

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Кафедра \_\_\_\_\_ Прикладная математика и информатика \_\_\_\_\_  
(наименование кафедры)

01.03.02 Прикладная математика и информатика  
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование  
(наименование профиля, специализации)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему: «Построение алгоритма и программная реализация решения стохастической задачи управления запасами»

Обучающийся	А.К.Кеворкова _____ (И.О. Фамилия)	_____ (личная подпись)
Руководитель	к.т.н., доцент, Н.А. Сосина _____ (ученая степень, звание, И.О. Фамилия)	
Консультант	к.п.н., доцент, А.В. Егорова _____ (ученая степень, звание, И.О. Фамилия)	

## Аннотация

Выпускная квалификационная работа (ВКР) посвящена теме: «Построение алгоритма и программная реализация решения стохастической задачи управления запасами». Работа была выполнена бакалавром кафедры прикладной математики и информатики Тольяттинского Государственного Университета Кеворковой Анастасией Константиновной.

Целью бакалаврской работы является разработка алгоритмов и создание программного продукта, способного помочь компаниям оптимизировать свои запасы, учитывая случайные факторы.

Задачи бакалаврской работы состоят в следующем:

- изучить существующие математические модели процессов управления запасами; изучить существующие математические методы оптимизации процессов управления запасам;
- построить математические модели стохастических задач управления запасами с отложенным спросом и потерянным спросом;
- разработать алгоритм для решения стохастических задач с применением математических методов;
- выполнить программную реализацию разработанных алгоритмов;
- выполнить тестирование и анализ полученных результатов.

Структура работы включает в себя введение, три раздела, заключение, список литературы. В первом разделе описываются математические модели процессов управления запасами, анализируются математические методы, применяемые для решения задач управления запасами, излагаются теоретические основы для решения стохастических задач с отложенным спросом и потерянным спросом. Во втором разделе приводится алгоритм решения стохастических задач управления запасами с отложенным спросом и потерянным спросом. В третьем разделе представлена программная реализация разработанных алгоритмов на языке программирования Python.

## Abstract

Graduate qualification work (GQW) is devoted to the topic: "Construction of algorithm and program realization of the solution of stochastic problem of inventory management". A bachelor of the Department of Applied Mathematics and Computer Science of Togliatti State University Kevorkova Anastasia Konstantinovna performed the work.

The purpose of the bachelor's work is to develop algorithms and create a software product that can help companies optimize their inventories, taking into account random factors.

The objectives of the bachelor's work are as follows:

- study existing mathematical models of inventory management processes; study existing mathematical methods for optimizing inventory management processes;
- build mathematical models of stochastic inventory management problems with deferred demand and lost demand;
- develop an algorithm for solving stochastic problems using mathematical methods;
- perform program implementation of the developed algorithms;
- perform testing and analyze the obtained results.

The structure of the work includes an introduction, three sections, conclusion, and a list of references. The first section describes mathematical models of inventory management processes, analyzes the mathematical methods used to solve inventory management problems, and outlines the theoretical basis for solving stochastic problems with pent-up demand and lost demand. The second section provides an algorithm for solving stochastic inventory management problems with pent-up demand and lost demand. The third section presents the program implementation of the developed algorithms in the Python programming language.

## Содержание

Введение.....	5
1 Введение в управление запасами.....	7
1.1 Описание и классификация управления запасами .....	7
1.2 Классификация моделей управления запасами .....	8
1.3 Стохастические модели управления запасами.....	10
1.3.1 Модели с учетом потерь от дефицита.....	10
1.3.2 Модели с потерянным спросом .....	13
2 Постановка задачи.....	17
2.1 Модель и формулировка задачи с отложенным спросом .....	17
2.2 Модель и формулировка задачи с потерянным спросом .....	23
3 Разработка алгоритма и программная реализация решения стохастической задачи управления запасами .....	30
3.1 Выбор языка программирования и среды разработки .....	30
3.2 Разработка алгоритма .....	31
3.2.1 Модель с отложенным спросом.....	31
3.2.2 Модель с потерянным спросом.....	39
Заключение .....	47
Список используемой литературы и используемых источников.....	48

## Введение

В современном бизнесе эффективное управление запасами играет важную роль в обеспечении стабильности операции и конкурентоспособности компании. Высокая сложность этой задачи возрастает при учете стохастических факторов: неопределенности в спросе, временных колебаний и других случайных воздействий на процесс управления запасами. В связи с этим, создание и внедрение эффективных методик и программного обеспечения для обработки стохастических задач в области управления запасами приобретают особую актуальность при решении практических задач.

В рамках выпускной квалификационной работы рассматривается построение алгоритма, позволяющего эффективно справляться с неопределенностями и случайностями при формировании стратегий управления ассортиментом. Кроме того, в работе осуществляется программная реализация метода стохастического моделирования, позволяющая усовершенствовать процесс оптимизации управления запасами в условиях стохастической неопределенности.

Целью бакалаврской работы является разработка алгоритма и создание программного продукта, способного помочь компаниям оптимизировать свои запасы, учитывая случайные факторы.

Задачи бакалаврской работы состоят в следующем:

- изучить существующие математические модели процессов управления запасами;
- рассмотреть существующие математические методы оптимизации процессов управления запасам;
- построить математические модели стохастических задач управления запасами с отложенным спросом и потерянным спросом;
- разработать алгоритм для решения стохастических задач управления запасами с отложенным спросом и потерянным спросом с применением математических методов;

- выполнить программную реализацию разработанных алгоритмов;
- выполнить тестирование и анализ полученных результатов.

Предполагается, что результаты, полученные в ходе этой работы, будут иметь практическое применение, а также помогут повысить эффективность процессов управления на предприятии.

Структура работы включает в себя введение, три раздела, заключение, список литературы.

В первом разделе описываются математические модели процессов управления запасами, анализируются математические методы, применяемые для решения задач управления запасами, излагаются теоретические основы для решения стохастических задач с отложенным спросом и потерянным спросом.

Во втором разделе приводится алгоритм решения стохастических задач управления запасами с отложенным спросом и потерянным спросом.

В третьем разделе представлена программная реализация разработанных алгоритмов на языке программирования Python.

## **1 Введение в управление запасами**

### **1.1 Описание и классификация управления запасами**

Управление запасами – ключевая функция логистики, играющая важную роль в обеспечении эффективного взаимодействия между производством, поставками и потреблением товаров. Главная задача управления запасами – поддержание оптимального уровня запасов, чтобы удовлетворить потребности клиентов при минимизации совокупных издержек. В условиях современного рынка, где конкуренция постоянно растет, управление запасами становится особенно значимым. Это позволяет компаниям снижать затраты, повышать качество обслуживания и укреплять свои конкурентные позиции. Успешное управление запасами способствует более гладкой и эффективной работе всей снабженческо-сбытовой цепи, что в конечном итоге ведет к улучшению финансовых результатов предприятия.

В управлении запасами различают несколько основных типов, каждый из которых выполняет определенные функции в цепочке поставок.

Текущие (регулярные) запасы: обеспечивают непрерывность производственного процесса и удовлетворение текущего спроса. Такие запасы формируются из-за несовпадения сроков и объемов поступления и потребления материальных ресурсов.

Страховые (буферные) запасы: используются для компенсации случайных колебаний спроса и предложения. Они помогают поддерживать непрерывность производственного или торгового процесса в случае неточного прогноза спроса или нарушении поставок.

Сезонные запасы: создаются с целью обеспечения нормального функционирования предприятия в периоды сезонных колебаний производства или потребления.

Спекулятивные запасы: создаются в ожидании изменения рыночных условий, таких как увеличение цен или дефицит товаров.

Перечислим основные задачи управления запасами.

Определение оптимального уровня запасов: Это включает в себя расчет необходимых объемов для каждого вида запасов с учетом специфики предприятия и рынка.

Разработка и внедрение систем контроля: Создание эффективных систем мониторинга и управления запасами для обеспечения своевременного пополнения и минимизации издержек.

Анализ и оптимизация процессов управления запасами: Постоянный анализ данных о запасах для выявления возможностей их оптимизации и снижения затрат.

## **1.2. Классификация моделей управления запасами**

Дадим классификацию моделям управления запасами.

Детерминированные и стохастические модели. Детерминированные модели – такие модели, в которых все параметры системы (спрос, время поставки, затраты) известны и остаются неизменными. Они часто применяются в условиях стабильного рынка. Стохастические модели – такие модели, в которых учитывают случайные колебания параметров системы, такие как изменчивость спроса и время поставки. Они являются более сложными и применяются в условиях неопределенности.

Статические и динамические модели. Статические модели – такие модели, в которых неизменность параметров системы с течением времени. Эти модели просты в реализации и применяются для анализа краткосрочных решений. Динамические модели – такие модели, в которых учитывают изменение параметров системы с течением времени и применяются для долгосрочного планирования.

Модели с учетом дефицита и без дефицита. Модели с учетом дефицита – такие модели, в которых возможно временное отсутствие товара на складе, и рассматриваются связанные с этим издержки. Модели без дефицита – такие

модели, в которых стремятся полностью исключить ситуации дефицита товара.

Перечислим некоторые из методов классификации запасов.

ABC-анализ. ABC-анализ – это метод классификации запасов, основанный на принципе Парето, который утверждает, что меньшая часть объектов (20%) обеспечивает большую часть результатов (80%). Этот метод широко используется для определения приоритетности управления различными категориями запасов.

ABC-анализ делит запасы на три А, В, С.

Группа А включает товары с наибольшей стоимостью и значимостью для предприятия, хотя они могут составлять небольшую часть общего количества товаров. Эти товары требуют тщательного контроля и частого мониторинга.

Группа В содержит товары средней значимости и стоимости, требующие умеренного уровня контроля.

Группа С включает товары с наименьшей стоимостью, которые составляют большую часть общего количества и требуют минимального контроля.

Метод ABC-анализа помогает предприятиям концентрировать усилия на наиболее значимых позициях запасов, улучшая управление и снижая издержки.

XYZ-анализ дополняет ABC-анализ, классифицируя товары по степени предсказуемости спроса:

Группа Х – это товары с равномерным, стабильным спросом, которые легко прогнозируются. Обычно это материалы или продукты, необходимые для непрерывного производства.

Группа Y – это товары с колеблющимся спросом, часто имеющие сезонные или трендовые изменения. Прогнозирование спроса для этих товаров требует дополнительных усилий.

Группа  $Z$  – это товары с эпизодическим и непредсказуемым спросом, прогнозирование которого затруднено. Обычно это товары с нерегулярным использованием.

XYZ-анализ помогает предприятиям лучше понимать структуру спроса на их товары и разрабатывать более эффективные стратегии управления запасами.

### **1.3. Стохастические модели управления запасами**

В работе подробно рассматриваются стохастические модели управления запасами. Такие модели основаны на том, что основные параметры систем управления запасами являются случайными величинами. Распределение этих параметров управления запасами подчинено, как правило, нормальному закону Гаусса или экспоненциальному закону.

Подробно рассмотрим следующие стохастические модели управления запасами:

- модели с учетом потерь от дефицита,
- модели с потерянным спросом.

#### **1.3.1 Модели с учетом потерь от дефицита**

Модели экономического размера заказа с учетом дефицита (EOQ with backorders) важны для предприятий, которые могут временно допустить дефицит товара, откладывая неудовлетворенный спрос на будущее. Эти модели помогают минимизировать совокупные издержки, включая затраты на размещение заказа, хранение и дефицит.

Основными допущениями модели являются:

- фиксированная задержка пополнения, где поставка товара осуществляется с фиксированной задержкой.
- затраты на упущенный спрос, где включают упущенную прибыль и затраты на привлечение новых клиентов.

В моделях с отложенным спросом, учитывается дефицит, возникающий во время поставки, при определении размера партии. Это позволяет удовлетворить текущий спрос, а также ранее заявленный, но неудовлетворенный.

Модели задач с отложенным спросом включают в себя ряд аспектов.

1. Спрос: интерес к продукту может возникнуть позже, когда покупатель примет решение о его покупке в будущем, когда это будет необходимо, при этом запрос зависит от времени, цены товара, сезонности и других факторов.

2. Время задержки: промежуток времени, который требуется для превращения отложенного запроса в реальный спрос на товар, зависит оно может от различных факторов, таких как время доставки, производственные циклы и другие.

3. Расходы на сохранение запасов: затраты, связанные с хранением товаров в ожидании будущего спроса, которые включают в себя затраты на хранение, страхование, потери от порчи или устаревания и т.д.

4. Управление запасами: определение оптимального уровня запасов и стратегии их управления в контексте отложенного спроса, которое включает в себя принятие решений о том, сколько товаров следует закупить сейчас и сколько оставить на будущее, чтобы снизить расходы на сохранение запасов и одновременно удовлетворить спрос.

Основной подход к решению проблем с накопленным спросом заключается в создании математических моделей и алгоритмов для оптимизации, которые помогают выявить наилучшие тактики управления запасами при наличии отсроченного спроса и прочих ограничений. Такие модели становятся незаменимыми инструментами во множестве сфер, включая ритейл, производственные процессы, логистику и прочие.

Математическая формулировка модели с учетом потерь от дефицита:

$$C(Q) = \frac{K*d}{Q} + \frac{h*Q}{2} \left(1 - \frac{d}{Q}\right) + \frac{p*d}{Q} \left(1 - \frac{Q}{d}\right), \quad (1)$$

где  $K$  – фиксированные затраты на заказ;

$d$  – интенсивность спроса;

$Q$  – размер заказа;

$h$  – удельные затраты на хранение;

$p$  – затраты от дефицита.

Из формулы (1) получаем оптимальный размер заказа

$$Q^* = \sqrt{\frac{2Kd}{h\left(1 - \frac{d}{Q}\right) + p}}, \quad (2)$$

где  $Q^*$  – размер производственного заказа, минимизирующий общие издержки в единицу времени;

$K$  – фиксированные затраты на заказ;

$d$  – интенсивность спроса;

$Q$  – размер заказа;

$h$  – удельные затраты на хранение;

$p$  – затраты от дефицита.

Модель с отложенным спросом представим на рисунке 1.

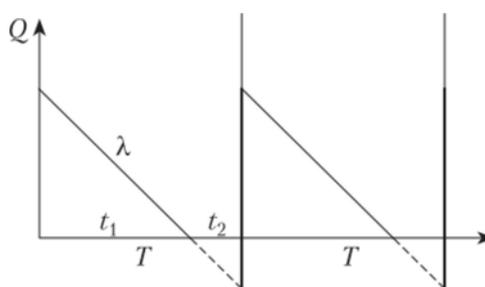


Рисунок 1 – Модель с отложенным спросом

На рисунке (1) изображено изменение количества продукта на складе за суммарное время  $t_x$  и  $t_2$ .

Каждый цикл состоит из двух этапов -  $t_x$  и  $t_2$ . В момент времени  $t_x$  происходит расход текущего заказа со скоростью  $X$ , а  $t_2$  - это период дефицита, когда товар отсутствует на складе, но поток заявок не уменьшается и продолжают поступать с той же интенсивностью  $X$ . При получении нового заказа для пополнения запасов, часть его мгновенно удовлетворяет заявки из дефицитного периода, а оставшаяся часть добавляется к текущему запасу, также мгновенно поступая на склад.

В моделях с дефицитом, заказ должен быть размещен на складе в момент, когда уровень дефицита (отрицательный запас) достигает максимального значения. Точка заказа может быть задана в отрицательных значениях.

### **1.3.2. Модели с потерянными спросом**

Модели с потерянными спросом (Lost Sales) актуальны для ситуаций, когда неудовлетворенный спрос не переносится на будущее, а теряется навсегда. Это критически важно для рынков с высокой конкуренцией, где потеря продаж может существенно повлиять на прибыльность.

Основными допущениями моделей являются:

- фиксированная задержка пополнения, где поставка товара осуществляется с фиксированной задержкой;
- затраты на упущенный спрос, где включают упущенную прибыль и затраты на привлечение новых клиентов.

В моделях оптимального размера заказа с потерянными спросом, дефицит рассматривается как невозможность удовлетворить заявки на отгрузку товаров. Клиентам отказывают, а восполнение запаса происходит в прежних размерах.

Модели задач с потерянными спросом включают в себя ряд аспектов.

1. Спрос: количество товара или услуги, которое потребители готовы купить или заказать в определенный период времени, при этом спрос может быть зависеть от цены, рекламы, сезонности и других факторов.

2. Потерянный спрос: количество спроса, которое не может быть удовлетворено из-за недостаточного предложения или невозможности удовлетворить спрос вовремя, что может привести к упущенным возможностям для бизнеса и потере дохода.

3. Последствия потерянного спроса: потерянный спрос может иметь различные последствия, такие как: потеря клиентов, снижение дохода, ущерб репутации бренда, из-за чего важно разработать стратегии управления запасами и производственными мощностями, чтобы минимизировать риск потери спроса.

4. Управление риском потерянного спроса: разработка стратегий и тактик, направленных на снижение риска потери спроса, таких как увеличение запасов, улучшение процессов производства и логистики, разработка альтернативных поставщиков или технологий и др.

Модели с потерянным спросом широко применяется в управлении запасами, логистике, маркетинге и других сферах бизнеса. Эффективное управление риском потерянного спроса помогает компаниям минимизировать потери и повышать уровень обслуживания клиентов.

Математическая формулировка модели с потерянным спросом:

$$C(Q) = \frac{K*d}{Q} + \frac{h*Q}{2} + \frac{b*d}{Q} \left(1 - \frac{Q}{d}\right), \quad (3)$$

где  $K$  – фиксированные затраты на заказ;

$d$  – интенсивность спроса;

$Q$  – размер заказа;

$b$  – затраты на упущенный спрос.

Из формулы (3) получаем оптимальный размер заказа:

$$Q^* = \sqrt{\frac{2Kd}{h+b}}, \quad (4)$$

где  $Q^*$  – размер производственного заказа, минимизирующий общие издержки единицу времени, равные сумме издержек хранения и издержек на возобновление производства;

$K$  – фиксированные затраты на заказ;

$d$  – интенсивность спроса;

$b$  – затраты на упущенный спрос.

Модель с потерянными спросом представим на рисунке 2.

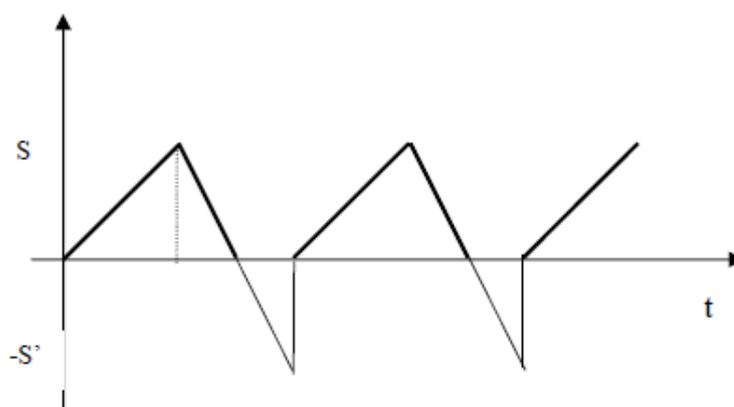


Рисунок 2 – Модель с потерянными спросом

На рисунке (2) изображена модель с потерянными спросом при условии постепенного пополнения.

Здесь  $S$  – будет означать максимальный уровень запаса на складе, а  $S'$  – максимальный уровень дефицита.

В моделях с потерянными спросом, уровень запаса не может быть отрицательным.

Модели управления запасами, учитывающие потери от дефицита и потерянный спрос, имеют критическое значение для эффективного

управления запасами, особенно в условиях высокой неопределенности спроса. Эти модели позволяют компаниям максимально точно оценивать и минимизировать риски, связанные с неудовлетворенным спросом, что в свою очередь помогает сохранить конкурентоспособность, удерживать клиентов и улучшать финансовые показатели.

Применение этих моделей требует от компаний не только глубокого понимания своих внутренних процессов, но и осведомленности о рыночных тенденциях. Аналитические данные и передовые технологии в области прогнозирования и анализа данных могут значительно улучшить точность прогнозов спроса, что позволяет более эффективно планировать запасы и снижать издержки.

В современной экономике, где скорость реакции и адаптивность играют ключевую роль, модели управления запасами, ориентированные на минимизацию потерь от дефицита и потерянного спроса, являются неотъемлемой частью стратегического планирования любого предприятия, стремящегося к эффективности и устойчивому развитию.

## 2 Постановка задачи

### 2.1 Модель и формулировка задачи с отложенным спросом

Сформулируем задачу с отложенным спросом.

Компания занимается производством деталей из системы транспортирования. Пусть каждый месяц используется –  $k$  гибких тросов в производственном процессе. Цена каждого троса, который будет поставлен на предприятие составляет –  $d$  тысяч рублей. Стоимость хранения одного троса –  $h$  тысяч рублей. Удельные потери прибыли от дефицита тросов будет оцениваться –  $b$  за штуку. Во время поставки спрос на тросы будет случайной величиной, которая будет распределена от 0 до  $p$  штук. Нужно найти оптимальную стратегию управления запасами.

Необходимо найти оптимальную стратегию управления запасами. Построим математическую модель задачи с отложенным спросом. Введем обозначения:

$m$  – число гибких тросов, используемых в партии в производственном процессе;

$e$  (д. е.) – затраты предприятия на заказ поставки партии тросов;

$c$  (д. е.) – стоимость хранения одного троса в месяц;

$n$  (д. е.) - удельные потери прибыли от дефицита тросов за штуку;

Для решения данной задачи требуется учитывать следующие факторы:

$d$  – (е.г) отражает интенсивность спроса;

$K$  – (е.г) соответствует объему заказа;

$h$  (е.г) – определяет стоимость хранения единицы товара за единицу времени.

Известно, что издержки, связанные с отсутствием товара, пропорциональны ожидаемому уровню дефицита.

$b$  (е.г) – обозначает потери от отсутствия единицы товара;

$\mu$  (е.г) – представляет собой ожидаемый спрос за весь период поставки;

$r$  – точка возобновления (восстановления) заказа;

$p$  – постоянная интенсивность (темп).

Перейдем к построению математической модели задачи.

Спрос подчиняется нормальному закону распределения.

Пусть  $f(x, t)$  – плотность распределения спроса за период  $t$ . Следовательно, плотность безусловного распределения спроса за время выполнения заказа будет вычисляться по формуле (5).

$$D_l(x) = \int_0^{\infty} f(x, l)g(l)dl, \quad (5)$$

где  $D_l(x)$  – функция плотности распределения спроса за время выполнения заказа;

$l$  – случайная величина с плотностью распределения  $g(l)$ ;

$f(x, l)$  – плотность распределения за период  $t$ .

Так как время выполнения одного заказа постоянно, то будем считать, что:

$$D_l(x) = f(x, l) = \frac{x}{p}, x \in [0, p], \quad (6)$$

$$\mu = \int_0^{\infty} xD_l(x)dx = \int_0^p \frac{xdx}{p}. \quad (7)$$

Формула (6) описывает линейное распределение спроса на товар в интервале от 0 до  $p$ . Формула (7) вычисляет среднее значение спроса  $\mu$ , когда спрос в период поставки считается равномерно распределённым от 0 до  $p$ .

Пусть  $r \leq p$ , тогда:

$$D_l(x)dx = \int_r^{\infty} f(x, l)g(l)dl = \int_r^p \frac{dx}{p} = 1 - \frac{r}{p}, \quad (8)$$

$$\int_r^{\infty} xD_l(x)dx = \int_r^p \frac{xdx}{p}. \quad (9)$$

По формуле (8) вычисляем вероятность того, что спрос за время поставки будет больше или равен заданному уровню  $r$ . Ожидаемое значение спроса  $x$  для значений, превышающих порог  $r$  будет вычисляться по формуле (9).

Для вычисления ожидаемого дефицита воспользуемся формулой (10).

$$\bar{B} = \int_r^{\infty} (x - r) D_l(x) dx = \int_r^p \frac{x dx}{p} - r \left(1 - \frac{r}{p}\right). \quad (10)$$

Оптимальный размер заказа с учетом ожидаемого спроса и дефицита будет вычисляться по формуле (11). Формула (12) вычисляет разницу между ожидаемым спросом и фактическим спросом.

$$Q^* = \sqrt{m(e + n\bar{B})}, \quad (11)$$

$$e - r^* = \frac{Q^*}{\int_0^p \frac{px dx}{p}}, \quad (12)$$

где  $Q^*$  - оптимальное значение заказа  $Q$ ;

$r^*$  - оптимальное значение  $r$ .

Так как

$$Q^*(0) = \sqrt{\frac{2d(K+b\mu)}{h}}, \quad (13)$$

$$Q_1^*(0) = \frac{bd}{h}. \quad (14)$$

По формуле (13) вычисляем оптимальный размер заказа, который минимизирует суммарные издержки на размещение заказов и хранение

запасов. По формуле (14) вычисляем оптимальный размер дополнительного заказа, который минимизирует издержки на дефицит и хранение.

Применяем итерационную процедуру, предполагаем, что:

$$Q_1 = \sqrt{2Kd/h}. \quad (15)$$

Формула (15) вычисляет оптимальный размер заказа, который минимизирует общие затраты на размещение заказов и хранение запасов.

Далее вычисляем  $r_1$ , используя второе уравнение в системе, где  $Q^* = Q_1$ . Затем определяем  $Q_2$  из первого уравнения системы, принимая  $r^* = r_1, r_2$  из второго уравнения, взяв  $Q^* = Q_2$  и так далее. Процесс итераций будет продолжаться до тех пор, пока различия между последовательными значениями оптимального размера заказа  $Q$  и точки возобновления заказа  $r$  не станут достаточно малыми.

Рассмотрим эту задачу с конкретными значениями параметров.

Компания занимается производством деталей из системы транспортирования. Пусть каждый месяц используется 2500 гибких тросов в производственном процессе. Цена каждого троса, который будет поставлен на предприятие составляет 150 тысяч рублей. Стоимость хранения одного троса 5 тысяч рублей. Удельные потери прибыли от дефицита тросов будет оцениваться в 15 тысяч рублей за штуку. Во время поставки спрос на тросы будет случайной величиной, которая будет распределена от 0 до 150 штук. Нужно найти оптимальную стратегию управления запасами. Используя формулы (5) - (9), которые представлены в математической модели решим задачу:

$$D_l(x) = f(x, l) = \frac{x}{150}, x \in [0, 150],$$

$$\mu = \int_0^{\infty} xD_l(x)dx = \int_0^{150} \frac{xdx}{150} = 75.$$

Пусть  $r \leq 150$ , тогда:

$$D_l(x)dx = \int_r^{\infty} f(x, l)g(l)dl = \int_r^{150} \frac{dx}{150} = 1 - \frac{r}{150},$$

$$\int_r^{\infty} xD_l(x)dx = \int_r^{150} \frac{xdx}{150} = 75 - \frac{r^2}{300}.$$

Ожидаемый дефицит вычисляется по формуле (10) и будет равняться:

$$\bar{B} = \int_r^{\infty} (x - r)D_l(x)dx = 75 - \frac{r^2}{300} - r\left(1 - \frac{r}{150}\right) = 75 + \frac{r^2}{300} - r.$$

На основании формул (11) и (12) можно вычислить оптимальное значение заказа  $Q$  и оптимальное значение  $r$ :

$$Q^* = \sqrt{2500(150 + 15\bar{B})} = 50\sqrt{150 + 15\bar{B}},$$

$$150 - r^* = \frac{Q^*}{75}.$$

Используем формулы (13) и (14), чтобы вычислить оптимальный размер заказа и размер дополнительного заказа:

$$Q^*(0) = 100\sqrt{1275} = 3571,$$

$$Q_1^*(0) = \frac{bd}{h} = \frac{15 * 2500}{5} = 7500.$$

Далее применяем итерационную процедуру:

$$Q_1 = 100\sqrt{150} = 1224,$$

$$r_1 = 100 - \frac{1224}{75} = 83,68,$$

$$\bar{B} = 75 + \frac{83,68^2}{300} - 83,68 = 15,$$

$$Q_2 = 100\sqrt{165} = 1284,$$

$$r_2 = 100 - \frac{1284}{75} = 83.$$

Анализируя решение, сделаем выводы. Так как  $r_1$  и  $r_2$  отличаются незначительно, можно предположить  $r^* = r_2 \approx 84$ ,  $Q^* = Q_2 = 1284$ . В системе поддерживается страховой запас  $\bar{I} = \bar{B} + r - \mu = 24$ , ожидаемая продолжительность цикла около 15 дней.

$$\begin{aligned} \frac{Kd}{Q} + h\left(\frac{Q}{2} + r - \mu\right) + \frac{bd}{Q} \int_r^{\infty} (x - r)D_l(x)dx = \\ = \frac{2500 * 150}{1284} + 5\left(\frac{1284}{2} + 24\right) + \frac{15 * 150}{1284} * 15 = 3648,34. \end{aligned}$$

Ожидаемые издержки будут равны 3648,34 т. р.

## 2.2 Модель и формулировка задачи с потерянным спросом

Сформулируем задачу с потерянным спросом.

В компании есть определенное количество специализированных каркасов сидений, которые используются для поставок. Годовой спрос составляет –  $d$  штук. Себестоимость каждого каркаса составляет –  $m$ . Приобретение необходимого количества каркасов выходит компании на –  $k$  тысяч рублей. Коэффициент издержек с учетом содержания запасов, оценивается в размере –  $b$ . Компания имеет возможность приобрести недостающие каркасы из запасов одного из производителей за дополнительные –  $s$  тысяч рублей на транспортировку. Распределение спроса в момент поставки новой партии каркасов соответствует нормальному распределению с математическим ожиданием –  $e$  и средним квадратическим отклонением –  $\sigma$  каркасов.

Требуется провести анализ для определения оптимального объема заказа и определения точки, учитывая возможные потери спроса и необходимость дополнительных заказов для его удовлетворения.

Построим математическую модель задачи с потерянным спросом. Необходимо найти оптимальную стратегию управления запасами.

Введем обозначения:

$d$  (шт.) – годовой спрос;

$m$  (д.е.) – себестоимость каждого каркаса;

$k$  (шт.) – необходимое количество каркасов;

$b$  – коэффициент издержек;

$s$  (д.е.) – дополнительная сумма на получение недостающих каркасов;

$e$  – математическое ожидание;

$\sigma$  – среднее квадратическое отклонение;

$r$  – точка возобновления (восстановления) заказа;

Спрос  $z$  является случайной величиной, имеющий стандартное нормальное распределение. Для вычисления  $z$  воспользуемся формулой (16).

$$z = \frac{D-e}{\sigma}, \quad (16)$$

Для вычисления оптимального размера заказа  $Q^*$  и точки восстановления запаса  $r^*$  воспользуемся формулам (17) и (18).

$$Q^* = \sqrt{2d(K + b \bar{B})/h}, \quad (17)$$

$$P(D < r^*) \equiv \int_{r^*}^{\infty} D_l(x) dx = \frac{hQ^*}{bd+hQ^*}, \quad (18)$$

где  $Q^*$  – оптимальное значение заказа  $Q$ ;

$P(D < r^*)$  – вероятность того, что спрос будет меньше  $r^*$ ;

$r^*$  – оптимальное значение  $r$ .

Ожидаемый дефицит вычислим по формуле (19).

$$\bar{B} = h \varphi(z_r) + (e - r)(-F(z_r)). \quad (19)$$

Используя, формулу (20) вычисляем оптимальный размер заказа. По формуле (21) вычисляем функцию распределения нормальной случайной величины.

$$Q^* = \sqrt{\frac{2d(K+b \bar{B})}{h}}, \quad (20)$$

$$F(z_r^*) = \frac{hQ^*}{bd+hQ^*}, \quad (21)$$

где  $z_r^*$  – критическое значение нормальной случайной величины;

$F(z_r^*)$  – функция распределения нормальной случайной величины.

Функция  $F(z_r^*)$  определяет вероятность того, что спрос не превысит уровень запаса  $z_r$ .

Применяем итерационный алгоритм:

$$Q_1 = \sqrt{\frac{2Kd}{h}}, \quad (22)$$

$$F(z_{r_1}) = \frac{hQ_1}{bd+hQ_1} = \frac{h\sqrt{\frac{2Kd}{h}}}{bd+h\sqrt{\frac{2Kd}{h}}}, \quad (23)$$

где  $Q_1$  – оптимальный размер заказа, который минимизирует общие затраты на размещение заказов и хранение запасов;

$F(z_r)$  – функция распределения нормальной случайной величины.

Далее считаем значение нормальной случайной величины  $z_{r_1}$  и точку возобновления заказа  $r_1$  соответственно.

После этого вычислим ожидаемый дефицит.

Итерации будут продолжаться до тех пор, пока разности между двумя последовательными значениями не станут достаточно малыми. Поскольку уровень запаса не может быть отрицательным, средний уровень запасов приблизительно равен:

$$(\bar{I} + Q + \bar{I})/2 = Q/2 + r - \mu + \bar{B}. \quad (24)$$

Для расчета ожидаемых годовых издержек будет использоваться следующая формула:

$$C = \frac{Kd}{Q} + h \left( \frac{Q}{2} + r - \mu \right) + \left( h + \frac{bd}{Q} \right) \int_r^\infty (x - r) D_l(x) dx. \quad (25)$$

Теперь рассмотрим эту задачу с конкретными значениями параметров.

В компании есть определенное количество специализированных каркасов сидений, которые используются для поставок. Годовой спрос составляет около 2800 штук. Себестоимость каждого каркаса составляет 50 тысяч рублей. Приобретение необходимого количества каркасов выходит компании на 600 тысяч рублей. Учитывая коэффициент издержек содержания

запасов, оцененный в размере 0,4, компания имеет возможность приобрести недостающие каркасы из запасов одного из производителей за дополнительные 400 тысяч рублей на транспортировку. Распределение спроса в момент поставки новой партии каркасов соответствует нормальному распределению с математическим ожиданием в 900 каркасов и средним квадратическим отклонением 50 каркасов.

Требуется провести анализ для определения оптимального объема заказа и определения точки, учитывая возможные потери спроса и необходимость дополнительных заказов для его удовлетворения.

Считаем годовые издержки хранения за единицу:

$$h = b * m = 0,4 * 50000 = 20000 \text{ рублей}$$

Случайная величина спроса  $z = \frac{D-900}{50}$  имеет стандартное нормальное распределение, следовательно:

$$\int_r^{\infty} D_l(x) dx = 1 - P(D < r) = 1 - P(z < z_r) = 1 - F(z_r),$$

где  $D_l(x)$  – функция плотности вероятности спроса  $x$  за период времени  $l$ ;

$$z_r = \frac{r-900}{50};$$

$z_r$  – случайная величина уровня запаса  $r$ ;

$F(z_r)$  – функция распределения нормальной случайной величины.

$$F(z) = \int_{-\infty}^z \varphi(\varepsilon) d\varepsilon = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z \exp\left(-\frac{\varepsilon^2}{2}\right) d\varepsilon,$$

где  $\varphi(\varepsilon)$  – плотность вероятности стандартного нормального распределения.

Далее вычисляем ожидаемое значение  $z$  для значений, превышающих порог  $z_r$ , в стандартном нормальном распределении:

$$\int_{z_r}^{\infty} z\varphi(z)dz = - \int_{z_r}^{\infty} \varphi'(z)dz = \varphi(z_r).$$

Вычисляем ожидаемое значение спроса  $x$  для  $x > r$ , используя стандартное нормальное распределение:

$$\int_r^{\infty} D_l(x)dx = \frac{1}{50} \int_r^{\infty} x\varphi\left(\frac{x-900}{50}\right)dx = 75\varphi(z_r) + 900(1 - F(z_r)).$$

Ожидаемый дефицит вычисляем по формуле (19).

$$\bar{B} = 50 \varphi(z_r) + (900 - r)(-F(z_r)).$$

На основании формул (20) и (21) можно вычислить оптимальное значение:

$$Q^* = \sqrt{2d(K + b\bar{B})/h} = \sqrt{2 * 2800\left(\frac{600000 + 400000}{20000}\right)} = 40\sqrt{(105 + 70\bar{B})},$$

$$F(z_r^*) = \frac{hQ^*}{bd + hQ^*} = \frac{20000Q^*}{400000 * 2800 + 20000Q^*} = \frac{56000}{56000 + Q^*}.$$

Применяем итерационный алгоритм:

$$Q_1 = \sqrt{\frac{2*2800*600000}{20000}} = \sqrt{168000} \approx 410 \text{ каркасов},$$

$$F(z_{r_1}) = \frac{56000}{56000 + 410} = 0,993.$$

Используем таблицу стандартного нормального распределения, где  $F(z_{r_1}) = 0,993$   $z_{r_1} = 2,5$ ,

$$r_1 \approx 2,5 * 50 + 900 = 1025,$$

$$\bar{B} = \frac{50}{\sqrt{2\pi}} \exp\left(-\frac{2,5^2}{2}\right) + (900 - 1025) * (1 - 0,993) \approx 0,014.$$

Проведем еще один итерационный алгоритм:

$$Q_2 = 40 \sqrt{(105 + 70\bar{B})} = 40 \sqrt{(105 + 70 * 0,014)} = 412,$$

$$F(z_{r_2}) = \frac{56000}{56000 + 412} = 0,992,$$

Используем таблицу стандартного нормального распределения, где  $F(z_{r_1}) = 0,99$ ,

$$z_{r_1} = 2,45,$$

$$r_2 \approx 2,45 * 50 + 900 = 1026,$$

$$\bar{B} = \frac{50}{\sqrt{2\pi}} \exp\left(-\frac{2,45^2}{2}\right) + (900 - 1026) * (1 - 0,992) \approx 0,008.$$

Проведем еще один итерационный алгоритм:

$$Q_2 = 40\sqrt{(105 + 70\bar{B})} = 40\sqrt{(105 + 70 * 0,008)} = 411,$$

$$F(z_{r_3}) = \frac{56000}{56000 + 411} = 0,992.$$

Используем таблицу стандартного нормального распределения, где  $F(z_{r_1}) = 0,992$ ,  $z_{r_1} = 2,45$ ,  $r_2 \approx 2,45 * 50 + 900 = 1026$ ,

$$\bar{B} = \frac{50}{\sqrt{2\pi}} \exp\left(-\frac{2,45^2}{2}\right) + (900 - 1026) * (1 - 0,992) \approx 0,008.$$

$$F(z_{r_3}) \approx F(z_{r_2}), \text{ поэтому } r_3 = r_2 = r^*, Q^* = Q_3.$$

Исходя из вышеперечисленных решений, можно вычислить гарантийный запас:

$$r - e = 1026 - 900 = 126.$$

А ожидаемые годовые издержки будут рассчитываться по формуле (24) и составят:

$$C = \frac{600000 * 2800}{411} + 20000 \left( \frac{411}{2} + 1026 - 900 \right) + \left( 20000 + \frac{400000 * 2800}{411} \right) * 0,008 \approx 10739551,73.$$

Таким образом, ожидаемые годовые издержки равны 10739551,73 т. р.

### **3. Разработка алгоритма и программная реализация решения стохастической задачи управления запасами**

#### **3.1 Выбор языка программирования и среды разработки**

Разработка алгоритма и его программная реализация для решения стохастической задачи управления запасами является важным этапом, позволяющим эффективно справляться с неопределенностью спроса и других параметров. В данной главе будет описан процесс создания алгоритмов управления запасами для двух различных моделей: модель с отложенным спросом и модель с потерянными спросом. Для реализации используется язык программирования Python и редактор кода Visual Studio Code (VS Code). Для реализации стохастической модели управления запасами был выбран язык программирования Python.

Перечислим основные причины выбора Python.

Простота и читаемость синтаксиса, что облегчает разработку и поддержку кода.

Наличие богатого набора библиотек для математических и статистических вычислений, таких как NumPy, SciPy, Matplotlib и Pandas.

Сообщество и документация, которые предоставляют множество примеров и решений.

Visual Studio Code (VS Code) был выбран в качестве текстового редактора по следующим причинам:

- бесплатность и открытый исходный код;
- многочисленные расширения;
- интеграция с терминалом;
- высокая производительность.

## 3.2 Разработка алгоритма

Стохастическая задача управления запасами заключается в оптимизации уровня запасов с учетом неопределенности спроса и времени поставки.

Основные параметры модели включают:

- средний спрос ( $\mu$ ) и его стандартное отклонение ( $\sigma$ ),
- время поставки ( $L$ ) и его стандартное отклонение,
- стоимость хранения ( $h$ ) и стоимость заказа ( $K$ ),
- штраф за дефицит ( $b$ ).

Целью является минимизация общих затрат, включающих затраты на хранение, размещение заказов и дефицит.

Алгоритмы решения задач включают следующие шаги:

- инициализация параметров модели;
- генерация случайного спроса на основе нормального распределения;
- вычисление уровня запасов и соответствующих затрат;
- оптимизация уровня заказа для минимизации совокупных затрат;

### 3.2.1 Модель с отложенным спросом

Модель с отложенным спросом предполагает, что неудовлетворенный спрос переносится на будущее. Это означает, что компания будет выполнять заказы, даже если это требует времени.

Исходя из данных условий и значений, напишем код для решения задачи управления запасами с моделью отложенного спроса.

Этот код представляет собой реализацию модели отложенного спроса с графическим интерфейсом на Python, используя библиотеки NumPy, Matplotlib, SciPy, StatsModels и Tkinter.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.stats import norm
4 from statsmodels.tsa.arima.model import ARIMA
5 import tkinter as tk
6 from tkinter import ttk
7
8 class DelayedDemandModel:
9     def __init__(self, d, K, h, b, μ, historical_data=None):
10         self.d = d # Интенсивность спроса
11         self.K = K # Фиксированные затраты на заказ
12         self.h = h # Удельные затраты на хранение
13         self.b = b # Удельные потери прибыли от дефицита
14         self.μ = μ # Математическое ожидание спроса за весь период поставки
15         self.historical_data = historical_data

```

Рисунок 3 – Класс DelayedDemandModel, Конструктор `_Init_`

На рисунке 3 используется класс `DelayedDemandModel`, который реализует модель отложенного спроса, предоставляя методы для прогнозирования спроса, расчета оптимального размера заказа, анализа чувствительности и визуализации результатов.

Далее представлен Конструктор `_Init_`, с помощью которого инициализируется модель с заданными параметрами.

```

17 def forecast_demand(self):
18     if self.historical_data is not None:
19         model = ARIMA(self.historical_data, order=(5, 1, 0))
20         model_fit = model.fit()
21         forecast = model_fit.forecast(steps=12)
22         self.forecasted_demand = forecast
23     else:
24         self.forecasted_demand = np.full(12, self.μ)
25
26 def calculate_optimal_order(self):
27     Q0 = np.sqrt((2 * self.d * self.K) / (self.h + self.b)) # Начальный размер заказа без учета дефицита
28     Q1 = np.sqrt((2 * self.d * self.K) / self.h) # Начальный размер заказа с учетом хранения
29     r1 = self.μ - Q1 / 2 # Уровень заказа при минимальном запасе
30
31     B = (self.μ - r1) * (1 - (r1 / self.μ))
32
33     self.Q_opt = Q0 + B
34     self.r_opt = r1 + B
35
36     self.C = self.K * self.d / self.Q_opt + self.h * (self.Q_opt ** 2) / 2 + self.b * B

```

Рисунок 4 – Методы `forecast_demand` и `calculate_optimal_order`.

На рисунке 4 представлен фрагмент кода, где используется метод `forecast_demand`, который использует модель `ARIMA` для прогнозирования

спроса на 12 месяцев вперед, если имеются исторические данные. В противном случае используется среднее значение спроса  $\mu$ .

Метод `calculate_optimal_order` рассчитывает оптимальный размер заказа ( $Q_{opt}$ ) и уровень дефицита ( $r_{opt}$ ), а также ожидаемые издержки ( $C$ ), используя заданные параметры модели.

```
38 def sensitivity_analysis(self, param_range, param_name):
39     results = []
40     original_value = getattr(self, param_name)
41     for value in param_range:
42         setattr(self, param_name, value)
43         self.calculate_optimal_order()
44         results.append((value, self.Q_opt, self.r_opt, self.C))
45     setattr(self, param_name, original_value)
46     return results
47
48 def display_results(self):
49     return f"Оптимальный размер заказа: {self.Q_opt:.2f}\nОжидаемый уровень дефицита: {self.r_opt:.2f}\nОжидаемые издержки: {self.C:.2f}"
```

Рисунок 5 – Методы `sensitivity_analysis` и `display_results`.

На рисунке 5 представлен метод `sensitivity_analysis`, который выполняет анализ чувствительности для заданного параметра модели в определенном диапазоне значений, возвращая список результатов (значение параметра, оптимальный размер заказа, уровень дефицита, ожидаемые издержки). А также метод `display_results` с помощью которого возвращается строка с результатами оптимизации: оптимальный размер заказа, ожидаемый уровень дефицита и ожидаемые издержки.

На рисунках 6-8 представлены методы визуализации `plot_inventory_levels`, `plot_forecasted_demand` и `plot_sensitivity_analysis`. Данные методы создают графики:

- уровня запасов с отложенным спросом;
- прогнозированного спроса на основе исторических данных;
- анализа чувствительности для заданного параметра.

```

51 def plot_inventory_levels(self):
52     time_period = np.linspace(0, 12, 100)
53     inventory_levels = self.Q_opt * np.sin(time_period) + self.r_opt
54     plt.plot(time_period, inventory_levels, label='Уровень запасов')
55     plt.axhline(y=0, color='r', linestyle='--', label='Уровень дефицита')
56     plt.xlabel('Время (месяцы)')
57     plt.ylabel('Уровень запасов')
58     plt.title('Уровень запасов с отложенным спросом')
59     plt.legend()
60     plt.grid(True)
61     plt.show()
62
63 def plot_forecasted_demand(self):
64     if self.historical_data is not None:
65         plt.plot(self.historical_data, label='Исторические данные')
66     plt.plot(np.arange(len(self.historical_data) + len(self.forecasted_demand)),
67             self.forecasted_demand, label='Прогнозированный спрос', linestyle='--')
68     plt.xlabel('Время')
69     plt.ylabel('Спрос')
70     plt.title('Прогноз спроса')
71     plt.legend()
72     plt.grid(True)
73     plt.show()

```

Рисунок 6 – Методы визуализации plot\_inventory\_levels и plot\_forecasted\_demand

```

75 def plot_sensitivity_analysis(self, results, param_name):
76     param_values, Q_opts, r_opts, costs = zip(*results)
77     plt.figure(figsize=(10, 6))
78     plt.subplot(3, 1, 1)
79     plt.plot(param_values, Q_opts, marker='o')
80     plt.xlabel(param_name)
81     plt.ylabel('Оптимальный размер заказа (Q_opt)')
82     plt.grid(True)
83
84     plt.subplot(3, 1, 2)
85     plt.plot(param_values, r_opts, marker='o')
86     plt.xlabel(param_name)
87     plt.ylabel('Ожидаемый уровень дефицита (r_opt)')
88     plt.grid(True)
89
90     plt.subplot(3, 1, 3)
91     plt.plot(param_values, costs, marker='o')
92     plt.xlabel(param_name)
93     plt.ylabel('Ожидаемые издержки (C)')
94     plt.grid(True)
95
96     plt.tight_layout()
97     plt.show()

```

Рисунок 7 – Метод визуализации plot\_sensitivity\_analysis

```

99 class App(tk.Tk):
100     def __init__(self):
101         super().__init__()
102         self.title("Delayed Demand Model")
103         self.geometry("400x300")
104
105         # Labels and entries for parameters
106         self.create_widgets()
107
108         self.model = None

```

Рисунок 8 – Класс App и конструктор `_Init_`

Далее представлен класс App, который реализует графический интерфейс приложения с использованием Tkinter. Конструктор `_Init_` инициализирует главное окно приложения с заголовком и размером, а также вызывает метод для создания виджетов.

На рисунках 9-10 представлены методы `create_widgets`, `run_model` и `plot_results`.

Метод `create_widgets` создает и размещает на окне виджеты для ввода параметров модели и кнопку для запуска модели. Также создается текстовое поле для отображения результатов.

С помощью метода `run_model` вызывается при нажатии кнопки «Запустить модель». Он считывает значения параметров из полей ввода, создает экземпляр `DelayedDemandModel`, выполняет прогнозирование спроса и расчет оптимального заказа, а затем отображает результаты в текстовом поле. Также вызывает метод `plot_results` для построения графиков.

Метод `plot_results` строит графики уровня запасов, прогнозированного спроса и анализа чувствительности для математического ожидания спроса  $\mu$ .

```

110 def create_widgets(self):
111     labels = ['Интенсивность спроса (d)', 'Фиксированные затраты на заказ (K)',
112             'Удельные затраты на хранение (h)', 'Удельные потери прибыли от дефицита (b)',
113             'Математическое ожидание спроса ( $\mu$ )']
114     self.entries = []
115
116     for i, label in enumerate(labels):
117         ttk.Label(self, text=label).grid(row=i, column=0, padx=10, pady=5, sticky='w')
118         entry = ttk.Entry(self)
119         entry.grid(row=i, column=1, padx=10, pady=5, sticky='ew')
120         self.entries.append(entry)
121
122     ttk.Button(self, text="Запустить модель", command=self.run_model).grid(row=len(labels), column=0, colspan=2, pady=10)
123
124     self.results_text = tk.Text(self, height=10, width=50)
125     self.results_text.grid(row=len(labels) + 1, column=0, colspan=2, padx=10, pady=10)
126
127 def run_model(self):
128     try:
129         d = float(self.entries[0].get())
130         K = float(self.entries[1].get())
131         h = float(self.entries[2].get())
132         b = float(self.entries[3].get())
133          $\mu$  = float(self.entries[4].get())
134
135         historical_data = np.random.poisson( $\mu$ , 50)
136
137         self.model = DelayedDemandModel(d, K, h, b,  $\mu$ , historical_data)
138         self.model.forecast_demand()
139         self.model.calculate_optimal_order()
140         results = self.model.display_results()
141
142         self.results_text.delete(1.0, tk.END)
143         self.results_text.insert(tk.END, results)
144
145         self.plot_results()

```

Рисунок 9 – Методы create\_widgets и run\_model

```

146         except ValueError as e:
147             self.results_text.delete(1.0, tk.END)
148             self.results_text.insert(tk.END, f"Ошибка ввода: {e}")
149
150     def plot_results(self):
151         self.model.plot_inventory_levels()
152         self.model.plot_forecasted_demand()
153         param_range = np.linspace(50, 100, 10)
154         results = self.model.sensitivity_analysis(param_range, ' $\mu$ ')
155         self.model.plot_sensitivity_analysis(results, 'Математическое ожидание спроса ( $\mu$ )')
156
157 if __name__ == "__main__":
158     app = App()
159     app.mainloop()
160
161

```

Рисунок 10 – Метод plot\_results

Пользователь может ввести значения для параметров модели через интерфейс, запустить расчет оптимального размера заказа и просмотреть

результаты, включая графики прогнозирования спроса, уровня запасов и анализа чувствительности.

Пользователь вводит значения параметров  $d$ ,  $K$ ,  $h$ ,  $b$ ,  $\mu$ , нажимает кнопку «Запустить модель», после чего получает результаты расчетов и визуализацию данных на графиках. На рисунке 11 представлен интерфейс для ввода параметров.

Параметр	Значение
Интенсивность спроса ( $d$ )	2500
Фиксированные затраты на заказ ( $K$ )	150
Удельные затраты на хранение ( $h$ )	5
Удельные потери прибыли от дефицита ( $b$ )	15
Математическое ожидание спроса ( $\mu$ )	75

**Запустить модель**

Оптимальный размер заказа: 693.65  
Ожидаемый уровень дефицита: 381.35  
Ожидаемые издержки: 1210913.54

Рисунок 11 – Интерфейс для ввода параметров

На рисунках 12-14 показаны результаты, которые будет выводить код:

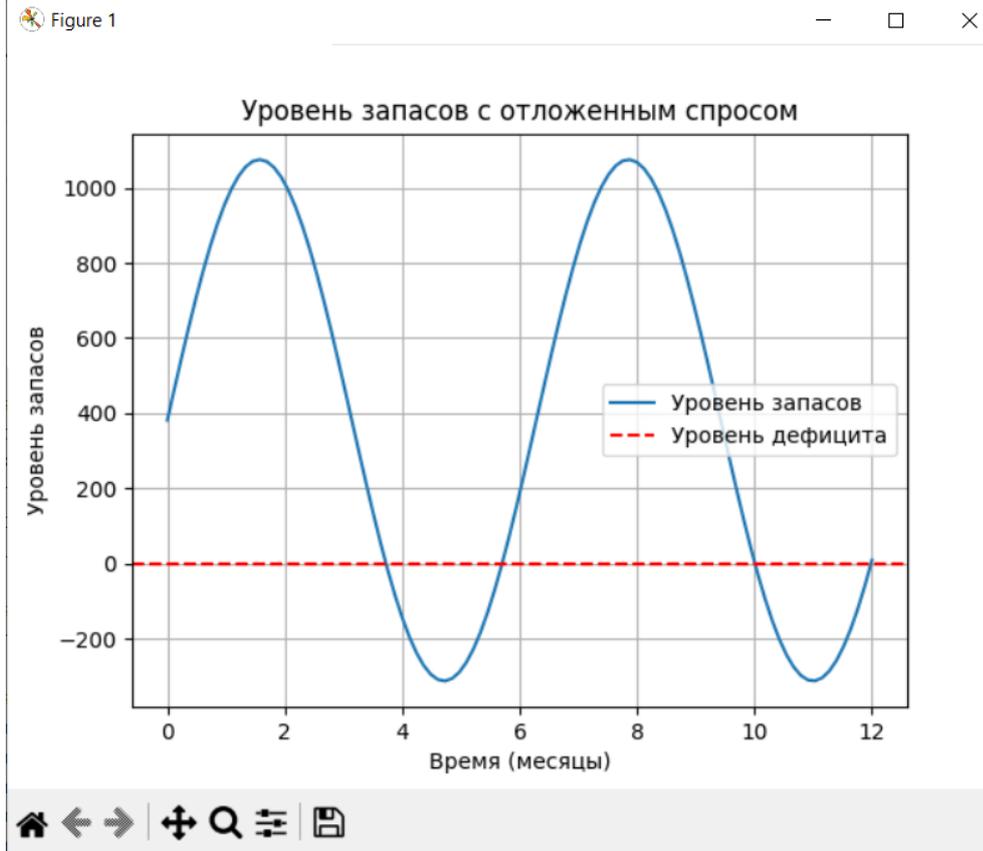


Рисунок 12 – Результаты задачи с моделью отложенного спроса

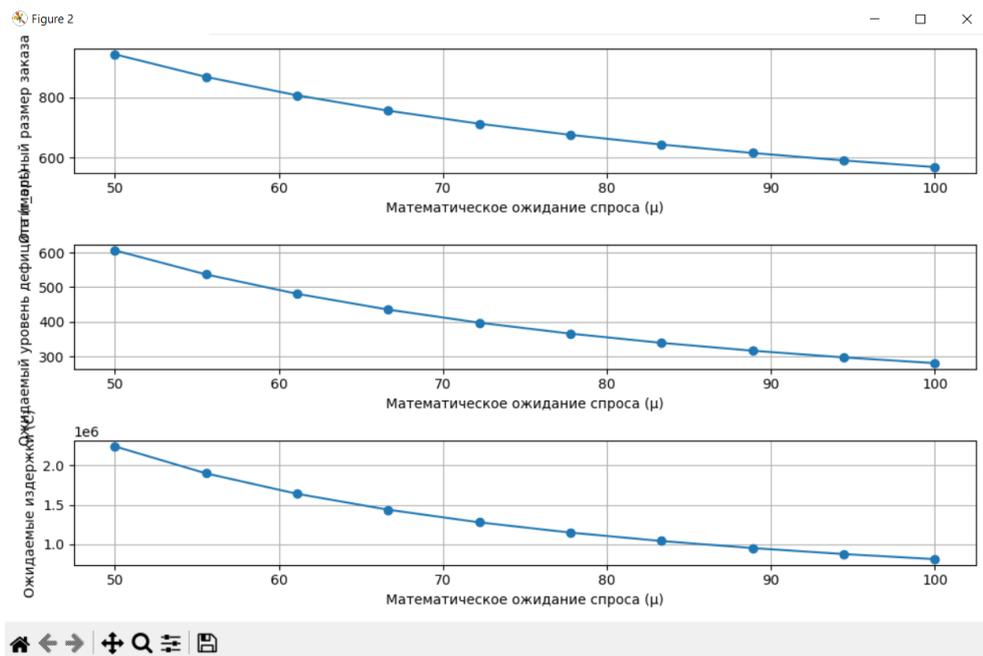


Рисунок 13 – Результаты задачи с моделью отложенного спрос

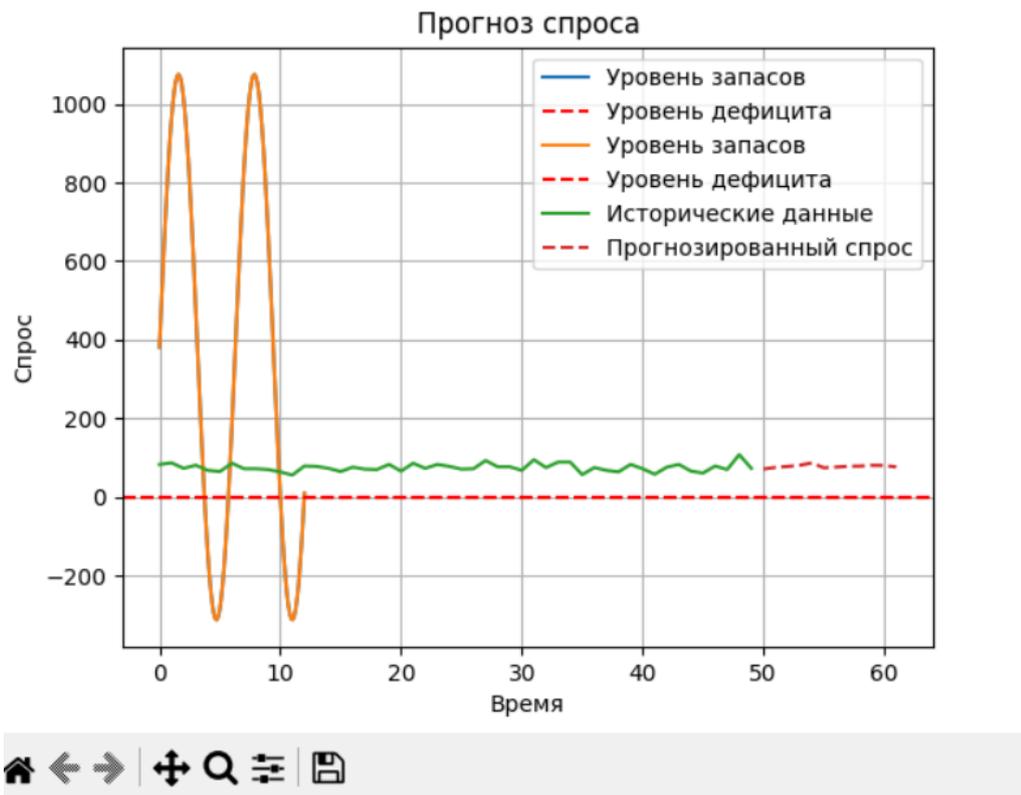


Рисунок 14 – Результаты задачи с моделью отложенного спроса

На основании графиков, представленных на рисунках (12) - (14) легко можно выполнить анализ и прогнозирование состояния складских запасов.

### 3.2.2 Модель с потерянными спросом

Модель с потерянными спросом предполагает, что неудовлетворенный спрос не переносится на будущее, а теряется навсегда, что может привести к упущенной прибыли и потере клиентов.

Исходя из данных условий и значений, напишем код для решения задачи управления запасами с моделью отложенного спроса.

Этот код представляет собой реализацию модели потерянного спроса с графическим интерфейсом на Python, используя библиотеки NumPy, Matplotlib, SciPy, и Tkinter.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.stats import norm
4 from scipy.optimize import minimize
5 import tkinter as tk
6 from tkinter import ttk
7
8 class LostSalesModel:
9     def __init__(self, d, m, K, b, s,  $\mu$ , h):
10         self.d = d # Годовой спрос
11         self.m = m # Себестоимость каждого каркаса
12         self.K = K # Необходимые затраты на заказ
13         self.b = b # Коэффициент издержек
14         self.s = s # Дополнительная сумма на получение недостающих каркасов
15         self. $\mu$  =  $\mu$  # Математическое ожидание спроса
16         self.h = h # Среднее квадратическое отклонение
17

```

Рисунок 15 – Класс LostSalesModel и конструктор `_Init_`

На рисунке 15 представлен фрагмент кода, где используется класс LostSalesModel, который реализует модель потеряннного спроса, предоставляя методы для расчета оптимального размера заказа, оптимизации заказа, анализа чувствительности и визуализации результатов.

Далее представлен Конструктор `_Init_`, с помощью которого инициализируется модель с заданными параметрами.

На рисунках 16-17 представлены методы `optimize_order`, `sensitivity_analysis` и `display_results`.

С помощью метода `optimize_order` происходит оптимизация размера заказа с использованием функции минимизации `minimize` из библиотеки SciPy. Функция минимизирует издержки, рассчитывая их на основе текущих параметров модели.

Метод `sensitivity_analysis` выполняет анализ чувствительности для заданного параметра модели в определенном диапазоне значений, возвращая список результатов (значение параметра, оптимальный размер заказа, уровень дефицита, ожидаемые издержки).

Метод `display_results` возвращает строку с результатами оптимизации: оптимальный размер заказа, ожидаемый уровень дефицита и ожидаемые издержки.

```

31
32     def optimize_order(self):
33         def cost(Q):
34             F_zr = norm.cdf(Q / (self.b * self.d + self.h * Q))
35             z_r = norm.ppf(F_zr)
36             B = self.h * norm.pdf(z_r) + self.μ - z_r
37             C = (self.K * self.d / Q + self.h * (Q ** 2) / 2 +

```

Рисунок 16 – Метод optimize\_order

```

38             self.s * norm.pdf(z_r) + self.b * self.d * norm.pdf(z_r))
39         return c
40
41         result = minimize(cost, x0=self.Q1, method='BFGS')
42         self.Q_opt = result.x[0]
43         self.C = result.fun
44         self.r_opt = norm.ppf(norm.cdf(self.Q_opt / (self.b * self.d + self.h * self.Q_opt)))
45
46     def sensitivity_analysis(self, param_range, param_name):
47         results = []
48         original_value = getattr(self, param_name)
49         for value in param_range:
50             setattr(self, param_name, value)
51             self.calculate_optimal_order()
52             results.append((value, self.Q_opt, self.r_opt, self.C))
53             setattr(self, param_name, original_value)
54         return results
55
56     def display_results(self):
57         return f"Оптимальный размер заказа: {self.Q_opt:.2f}\nОжидаемый уровень дефицита: {self.r_opt:.2f}\nОжидаемые издержки: {self.C:.2f}"

```

Рисунок 17 – Метод sensitivity\_analysis и display\_results

На рисунках 18-19 представлены методы визуализации plot\_inventory\_levels и plot\_sensitivity\_analysis. Данные методы создают графики:

- уровня запасов с отложенным спросом;
- анализа чувствительности для заданного параметра.

```

59     def plot_inventory_levels(self):
60         time_period = np.linspace(0, 12, 100)
61         inventory_levels = self.Q_opt * np.sin(time_period) + self.r_opt
62         plt.plot(time_period, inventory_levels, label='Уровень запасов')
63         plt.axhline(y=0, color='r', linestyle='--', label='Уровень дефицита')
64         plt.xlabel('Время (месяцы)')
65         plt.ylabel('Уровень запасов')
66         plt.title('Уровень запасов с потеряннным спросом')
67         plt.legend()
68         plt.grid(True)
69         plt.show()

```

Рисунок 18 – Метод визуализации plot\_inventory\_levels

```

71     def plot_sensitivity_analysis(self, results, param_name):
72         param_values, Q_opts, r_opts, costs = zip(*results)
73         plt.figure(figsize=(10, 6))
74         plt.subplot(3, 1, 1)
75         plt.plot(param_values, Q_opts, marker='o')
76         plt.xlabel(param_name)
77         plt.ylabel('Оптимальный размер заказа (Q_opt)')
78         plt.grid(True)
79
80         plt.subplot(3, 1, 2)
81         plt.plot(param_values, r_opts, marker='o')
82         plt.xlabel(param_name)
83         plt.ylabel('Ожидаемый уровень дефицита (r_opt)')
84         plt.grid(True)
85
86         plt.subplot(3, 1, 3)
87         plt.plot(param_values, costs, marker='o')
88         plt.xlabel(param_name)
89         plt.ylabel('Ожидаемые издержки (C)')
90         plt.grid(True)
91
92         plt.tight_layout()
93         plt.show()

```

Рисунок 19 – Метод визуализации plot\_sensitivity\_analysis

На рисунке 20 представлен класс App, который реализует графический интерфейс приложения с использованием Tkinter. Конструктор `_Init_` инициализирует главное окно приложения с заголовком и размером, а также вызывает метод для создания виджетов.

```

95     class App(tk.Tk):
96         def __init__(self):
97             super().__init__()
98             self.title("Lost Sales Model")
99             self.geometry("400x400")
100
101             self.create_widgets()
102
103             self.model = None
104

```

Рисунок 20 – Класс App и конструктор `_Init_`

На рисунках 21-22 представлены методы `create_widgets`, `run_model` и `plot_results`. Метод `create_widgets` создает и размещает на окне виджеты для ввода параметров модели и кнопку для запуска модели. Также создается текстовое поле для отображения результатов. С помощью метода `run_model`

вызывается при нажатии кнопки «Запустить модель». Он считывает значения параметров из полей ввода, создает экземпляр `LostSalesModel`, выполняет прогнозирование спроса и расчет оптимального заказа, а затем отображает результаты в текстовом поле. Также вызывает метод `plot_results` для построения графиков.

Метод `plot_results` строит графики уровня запасов, прогнозируемого спроса и анализа чувствительности для математического ожидания спроса  $\mu$ .

```

105 def create_widgets(self):
107     """Необходимые затраты на заказ (K), 'Коэффициент издержек (b)',
108     'Дополнительная сумма на получение недостающих каркасов (s)',
109     'Математическое ожидание спроса ( $\mu$ )', 'Среднее квадратическое отклонение (h)']
110     self.entries = []
111
112     for i, label in enumerate(labels):
113         ttk.Label(self, text=label).grid(row=i, column=0, padx=10, pady=5, sticky='w')
114         entry = ttk.Entry(self)
115         entry.grid(row=i, column=1, padx=10, pady=5, sticky='ew')
116         self.entries.append(entry)
117
118     ttk.Button(self, text="Запустить модель", command=self.run_model).grid(row=len(labels), column=0, colspan=2, pady=10)
119
120     self.results_text = tk.Text(self, height=10, width=50)
121     self.results_text.grid(row=len(labels) + 1, column=0, colspan=2, padx=10, pady=10)
122
123     def run_model(self):
124         try:
125             d = float(self.entries[0].get())
126             m = float(self.entries[1].get())
127             K = float(self.entries[2].get())
128             b = float(self.entries[3].get())
129             s = float(self.entries[4].get())
130              $\mu$  = float(self.entries[5].get())
131             h = float(self.entries[6].get())
132
133             self.model = LostSalesModel(d, m, K, b, s,  $\mu$ , h)
134             self.model.calculate_optimal_order()
135             self.model.optimize_order()
136             results = self.model.display_results()
137
138             self.results_text.delete(1.0, tk.END)
139             self.results_text.insert(tk.END, results)
140
141             self.plot_results()

```

Рисунок 21 – Методы `create_widgets` и `run_model`

```

142         except ValueError as e:
143             self.results_text.delete(1.0, tk.END)
144             self.results_text.insert(tk.END, f"Ошибка ввода: {e}")
145
146     def plot_results(self):
147         self.model.plot_inventory_levels()
148         param_range = np.linspace(50, 150, 10)
149         results = self.model.sensitivity_analysis(param_range, ' $\mu$ ')
150         self.model.plot_sensitivity_analysis(results, 'Математическое ожидание спроса ( $\mu$ )')
151
152     if __name__ == "__main__":
153         app = App()
154         app.mainloop()
155

```

Рисунок 22 – Метод `plot_results`

```
152 if __name__ == "__main__":  
153     app = App()  
154     app.mainloop()  
155
```

Рисунок 23 – Запуск приложения

На рисунке 23 создается экземпляр App и запускается главный цикл Tkinter (app.mainloop()), что позволяет взаимодействовать с графическим интерфейсом.

Пользователь может ввести значения для параметров модели через интерфейс, запустить расчет оптимального размера заказа и просмотреть результаты, включая графики уровня запасов и анализа чувствительности.

Пользователь вводит значения параметров  $d$ ,  $m$ ,  $K$ ,  $h$ ,  $b$ ,  $\mu$ ,  $s$ , нажимает кнопку «Запустить модель», после чего получает результаты расчетов и визуализацию данных на графиках. На рисунке 24 представлен интерфейс для ввода параметров.

Параметр	Значение
Годовой спрос (d)	2800
Себестоимость каждого каркаса (m)	50
Необходимые затраты на заказ (K)	600
Кoeffициент издержек (b)	0.04
Дополнительная сумма на получение недостающих каркасов (s)	400
Математическое ожидание спроса ( $\mu$ )	900
Среднее квадратическое отклонение (h)	50

Запустить модель

Оптимальный размер заказа: 32.27  
Ожидаемый уровень дефицита: 0.02  
Ожидаемые издержки: 78298.78

Рисунок 24 – Интерфейс для ввода параметров

На рисунках 25-27 показаны результаты, которые будет выводить код:

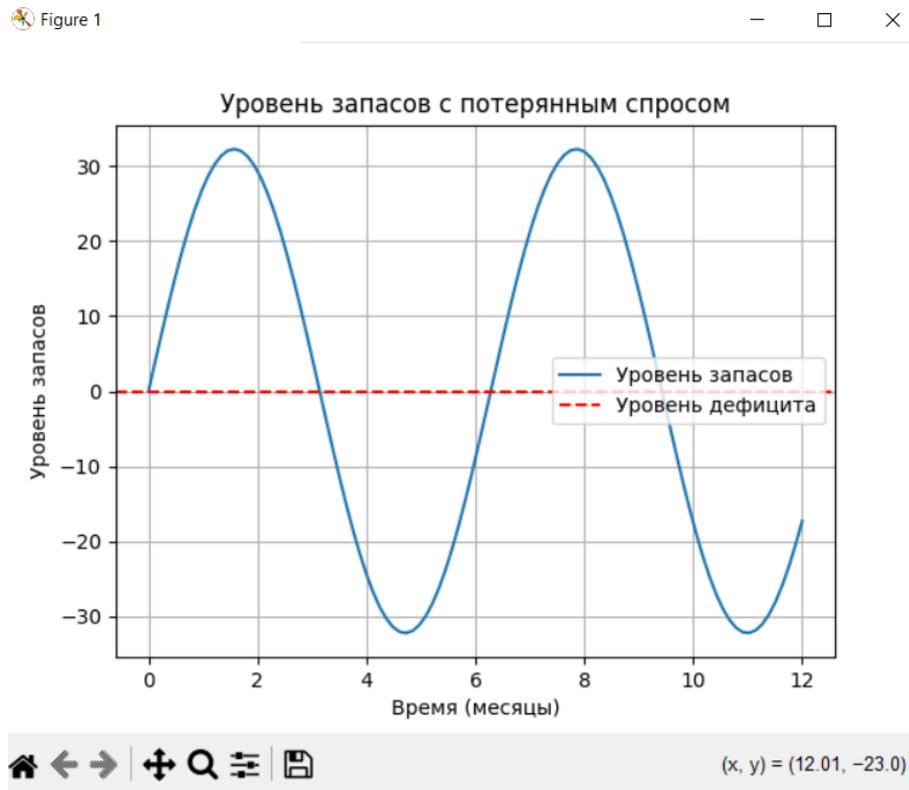


Рисунок 25 – Результаты задачи с моделью потерянного спроса

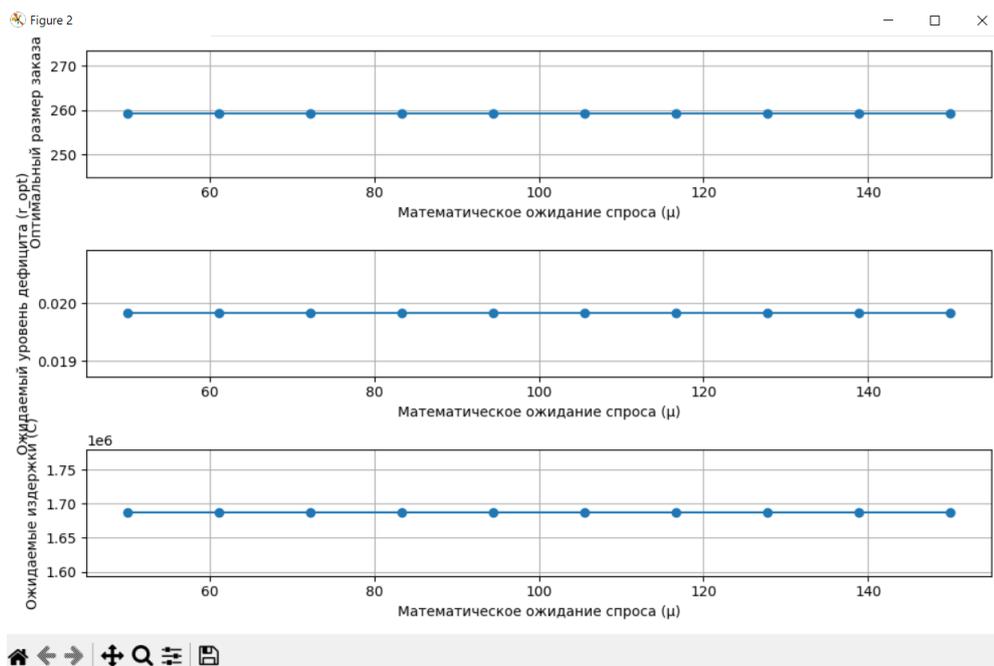


Рисунок 26 – Результаты задачи с моделью потерянного спроса



Рисунок 27 – Результаты задачи с моделью потерянного спроса

На основании графиков, представленных на рисунках (25) - (27) легко можно выполнить анализ и прогнозирование состояния складских запасов.

## Заключение

В выпускной квалификационной работе подробно рассмотрены стохастические модели управления запасами с отложенным спросом и потерянным спросом.

На основе стохастических моделей управления запасами разработаны алгоритмы нахождения оптимальных значений параметров.

Программная реализация алгоритмов представлена на языке Python. Программа выполняет расчет оптимального заказа, прогнозирование спроса, уровень дефицита, ожидаемые издержки, а также выполняет анализ чувствительности для заданных значений параметров. Выполняется визуализация полученных результатов в виде графиков и диаграмм, что позволяет эффективно осуществить прогноз спроса на основе статистических данных.

Разработан удобный интерфейс для пользователей. Программа протестирована для реальных значений параметров, которые были предоставлены ООО «Нобель Автомотив Русиа». Пользователь может ввести значения параметров модели через интерфейс, запустить расчет оптимального размера заказа и просмотреть результаты, включая графики прогнозирования спроса, уровня запасов и анализа чувствительности.

По результатам тестирования ошибок и сбоев в работе программы не выявлено.

Результаты работы могут быть использованы на практике, для повышения эффективности процессов управления и снижению издержек на предприятии. Применение разработанных алгоритмов позволит компаниям лучше управлять своими запасами, сократить расходы на их хранение и уменьшить вероятность дефицита продукции. Также использование данных алгоритмов может помочь точнее прогнозировать спрос, что приведет к более эффективному планированию закупок и производства.

## Список используемой литературы и используемых источников

1. Денисова, А. Л. Организация коммерческой деятельности. Управление запасами [Электронный ресурс]: учебное пособие / Денисова А.Л., Дюженкова Н.В.— Электрон. текстовые данные. — Тамбов: Тамбовский государственный технический университет, ЭБС АСВ, 2012. — 81 с. — Режим доступа: <http://www.iprbookshop.ru/64143.html>. — ЭБС «IPRbooks» (дата обращения: 18.04.2017).
2. Ермольев, Ю. М. Методы стохастического программирования / Ю. М. Ермольев. – Москва: Наука, 1976. – 239 с
3. Карманов, В. Г. Математическое программирование: учеб. пособие / В. Г. Карманов. – М.: ФИЗМАТЛИТ, 2004. – 264 с.
4. Котов, В. П. Математическое программирование: учеб. пособие / В.П. Котов, Н.А., Адрицкая. – СПб.: Лань, 2014. – 432 с.
5. Лукинский В.С., Цвиринько И.А., Малевич Ю.В. Логистика. // СПб: Издательство СПбГИЭУ, 2000.
6. Применение методов ABC и XYZ анализа в системе бюджетных закупок лекарственных средств [Электронный ресурс] // SciUp URL: <https://sciup.org/140104743> (дата обращения 16.04.2024).
7. Скородумов А.А. КРАТКИЙ КУРС молодого менеджера по закупкам & ЛИКБЕЗ маркетолога-аналитика [Электронный ресурс] // MatrixCheck URL: [https://matrixcheck.ru/wp-content/uploads/2015/01/Aleksey\\_Skorodu](https://matrixcheck.ru/wp-content/uploads/2015/01/Aleksey_Skorodu)
8. Социально-экономическое развитие: Системный подход и информационные технологии / Под ред. М.Г. Завельского. - М.: КД Либроком, 2008. - 176 с.
9. Тюхтина, А. А. Модели управления запасами: учебно-методическое пособие / А. А. Тюхтина. — Нижний Новгород: ННГУ им. Н. И. Лобачевского, 2017. — 84 с

10. Управление запасами: многофакторная оптимизация процесса поставок: учебник для среднего профессионального образования / Г. Л. Бродецкий, В. Д. Герами, А. В. Колик, И. Г. Шидловский. — Москва: Издательство Юрайт, 2024. — 322 с
11. Управление запасами в цепях поставок: учебник и практикум для вузов / В. С. Лукинский [и др.]. — Москва: Издательство Юрайт, 2024. — 625 с. Управление запасами широкой номенклатуры: с чего начать? [Электронный ресурс] // HSE URL: <https://www.hse.ru/data/391/032/1239/ABC-XYZ%20-%20для%20Логинфо.pdf> (дата обращения 18.04.2024).
12. Фишер А., «Методы выделения групп в ABC-анализе» // журнал «Логистика и Управление», № 1-2008.
13. Фёдоров Д. Ю. Основы программирования на примере языка Python / Учебное пособие. — СПб.: Юрайт, 2018. — 167 с.
14. Чаплыгин А.Н. Учимся программировать вместе с Питоном. Учебник. — ревизия 226. — 135 с.
15. Ширяев В.И. Исследование операций и численные методы оптимизации: учебное пособие / В.И. Ширяев. Москва: Ленанд, 2017. — 224 с.
16. Шведова И.А. ABC-анализ: полный гид [Электронный ресурс] // Литрес URL: <https://www.litres.ru/book/irina-aleksandrovna-shvedova/avs-analiz-polnyu-gid-66677318/chitat-onlayn/> – (дата обращения 15.03.2024).
17. Экономический риск и методы его измерения Ч2 [Электронный ресурс] // Studfile URL: <https://studfile.net/preview/1100371/page:12/> (дата обращения 06.05.2024).
18. Юрьева, А. А. Математическое программирование: учеб. пособие / А.А. Юрьева. – СПб.: Лань, 2014. – 432 с.
19. SciPy documentation [Электронный ресурс]. URL: <https://docs.scipy.org/doc/scipy/> (дата обращения: 30.04.2024).

20. А.И. Маркушевич - Краткий курс теории аналитических функций [Электронный ресурс]. URL: mov\_Kratky\_Kurs\_Analitika.pdf (дата обращения 03.04.2024).
21. Winston W.L. Introduction to Mathematical Programming: Applications and Algorithms. Boston (Mass.): PWS–KENT Publ., 1991. – 794 p.
22. Progress in inventory research. //Proc. of the 4-th Internat. Symp Budapest: Akad. Kiado, 1989. – 446 p.
23. Winston W.L. Operations Research: Applications and Algorithms Boston (Mass.): PWS–KENT Publ., 1990. – 1434p.
24. 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. – 423–434 p.
25. Jaworski M., Ziadé T. Expert Python Programming Packt Publishing. 2021 – 630 p.