеих версий Regularized SD (с повторной выборкой и без нее) и выполняем 20 реплицированных прогонов. В версии с передискретизацией мы запускаем процесс передискретизации после 300 итераций и выбираем вероятность принятия р равной 0,7. Наконец, в нашем вычислении LP-вычислитель использует вызываемую библиотеку ILOG CPLEX версии 10.0, и все программы были скомпилированы и запущены в среде Unix, работающей на рабочей станции Sun (Sun Fire V440).

На рисунке 1 показано время решения для двух сравниваемых версий. Мы записываем время процессора (в секундах) каждые 100 итераций. Как показано на рисунке 1, нет никакой разницы между двумя версиями для первых 300 итераций, потому что повторная выборка была начата только после 300 итераций. Однако после 300 итераций измененная версия работает быстрее по мере выполнения итераций. Более того, на итерации 800 версия с обновленной дискретизацией занимает 62 секунды по сравнению с 84,5 с для Regularized SD. Таким образом, пересчитанная версия приводит к сокращению времени вычислений на 26,7%.

Рисунок 2 демонстрирует качество решения, полученного двумя версиями. Как и ожидалось, нет никакой разницы между двумя версиями SD для первых 300 итераций. Итераций 400, там есть ошибка объективной функции, оцениваемой для повторной выборки. Эти значения были получены путем запуска оценщика вне выборки, который выбирает

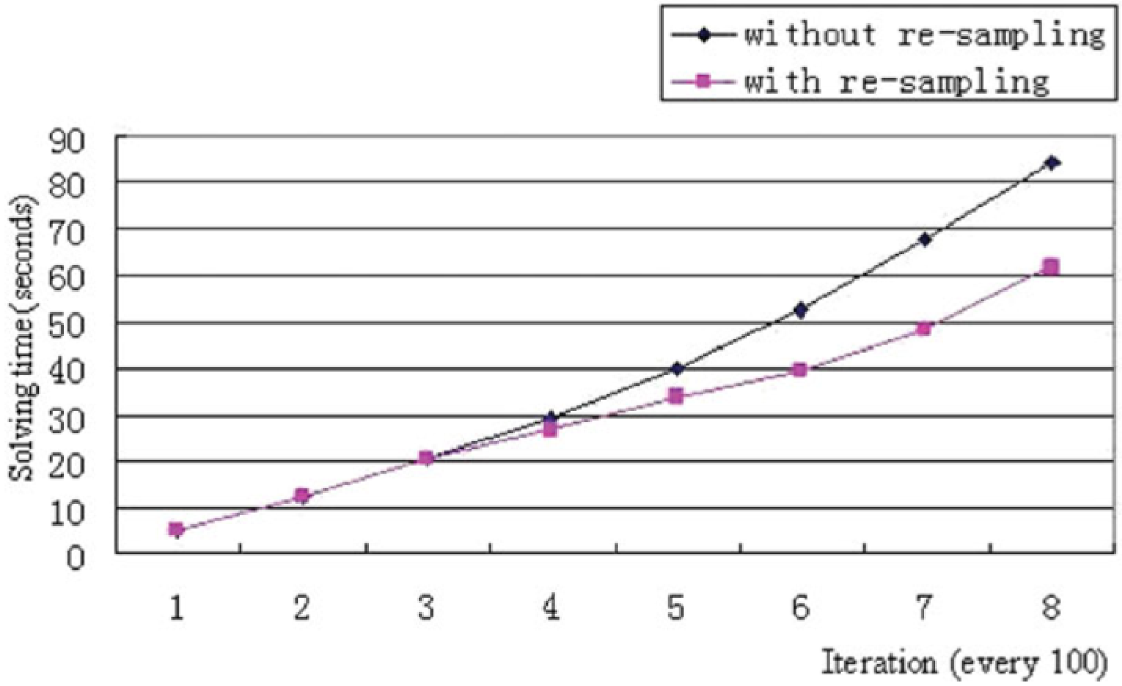


Рисунок 1 – Время решения для регуляризованного SD: с повторной выборкой и без нее

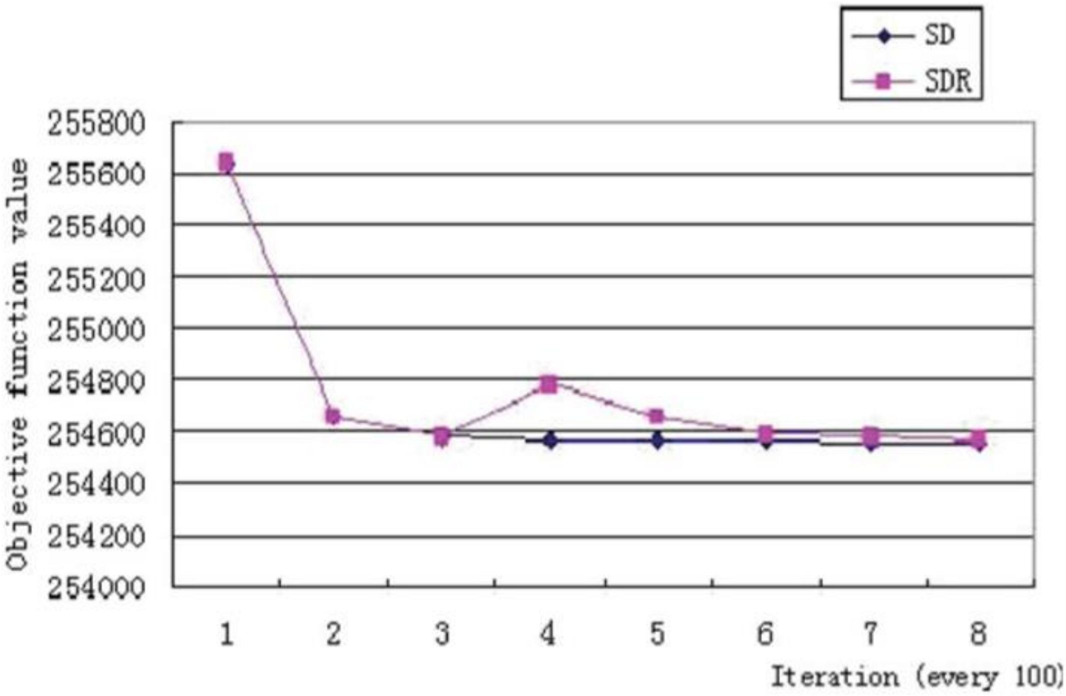


Рисунок 2 – Сравнение качества решения: SD и SDR

целевая функция, заданная на первом этапе решения, используемого в конкретной итерации. Интересно отметить, что хотя объективные значения, полученные с помощью версии с повторной выборкой, не так хороши на ранних итерациях повторной выборки, две версии начинают сходиться к одному и тому же значению по мере того, как итерации продолжаются. На самом деле, на итерации 800 наблюдаются незначительные различия в значении целевой функции между двумя версиями, с исходной версией Regularized SD, дающей 254561.8310, и сэмплированной версией, обеспечивающей немного более высокое значение на 254572.1976.

Исследованные расширения охватывают как теорию сходимости, так и алгоритмические соображения для SD. В области теории сходимости показано, что дифференцируемость функции регресса в любой точке накопления достаточна для единственного предела. Хотя это не может быть проверено априори, можно оценить вероятность уникальности при прекращении. С алгоритмической стороны мы обращаемся к проблеме масштабируемости. Показано, что при потере несимметричных свойств возможно контролировать количество результатов, полученных в процессе «argmax». Эта методика помогает алгоритмам SD преодолеть необходимость использования каждого ранее сгенерированного результата в процессе [30].

2.4Метод возмущений решения задач стохастического программирования

«Не исключен тот факт, что задачи стохастического программирования являются весьма трудноразрешимыми, или неразрешимыми вовсе. Благодаря рассматриваемому методу возмущений, задачи стохастического программирования можно разделить на основную задачу, детерминированную и стохастическую. Переход от основной задачи к детерминированной, позволяет исключить необходимость в решении сложной задачи с вероятностными ограничениями» [9].

Итак, существует некая стохастическая задача:

$$\max f(x, w)$$
$$g(x, w) \leq 0.$$

Ниже будет представлен подход к решению задачи, который был предложен в работе [27]. «Идея подхода состоит в следующем: выделить в сложной задаче основную часть, которая рассматривается как малое возмущение основной системы.

Задача:

$$\begin{gathered} \max f(x) \\ g(x) \leq 0 \end{gathered} \tag{2.33}$$

Функции $f(x)$ и $g(x)$ первоначальной задачи выглядят таким образом:

$$\begin{gathered} \varepsilon f_1(x) = f(x) - f_0(x) \\ \varepsilon g_1(x) = g(x) - g_0(x). \end{gathered}$$

где $f_0(x)$, $g_0(x)$– основная составляющая, которая описывает изучаемую систему;

$\varepsilon g_1(x), \varepsilon f_1(x)$ – настраиваемая часть, которая описывает некую возмущающую, зависящую от случайных величин компоненту» [27].

При решении задачи вводятся функции$f_0(x), g_0(x)$ и ε (ε может быть константой, функцией, случайным вектором, случайной вектор – функцией). Было решено использовать для нахождения решения не исходную задачу, а систему:

$$\begin{gathered} \max f_0(x) \\ g_0(x) \leq 0 \end{gathered} \tag{2.34}$$

и семейство задач:

$$\begin{gathered} \max\{\varepsilon f_1(x) + f_0(x)\} \\ \varepsilon g_1(x) + g_0(x) \leq 0, \end{gathered} \tag{2.35}$$

для того чтобы найти решение исходной задачи (2.33) необходимо решить простую задачу (2.34), а после внести поправки. Итак, задача (2.34) - порождающая, а задача (2.35) – возмущенной. Хотя элемент случайности (стохастики) очень часто присутствует в прикладных задачах, тем не менее, та вероятностная часть, которая, например, описывает отклонение спроса от обычной линии или внезапную поломку, является как раз возмущающей частью. Возмущения предполагаются малыми, а величина ε характеризует уровень малости этих возмущений [25]. Таким образом, предполагаем, что наряду со стабильно действующими факторами существуют возмущающие факторы, которые являются стохастическими.

Вывод соотношений.

Рассмотрим задачу:

$$\max c_0 x$$
$$Ax_0 \le b_0$$

и возмущенную задачу:

$$\max(c_0 + \varepsilon c_1)\, x \qquad (2.36)$$
$$p\{A_0 x + \varepsilon A_1 x \le b_0 + \varepsilon b_1(w)\} \ge a$$

«Чтобы была предоставлена возможность решения задачи при помощи метода возмущений, необходимо чтобы в исходной задаче присутствовали ряд свойств. Важно заметить, то, что функции, которые представляют ограничения возмущенной и порождающей частей, были непрерывными» [25].

Рассмотрим i-ую строку ограничений задачи (2.36):

$$a_{i1}^0 x_1 + a_{i2}^0 x_2 + \cdots + a_{im}^0 x_m + \varepsilon(a_{i1}^1 x_1 + a_{i2}^1 x_2 + \cdots + a_{im} x_m) \le b_i^0 + \varepsilon b_i^1(w).$$

Допустим, что величина $b_i^1(w) \infty N(b_i^1, \sigma_b)$. «Использование нормального закона распределения является целесообразным, так как случайные величины, подчиняющиеся нормальному закону, часто встречаются в прикладных задачах. После проведения преобразований получим:

$$\frac{b_i^1(w) - \overline{b_i^1}}{\sigma_b} \infty N(0,1)$$

Введем обозначения:

$$I = \left(\frac{a_{i1}^1 x_1 + a_{i2}^1 x_2 + \cdots + a_{im}^1 x_m - \overline{b_i^1}}{\sigma_b}\right) \le \frac{b_i^1(w) - b_i^1}{\sigma_b},$$

$$z = \left(\frac{a_{i1}^1 x_1 + a_{i2}^1 x_2 + \cdots + a_{im}^1 x_m - \overline{b_i^1}}{\sigma_b}\right).$$

Следовательно, для i-ой строки ограничений можно записать следующее выражение $(P(I) \ge a_i)$:

$$P(I) = \varphi(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} \exp\left\{-\frac{1}{2}\left(\frac{\sum_{j=1}^{m} a_{ij}x_j - \overline{b_i^1}}{\sigma_b}\right)^2\right\} d\left(\frac{\sum_{j=1}^{m} a_{ij}x_j - \overline{b_i^1}}{\sigma_b}\right),$$

здесь

$$\varphi(z) = 1 - \Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} exp - \frac{Z^2}{2} dZ,$$

т.е. условие непрерывности выполняется. Решить задачу до реализации случайной величины данным методом невозможно, так как для i-ой строки получаем следующее выражение:

$$P\{a_{i1}^0 x_1 + a_{i2}^0 x_2 + \cdots + a_{im}^0 x_m + \varepsilon\left(a_{i1}^1(w)x_1 + a_{i2}^2(w)x_2 + \cdots + a_{im}^1(w)x_{m1}\right)$$
$$\le b_1^0 + \varepsilon b_i^1(w)\} \ge a.$$

Такое метод целесообразнее использовать при наличии случайных переменных в ограничениях задачи. В случае порождающей задачи более одного решения, необходимо иметь возможность решить следующую вспомогательную систему:

$$F_1^* = \max\{c^{1T}x - \Lambda^T A^1 x + \Lambda^T b^1(w) | x \in \theta^*\}, \qquad (2.37)$$

где θ^* – множество решений порождающей задачи;

Λ^T – вектор Куна-Таккера» [24].

Когда матрица А является стохастической, решить вспомогательную систему невозможно. В том случае, когда матрица А является детерминированной, по-прежнему остается необходимым решать сложную стохастическую задачу на определенном множестве величин x, которые являются решениями порождающей системы. Чтобы избежать процедуры решения сложной стохастической задачи, можно оценить вероятность того, что максимизируемое значение (2.37) окажется больше некоторого β. То есть интересно, с какой возможной вероятностью будет выполняться данное выражение:

$$c^{1T}x - \Lambda^T A^1 x + \Lambda^T b^1(w) \ge \beta.$$

Для оценки выполним преобразования и, обозначив,

$$(\gamma_1, \gamma_2, \dots, \gamma_n)^T = [\Lambda\Lambda^T]^{-1}\Lambda,$$

получим

$$b^1(w) \geq \gamma\beta - \gamma cx + A_1 x.$$

Проведем оценку выражения $b_i^1(w) - \overline{b_i^1} \geq \gamma_i\beta - \gamma_i cx + a_i^1 x - \overline{b_i^1}$ с помощью неравенств Чебышева и обозначим $\gamma_i\beta - \gamma_i cx + a_i^1 x - \overline{b_i^1} = C_1$. Получим в случае, когда $b_i^1(w) \geq \overline{b_i^1}$:

$$P\left(b_i^1(w) - \overline{b_i^1}\right) \leq \frac{\sigma_{b(w)}}{C_1^2}.$$

Таким образом, учитывая, что все значения кроме стохастической величины $b(\omega)$ известны, можно, подставляя различные значения β, вычислить с какой вероятностью максимизируемая величина окажется больше некоторого значения. Величину β можно увеличивать до тех пор, пока вероятность будет оставаться необходимо большой. Так можно максимизировать необходимую величину с некоторой наперед заданной вероятностью, избежав необходимости решения первоначальной сложной стохастической задачи [37].

Пусть целевая функция стохастической задачи линейного программирования имеет вид:

$$f(x, w) = c_1^0 x_1 + c_2^0 x_2 + \dots + c_n^0 x_n + \varepsilon c_1^1(w) x_1 + \dots + \varepsilon c_n^1(w) x_n. \qquad (2.38)$$

«Интересным является вопрос о постоянстве базисных переменных, которые образуют оптимальное базисное решение при различных реализациях случайной величины. Будем рассматривать устойчивость относительно базисных переменных, найденных для задачи, в которой случайные параметры целевой функции заменены их математическими ожиданиями

$$f(x, \overline{w}) = c_1^0 x_1 + c_2^0 x_2 + \dots + c_n^0 x_n + \varepsilon \overline{c_1^1} x_1 + \dots + \varepsilon \overline{c_n^1}(w) x_n. \qquad (2.39)$$

Итак, переходим к нахождению таких условий и вероятностей, с помощью которых можно обеспечить постоянство базисных переменных. На каждом шаге при использовании симплекс-метода, определяется вводимая и выводимая из базисного решения переменная. Индекс базисной переменной напрямую зависим от коэффициентов целевой функции.

Перейдем к рассмотрению i-шага: допустим, что на i-ом шаге при решении задачи (2.39) наименьшим был t-ый столбец, следовательно, для решения (2.38) индекс ведущего столбца должен остаться таким же. Выражение, по которому формируются на i-ом шаге коэффициенты целевой функции выглядит так:

$\gamma_t = \frac{\gamma_t}{\gamma_k} - 1$,где t является индексом ведущего столбца» [15].

Заметим, что при переходе на новый шаге, коэффициенты целевой функции переименовываются. Допустим на $(i+1)$ -ом шаге наименьшим коэффициентом при решении (2.39) был коэффициент $\widehat{\gamma_\iota}$. Для задачи (2.38) индекс наименьшего коэффициента должен остаться тем же, т.е. должно выполняться условие:

$$\widehat{\gamma_t} \leq \widehat{\gamma_{t+j}},$$

или

$$\frac{c_t^0 + \varepsilon c_t^1(w)}{c_k^0 + \varepsilon c_k^1(w)} < \frac{c_{t+j}^0 + \varepsilon c_{c+j}^1(w)}{c_k^0 + \varepsilon c_k^1(\omega)},$$

где j – любое целое число: $1 \leq t + j \leq n$.

Так как знаменатель в обеих частях неравенства один и тот же, будем анализировать числитель. Проведя преобразования и обозначив

$$C_1 = c_{t+j}^0 - c_t^0 + \varepsilon\left[\overline{c_{t+j}^1}(w) - \overline{c_t^1}(w)\right],$$

получим

$$\varepsilon\left[c_t^1(w) - \overline{c_t^1}(w)\right] < \varepsilon\left[c_{t+j}(w) - \overline{c_{t+j}^1}(w)\right] + C_1.$$

Правая часть выражения в случае $c_t^1(w) - \overline{c_t^1}(w) > 0$ будет меньше некоторого ς_1 с вероятностью:

$$P\left(\left|c_t^1(w) - \overline{c_t^1}(w)\right| < \varsigma_1\right) \geq 1 - \frac{D(w)}{\varsigma_1^2}.$$

Итак, с вероятностью, большей $\frac{D(w)}{\varsigma_1^2}$, будет выполнено неравенство:

$$\varsigma_1 - \frac{C_1}{\varepsilon} < \left[c_{t+j}^1(w) - \overline{c_{t+J}^1}(w)\right].$$

С помощью неравенства Чебышева получаем оценку для случая, когда $c_{t+j}(w) - \overline{c_{t+J}^1}(w) < 0$:

$$P\left(\left|c_{t+j}^1(w) - \overline{c_{t+J}^1}(w)\right| < -\varsigma_2\right) \geq 1 - \frac{D(w)}{\varsigma_2^2},$$

где $\varsigma_2 = \varsigma_1 - \frac{C_1}{\varepsilon}$.

«Таким образом, получили вероятностную оценку возможности выполнения необходимых неравенств: это вероятность большая $1 - \frac{D(w)}{\varsigma_1^2}$ и $1 - \frac{D(w)}{\varsigma_2^2}$. Учитывая, что таких неравенств будет $\leq n$), выбираем величины $\varsigma_{1\text{Н}}, \varsigma_{2H},$ удовлетворяющие всем выражениям. Получим, что на $(i+1)-$ом шаге с вероятностью большей

$$P_H = \left(1 - \frac{D(w)}{\varsigma_{1H}}\right)\left(1 - \frac{D(w)}{\varsigma_{2H}}\right)$$

базисные переменные останутся такими же, как и при решении задач стохастического программирования, в которой случайные величины были заменены их средними значениями [31]. Таким образом, регулируя величины ς_{1H} и ς_{2H} , можно получить постоянство базисных переменных с определенной вероятностью» [30].

Глава 3 ПОСТРОЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ МАКСИМИЗАЦИИ СРЕДНЕЙ ПРИБЫЛИ ПРОИЗВОДСТВА

3.1 Анализ предметной области

Главная задача в планировании работы на предприятии состоит в том, чтобы оценить будущее состояние, взяв за основу состояние нынешней работы. Следовательно, для этого нужно учесть ряд случайных факторов, возникающих в процессе производства, и которые соответственно, влияют на сам процесс. Процесс устранения случайных факторов, идет за счет глубокого анализа работы предприятия, которое включает в себя производство, сбыт продукции. Это также влияет на получение более достоверной оценки будущих процессов на предприятии. Планирование будущего состояния, в особенности, является актуальным для предприятий с непрерывным циклом производства, т.к. сам технологический процесс очень сложный и подвергается частым изменениям, сложившийся под действием внешних и внутренних факторов (работа технологических установок, спрос, сбои в поставках сырья).

«Для того, чтобы заранее выявить возможные отклонения при реализации планов, учитывая при этом случайные компоненты предлагается использовать модели и методы стохастического программирования.

Что из себя представляет планирование на предприятии? Это реализация комплекса планов необходимых для определенных целей и задач. Различают несколько типологий планов предприятия. В данном же исследовании использована такая система планов, которая включает в себя 3 уровня:

- стратегический уровень – планирование концепции развития предприятия на долгосрочную перспективу (свыше трех лет);
- тактический уровень – объемное планирование мероприятий по реализации стратегии, выработка финансовой политики, бизнес-план сроком на 1-3 лет;

- оперативный уровень – планирование осуществления производства, включая формирование производственных, коммерческих, административно-хозяйственных планов, бюджет предприятия сроком до 1 года» [39].

Рисунок 3 – Система производственных планов предприятия

Критерием оптимальности в данной ситуации можно считать максимизацию прибыли от производства, снижение расходов, а ограничениями здесь будут являться производственные мощности, финансовые и человеческие ресурсы.

Итак, план производства определяет оптимальный план выпуска продукции. Благодаря которому можно уже с меньшими рисками определить более точное количество производимой продукции, которая пойдет на прибыль компании, а не в убыток.

Компания, выбранная для рассмотрения максимизации прибыли производства является компания «Роснефть». Компания «Роснефть» занимает лидирующую позицию нефтяной отрасли в Российской Федерации, а также является одной из крупнейших корпораций мира. «Данная нефтеперерабатывающая организация занимается поиском и разведкой месторождений углеводородов, добычей нефти, газа, а также переработкой, реализацией нефти, газа.

Основной акционер компании, а именно владелец 40,4 процентов является АО «Роснефтегаз». Также одним из акционеров является государство в лице Федерального агентства по управлению государственным имуществом, в том числе в составе акционеров входят такие компании как, BP, КьюЭйч Оил Инвестментс ЛЛК / QH Oil Investments LLC.

Цели и задачи компании, заключается в следующем:

- обновление запасов не менее 100%;
- эффективная добыча на зрелых месторождениях;
- создание новых кластеров добычи на шельфе;
- развитие технологий и внедрение практик проектного управления мирового уровня;
- монетизация газовых запасов и конкурентный рост добычи;
- максимально прибыльная реализация продукции.

Уточним что, история компании связана с историей отечественной нефтяной промышленности. Впервые о предприятиях, входящих в состав компании «Роснефть» было упомянуто в конце 19-го века. Уже в 1889 году, разведка нефтяных месторождений началась на Сахалине» [43].

Основные активы компании были созданы еще в советское время, когда началось освоение новых месторождений. Однако в 1990-е года большая часть промышленности была приватизирована. Уже в сентябре 1995 года было создано ОАО «Роснефть». Итоге добычи нефти и жидких углеводородов в 1995 году – 12,7 миллиона тонн.

За достаточно короткий срок «Роснефть» значительно повысила эффективность корпоративного управления, была также проведена работа по консолидации нефтедобывающих и перерабатывающих активов. 2005 год стал знаменательным для компании, так как «Роснефть» стала лидером среди нефтяных компаний по добыче нефти [36].

Стратегия компании состоит в следующим:

«Цель компании – рациональное освоение природных богатств, выпуск наиболее экологичных видов топлив. В рамках стратегии компания реализует инициативы, благодаря которым можно попасть в первый квартиль международных нефтегазовых компаний в области промышленной безопасности, охраны труда и окружающей среды».

Так как компания «Роснефть» занимает лидирующие позиции, соответственно, у нее большая география деятельности, а именно реализует свою деятельность в 25 странах, в 78 регионов Российской Федерации. Также компания имеет 6% доли в мировой добыче нефти, а в России 41 %.

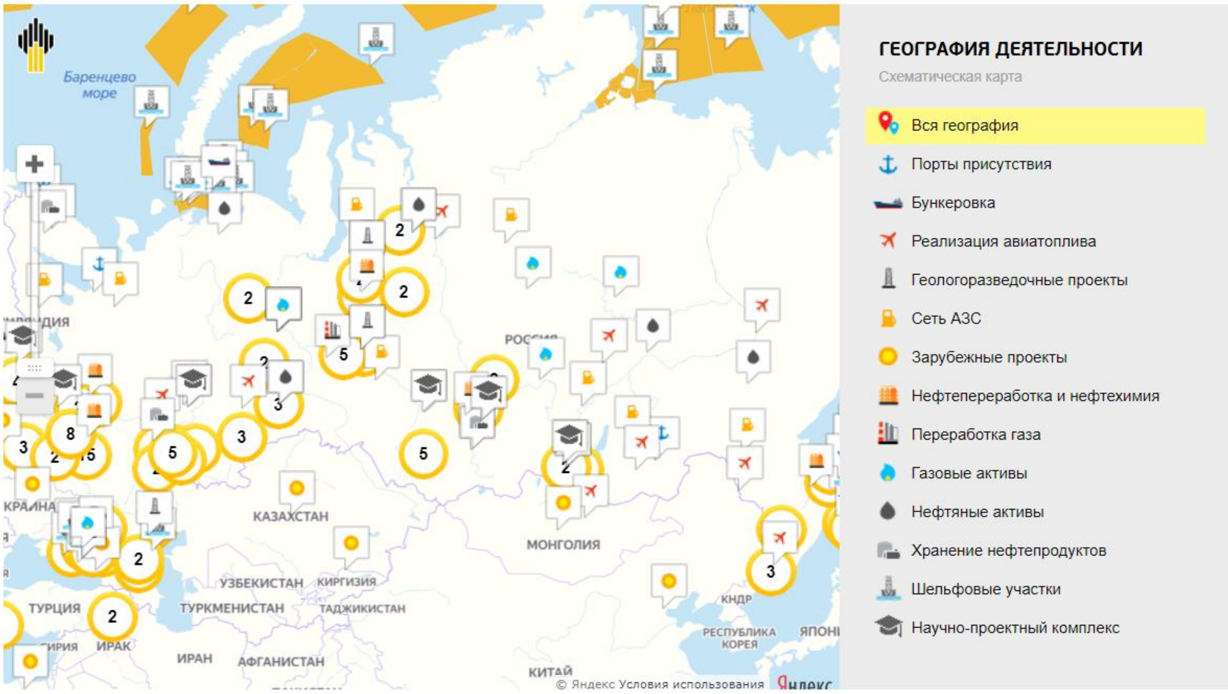


Рисунок 4 – География деятельность компании «Роснефть»

3.2 Математическая модель задачи максимизации прибыли производства

Итак, перейдем к постановке задачи.

Возьмем следующие входные данные:

- нефтепродукты, их объем и цена поставок;

- норма расхода, нефтепродукты на входе и на выходе, номинальная мощность и другие;
- расчетные параметры, параметры требуемые для анализа экономико-технических показателей.

Итоги поставленной задачи на основе входных данных:

- технико-экономические показатели в условиях вероятностной неопределенности;
- достаточный объем производственных ресурсов;
- определенная нагрузка мощностей на каждой установке;
- оптимальный план направленный на максимизацию прибыли производства продукции.

Необходимо построить математическую модель оптимального плана прибыли производства, которая позволить моделировать разные варианты производства, объемы выпускаемой продукции. Случайные факторы способствуют тому, чтобы поставленные задачи были исследованы при помощи стохастического программирования.

Итак, задача, которую необходимо максимизировать будет представлена в М-постановке с построчными вероятностными ограничениями. Задача состоит в том, найти объемы производства, которые максимизируют математическое ожидания прибыли при условии, что израсходованный ресурс не превысит имеющийся с некоторой заданной вероятностью и мощности будут использованы в полном объеме.

Таким образом NT, NP являются номенклатурой товарной продукции, полуфабрикатов собственного потребления (N= NT+NP);

M,K это количество типов оборудования и видов сырья (используемых материалов);

R1 – мощность $l-$ го типа оборудования;

$\bar{a}_{ij}$ – математическое ожидание потребности $i-$ го типа материала для производства единицы $j-$ го вида продукции;

$\bar{B}_i$ – математическое ожидание величины располагаемого $i -$го ресурса за весь период планирования;

x_j-количество выпускаемой продукции $j -$ го вида;

d_{sj} – технологический коэффициент затрат $s -$ го вида продукции на единицу $j -$ го вида сырья и используемых материалов;

$\sigma_{ij}^2[a_{ij}], \sigma_{ij}^2[b_i], j = \overline{1,n}, i = \overline{1,m}$ - дисперсии значений норм расхода и имеющегося объема ресурсов;

$t(\alpha_i)$ - значение центрированной нормированной случайной величины, которое соответствует заданному уровню вероятности соблюдения ограничения $\alpha_i, w_i, -$ величина дополнительного ресурса, необходимого для гарантированного выполнения плана при разных уровнях вероятности,

$$r_{lj} = \begin{Bmatrix} 1, \text{если продукт } j \text{ изготовляется на установке } l \\ 0, \text{в противном случае} \end{Bmatrix}, l = \overline{1,M}, j = \overline{1,N}.$$

Необходимо определить вектор $X = (x_1, x_2, \dots, x_N)$, который находит максимальное математическое ожидание прибыли

$$M[\sum_{j=1}^{NT} ct_j * x_j - \sum_{i=1}^{K} \sum_{j=1}^{N} a_{ij} * x_j * cm_i] \to \max \qquad (3.1)$$

Ограничения:

1. По мощности

$$\sum_{j=1}^{N} r_{lj} x_j \leq R_l, l = \overline{1,M} \qquad (3.2)$$

2. По ресурсам, обозначающая, что вероятность выполнения каждого заданного ограничения должна быть не менее назначенной величины α_i,

$$P[\sum_{j=1}^{n} a_{ij} * x_j \le B_i] \ge a_i \tag{3.3}$$

3. На неотрицательность вектора

$$x_j \ge 0, j = \overline{1, N}, \tag{3.4}$$

Ниже представлена модель, которая приведена к детерминированной задаче, состоящая из линейной целевой функции с нелинейными ограничениями:

$$F = \sum_{j=1}^{NT} \overline{ct_J} * x_j - \sum_{i=1}^{K} \sum_{j=1}^{N} \overline{a_{\iota J}} * x_j * \overline{cm_\iota} \to \max \tag{3.5}$$

при следующих ограничениях

1. По мощности

$$\sum_{j=1}^{N} r_{lj} x_j \le R_l, \qquad l = \overline{1, M}, \tag{3.6}$$

2. По ресурсам

$$\sum_{j=1}^{m} \overline{a_{\iota J}}\, x_j \le \overline{B_{\iota_\iota}} - w_i, \tag{3.7}$$

$$\text{где } w_i = t(\alpha_i)(\sum_{j=1}^{n} \sigma_{ij}^2 [a_{ij}] x_j^2 + \sigma_i^2 [b_i])^{\frac{1}{2}},$$

3. На неотрицательность вектора

$$x_j \ge 0, \;\; j = \overline{1, N} \tag{3.8}$$

Рассмотрим производство, которое занимается нефтепереработкой, а именно компанию «Роснефть». Рассмотрим производство, которое занимается нефтепереработкой, а именно компанию «Роснефть». Основным производством компании, которое идет на продажу, является бензин, дизельное топливо, газ. Главное сырье для готовой продукции – нефть, а также дополнительно в производство включаются следующие ресурсы: пар, вода, сжатый воздух, аммиак, сода каустическая и др.

Цены на выпускающую продукцию взяты с сайта компании «Роснефть». Соответственно доходы - это сумма продажи продукции. Затраты же включаются в стоимость всех ресурсов.

Сравнительная оценка относительного изменения прибыли компании и увеличения затрат при разных уровнях вероятности приведены на рисунке 5.

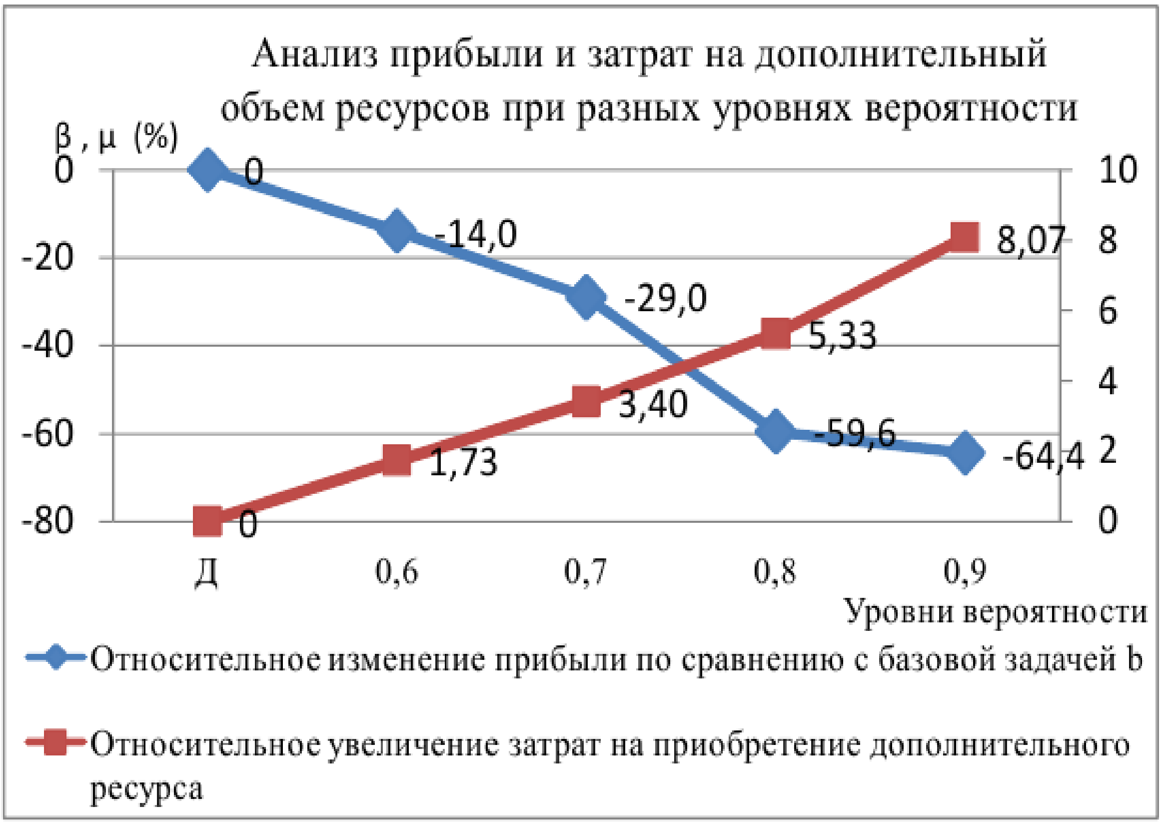


Рисунок 5 – Анализ прибыли и затрат

Итак, в процессе анализа полученных данных выявлено то, что учет случайных величин оказывает огромное влияние на дальнейшую прибыль производства. В случае их пренебрежения, предприятие может потерпеть существенное снижение прибыли.

К примеру, чтобы обеспечить 60% объема прибыли, необходимо увеличить затраты на ресурсы на 1.74 %, при отсутствии такого увеличения, предприятие может ожидать снижение прибыли на 14 %.

Вывод проделанной работы:

В ходе проведения анализа работы в 3 главе, было выявлено что, план производства характеризуется высоким уровнем неопределенности, тем самым это приводит к принятию ошибочных решений. Следовательно, чтобы уменьшить риск, необходимо разработать такую математическую модель, чтобы была возможность определить его оптимальный план максимизации прибыли производства. Такую математическую модель можно построить благодаря стохастическому программированию.

Математическая модель в свою очередь будет сведена к задаче линейного программирования с нелинейными ограничениями, ставя при этом перед собой цель - нахождение такого объема продукции, которая приведет к максимальной прибыли производства. Необходимым является обязательный учет случайных параметров, в противном случае это приведет к снижению прибыли, а также к риску множества невыполненных задач.

Глава 4 РЕАЛИЗАЦИЯ АЛГОРИТМА СТОХАСТИЧЕСКОГО ПРОГРАММИРОВАНИЯ

4.1 Выбор программного обеспечения

Программным обеспечением является совокупность программ, предназначенных для отладки, функционирования, проверки работоспособности системы.

Начальные этапы проектирования программного обеспечения включают в себя принятия важных решений, которые определяют процесс проектирования. Такими решениями являются выбор:

- подхода к разработке;
- архитектуры программного обеспечения;
- систем управления базами данных;
- языка и среды программирования.

Следующим шагом является выбор языка программирования. Языков программирования большое количество, они различаются по предоставляемым возможностям. К выбору языка программирования необходимо отнестись ответственно, так как это влияет на качественный результат, скорость и удобства самого процесса разработки. Исходя из возможностей будущей реализации алгоритма и его свойств, которые будут получены в процессе разработки, определены некоторые критерии выбора языка программирования:

- невысокая цена;
- портативность разработанного продукта под различные системы;
- простота настройки;
- скорость выполнения команд;
- универсальный и ясный синтаксис.

Всем необходимым требованиям соответствует несколько языков: PHP, Java, ColdFusion, Matlab.

Перейдем к рассмотрению данных языков.

«PHP язык довольно молодой, он установлен уже на порядка миллиона серверов по всему миру. Новое поколение PHP должно вообще стереть все преимущества Perl перед PHP с точки зрения быстродействия обработки программ. Наконец, большинство PHP-сценариев работают очень быстро. это молодой, перспективный и быстроразвивающийся языков программирования, доля его использования по сравнению с другими языками набирает большие обороты» [34].

«Пакет ColdFusion, разработанный Allaire, предназначен для быстрой разработки и интерактивных динамических веб-документов, обрабатывая информацию, полученную из базы данных. Существенный недостаток - низкая допустимость. ColdFusion работает только на четырех платформах: Win32, Solaris, HP / UX и Linux. Кроме того, ColdFusion является коммерческой разработкой. Преимуществом является хорошая среда разработки» [35].

«MATLAB – это язык высокого уровня и интерактивная среда для программирования, численные вычисления и визуализация результатов. Благодаря MATLAB анализируются данные, разрабатываются алгоритмы, создаются модели, приложения» [36].

Язык Java является C-подобным языком. «Технология Java включает в себя клиентские и серверные части, а также доступ к базам данных, поэтому наиболее корректно сравнивать Java с Apache / PHP / MySQL. Технология Java была разработана как кросс-платформенная технология, позволяющая создавать веб-приложения масштаба предприятия. Основными преимуществами технологии является переносимость и кросс-платформенный объектно-ориентированный язык, позволяющий создавать сложные и крупные приложения. Одним из недостатков является медленное выполнение, потребление больших объемов памяти и сложность разработки приложений. Использование Java для разработки довольно простого приложения, вряд ли оправдано, и при работе над серьезными задачами Java стоит дороже» [33].

Для того, чтобы объективно сделать выбор, необходимо провести сравнительный анализ предложенных языков. Соответствующие результаты приведены в таблице 4.1.1 причем наличие того или иного критерия будем обозначать числом – баллом в интервале от 0 (не поддерживается) до 10 (полная поддержка).

Таблица 4.1.1 – Анализ языков программирования

Анализ языков программирования				
Критерии	PHP	Java	ColdFusion	Matlab
Быстродействие	10	10	5	10
Простота синтаксиса	5	8	5	10
Межплатформенная переносимость	10	10	5	10
Доступность	10	10	3	10
Итого:	35	38	18	40

Из приведенного выше сравнения языков, можно сделать вывод, что Matlab обладает всеми необходимыми требованиями к языку программирования разрабатываемой системы, по сравнению с другими языками программирования.

4.2 Обзор и обоснование выбора среды разработки Matlab

MATLAB – это язык высокого уровня и интерактивная среда для программирования, численные вычисления и визуализация результатов. Благодаря MATLAB анализируются данные, разрабатываются алгоритмы, создаются модели, приложения [13]. MATLAB одна из широко используемых математических вычислительных сред. В 1970-х годах MATLAB начинался как интерактивный интерфейс для EISPACK и LINPACK. С того момента среда значительно расширилась в ней появились инструменты для решения задач в различных прикладных областях. MATLAB подходит для программирования матричных задач, так как он дает гибкую индексацию матриц. MATLAB имеет удобному интерфейс, хорошие графические возможности, имеющие библиотеки ускоряющие векторные матричные вычисления. Кроме того, исследователи строят очень сложные системы с использованием языка и инструментов MATLAB [6]. MATLAB предоставляет большое количество методов для анализа данных, разработки алгоритмов и создания моделей. MATLAB включает также математические функции для инженерных и научных операций.

Особенности MATLAB:

- ✓ является высокоуровневым языком программирования;
- ✓ работа с матрицами;
- ✓ разработка алгоритмов;
- ✓ среда разработки кода – диалоговая;
- ✓ возможность управления файлами и данными;
- ✓ математические методы: статистика, дифференциальные уравнения, анализ Фурье и другие;
- ✓ 2D,3D графика;
- ✓ встроенные средства разработки пользовательского интерфейса для создания законченных приложений на MATLAB.

MATLAB это инструмент, который обеспечивает взаимодействие оператора со всеми доступными возможностями анализа, сбора данных и презентаций. У него также есть свои минусы и плюсы, как практически и во всех языках программирования.

Недостатками языка MATLAB являются:

- ▪ медленный и перегруженный операторами, командами, функциями язык, основной целью которого является улучшение визуального восприятия;
- ▪ узконаправленный;
- ▪ невысокий спрос.

В достоинства MATLAB входят следующие пункты:

- ▪ простой, понятный синтаксис;
- ▪ легок в изучении;

- 2-3 раза в год соответствующие обновления;
- Способность среды к преобразованию в код на C, C++.

4.3 Реализация алгоритма

Определение количественных характеристик и функций случайной величины.

Рассмотрим одно из основных нефтеперерабатывающих предприятий в России, компания «Роснефть». Данные по прибыли от продукции за литр взяты с официального сайта компании «Роснефть». Цена указана по Самарской области. В качестве продукции взята продукция производимая на данном нефтеперерабатывающем предприятии. Продукция именуется следующим образом: бензин (взят наиболее ходовой автомобильный бензин), дизельное топливо (в ниже приведенной таблице сокращено до аббревиатуры ДТ), а также сжиженный углеводородный газ. Использованы ограничения по сырью (углеводороды), и по мощности переработки. Данные использованные для определения максимизации прибыли производства представлены в таблице 4.3.1.

Таблица 4.3.1 – Ресурсы для получения продукции нефтеперабатывающего производства компании «Роснефть»

Ограничения	Продукция			Объемы
	Бензин(АИ-92)	ДТ	СУГ	
Углеводороды(парафиновые)(%)	0	0,25	0	38
Углеводороды(нафтеновые) (%)	0,40	0,40	0	44
Углеводороды(предельные) (%)	0,43	0	0,95	50
Углеводороды(непредельные) (%)	0,29	0	0,01	30
Углеводороды(ароматические) (%)	0,10	0,22	0	40
Мощность	1,40	1,78	42,3	62
Прибыль(руб/литр)	41	46	17	

В условии задачи сформулирована цель - добиться максимального дохода от реализации продукции, т.е. критерием эффективности служит параметр суточного дохода, который должен стремиться к максимуму. Чтобы рассчитать величину суточного дохода от продажи продукции обоих видов, необходимо знать объемы производства. Таким образом, доход от продажи суточного объема производства бензина равен $41x_1$ руб. в сутки, от продажи дизельного топлива - $46x_2$ руб. в сутки, а от продажи газа – $17x_3$ руб. в сутки. Поэтому запишем целевую функцию в виде суммы дохода от продажи представленной продукции.

$$Z(x) = 41x_1 + 46x_2 + 17x_3 \to max$$

Ограничения.

Возможные объемы производства продукции $x_{1,}, x_2, x_3$ ограничиваются следующими условиями:

$$\begin{cases} 0.25x_2 \leq 38 \\ 0.40x_1 + 0.40x_2 \leq 44 \\ 0.43x_1 + 0.95x_3 \leq 50 \\ 0.29x_1 + 0.01x_3 \leq 30 \\ 0.1x_1 + 0.22x_2 \leq 40 \\ 1.40x_1 + 1.78x_2 + 42.3x_3 \leq 62 \\ x_{i,j} \geq 0, j = \overline{1,6} \end{cases}$$

Математическая модель задачи состоит в том, чтобы найти такой план производства продукции $\overline{x} = (x_1, x_2, ..., x_n)$, удовлетворяющий системе ограничений, при котором целевая функция принимает максимальное значение.

Определение оптимального плана производства

Аналитическое решение задачи

Будем считать числовые значения $c_j, a_{ij}, b_i (i = \overline{1,6}; j = \overline{1,6})$ эти величины случайными и в математической модели представлены их средние значения.

Пусть наблюдения за величиной прибыли от продукции в течении 2019-2020 гг распределились следующим образом как указано в таблице 4.3.2:

Таблица 4.3.2 – Данные о прибыли по кварталам за 2019-2020 год

	Данные о прибыли по кварталам за 2019-2020 год				
квартал	I кв 2019	II кв 2019	III кв 2019	IV кв 2019	I кв 2020
Прибыль (млрд.руб)	131	194	225	158	156

Найдем количественные характеристики случайных величин. Количественными характеристиками являются математическое ожидание $E[x] = \frac{1}{n}\sum_{i=1}^{n} x_i$; среднее квадратическое отклонение

$$\sigma[x] = \sqrt{\frac{\sum_{l=1}^{n}(x_i - E[x])^2}{n}}$$

где x_i- значение случайной величины прибыли).Вычислим также коэффициент вариабельности, характеризующий относительный разброс случайных величин $V[x] = \sigma[x]/E[x]$.

Полученные значения приведены в таблице 4.3.2:

Таблица 4.3.3 – Количественные характеристики

Количественные характеристики		
мат.ожидание	ср.кв.отклонение	коэф.вариабельности
172,8	36,8	0,21

Далее формируем исходных данных для детерминированного эквивалента задаче в М-постановке. Для формирования детерминированного эквивалента, с помощью коэффициента вариабельности вычислим $\sigma[a_{ij}], \sigma[b_{ij}]$.

Полученные данные представлены в таблицах 4.3.4, 4.3.5, 4.3.6:

Таблица 4.3.4 – Данные для формирования детерминированного эквивалента

	Формирование детерминированного эквивалента	$\sigma(a_{ij})$		$\sigma(b_i)$
угл.(параф)	0	0,0525	0	7,98
угл.(нафтен)	0,084	0,084	0	9,24
угл.(пред.)	0,0903	0	0,1995	10,5
угл.(непред.)	0,0609	0	0,0021	6,3
угл.(аромат.)	0,021	0,0462	0	8,4
мощность	0,294	0,3738	8,883	13,02

Таблица 4.3.5 – Детерминированный эквивалент

	$\sigma^2\,[a_{ij}]$			$\sum_{j=1}^{6}\sigma^2[a_{ij}]$	$\sigma^2[b_j]$
угл.(параф)	0	0,275625	0	0,275625	63,6804
угл.(нафтен)	0,7056	0,7056	0	1,4112	85,3776
угл.(пред.)	0,815409	0	3,980025	4,795434	110,25
угл.(непред.)	0,370881	0	0,000441	0,371322	39,69
угл.(аромат.)	0,0441	0,213444	0	0,257544	70,56
мощность	8,6436	13,972644	7890,7689	7913,385144	169,5204

Таблица 4.3.6 – Вычисление стохастического параметра и вычисление левой части ограничений

W_i	$t[\beta]$	Лев.часть	Знак	
7,997251	6,73065633	9,2306563	<=	38
9,316051	7,84058605	15,840586	<=	44
10,72592	9,02716493	22,827165	<=	50
6,329401	5,32695861	8,3269586	<=	30
8,415316	7,08250866	10,282509	<=	40
89,90498	75,6659407	530,46594	<=	62

где $t[\beta]W_i$- стохастический параметр.

Решение данной стохастической задачи (ее детерминированного эквивалента) свелось к решению задачи нелинейного программирования. Согласно полученному решению объемы производства продукции и прибыль характеризуется следующими значениями: бензин (Аи-92)=34,37; дт=0; суг =0; прибыль=1409,45. Таким образом, оптимальный вариант – сконцентрироваться на выпуске продукции бензина, дт и суг производить не стоит.

Алгоритм нахождения максимальной прибыли производства имеет вид, представленный в листинге 1.

Листинг 1. Алгоритм нахождения максимальной прибыли производства

```
function [ val mat ]  = simplex_min(A,C )
```

```
A =[0 0,25 0 ; 0,40 0,40 0 ;0,43 0 0,95; 0.29 0 0.01; 0.1
0.22 0; 1.40 1.78 42.3]
C = [38 44 50 30 40 62]
[ na ma] = size(A);
[ nc mc] = size(C);
% check for matrix C

if nc ~= 1
    disp('Pls check the given objective function.It should
be row matrix ')
    return
end
if ma-1 ~= mc
    disp('Check the given objective function or augmented
matrix')
    return
end

X = [ A(:,1:ma-1) eye(na) A(:,ma) ];
X(na+1,:) = zeros(1,na+ma);
X(na+1,1:mc) = -C;% Indicator row.

while sum(X(na+1,1:na+ma-1) > zeros(1,na+ma-1)) ~= 0
    % finding the largest matrix element in the co-
efficient matrix
    xw = X(1:na , 1:na + ma - 1);
    [ v1 i1 ] = max(xw);
    [ v2  j ] = max(v1);% determining j and hence the pivot
coloumn
    i = i1(1,j);
    Y = X(1:na,na+ma)./X(1:na,j);% determining lowest
positive ratio for finding pivot row
    a1 = sign(Y);
    a1 = a1 + ones(na,1);
    y1 = Y.*a1/2;
    [ v3 i ] = min(y1);% finding lowest non -ve no
        if v3 == 0
            ys = sort(y1);
            k = 1;
                while ys(k,1) <= 0
                       k = k + 1;
                end
            b = ys(k,1);
            [ i j1 ] = find( y1 == b );
        end
        X = elimination(X,i,j); % Pls see the function '
elimination '

        % finding -ve no in the column matrix
```

```
        ele = find(sign(X(na+1,1:na+ma-1))== -1);
        [ ne me ] = size(ele);
        if me == 0
            break
        else
             j = ele(1,1);% fixing  pivot column
            Y = X(1:na,na+ma)./X(1:na,j);
            a1 = sign(Y);
            a1 = a1 + ones(na,1);
            y1 = Y.*a1/2;
            [ v3 i ] = min(y1);
                if v3 == 0
                ys = sort(y1);
                k = 1;
                    while ys(k,1) <= 0
                               k = k + 1;
                    end
                b = ys(k,1);
                [ i j1 ] = find( y1 == b );
                X = elimination(X,i,j);
                end
        end
        % for checking boundedness of solution
        for k = 1:na+ma-1
            un = sign(X(:,k));
            if un == - ones(na+1,1)
                disp(' The solution is not bounded')
                return
            end
        end
end

% Obtaining the solution from Final tebula

opt = X( na+1, ma+na);
sol = X(1:na , 1:ma-1);
for k = 1: ma-1
        % looking for the column which forms the rrel for
matix A
        t = roots( [sol(:,k);0] );
        [ nt mt ] = size(t);
        if t == zeros(nt,1)
            mat(1,k) = X(na - nt +1, na+ma);
        else
            mat(1,k) = 0;
        end
end
disp('Co-efficient matrix correspond to optimum solution ')
mat
```

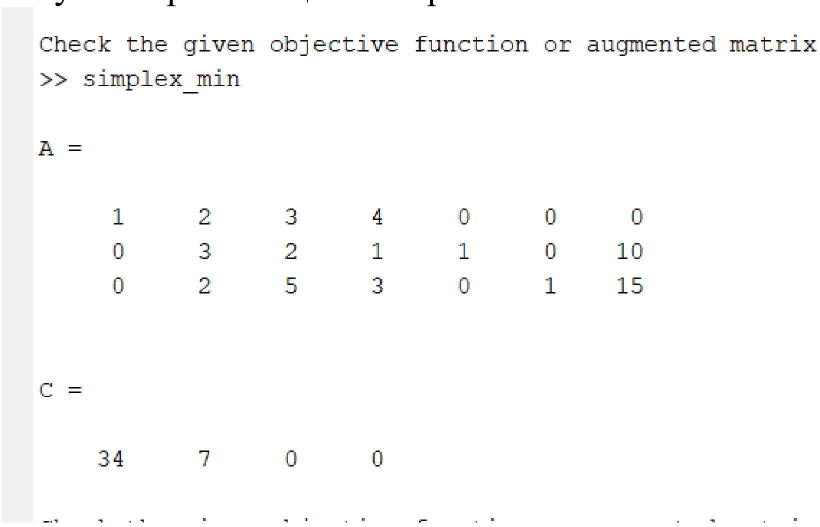
```
disp('and optimum value is')
opt

function X = elimination(X,i,j)
% Pivoting (i,j) element of matrix X and eliminating other
column

% elements to zero

[ nX mX ] = size( X);
a = X(i,j);
X(i,:) = X(i,:)/a;
for k =  1:nX
    if k == i
        continue
    end
    X(k,:) = X(k,:) - X(i,:)*X(k,j);end
```

Результат реализации алгоритма:



```
Check the given objective function or augmented matrix
>> simplex_min

A =

     1     2     3     4     0     0     0
     0     3     2     1     1     0    10
     0     2     5     3     0     1    15


C =

    34     7     0     0
```

Рисунок 6 – Результат программной реализации алгоритма

ЗАКЛЮЧЕНИЕ

Стохастическое программирование - это область математического программирования, в котором в отличии от математического программирования содержаться случайные параметры. Тем самым, определяется важность стохастического программирования как одной из основных областей математического программирования, так как в реальных задачах вероятность появления случайных параметров больше, чем фиксированных. Конечно, нахождения этих неизвестных переменных не обходится без применения методов математического программирования.

Для того, чтобы решить задачу стохастического программирования необходимо выбрать постановку задачи, благодаря которой происходит переход к детерминированному эквиваленту, и уже после того, как это переход осуществился, применяется, в частности, линейное программирование. Решение таких задач, как правило, происходит в один или в два этапа, тем самым они подразделяются на одноэтапные и двухэтапные задачи стохастического программирования. Благодаря задачам стохастического программирования можно предугадать к примеру, спрос на продукцию.

В ходе написания магистерской диссертации была выполнена поставленная цель, а именно, была построена математическая модель максимизации прибыли на основе стохастического программирования.

Также были выполнены следующие задачи:

- проанализированы различные модели стохастического программирования;
- проведен анализ предметной области деятельности предприятия с точки зрения максимизации прибыли;
- построена математическая модель максимизации прибыли с использованием стохастического программирования;

- осуществлена программная реализация математической модели максимизации прибыли предприятия.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Азанов В. М. Оптимизация коррекции околокруговой орбиты искусственного спутника Земли по вероятностному критерию / Кан Ю.С. // Труды ИСА РАН. – 2015. – Т. 65. – №2. – С. 18–26.
2. Акулич, И. Л. Математическое программирование в примерах и задачах: учеб. пособие для студентов эконом. спец. вузов / И. Л. Акулич. – М.: Высшая школа, 1986. – 319 с.
3. Аттетков, А. В. Методы оптимизации / Галкин С. В., Зарубин В. С. – М.: МГТУ им. Баумана, 2001. – 440 с.
4. Ермольев, Ю. М. Методы стохастического программирования / Ю. М. Ермольев. – Москва: Наука, 1976. – 239 с.
5. Ермольев, Ю. М., Некрылова, 3. В. О некоторых методах стохастической оптимизации /Ю. М. Ермольев, Некрылова 3. В.// Кибернетика. – 1966. – № 6. – С. 96—98.
6. Кан, Ю. С. Задачи стохастического программирования с вероятностными критериями / Кибзун А. И. – М.: Физматлит, 2013. – 542 с.
7. Кан, Ю. С. Минимизация квантили нормального распределения билинейной функции потерь / Тузов Н. В. // Автоматика и телемеханика. – 1998. – № 11. – С. 82–92.
8. Канторович, Л. В. Математические методы организации и планирования производства / Л. В. Канторович. – Л.: ЛГУ, 1939. – 352 с.
9. Карманов, В. Г. Математическое программирование: учеб. пособие / В. Г. Карманов. – М.: ФИЗМАТЛИТ, 2004. – 264 с.
10. Кибзун, А. И. Сведение двухшаговой задачи стохастического оптимального управления с билинейной функцией дохода к задаче

смешанного целочисленного линейного программирования / Игнатов А. Н. // Автоматика и телемеханика. – 2016. – №12. – С. 80–101.

11. Колбин, В. В. Двухэтапная задача стохастического программирования и многоэкстремальность. Процессы управления и устойчивость / В. В. Колбин, Е. С. Савкина, М. А. Суворова // Труды XXXII научной конференции студентов и аспирантов. – СПб.: СПбГУ, 2001. – С. 14–18.

12. Котов, В. П. Математическое программирование: учеб. пособие / В.П. Котов, Н.А., Адрицкая. – СПб.: Лань, 2014. – 432 с.

13. Лэсдон, Л. С. Оптимизация больших систем / Л. С. Лэсдон. – М.: Наука, 1977. – 655 с.

14. Лю, Б. Теория и практика неопределенного программирования / Б. Лю. – М.: БИНОМ, 2005. – 416 с.

15. Мину, М. Н. Математическое программирование. Теория и алгоритмы / М. Н. Мину. – М.: Наука, 1990. – 487 с.

16. Моисеев, Н. Н. Методы оптимизации / Н. Н. Моисеев, Ю. П. Иванилов, Е. М. Столярова. – М.: Наука, 1978. – 520 с.

17. Первозванская, Т. Н. Стохастическое линейное программирование. Проблемы применения математики в социалистической экономике / Т.Н. Первозванская. – Л.: ЛГУ, 1965. – 254 с.

18. Первозванский, А. А. Декомпозиция, агрегирование и приближенная оптимизация / В.Г. Первозванский, А.А. Гайцгори. – М., 1979. – 425 с.

19. Поляк, Б. Т. Введение в оптимизацию / Б. Т. Поляк. – М.: Наука, 2012. – 452 с.

20. Соколов, А. В. Методы оптимальных решений. Общие положения. Математическое программирование / А.В. Соколов. – М.: Физматлит, 2012. – 564 с.

21. Солдатов, В. Е. О задачах линейного программирования со случайными данными / В. Е. Солдатов // Математические модели и методы оптимального планирования. – Новосибирск: Наука, 1966. – С. 54 – 64.

22. Солдатов, В. Е. О некоторых задачах стохастического программирования / В. Е. Солдатов // Математическое программирование. – М. : Наука, 1966. – № 2. – С. 78—89.

23. Таха, Х. А. Введение в исследование операций, 7-е издание / Х. А. Таха. – М.: Вильямс, 2005. – 912 с.

24. Шор, Н. 3., Щепакин, М. Б. Алгоритм решения двухэтапиой задачи стохастического программирования / Н. З. Шор, М. Б. Щепакин // Кибернетика. – 1968. – № 3. – С. 56—58.

25. Юдин, Д. Б. Задачи и методы стохастического программирования / Д. Б. Юдин. – Сов. радио, М., 1979. – 425 с.

26. Юдин, Д. Б. Новые подходы к стохастическому программированию / Д. Б. Юдин // Экономика и математические методы. – 1968. – № 6. – С. 907—919.

27. Юдин, Д. Б. Об одном классе задач стохастического программирования /Д. Б. Юдин // Докл. АН СССР. – 1967. – № 6. – С. 1292—1293.

28. Юдин, Д. Б. Математические методы управления в условиях неполной информации / Д. Б. Юдин. – М.: Сов. радио, 1974. – 400 с.

29. Юдин, Д. Б. Обобщенное математическое программирование / Д. Б. Юдин // Экономика и математические методы. – 1984. – Т. 20. – № 1. – С. 148-168.

30. Юдин, Д. Б. Линейное программирование. Теория, методы, приложения / Д. Б. Юдин. – М.: Наука, 1969. – 424 с.

31. Юрьева, А. А. Математическое программирование: учеб. пособие / А.А. Юрьева. – СПб.: Лань, 2014. – 432 с.

Электронные ресурсы

32. MATLAB. – 2020 [Электронный ресурс]. Дата обновления: 28.04.2020. – URL: https://ru.wikipedia.org/wiki/MATLAB (дата обращения: 17.05.2020).

33. Дементий, Д. Язык программирования Java: особенности, популярность, ситуация на рынке труда. / Д. Дементий. – 2019 [Электронный ресурс]. Дата обновления: 18.11.2019. – URL: https://yandex.ru/turbo?text=https%3A%2F%2Fru.hexlet.io%2Fblog%2Fposts%2Fyazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda (дата обращения: 17.05.2020).

34. PHP. – 2020 [Электронный ресурс]. Дата обновления: 10.06.2020. – URL: https://ru.wikipedia.org/wiki/PHP (дата обращения: 17.05.2020).

35. ColdFusion. – 2019 [Электронный ресурс]. Дата обновления: 30.09.2019. – URL: https://ru.wikipedia.org/wiki/ColdFusion (дата обращения: 17.05.2020).

36. «Роснефть» сегодня – 2020 [Электронный ресурс]. Дата обновления: 01.03.2020 – URL: https://www.rosneft.ru/about/Glance/ (дата обращения: 10.04.2020).

Литература на иностранном языке

37. Artzner, P. T. Coherent measures of risk / P. T. Artzner // Mathematical Finance. –2014. – Vol. 9, № 3. – P. 203 – 228.

38. Barmish, B. R. The uniform distribution: a rigorous justification for its use in robustness analysis / Lagoa C. M. // Math. Control, Signals Systems. – 1997. – Vol. 10. – P. 203 – 222.

39. Beale, E. M. L. On Minimizing a Convex Function Subject to Linear Inequalities / E. M. L. Beale // Journal of Royal Statistical Society. – 1955. – Vol. 17, Series B. – P. 173–184.

40. Benati, S. A. Mixed integer linear programming formulation of the optimal of the optimal mean / ValueatRisk portfolio problem / Rizzi R // European Journal of Operational Research. – 2007. – Vol. 176, № 1. – P. 423–434.

41. Dyer Martin Computational complexity of stochastic programming problems [Text] / Dyer Martin , Stouge Leen // Eindhoven: Technishe Universiteit Eindhoven. – 2003.