

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	2
1. Анализ особенностей цепей поставок	5
1.1. Анализ задачи оптимизации цепи поставок	5
1.2. Математическая модель цепи поставок	11
2. Разработка алгоритма решения	19
2.1. Пример решения задачи	21
2.2. Поведение при отрицательных объемах	26
2.3. Пример решения задачи с отрицательными объемами	29
3. Разработка и проектирование программного обеспечения	31
3.1. Функциональные требования к разрабатываемой системе	31
3.2. Нефункциональные требования к разрабатываемой системе.....	32
3.3. Требования к организации входных и выходных данных	33
3.4. Варианты использования системы.....	37
3.5. Выбор и обоснование используемых инструментов.....	50
3.6. Инструкция к запуску программы	57
3.7. Пример работы ПО со скриншотами интерфейса	58
ЗАКЛЮЧЕНИЕ	67
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	68

ВВЕДЕНИЕ

Цепи поставок — неотъемлемая часть мировой экономики и бизнеса, которая тесно связана с торговлей и производством. Компаниям, находящимся на рынке, необходимо налаживать операционную деятельность для реализации собственных продуктов путем создания системы товарных потоков и торговых связей в рамках цепи поставок. Последние исследования в области управления цепочками поставок показали, что в структуре, состоящей из независимых фирм, принимаемые решения оказывают влияние не только на прибыль самих компаний, но и других участников цепи. Это в свою очередь воздействует на всю цепь поставок. Именно данный момент послужил причиной к изучению стратегических решений, принимаемых участниками цепи, и их влиянию на всю цепь поставок.

С течением времени численность и сложность подобных систем увеличиваются под воздействием технологического развития и процесса мировой интеграции. А также налаживание и оптимизация уже существующих цепей поставок становятся еще более трудоемкой и комплексной задачей. Однако, важность ее решения часто недооценивается, что приводит к финансовым потерям и упущенной выгоде. Поэтому необходимо оптимизировать цепь поставок по критерию прибыли и координировать действия участников цепи. Данная задача, без сомнения, становится особенно актуальной в связи с широкой распространенностью цепей поставок и их важностью в современном бизнесе [6].

Для анализа ситуаций, где результат зависит от принимаемых решений каждой из участвующих сторон, наиболее подходящим инструментом является теория игр. В настоящее время исследователи управления цепями поставок все чаще используют теорию игр для изучения, прогнозирования и помощи управленцам в принятии стратегических решений в сложных цепях поставок. Теория игр позволяет учесть воздействие каждого участника цепи на общую прибыль (или другую меру эффективности работы цепи поставок),

а также способствует прогнозированию результатов применения участниками различных стратегий.

Целью данной работы является разработка и программная реализация алгоритма оптимизации цепи поставок древовидной структуры по критерию прибыли.

Для более эффективного изучения цепей поставок важно установить их структуру и взаимосвязи между элементами. Один из практических подходов формализации заключается в представлении цепей в виде древовидного графа. Это позволяет лучше понять и управлять процессом поставок.

Нами будет рассмотрен древовидный граф. Внутри каждого из его узлов находятся несколько фирм, между которыми происходит конкуренция по модели Курно [22]. В корневом узле или, говоря иначе, узле-дистрибьюторе, распределяется товар между дочерними узлами. В последующих узлах, если они не являются терминальными, происходит модернизация исходного продукта, после чего осуществляется продажа своим дочерним узлам. Терминальные узлы реализуют товар на рынках.

Ключевым задачами для достижения цели исследования, являются:

- произвести анализ и обзор литературных источников, касающихся методов координации, теоретических аспектов взаимодействия и конкуренции в цепях поставок;
- построить теоретико-игровую модель для оптимизации цепи поставок по критерию прибыли;
- разработать алгоритм, который позволит построить оптимальное решение для данной цепи;
- создать реализацию разработанного алгоритма в программном виде;

Работа содержит 3 раздела. Основная часть работы изложена на 94 страницах, содержит в себе 19 рисунков и 11 таблиц.

В первом разделе представлены обзор литературы, постановка задачи, математическая модель, проанализированы аналоги программного обеспечения (ПО) на рынке.

Во втором разделе разработан алгоритм решения задачи, а также разобран аналитический пример решения.

В третьем разделе представлены функциональные и нефункциональные требования к разрабатываемой системе, а также выбор и обоснование программно-технических средств, инструкция по запуску ПО, результаты и скриншоты работы программного продукта.

В заключение приведены выводы по результатам работы, список использованной литературы и программный листинг (приложение А).

1. Анализ особенностей цепей поставок

1.1. Анализ задачи оптимизации цепи поставок

Цепи поставок являются интегральной частью функционирования современных компаний и бизнеса в целом. В расширяющемся и все более глобализированном мире, преодоление коммуникационных, производственных и логистических барьеров, а также стремление к оптимизации и упрощению реализации товаров и услуг становятся ключевыми задачами всех участников рынка. Цепь поставок представляет собой совокупность всех этапов, связанных с передачей товаров и услуг от производителя к конечному потребителю. Включая поставщиков сырья, производство, хранение, транспортировку, дистрибуцию, а также более широкий круг внутренних и внешних связей, необходимых для функционирования и развития предприятий. Цепи поставок, как правило, включают в себя несколько международных, национальных и/или региональных организаций, а также инфраструктурных элементов, которые обеспечивают непрерывность в создании, производстве и продаже продукции и услуг.

Цепь поставок является сложной сетью взаимосвязанных фирм, где каждая из них стремится максимизировать свою прибыль, принимая во внимание действия и решения конкурентов. В задаче оптимизации цепи поставок необходимо найти наилучшую стратегию для каждой фирмы с целью достижения общего экономического выгодного состояния.

В контексте нашей задачи оптимизации цепи поставок рассмотрим несколько моделей конкуренции, которые часто применяются в экономической теории:

1. Модель Бертрана: В этой модели предполагается, что фирмы конкурируют по цене, а не по количеству производимого товара. При наличии нескольких фирм, каждая из них ставит цену ниже своих конкурентов, чтобы привлечь больше покупателей. Модель Бертрана

применяется, когда товары являются однородными, а стоимость производства не зависит от объема.

2. Модель Курно: В этой модели предполагается, что фирмы конкурируют по количеству производимого товара, а не по цене. Фирмы выбирают свои производственные объемы с учетом ожидаемого спроса и реакции конкурентов.
3. Модель Штакельберга: Эта модель подразумевает наличие монопольного производителя, который является первопроходцем на рынке, и других производителей, которые присоединяются к нему. В этой модели монопольный производитель может устанавливать цены, тогда как другие производители должны приспосабливаться к этим ценам.

Каждая из этих моделей имеет свои особенности и применяется в зависимости от конкретных условий рынка и типа конкуренции, которая на нем происходит. Например, модель Бертрана может быть применима в случаях, когда производители имеют одинаковые затраты на производство и конкурируют на ценах. Модель Курно часто используется для моделирования рынков, где производители имеют непостоянные затраты на производство, а модель Штакельберга для моделирования монопольных рынков, где производитель устанавливает цены и определяет уровень производства.

При анализе рыночных моделей конкуренции существуют различные подходы, которые описывают различные виды взаимодействия между фирмами. Одной из наиболее распространенных моделей является модель Курно. Она основывается на предположении, что фирмы в цепи поставок выбирают свой производственный объем независимо друг от друга, принимая во внимание решения своих конкурентов. Это означает, что каждая фирма стремится максимизировать свою прибыль, учитывая предполагаемый отклик конкурентов на свои действия.

Выбор модели Курно в нашей задаче оптимизации цепи поставок обусловлен несколькими факторами. Во-первых, модель Курно широко

используется в исследованиях и приложениях, связанных с конкуренцией на рынке. Она является стандартной моделью, которая позволяет анализировать взаимодействие фирм в условиях несовершенной конкуренции.

Во-вторых, она предоставляет возможность изучить стратегии производственного объема и ценообразования фирм, учитывая их взаимодействие и реакцию конкурентов. Это позволяет оценить эффекты таких факторов, как снижение цены, увеличение объема производства и изменение доли рынка.

Таким образом, модель Курно представляет собой релевантный инструмент для решения задачи оптимизации цепи поставок, где фирмы конкурируют за рыночную долю и стремятся максимизировать свою прибыль, принимая во внимание решения конкурентов.

Экономическая модель Курно является одной из основных моделей теории игр и используется для описания рыночных ситуаций, где фирмы конкурируют друг с другом, устанавливая цены на товары.

В модели Курно предполагается, что на рынке присутствуют несколько фирм, каждая из которых может устанавливать свою цену на продукт. Однако, в отличие от идеальной конкуренции, цены не являются равными стоимости производства товара. Вместо этого каждая фирма принимает во внимание реакцию других фирм на свою цену [19].

В модели Курно фирмы максимизируют свою выручку, выбирая цену, которая дает наибольшую прибыль при условии, что остальные фирмы удержат свои цены неизменными. В результате каждая фирма устанавливает цену, которая ниже оптимальной в условиях идеальной конкуренции.

Кроме того, в модели Курно предполагается, что фирмы действуют независимо друг от друга и не сотрудничают между собой, что может привести к неэффективности рынка и снижению общественного благосостояния.

Особенности модели Курно состоят в следующем [20]:

1. Несколько фирм на рынке: модель Курно предполагает, что на рынке действуют несколько фирм, каждая из которых конкурирует с другими.
2. Фирмы устанавливают цены: в отличие от модели идеальной конкуренции, где цены устанавливаются на основе стоимости производства, в модели Курно каждая фирма устанавливает свою цену на продукт.
3. Фирмы максимизируют прибыль: цель каждой фирмы – максимизировать свою выручку, а не свою долю рынка, посредством выбора цены, которая дает наибольшую прибыль при условии, что остальные фирмы удержат свои цены неизменными.
4. Каждое предприятие знает, что другие предприятия будут реагировать на его решения, и, следовательно, на различные стратегии, которые оно может выбрать.
5. Предприятия принимают решение о выпуске продукции до того, как они узнают стратегию других участников рынка.
6. Каждому предприятию выгодно действовать, учитывая реакции других участников рынка, то есть использовать игровую теорию для принятия решений.
7. Неэффективность рынка: в модели Курно фирмы действуют независимо друг от друга и не координируют свои действия, что может привести к неэффективности рынка и снижению общественного благосостояния.
8. Необходимость анализа равновесия Нэша: в модели Курно необходимо анализировать равновесие Нэша, которое возникает при определенных стратегиях фирм и описывает ситуацию, при которой ни одна фирма не может увеличить свою прибыль, изменив свою стратегию.

Для изучения и оптимизации цепей поставок важно формализовать их структуру. Одним из возможных подходов к такой формализации может быть

представление их в виде древовидного графа или дерева решений. В структуре древовидного графа, узлы представляют собой отдельных участников (т.е., производители, поставщики, другие фирмы и т.д.), а рёбра соединяют этих участников и символизируют потоки товаров, сырья и услуг между ними.

Рассмотрим пример цепи поставок лесной промышленности (см. рисунок 1).

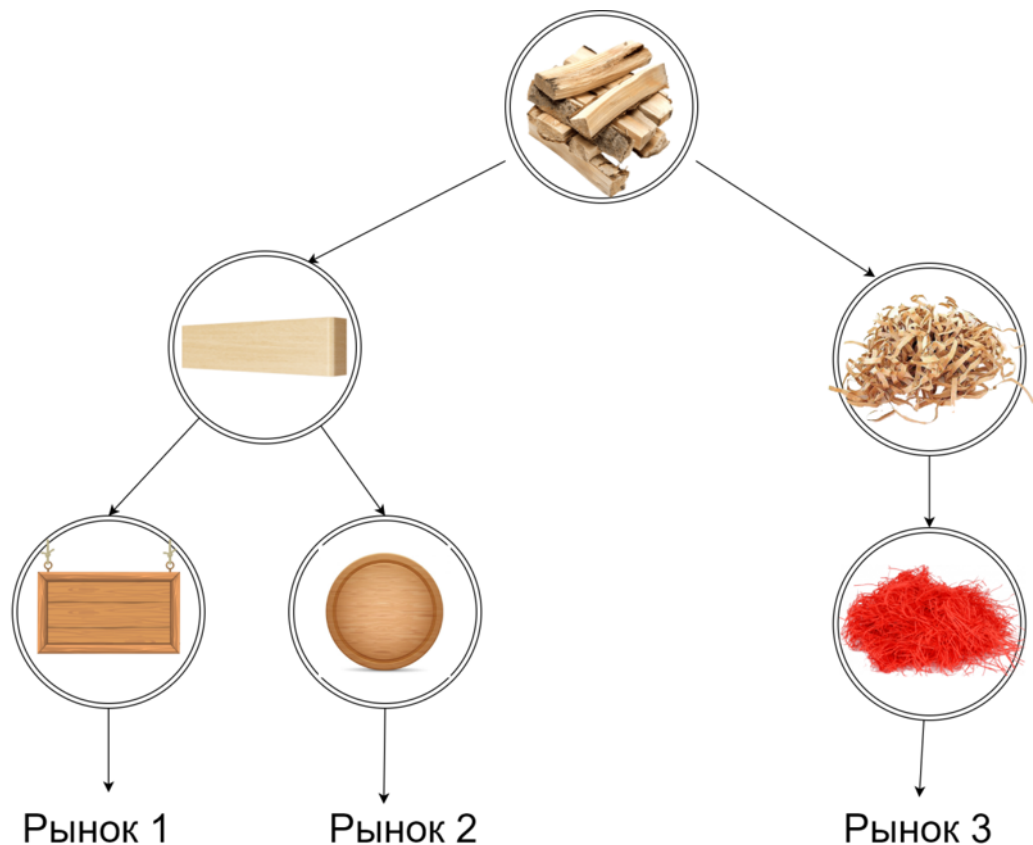


Рисунок 1 – цепь поставок лесной промышленности

У нас есть оптовые дистрибьюторы, которые поставляют необработанную древесину розничным дистрибьюторам. Они, в свою очередь, модернизируют полученный товар для продажи ритейлерам. Последние также вносят изменения в продукцию и реализуют товар на рынках.

Формализуем эту ситуацию, используя древовидный граф. Внутри каждого из его узлов находятся несколько фирм, между которыми происходит конкуренция по модели Курно.

Рассмотрим древовидный граф на рисунке 2.

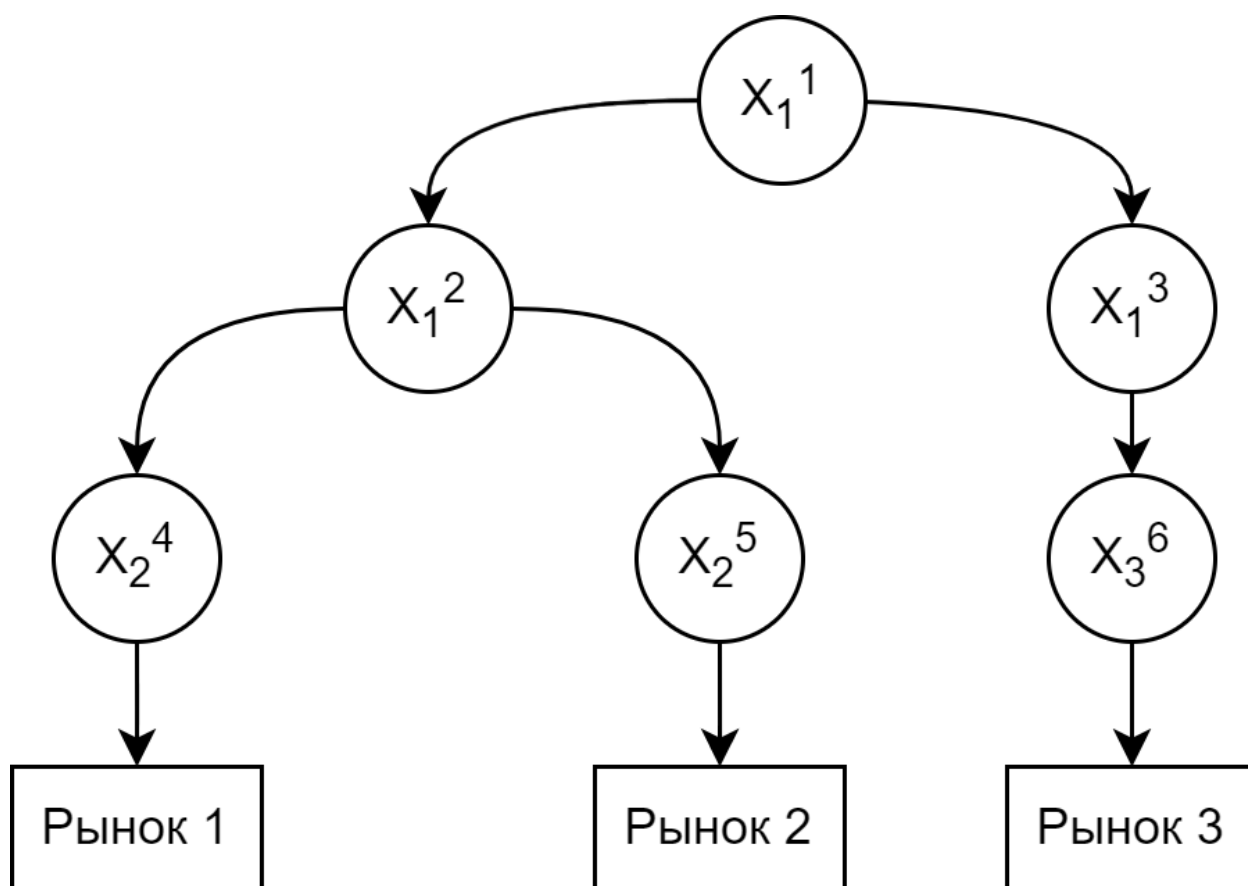


Рисунок 2 – Пример цепи поставок с древовидной структурой.

Допустим, что каждый узел в цепи поставок представляет группу компаний, которые соперничают друг с другом, производя одинаковый продукт с использованием одинаковых ресурсов. Хотя переменные издержек у фирм могут отличаться, они являются общедоступными параметрами в сети. Количество компаний в каждом узле также известно. Предполагается, что компании, находящиеся в одном узле, продают продукт по одной и той же цене.

В корневом узле или, говоря иначе, узле-дистрибьюторе (X_1^1), распределяется товар между дочерними узлами (X_1^2 и X_1^3). В последующих узлах, если они не являются терминальными, происходит модернизация исходного продукта, после чего осуществляется продажа уже своим дочерним узлам. Терминальные узлы реализуют товар на рынках. Предполагается, что данные рынки, не конкурируют между собой и работают в соответствии с

моделью Курно, где используется линейная функция спроса и фиксированные издержки.

Все фирмы стремятся к максимизации собственной выручки, однако на значение прибыли отдельно взятого участника цепочки поставок влияют не только его собственные решения, но и действия других участников. Исходя из этого было решено рассматривать данную задачу в контексте теоретико-игрового подхода.

1.2. Математическая модель цепи поставок

Разберем ориентированный граф, имеющий древовидную структуру, в котором каждый узел представляет несколько компаний, функционирующих между собой по экономической модели рыночной конкуренции Курно.

Корневой узел, или узел-дистрибьютор, делит произведенный им товар между своими дочерними узлами. При условии того, что дальнейшие узлы не являются терминальными, в них происходит модернизация исходного продукта, а затем продажа произведенного результата своим дочерним узлам. Терминальные узлы реализуют товар на рынках.

Данный способ устройства экономических отношений является наиболее распространённым в современном бизнесе. В качестве примера можно рассмотреть цепочку поставок в производстве электроники.

На корневой вершине ориентированного древовидного графа расположены производители полупроводниковых материалов и компонентов, например, Qualcomm, Intel, Micron Technologies, Broadcom и т.д. Они поставляют свои продукты, такие как процессоры, чипы памяти, модули связи, на следующий уровень цепочки поставок.

На следующем уровне расположены производители микросхем и компонентов для электроники, такие как Texas Instruments, Broadcom, Analog Devices и т.д. Они используют продукты, полученные от производителей полупроводниковых материалов, чтобы создавать микросхемы, процессоры и

т.д., которые в свою очередь поставляются на следующий уровень цепочки поставок.

Наконец, на терминальном уровне находятся производители конечной электроники, такие как Apple, Samsung, Dell и т.д., которые используют поставленные дистрибьюторами микросхемы и компоненты, чтобы создать итоговые продукты, такие как смартфоны, ноутбуки, планшеты и т.д.

Введем ряд определений и понятий, чтобы упростить понимание дальнейших положений.

Будем рассматривать ориентированный граф, который *обозначим* через $G = (X, U)$, где X – множество вершин, а U – множество ребер.

Определение 1. Если существует ребро $u_{ij} \in U$, связывающее вершины X^i и X^j , где $X^i, X^j \in X$, тогда вершина X^j называется дочерней вершиной X^i , а вершина X^i будет называться **родительской по отношению к вершине** X^j .

Обозначим через F_i множество индексов дочерних узлов, для вершины $i : F_i \subset I, i = \overline{1, L}, I = \{1, 2, \dots, L\}$, где L – количество узлов в графе.

Замечание 1. Если X^j является дочерней вершиной X^i , тогда $j \in F_i$.

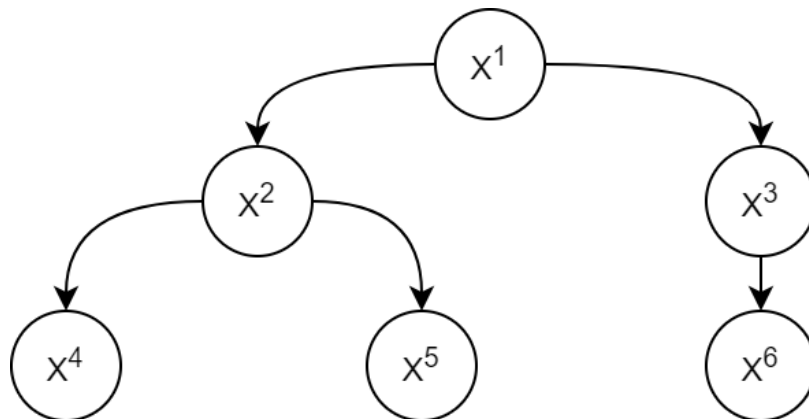


Рисунок 3 – Древовидная модель цепи поставок с номерными индексами.

Для цепи поставок, изображенной на рис. 3, имеем:

$$u_{12} = \{X^1, X^2\}, u_{13} = \{X^1, X^3\}, \text{ тогда } F_1 = \{2, 3\};$$

$$u_{24} = \{X^2, X^4\}, u_{25} = \{X^2, X^5\}, \text{ тогда } F_2 = \{4, 5\};$$

$$u_{36} = \{X^3, X^6\}, \text{ тогда } F_3 = \{6\}.$$

Замечание 2. Если для $X^i \in X$ множество $F_i \neq \emptyset$ и существует некоторая вершина X^j такая, что $j \in F_i$, тогда можно реализовать следующую запись узла: X_i^j .

Дополним рисунок 3, исходя из замечания 2.

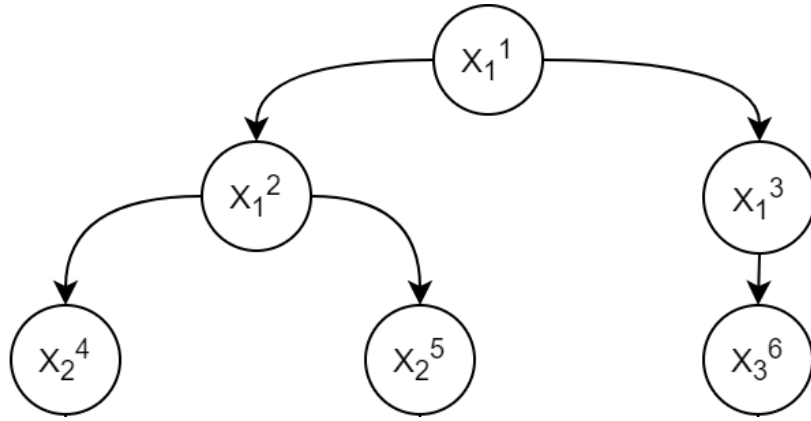


Рисунок 4 – Древовидная модель цепи поставок с номерными верхними индексами и нижними ссылками на родителей.

Для классификации узлов внутри цепи поставок введем ряд обозначений.

Обозначим корневую вершину через X_1^1 : $X_{кор} = \{X_1^1\}$.

Множество терминальных узлов обозначим через $X_{терм}$:

$$X_{терм} = \{X_j^i \in X \mid F_i = \emptyset\}.$$

Множество промежуточных узлов обозначим через $X_{пром}$:

$$X_{пром} = \{X \setminus (X_{кор} \cup X_{терм})\}.$$

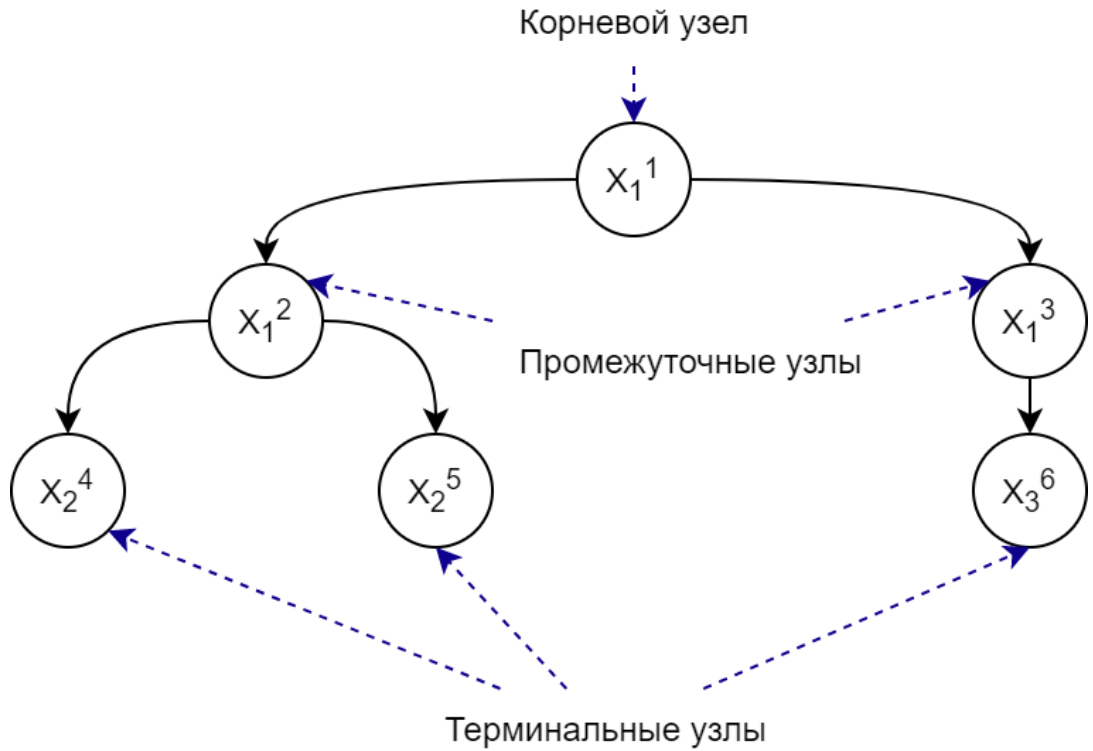


Рисунок 5 – Классификация узлов на примере

Определение 2. Под **мощностью** n_{ij} узла X_j^i будем понимать количество фирм в узле $n_{ij} = |X_j^i|$, где $n_{ij} \in \mathbb{Z} > 0$.

Построим теоретико-игровую модель для рассматриваемой задачи [18].

Определим **множество игроков**, как $N = \{x_{jk}^i\}_{k=1}^{n_{ij}}$, где x_{jk}^i фирма внутри узла X_j^i , $i = \overline{1, L}$, где L – количество узлов в графе. Иначе говоря, каждый игрок представляет собой компанию, находящуюся внутри каждого узла.

Определим **множество стратегий** для каждого игрока из множества N следующим образом:

Для игроков, находящихся в узлах **корневого** и **промежуточного** типа:

$$S_{ijk} = \{q_{ijk}, p_{ij}\},$$

Для игроков, находящихся в узлах *терминального* типа:

$$S_{ijk} = \{q_{ijk}\},$$

где $q_{ijk} \in \mathbb{R} \geq 0$ — это объем однородной продукции выпускаемой фирмой

x_{jk}^i узла X_j^i , а $p_{ij} \in \mathbb{R} \geq 0$ — это цена за единицу товара узла X_j^i .

Определим **функции выигрыша** для каждого из игроков, как прибыль, которую может получить фирма, следующим образом.

Для игроков, находящихся в узлах *терминального* типа

$$\pi_{ijk} = q_{ijk} (a_{ij} - b_{ij}Q_{ij} - p_{jh} - c_{ijk}).$$

Для игроков, находящихся в узлах *корневого* типа

$$\pi_{ijk} = q_{11k} (p_{11} - c_{11i}).$$

Для игроков, находящихся в узлах *промежуточного* типа

$$\pi_{ijk} = q_{ijk} (p_{ij} - p_{jh} - c_{ijk}),$$

где

$c_{ijk} \in \mathbb{R} > 0$ – значение издержек определенной фирмы при реализации одной единицы товара.

$a_{ij} \in \mathbb{R} > 0$ – свободный член, константа, обозначающая цену товара на узле X_j^i при нулевом количестве продаж.

$b_{ij} \in \mathbb{R} > 0$ – коэффициент пропорциональности, обозначает скорость изменения цены товара в зависимости от изменения количества продаж.

Рассматривается игра с полной информацией, то есть все игроки имеют полную информацию о характеристиках игры и выбранных стратегиях других игроков.

Таким образом, теоретико-игровая модель для задачи может быть представлена следующим образом:

$$G = \{N, \{S_l\}_{l \in N}, \{\pi_l\}_{l \in N}\}, \quad (1)$$

где N – множество игроков, участвующих в игре. Каждый игрок имеет свои собственные цели и интересы и выбирает свою стратегию, чтобы достичь их.

$\{S_l\}_{l \in N}$ – множество стратегий, доступных каждому игроку. Стратегия — это полный набор действий, которые игрок может выбрать в любой ситуации в игре.

$\{\pi_l\}_{l \in N}$ – функция выигрыша каждого игрока, зависящая от выбранных ими стратегий.

Также представим формулы линейной функции цены и условия отсутствия излишков и дефицита. Каждая из фирм стремится к максимизации собственной прибыли [2].

Линейная функция цены:

$$p_{ij} = a_{ij} - b_{ij}Q_{ij}. \quad (2)$$

Функция условия отсутствия излишков и дефицита:

$$Q_{ij} = \sum_{k=1}^{n_{ij}} q_{ijk} = \sum_{h \in F_i} Q_{hi} = \sum_{h \in F_i} \sum_{t=1}^{n_{hi}} q_{hit}. \quad (3)$$

Условие отсутствия излишков и дефицита в цепи поставок — это ситуация, когда производитель способен оценить спрос на свой товар, точно распланировать производственную программу, чтобы покупателям хватало товара без излишков или дефицита. То есть значение объема родительского узла представляет собой сумму объемов дочерних узлов. Обеспечивается эффективный поток товаров без задержек на всех этапах поставки. Таким образом, гарантируется, что количество товаров, поступившее на вход, равно количеству товаров, перешедшему на следующий уровень, что позволяет избежать потерь и оптимизировать производственный процесс.

Опишем процесс принятия решения в рассматриваемой модели.

Шаг 1. Корневой узел определяет цену, по которой он продает товар своим дочерним узлам.

Шаг 2. Дочерние узлы корневого узла цепи поставок, если они не являются терминальными узлами, получая информацию от родителя, назначают цену товара уже своим дочерним вершинам.

Шаг 3. Терминальные узлы на основе цен, полученных от своих родителей, и функций спроса определяют объемы выпуска товара на рынок.

Шаг 4. Происходит процедура распределения объемов между фирмами в каждом из терминальных узлов.

Шаг 5. Информация об объемах поступает на все верх лежащие узлы и внутри каждого происходит процедура распределения объемов между фирмами.

Шаг 6. Происходит подсчет прибыли каждого из участников цепи поставок.

Данная процедура принятия решений может быть представлена многошаговой некооперативной иерархической игрой n лиц. Игроками являются фирмы, находящиеся в каждом из узлов. Стратегиями игроков, располагающихся в корневом и промежуточных узлах, являются объемы производства и цены, по которым продается товар, а для игроков конечных вершин – только объёмы. Функции выигрыша – это функции прибыли каждой фирмы-игрока. При этом внутри каждого узла фирмы участники конкурируют по модели Курно [22-23]. В качестве принципа оптимальности было выбрано равновесие по Нэшу [21]. Принцип оптимальности был выбран, исходя из того, что нас интересует оптимизация действий на уровне отдельного игрока (фирмы) и в рассматриваемой задаче на данный момент не предусматриваются коалиции, т.е. каждый игрок действует независимо от других в угоду только собственным интересам

Равновесие по Нэшу [2]:

Ситуация $q_N^* = (q_1^*, \dots, q_{i-1}^*, q_i^*, q_{i+1}^*, \dots, q_N^*) \in \widetilde{q_N}$ в игре G_N называется равновесием по Нэшу, если для каждого игрока i и любой стратегии $q_i \in \widetilde{q_i}$ этого игрока выполняется неравенство:

$$\pi_i(q_1^*, \dots, q_{i-1}^*, q_i^*, q_{i+1}^*, \dots, q_N^*) \geq \pi_i(q_1^*, \dots, q_{i-1}^*, q_i, q_{i+1}^*, \dots, q_N^*)$$

2. Разработка алгоритма решения

Опишем алгоритм построения оптимального решения.

1. Представим исходные данные в качестве древовидного графа, где в узлах хранится изначальная информация о количестве фирм и их издержек. Терминальные узлы выходят на рынки, где нам известна функция спроса на товар. Выразим из функции спроса функцию цены.
2. Будем производить обратный обход дерева:
 - 2.1. Если текущая вершина принадлежит к терминальному множеству:
 - 2.1.1. Функция цены нам уже известна.
 - 2.1.2. Формируем систему функций прибыли.
 - 2.1.3. Применяем к системе условия максимума.
 - 2.1.4. Из полученной системы производных выражаем функции оптимальных объемов фирм.
 - 2.1.5. Суммируем функции оптимальных объемов фирм, чтобы получить функцию объема текущего узла.
 - 2.2. Если текущая вершина принадлежит к промежуточному множеству:
 - 2.2.1. Применяем условие отсутствия дефицита и излишков.
 - 2.2.2. Выражаем функцию цены текущего узла.
 - 2.2.3. Формируем систему функций прибыли.
 - 2.2.4. Применяем к системе условия максимума.
 - 2.2.5. Из полученной системы производных выражаем функции оптимальных объемов фирм.
 - 2.2.6. Суммируем функции оптимальных объемов фирм, чтобы получить функцию объема текущего узла.
 - 2.3. Если текущая вершина принадлежит к корневому множеству:
 - 2.3.1. Применяем условие отсутствия дефицита и излишков.
 - 2.3.2. Выражаем функцию цены текущего узла.
 - 2.3.3. Формируем систему функций прибыли.
 - 2.3.4. Применяем к системе условия максимума.

2.3.5. Получаем значения объемов фирм.

2.3.6. Рассчитываем значение цены.

2.3.7. Рассчитываем значения прибыли фирм.

3. Спускаясь по дереву вниз, рассчитываем значения цены, объёмов и прибыли для всех оставшихся узлов.

Покажем, что данный алгоритм приведёт к построению равновесия по Нэшу. Рассмотрим терминальную вершину 2 цепи поставок, являющейся дочерней к корневой вершине 1. Будем считать для определённости, что имеем дело со случаем дуополии, т.е. предположим, что в узле действуют всего две фирмы, т.е. $k = 2$. Пусть эти фирмы конкурируют в соответствии с моделью Курно.

Функция цены для обоих игроков, в соответствии с формулой (2), имеет вид: $p_{21} = a_{21} - b_{21}(q_{211} + q_{212})$. А функции прибыли, для игроков 1 и 2, находящихся в узлах терминального типа, имеют следующий вид:

$$\begin{cases} \pi_{211} = q_{211}(a_{21} - b_{21}(q_{211} + q_{212}) - p_{11} - c_{211}) \\ \pi_{212} = q_{212}(a_{21} - b_{21}(q_{211} + q_{212}) - p_{11} - c_{212}) \end{cases}$$

Применим необходимое условие экстремума первого порядка и получим систему (4):

$$\begin{cases} \frac{\partial \pi_{211}}{\partial q_{211}} = a_{21} - 2b_{21}q_{211} - b_{21}q_{212} - p_{11} - c_{211} = 0 \\ \frac{\partial \pi_{212}}{\partial q_{212}} = a_{21} - 2b_{21}q_{212} - b_{21}q_{211} - p_{11} - c_{212} = 0 \\ q_{211}, q_{212} \geq 0 \end{cases} \quad (4)$$

Решая систему (4), получим функции реакции конкурирующих фирм (5) [3]:

$$\begin{cases} q_{211} = R_1(q_{212}) = \frac{a_{21} - b_{21}q_{212} - p_{11} - c_{211}}{2b_{21}} \\ q_{212} = R_2(q_{211}) = \frac{a_{21} - b_{21}q_{211} - p_{11} - c_{212}}{2b_{21}} \end{cases} \quad (5)$$

Решим систему (4), подставив функции (5). Имеем:

$$\left\{ \begin{array}{l} 2a_{21} - 2p_{11} - c_{211} - c_{212} - 3q_{211}b_{21} - 3q_{212}b_{21} = 0 \\ q_{211} = \frac{a_{21} - b_{21}q_{212} - p_{11} - c_{211}}{2b_{21}} \\ q_{212} = \frac{a_{21} - b_{21}q_{211} - p_{11} - c_{212}}{2b_{21}} \end{array} \right. \quad (6)$$

Проанализируем вторые производные от функций прибыли:

$$\frac{\partial^2 \pi_{211}}{\partial q_{211}^2} = -2 < 0, \quad \frac{\partial^2 \pi_{212}}{\partial q_{212}^2} = -2 < 0$$

Они отрицательны. Это означает, что функции прибыли обеих фирмы строго вогнуты по объемам выпуска q_{211}, q_{212} , соответственно, этих фирм. Это означает, что условия первого порядка являются необходимыми и достаточными для равновесия по Нэшу [3].

Таким образом, мы показали, что найденные объёмы в системе (6) являются равновесием по Нэшу:

$$q_{211}^* = \frac{a_{21} - p_{11} + c_{212} - c_{211}}{3b_{21}}, \quad q_{212}^* = \frac{a_{21} - p_{11} + c_{211} - c_{212}}{3b_{21}}$$

Можно показать, что данный результат имеет место как для большего числа фирм внутри одной вершины, т.е., когда $k > 2$, так и для всех остальных множеств вершин в рассматриваемой цепи, а именно корневых и промежуточных.

Таким образом, реализация, описанного выше алгоритма, позволит рассчитать оптимальные объёмы в цепи поставок и соответствующие цены.

2.1. Пример решения задачи

Рассмотрим теоретико-игровую модель цепи поставок с шестью узлами изображенную на рисунке 6. В таблице 2 представлены исходные данные.

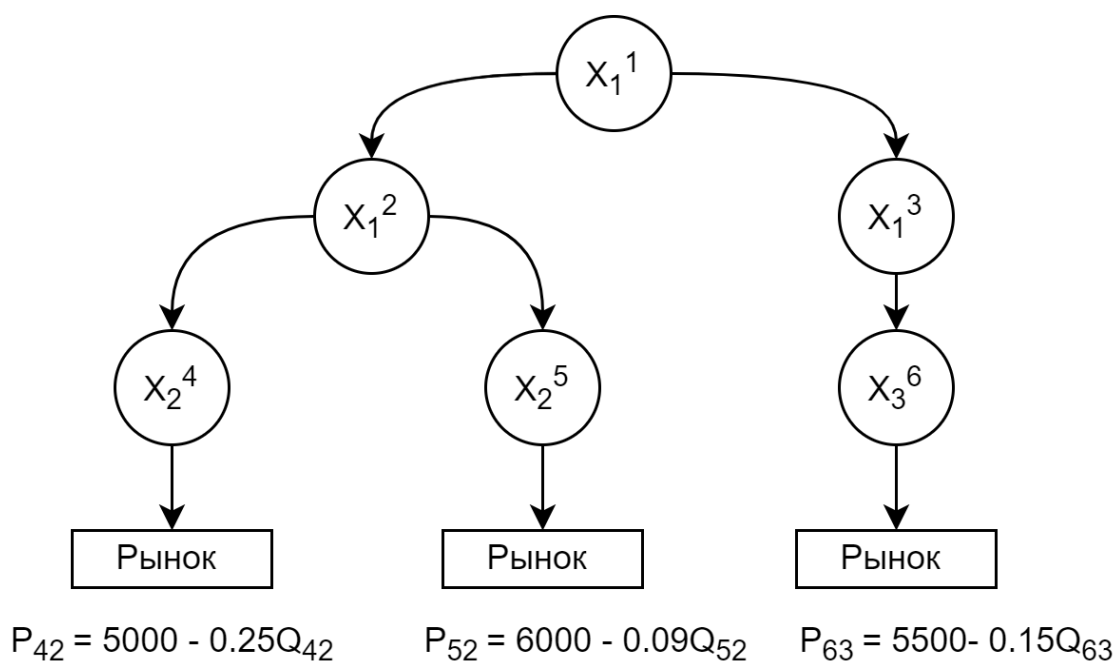


Рисунок 6 – Цепь поставок с тремя узлами дистрибьюторов и тремя узлами ретейлеров

Таблица 1 — Обозначения

Обозначения	Пояснение
X_j^i	Узел номер i с родителем j
x_{jk}^i	Фирм k внутри узла X_j^i
$Q_{ij} = \sum_{k=1}^{n_{ij}} q_{ijk}$	Суммарный объём однородной продукции в узле X_j^i
q_{ijk}	Объём выпуска продукции фирмы k в узле X_j^i
π_{ijk}	Прибыль фирмы k в узле X_j^i
P_{ij}	Цена в узле X_j^i

Таблица 2 — Исходные данные

Узел	X_1^1	X_1^2	X_1^3	X_2^4	X_2^5	X_3^6
Кол-во фирм в узле	$n_{11} = 2$	$n_{21} = 1$	$n_{31} = 2$	$n_{42} = 4$	$n_{52} = 2$	$n_{63} = 1$
Затраты	$C_{111} = 1500$ $C_{112} = 1505$	$C_{211} = 700$	$C_{311} = 720$ $C_{312} = 710$	$C_{421} = 342$ $C_{422} = 340$ $C_{423} = 338$ $C_{424} = 345$	$C_{521} = 120$ $C_{522} = 122$	$C_{631} = 200$

Для построения равновесия по Нэшу в рассматриваемой игре распишем функции прибыли для всех фирм из терминальных узлов:

$$\left\{ \begin{array}{l} \pi_{421} = q_{421} \left(5000 - 0,25 \sum_{i=1}^4 q_{42i} - p_{21} - 342 \right) \\ \pi_{422} = q_{422} \left(5000 - 0,25 \sum_{i=1}^4 q_{42i} - p_{21} - 340 \right) \\ \pi_{423} = q_{423} \left(5000 - 0,25 \sum_{i=1}^4 q_{42i} - p_{21} - 340 \right) \\ \pi_{424} = q_{424} \left(5000 - 0,25 \sum_{i=1}^4 q_{42i} - p_{21} - 345 \right) \end{array} \right. \quad (7)$$

$$\left\{ \begin{array}{l} \pi_{521} = q_{521} \left(6000 - 0,09 \sum_{i=1}^2 q_{52i} - p_{21} - 120 \right) \\ \pi_{522} = q_{522} \left(6000 - 0,09 \sum_{i=1}^2 q_{52i} - p_{21} - 122 \right) \end{array} \right. \quad (8)$$

$$\pi_{631} = q_{631} (5500 - 0,15 q_{621} - p_{31} - 200) \quad (9)$$

Применив к функциям (7), (8) и (9) необходимое условие максимума, получаем системы уравнений:

$$\begin{pmatrix} 0,5 & 0,25 & 0,25 & 0,25 \\ 0,25 & 0,5 & 0,25 & 0,25 \\ 0,25 & 0,25 & 0,5 & 0,25 \\ 0,25 & 0,25 & 0,25 & 0,5 \end{pmatrix} \begin{pmatrix} q_{421} \\ q_{422} \\ q_{423} \\ q_{424} \end{pmatrix} = \begin{pmatrix} 4658 - p_{21} \\ 4660 - p_{21} \\ 4662 - p_{21} \\ 4655 - p_{21} \end{pmatrix} \quad (10)$$

$$\begin{pmatrix} 0,18 & 0,09 \\ 0,09 & 0,18 \end{pmatrix} \begin{pmatrix} q_{521} \\ q_{522} \end{pmatrix} = \begin{pmatrix} 5880 - p_{21} \\ 5878 - p_{21} \end{pmatrix} \quad (11)$$

$$(0,3)(q_{631}) = (5300 - p_{31}) \quad (12)$$

Также возьмем вторую производную от функций (7), (8) и (9), чтобы показать ее отрицательность, это доказывает, что необходимые условия максимума являются достаточными:

$$\begin{cases} \frac{d^2 \pi_{421}}{dq_{421}^2} = -0,5 \\ \frac{d^2 \pi_{422}}{dq_{422}^2} = -0,5 \\ \frac{d^2 \pi_{423}}{dq_{423}^2} = -0,5 \\ \frac{d^2 \pi_{424}}{dq_{424}^2} = -0,5 \end{cases}$$

$$\begin{cases} \frac{d^2 \pi_{521}}{dq_{521}^2} = -0,18 \\ \frac{d^2 \pi_{522}}{dq_{522}^2} = -0,18 \end{cases}$$

$$\frac{d^2 \pi_{631}}{dq_{631}^2} = -0,3$$

В результате решения систем (10) – (12) получим выражения для объёма поставок (q_{ijk}) . Затем, исходя из условия отсутствия дефицита и излишков, получаем соотношение:

$$Q_{21} = q_{211} = Q_{42} + Q_{52} = \sum_{i=1}^4 q_{42i} + \sum_{i=1}^2 q_{52i} = 58456,1 - 10,6p_{21} \quad (13)$$

$$Q_{31} = q_{311} + q_{312} = Q_{63} = q_{631} = 17,666,67 - 3,3p_{31} \quad (14)$$

Из соотношения (13) выразим p_{21} :

$$p_{21} = 5510,88 - \frac{q_{211}}{10,6}. \quad (15)$$

Из соотношения (14) выразим p_{31} :

$$p_{31} = 5300 - \left(\frac{q_{311}}{3,3} + \frac{q_{312}}{3,3} \right). \quad (16)$$

Зная функции цен, мы можем выражать функции прибыли для фирм родительских узлов. Дальнейшие действия осуществляются по аналогии с приведенными, вплоть до множества корневых узлов, при достижении которого получаем численные значения объёмов для фирм корневого узла. Зная данные значения, подставляем их в ранее выведенные формулы для получения информации об объёмах и прибыли.

Таблица 3 — Результаты

Узел	X_1^1	X_1^2	X_1^3	X_2^4	X_2^5	X_3^6
Объём выпуска продукции для фирмы	$q_{111} \approx 8151$ $q_{112} \approx 8113$	$q_{211} \approx 11815$	$q_{221} \approx 2207$ $q_{222} \approx 2241$	$q_{311} \approx 206$ $q_{312} \approx 214$ $q_{313} \approx 222$ $q_{314} \approx 194$	$q_{321} \approx 5500$ $q_{322} \approx 5477$	$q_{331} \approx 4448$

Цена	$p_{11} \approx 2583$	$p_{21} \approx 4396$	$p_{22} \approx 3965$	$p_{31} \approx 4790$	$p_{32} \approx 5011$	$p_{33} \approx 4832$
Прибыль	$\pi_{111} \approx 8828070$ $\pi_{112} \approx 8746748$	$\pi_{211} \approx 13161609$	$\pi_{221} \approx 1462195$ $\pi_{222} \approx 1506682$	$\pi_{311} \approx 10652$ $\pi_{312} \approx 11494$ $\pi_{313} \approx 12368$ $\pi_{314} \approx 9450$	$\pi_{321} \approx 2722608$ $\pi_{322} \approx 2700652$	$\pi_{311} \approx 2968710$

Ручное решение данной проблемы потребовало значительного количества затраченного времени и вычисления математических операций. Однако, время, необходимое для получения решения и сложность выполняемых человеком операций, несомненно, будет значительно сокращено с разработкой программного обеспечения для получения оптимального, с точки зрения прибыли, решения в таких цепях.

2.2. Поведение при отрицательных объемах

При поиске равновесия Нэша в игре с нелинейными функциями выигрыша для каждого участника мы решаем задачу оптимизации с ограничениями, где каждый участник максимизирует свою функцию выигрыша, учитывая действия других участников. При этом существуют дополнительные ограничения в виде неравенств, которые гарантируют, что объемы производства (или цены, если они являются стратегиями) не могут быть отрицательными. В общем виде можно записать задачу нелинейного программирования следующим образом:

$$\begin{aligned} \max(\pi_{ijk}(q_{ij1}, q_{ij2}, \dots, q_{ijk}, p_{ij})), \\ q_{ijn} \geq 0, p_{ij} \geq 0, \\ i = \overline{1, L}, j \subseteq i, k = \overline{1, n_{ij}}, \end{aligned} \quad (17)$$

где L – количество узлов в графе, n_{ij} – мощность конкретного узла.

Сформулированная задача (17) является стандартной задачей нелинейной оптимизации. Необходимыми условиями оптимальности для неё

являются условия Куна-Таккера [3, 31]. Из этих условий следует, что если при решении задачи мы получаем отрицательное значение объема производства или цены, то оно приравнивается к нулю. С практической точки зрения это может происходить в связи с разными факторами, такими как, существенные издержки по сравнению с конкурентами, завышенные цены у фирм родительского узла и т.д. Но приравняв к нулю объем производства, мы фактически исключаем из процесса одного из игроков, что может привести к несбалансированности в игре и необходимости пересчета решения, так как участник был учтен в исходной модели, и его исключение может нарушить равновесие между оставшимися игроками.

Объяснение положений исходя из логики предметной области:

- Ситуация 1. Фирме невыгодно производить и участвовать в рыночных отношениях. Связи с этим она выпадает из товарно-денежного оборота. Она ничего не тратит, ничего не производит, ничего не продает.
- Ситуация 2.1. В рамках цепи поставок целому узлу невыгодно производить свой товар и участвовать в рыночных отношениях. Связи с этим он упраздняется из цепочки, а его дочерние узлы получая сырье от нового родительского узла изменяют собственное производство, не изменяя своих издержек.
- Ситуация 2.2. В рамках цепи поставок нерентабельным является выход терминального узла на определенный рынок товаров. В таком случае, если нет возможности перейти на другой рынок, целое поддерево цепи поставок может быть закрыто.
- Ситуация 2.3. В рамках цепи поставок корневому узлу невыгодно производить свой товар и участвовать в рыночных отношениях. Это приводит к невозможности функционирования низлежащих узлов, т.к. отсутствует исходное сырье.

Рассмотрим возникающие ситуации и предлагаемые действия устранения проблемы:

1. *Если хотя бы одна фирма в узле имеет неотрицательный объем.*
Предлагается вычеркнуть фирмы с отрицательным объёмом из рассмотрения и провести поиск решения сначала изменив исходные данные. В результирующем ответе считать объем вычеркнутых фирм равным нулю.

2. *Если все фирмы узла имеют отрицательные объёмы.*

2.1. *Если узел является промежуточным.*

Предлагается вычеркнуть данный узел из дальнейшего рассмотрения. Произвести реструктуризацию цепи поставок, таким образом, чтобы родительский узел удаляемой вершины, стал родителем для всех его дочерних узлов. Провести поиск решения с изменёнными значениями и реконструированной цепочкой поставок. В результирующем ответе считать объем вычеркнутых фирм равным нулю.

2.2. *Если узел является терминальным*

Предлагается вычеркнуть узел и рынок, на который он выходит из дальнейшего рассмотрения. А также родительские узлы, если у них нет альтернативных выходов на рынки. Провести поиск решения с изменёнными значениями и реконструированной цепочкой поставок. В результирующем ответе считать объем вычеркнутых фирм равным нулю.

2.3. *Если узел является корневым.*

Предлагается считать всю цепочку поставок нерентабельной. Обозначить объемы всех фирм равными нулю. Не производить дополнительных вычислений.

2.3. Пример решения задачи с отрицательными объемами

Рассмотрим ситуации возникновения отрицательных объемов. Для этого изменим исходные данные задачи из предыдущего пункта, таким образом, чтобы издержки фирмы C_{311} возросли с 720 до 2000 усл. ед.

Таблица 4 — Исходные данные

Узел	X_1^1	X_1^2	X_1^3	X_2^4	X_2^5	X_3^6
Кол-во фирм в узле	$n_{11} = 2$	$n_{21} = 1$	$n_{31} = 2$	$n_{42} = 4$	$n_{52} = 2$	$n_{63} = 1$
Затраты	$C_{111} = 1500$ $C_{112} = 1505$	$C_{211} = 700$	$C_{311} = 2000$ $C_{312} = 710$	$C_{421} = 342$ $C_{422} = 340$ $C_{423} = 338$ $C_{424} = 345$	$C_{521} = 120$ $C_{522} = 122$	$C_{631} = 200$

Ход решения аналогичен ранее рассмотренной задачи. В результате вычислений получим значения, записанные в таблице 5.

Таблица 5 — Результаты

Узел	X_1^1	X_1^2	X_1^3	X_2^4	X_2^5	X_3^6
Объём выпуска продукции для фирмы	$q_{111} \approx 7677$ $q_{112} \approx 7639$	$q_{211} \approx 12150$	$q_{221} \approx -567$ $q_{222} \approx 3733$	$q_{311} \approx 232$ $q_{312} \approx 240$ $q_{313} \approx 247$ $q_{314} \approx 220$	$q_{321} \approx 5617$ $q_{322} \approx 5594$	$q_{331} \approx 3166$
Цена	$p_{11} \approx 2520$	$p_{21} \approx 4366$	$p_{22} \approx 4350$	$p_{31} \approx 4765$	$p_{32} \approx 4991$	$p_{33} \approx 5025$

Прибыль	$\pi_{111} \approx 7831031$ $\pi_{112} \approx 7754449$	$\pi_{211} \approx 13916428$	$\pi_{221} \approx 96359$ $\pi_{222} \approx 4181162$	$\pi_{311} \approx 13412$ $\pi_{312} \approx 14354$ $\pi_{313} \approx 15329$ $\pi_{314} \approx 12058$	$\pi_{321} \approx 2839320$ $\pi_{322} \approx 2816898$	$\pi_{311} \approx 1504021$
---------	--	------------------------------	--	--	--	-----------------------------

Как мы можем видеть, объем для фирмы C_{311} составляет -567, что является недопустимым в меру условия неотрицательности данного параметра.

Вычеркнем данную фирму из рассмотрения и снова проведем вычисления. Итоговые результаты представлены в таблице 6, также учтем ранее удаленную компанию, присвоив ей нулевое значение объемов продаж и прибыли соответственно.

Таблица 6 — Результаты

Узел	X_1^1	X_1^2	X_1^3	X_2^4	X_2^5	X_3^6
Объем выпуска продукции для фирмы	$q_{111} \approx 7581$ $q_{112} \approx 7547$	$q_{211} \approx 11791$	$q_{221} = 0$ $q_{222} \approx 3337$	$q_{311} \approx 205$ $q_{312} \approx 213$ $q_{313} \approx 221$ $q_{314} \approx 192$	$q_{321} \approx 5491$ $q_{322} \approx 5469$	$q_{331} \approx 3337$
Цена	$p_{11} \approx 2587$	$p_{21} \approx 4399$	$p_{22} \approx 4298$	$p_{31} \approx 4792$	$p_{32} \approx 5013$	$p_{33} \approx 4999$
Прибыль	$\pi_{111} \approx 8246412$ $\pi_{112} \approx 8170770$	$\pi_{211} \approx 13106988$	$\pi_{221} = 0$ $\pi_{222} \approx 3341042$	$\pi_{311} \approx 10462$ $\pi_{312} \approx 11296$ $\pi_{313} \approx 12163$ $\pi_{314} \approx 9271$	$\pi_{321} \approx 2714130$ $\pi_{322} \approx 2692209$	$\pi_{311} \approx 1670521$

3. Разработка и проектирование программного обеспечения

3.1. Функциональные требования к разрабатываемой системе

В таблице 7 представлены основные функциональные требования к разрабатываемому программному продукту. Реализация спецификации осуществлялась исходя из требований к ПО [25;26;27].

Таблица 7 — Функциональные требования

FUN-ID	Описание
FUN-01	Система должна иметь графический интерфейс (GUI).
FUN-02	Пользователь должен иметь возможность добавлять новые узлы в цепь поставок.
FUN-03	Пользователь должен иметь возможность удалять узлы из цепи поставок.
FUN-04	Пользователь должен иметь возможность по двойному клику на узел цепи вносить в него данные о фирмах (Название компании, значение издержек за единицу товара).
FUN-05	Пользователь должен иметь возможность по двойному клику на терминальный узел цепи вносить данные о коэффициентах линейной функции цены.
FUN-06	Система должна иметь возможность сбросить всю внесенную в нее информацию по нажатию одной кнопки (Сбросить). После чего на экране отображается состояние системы по умолчанию (пустой шаблон).
FUN-07	Пользователь должен иметь возможность внешне моделировать цепь поставок (перемещать узлы зажав ЛКМ по плоскости).
FUN-08	Система должна рассчитывать и выводить на экран оптимальные значения для смоделированной пользователем цепи поставок используя алгоритм, описанный в пункте 1.3.

FUN-09	Система должна осуществлять валидацию вводимых пользователем данных в соответствующие поля.
FUN-10	Пользователь должен иметь возможность изменять вводимые ранее данные о фирмах внутри узлов.
FUN-11	Пользователь должен иметь возможность закрыть поле ввода, не изменяя его, то есть только просмотреть его.
FUN-12	Система должна отображать цепь поставок в виде древовидного графа.
FUN-13	Система должна иметь возможность сохранения структуры цепи поставок и информации о фирмах, содержащихся в ней в формат json.
FUN-14	Система должна иметь возможность загрузки структуры цепи поставок и информации о фирмах, содержащихся в ней из формата json.
FUN-15	Система должна иметь возможность сохранения рассчитанных результатов в формате xlsx.
FUN-16	Пользователь должен иметь возможность по одинарному клику выбирать узел.

3.2. Нефункциональные требования к разрабатываемой системе

Нефункциональные требования (NF-требования) определяют, как программное обеспечение должно работать и какими свойствами оно должно обладать, чтобы удовлетворять нужды пользователей [28]. В таблице 8 представлены основные нефункциональные требования к разрабатываемому программному продукту.

Таблица 8 — Нефункциональные требования

NF-ID	Описание
NF-01	Программа должна быть стабильной и надежной, не должна выдавать ошибки или аварийно завершаться (Надежность и безопасность).
NF-02	Программа должна быстро загружаться, быстро открывать и обрабатывать файлы, работать без промедлений и нареканий (Производительность).
NF-03	Интерфейс программы должен быть интуитивно понятным и минималистичным, что позволит пользователю быстро освоить программу и начать ее использовать (Удобство использования).
NF-04	Программа должна хорошо работать с другими программами и операционными системами, чтобы пользователи могли легко обмениваться информацией и файлами.
NF-05	Приложение должно работать на различных операционных системах.
NF-06	Программа должна уметь адаптироваться к разным размерам экранов и разрешениям, чтобы пользователи могли использовать ее на разных устройствах (например, на мониторе, на ноутбуке).

3.3. Требования к организации входных и выходных данных

Для организации входных данных необходимо разработать графический интерфейс, который позволит пользователю схематично добавлять новые узлы и заполнять такие поля как: Название фирмы, Издержки фирмы для каждого узла, а также рыночные коэффициенты А и Б для терминальных узлов. Необходимо предусмотреть возможность редактирования и удаления

уже существующих узлов. Данные, вводимые вручную, проверяются на корректность после попытки сохранения, также предусмотрен запрет в конкретных полях на определенные символы.

Для организации выходных данных нужно создать таблицу, которая будет содержать информацию о всех фирмах во всех узлах цепочки поставок, включая их цены, объемы товаров и прибыль. Возможно, также стоит предусмотреть возможность сортировки и фильтрации данных в таблице для удобства пользователя.

При организации входных и выходных данных необходимо учитывать, что они должны быть четко структурированы и удобны для работы пользователя. Необходимо предусмотреть все возможные варианты заполнения полей и обработки данных, чтобы избежать ошибок и искажений информации. Используемые типы данных представлены в таблице 9.

Таблица 9 — Принятые обозначения и типы данных

№	Название	Описание
1	float	Представляет собой числа с плавающей точкой двойной точности (64 бита) и имеет ограничения на точность и диапазон значений. Диапазон значений типа float составляет от приблизительно $-1,8 \times 10^{308}$ до приблизительно $1,8 \times 10^{308}$, что обеспечивает возможность представления очень больших и очень маленьких чисел. Однако точность типа float ограничена 16 десятичными знаками.
2	string	Представляет собой последовательность символов Unicode произвольной длины, которые используются для хранения текстовой информации.

3	int	Целочисленные значения в диапазоне от -2^{31} (-2 147 483 648) до $2^{31}-1$ (2 147 483 647)
---	-----	--

Таблица 10 — Словарь данных

Структура или элемент данных	Тип данных	Длина	Пояснение
nodeNumber	int	max	Генерируемый системой порядковый номер узла в цепи поставок, начиная с 1. (Not Null)
nodeParent	int	max	Номер родительской вершины, необходимый для индексирования наследственности. Принимает значения из множества номеров узлов. (Not Null)
firmName	string	400 символов	Название фирмы, может содержать буквы латиницы и кириллицы числа и символы из следующего после двоеточия списка: [0-9] / ! * ' ; : @ & = + \$? # _
firmCost	float	max	Издержки фирмы, которые являются строго положительным числом с плавающей запятой

a	float	max	Свободный член, константа, обозначающая цену товара на узле при нулевом количестве продаж, который является строго положительным числом с плавающей запятой
b	float	max	Коэффициент пропорциональности, обозначает скорость изменения цены товара в зависимости от изменения количества продаж, который является строго положительным числом с плавающей запятой
price	float	max	Цена за ед. Продукцию фирмы, которые являются строго положительным числом с плавающей запятой.
volume	float	max	Объем однородной продукции фирмы, которые являются строго положительным числом с плавающей запятой.

profit	float	max	Итоговая прибыль фирмы, которые являются строго положительным числом с плавающей запятой.
--------	-------	-----	---

3.4. Варианты использования системы

Словесные представления вариантов использования приведены в таблице 11.

Таблица 11 — Варианты использования системы.

Идентификатор ВИ	UC-01
Наименование:	Создание дочерней вершины
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-01, FUN-02, FUN-12, FUN-16
Действующее лицо:	Пользователь
Описание:	Создает новую дочернюю вершину для выбранной исходной вершины
Условие-триггер:	Пользователь вошел в приложение, чтобы смоделировать цепь поставок.
Предварительные условия:	Пользователь выбрал уже имеющуюся вершину в цепи.
Входные данные	Выбранная вершина, ее номер и расположение в цепи.
Выходные условия:	Создание новой вершины в цепи поставок.

Основной поток:	<p>1. Пользователь, используя графическое представление цепи поставок, выбирает конкретный узел и нажимает по нему 1 раз ЛКМ.</p> <p>2. Пользователь выбирает опцию создания новой вершины на панели инструментов.</p> <p>3. Приложение добавляет новую вершину в древовидный граф.</p>
Исключения:	<p>Пользователь пытается использовать функцию добавления без выбора узла.</p> <p>1. Система ничего не делает, ожидает дальнейшего выбора пользователя.</p>
Идентификатор ВИ	UC-02
Наименование:	Удаление вершины
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-01, FUN-03, FUN-12, FUN-16
Действующее лицо:	Пользователь
Описание:	Позволяет пользователю удалить выбранную вершину из древовидного графа.
Условие-триггер:	Пользователь вошел в приложение, чтобы смоделировать цепь поставок.

Предварительные условия:	Пользователь выбрал уже имеющуюся вершину в цепи.
Входные данные	Выбранная вершина, ее номер и расположение в цепи.
Выходные условия:	Удаление вершины и ее связей, а также всех дочерних узлов
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию удаления вершины на панели инструментов 2. Пользователь выбирает вершину для удаления 3. Система удаляет выбранную вершину из графа и отображения на экране, а также ее дочерние узлы если те имеются.
Исключения:	<p>Пользователь пытается использовать функцию удаления без выбора узла.</p> <ol style="list-style-type: none"> 1. Система ничего не делает, ожидает дальнейшего выбора пользователя.
Идентификатор ВИ	UC-03
Наименование:	Управление графом
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-01, FUN-07, FUN-12, FUN-16
Действующее лицо:	Пользователь
Описание:	Изменение расположения в пространстве вершин графа

Условие-триггер:	Пользователь вошел в приложение
Предварительные условия:	Пользователь вошел в приложение, чтобы смоделировать цепь поставок.
Выходные условия:	Измененный вид древовидного графа
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию управления на панели инструментов 2. Пользователь выбирает вершину для перемещения 3. Пользователь перемещает вершину 4. Пользователь повторяет пункт 2-3 пока его не устроит внешний вид 5. Система фиксирует изменение графа и отображает их на экране
Исключения:	—
Идентификатор ВИ	UC-04
Наименование:	Внесение данных о цепи поставок
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-01, FUN-04, FUN-05, FUN-09, FUN-16
Действующее лицо:	Пользователь
Описание:	Пользователь вносит данные о фирмах, находящихся внутри узлов цепи поставок
Условие-триггер:	Пользователь вошел в приложение, чтобы получить оптимальные по

	критерию прибыли значения для фирм определенной цепи поставок.
Предварительные условия:	Пользователь сформировал древовидный граф цепи поставок
Входные условия:	Название фирмы; Издержки фирмы; Коэффициенты a , b линейных функций цен терминальных рынков;
Выходные условия:	Сохраненные данные о фирмах хранящийся внутри узлов
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь дважды нажимает на выбранную <i>корневую</i> или <i>промежуточную</i> вершину ЛКМ 2. В открывшемся окне он вбивает в таблицу названия фирм и значения их издержек 3. После того как все данные были внесены, пользователь нажимает «ОК» 4. Система сохраняет информацию о фирмах и привязывает их к определенному узлу цепи поставок
Альтернативный поток:	<ol style="list-style-type: none"> 1. Пользователь дважды нажимает на выбранную <i>терминальную</i> вершину ЛКМ 2. В открывшемся окне он вбивает коэффициенты линейной

	<p>функции цены для рынка, на который выходит данный узел</p> <p>3. После того как все данные были внесены, пользователь нажимает «ОК»</p> <p>5. В открывшемся окне он вбивает в таблицу названия фирм и значения их издержек</p> <p>6. После того как все данные были внесены, пользователь нажимает «ОК»</p> <p>4. Система сохраняет информацию о фирмах и привязывает их к определенному узлу цепи поставок. Также сохраняет данные рынка.</p>
Исключения:	<p>Пользователь не ввел никаких данных.</p> <p>1. Система ничего не делает, ожидает дальнейшего выбора пользователя.</p>
Идентификатор ВИ	UC-05
Наименование:	Расчет оптимальных значений
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-08
Действующее лицо:	Пользователь

Описание:	Расчет оптимальных значений объемов и цен с точки зрения прибыли.
Условие-триггер:	Пользователь вошел в приложение, чтобы получить оптимальные по критерию прибыли значения для фирм определенной цепи поставок
Предварительные условия:	Пользователь сформировал древовидный граф цепи поставок. Также внес информацию о всех фирмах и рынках в цепи
Выходные условия:	Таблица с рассчитанными значениями цен, объемов, прибыли для каждой фирмы всех узлов
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь сформировал и заполнил данные в цепи поставок 2. Пользователь выбирает опцию «Рассчитать» на панели инструментов 3. Открывается таблица с результирующими данными и двумя кнопками «Заккрыть» и «Экспорт в Excel»
Исключения:	—
Идентификатор ВИ	UC-06
Наименование:	Сохранение рассчитанных данных в Excel-файл
Автор:	Владислав Чепусов
Дата создания	14.04.2023

Требования:	FUN-15
Действующее лицо:	Пользователь
Описание:	Экспортирование значений, рассчитанных программой в Excel-файл
Условие-триггер:	Пользователь захотел сохранить данные для дальнейших манипуляций в среде Excel
Предварительные условия:	Был произведен расчет оптимальных значений программой
Выходные условия:	Excel-файл с данными
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию экспорта данных. 2. Приложение отображает диалоговое окно для выбора имени и места сохранения файла. 3. Пользователь выбирает имя и место сохранения файла и нажимает кнопку "Сохранить". 4. Приложение экспортирует данные в формат .xlsx и сохраняет файл.
Исключения:	<p>Файл с таким именем уже существует</p> <ol style="list-style-type: none"> 1. Система предлагает заменить существующий файл или переименовать текущий.

	2. В случае замены файла система спрашивает точно ли пользователь этого хочет.
Идентификатор ВИ	UC-07
Наименование:	Загрузить цепь поставок
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-14
Действующее лицо:	Пользователь
Описание:	Позволяет пользователю загружать ранее сохраненные цепи поставок из файла.
Условие-триггер:	Пользователь вошел в приложение и хочет продолжить работу с ранее сохраненной цепочкой поставок.
Предварительные условия:	—
Выходные условия:	Отображается древовидный граф цепи поставок, внутри узлов хранятся данные о фирмах.
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает на опцию «Загрузить» на панели инструментов. 2. Приложение отображает диалоговое окно для выбора файла в формате json. 3. Система отображает загруженный граф на экране.
Исключения:	Ошибка при загрузке файла.

	<ol style="list-style-type: none"> 1. Выводится сообщение об ошибке. 2. Пользователю предлагают попробовать загрузить снова файл.
Идентификатор ВИ	UC-08
Наименование:	Сохранить цепь поставок
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-13
Действующее лицо:	Пользователь
Описание:	Позволяет пользователю сохранить данные о текущей цепи поставок в формате json для дальнейшей работы с ней.
Условие-триггер:	Пользователь хочет сохранить исходные данные.
Предварительные условия:	Пользователь сформировал древовидный граф цепи поставок.
Выходные условия:	Файл с сохраненным графом формата json.
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь выбирает опцию «Сохранить» на панели инструментов. 2. Приложение отображает диалоговое окно для выбора имени и места сохранения файла.

	<p>3. Пользователь выбирает имя и место сохранения файла и нажимает кнопку "Сохранить".</p> <p>4. Приложение сохраняет файл формата json.</p>
Исключения:	<p>Файл с таким именем уже существует</p> <p>1. Система предлагает заменить существующий файл или переименовать текущий.</p> <p>2. В случае замены файла система спрашивает точно ли пользователь этого хочет.</p>
Идентификатор ВИ	UC-09
Наименование:	Сброс в начальное состояние
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-06
Действующее лицо:	Пользователь
Описание:	Система сбрасывает всю внесенную в нее информацию о цепи поставок. Отображая граф по умолчанию.
Условие-триггер:	Пользователь вошел в приложение
Предварительные условия:	—
Выходные условия:	Начальный граф, состоящий из одной корневой вершины

Основной поток:	<p>1. Пользователь выбирает опцию «Заново» на панели инструментов.</p> <p>2. Приложение удаляет информацию о текущей цепи поставок</p> <p>3. Система отображает граф по умолчанию из одной вершины</p>
Исключения:	—
Идентификатор ВИ	UC-10
Наименование:	Изменить данные в узле
Автор:	Владислав Чепусов
Дата создания	14.04.2023
Требования:	FUN-01,FUN-04,FUN-05,FUN-09 FUN-10,FUN-11
Действующее лицо:	Пользователь
Описание:	Позволяет пользователю изменять данные вершины в цепи поставок
Условие-триггер:	Пользователь решил изменить исходные данные, чтобы получить оптимальные по критерию прибыли значения для фирм определенной цепи поставок.
Предварительные условия:	Данные в узле были внесены до текущей операции
Входные условия:	<p>Название фирмы;</p> <p>Издержки фирмы:</p>

	Коэффициенты a , b линейных функций цен терминальных рынков;
Выходные условия:	Измененные сохраненные данные о фирмах хранящийся внутри узлов
Основной поток:	<ol style="list-style-type: none"> 1. Пользователь дважды нажимает на выбранную <i>корневую</i> или <i>промежуточную</i> вершину ЛКМ 2. В открывшемся окне он изменяет в таблице названия фирм и/или значения их издержек 3. После того как все данные были внесены, пользователь нажимает «ОК» 4. Система сохраняет информацию о фирмах и привязывает их к определенному узлу цепи поставок
Альтернативный поток:	<ol style="list-style-type: none"> 1. Пользователь дважды нажимает на выбранную <i>терминальную</i> вершину ЛКМ 2. В открывшемся окне он изменяет коэффициенты линейной функции цены для рынка, на который выходит данный узел 3. После того как все данные были внесены, пользователь нажимает «ОК»

	<p>5. В открывшемся окне он изменяет в таблице названия фирм и/или значения их издержек</p> <p>6. После того как все данные были внесены, пользователь нажимает «ОК»</p> <p>4. Система сохраняет измененную информацию о фирмах и привязывает их к определенному узлу цепи поставок. Также сохраняет данные рынка.</p>
Исключения:	<p>Пользователь не захотел изменять данные.</p> <p>1. Пользователь может нажать на кнопку отмены, чтобы отменить внесенные им изменения.</p> <p>2. Система сохранит исходные данные</p>

3.5.Выбор и обоснование используемых инструментов

При изучении и анализе возможных инструментов для разработки программного продукта, был сделан выбор в пользу следующих средств, библиотек и языка программирования:

- ЯП — Python (версия 3.9.13)

Python является одним из наиболее популярных языков программирования для разработки научных и инженерных приложений, включая приложения для оптимизации различных алгоритмов. Ниже приведены несколько причин, по которым Python может быть хорошим выбором для разработки десктопного приложения:

1. *Экономия рабочего времени:* Python имеет простой и понятный синтаксис, который позволяет быстрее создавать программы. Кроме того, Python имеет множество библиотек и фреймворков, помогающих автоматизировать задачи и ускорять разработку.
2. *Мощность и гибкость:* Python является мощным языком программирования, который поддерживает многопоточность и многозадачность, что делает его очень гибким и универсальным для разработки различных приложений. Python также может использоваться для создания приложений с графическим интерфейсом, что делает его подходящим для создания десктопных приложений.
3. *Простота и удобство в использовании:* Python имеет простой и понятный синтаксис, что делает его очень легким для изучения и использования. Python также имеет обширную библиотеку, которая содержит множество инструментов и функций, которые могут использоваться для разработки научных и инженерных приложений.
4. *Многофункциональность:* Python может использоваться для разработки широкого спектра приложений, включая создание веб-сайтов, научных вычислений, игр и многого другого.
5. *Математические и графические библиотеки:* Python имеет множество библиотек, включая NumPy, SciPy и Matplotlib, которые позволяют работать с математическими функциями и создавать красивые графики и диаграммы.
6. *Большое и активное сообщество:* Python имеет огромное сообщество разработчиков, которые создают и поддерживают множество библиотек и инструментов. Это делает Python идеальным выбором для разработки научных и инженерных приложений, так как вы можете легко получить доступ к широкому спектру библиотек и инструментов, которые могут помочь в оптимизации математических алгоритмов.

7. *Переносимость*: Python является кроссплатформенным языком программирования, что означает, что приложения, созданные на Python, могут работать на различных операционных системах и архитектурах, включая Windows, macOS и Linux.
8. *Библиотеки и фреймворки*: Python имеет огромную коллекцию библиотек и фреймворков, которые позволяют упростить разработку и повысить производительность приложения.
9. *Популярность*: Python является одним из самых популярных языков программирования в мире, и его использование может существенно улучшить карьерные перспективы программиста.

Python — это язык программирования, который часто используется при работе с математическими и научными расчетами. Он имеет широкий набор библиотек, таких как NumPy, SciPy, Matplotlib, которые облегчают решение сложных математических и инженерных задач. Кроме того, Python — это интерпретируемый язык, что означает, что он может очень быстро выполнить простые операции и не требует компиляции перед запуском программы.

Все эти преимущества делают его отличным выбором для математических и инженерных задач благодаря широкому набору библиотек, простому синтаксису и скорости интерпретации. Он делает решение задач легким и эффективным для программистов на любом уровне опыта и навыков.

- IDE — PyCharm (версия 2022.3.3)

PyCharm - это интегрированная среда разработки (IDE) для языка программирования Python, разработанная компанией JetBrains. Вот ряд преимуществ, по которым PyCharm может быть лучше аналогов [29]:

1. *Интеллектуальная подсветка синтаксиса и автодополнение*: PyCharm предоставляет широкие возможности, такие как автоматическое завершение кода, быстрые переходы по определениям функций, классов и модулей, более точную подсветку

синтаксиса, которая позволяет различать типы переменных (локальные, глобальные и т. д.).

2. *Инструменты анализа кода:* PyCharm включает в себя инструменты анализа кода, которые помогают выявить ошибки во время написания. Такие инструменты как Linting, проверка кода на PEP-стандарты, статический анализ и другие, повышают качество и читабельность кода.
3. *Интеграция с Git и другими системами контроля версий:* PyCharm имеет встроенную поддержку Git, Mercurial, Subversion и других систем контроля версий, что позволяет быстро и удобно работать с кодом, отслеживать изменения и управлять версиями.
4. *Интеграция с виртуальными окружениями:* PyCharm поддерживает создание и управление виртуальными средами Python. Это позволяет изолировать приложение от других и повысить его устойчивость.
5. *Наличие ряда полезных плагинов:* PyCharm предоставляет большое количество плагинов, например, Pytest, Django Support, Docker Support, и многие другие плагины, которые упрощают разработку и автоматизацию работы в IDE.
6. *Облегчение отладки:* PyCharm имеет встроенный отладчик, который помогает предотвращать ошибки, и обеспечивает более быстрый и удобный процесс отладки. Также предоставляется множество иных возможностей для анализа кода, включая поддержку отладчика PDB, условных точек останова, встроенную поддержку юнит-тестирования и многие другие инструменты. Это позволяет разработчикам быстро находить и устранять ошибки.
7. *Поддержка различных операционных систем:* PyCharm работает на Windows, Mac OS и Linux, что позволяет использовать её на различных платформах.

8. *Популярность*: PyCharm обладает широкой пользовательской базой и активным сообществом, что делает его более удобным и популярным для разработчиков различных уровней.
9. *Удобство использования*: PyCharm имеет удобный интерфейс, интуитивно понятные команды и шаблоны кода, что упрощает процесс разработки и снижает количество ошибок.
10. *Поддержка многопоточности*: PyCharm поддерживает разработку многопоточных приложений и обеспечивает удобные инструменты для отслеживания и отладки многопоточных приложений.

Таким образом, PyCharm — это один из лучших выборов для разработки информационного продукта, так как он обладает множеством инструментов, является мощной и удобной средой разработки для языка Python, идеально подходит для командной работы и имеет широкое сообщество разработчиков, которые могут помочь в случае необходимости.

- Графическая библиотека – PyQt (Версия 5.15.7)

PyQt — это библиотека Python, которая предоставляет инструменты для создания графических пользовательских интерфейсов (GUI) на основе фреймворка Qt. К значительным преимуществам PyQt можно отнести:

1. *Кроссплатформенность*: PyQt работает на всех платформах, где есть Python и Qt. Это позволяет создавать приложения для Windows, Mac, Linux, Android и iOS.
2. *Большое количество возможностей*: PyQt позволяет создавать различные типы приложений, от простых форм до сложных мультимедийных интерфейсов. Библиотека предоставляет разнообразные инструменты, такие как дизайнеры интерфейсов и инструменты редактирования CSS, которые упрощают и ускоряют процесс разработки.
3. *Открытое программное обеспечение*: PyQt имеет открытый исходный код, который может быть доступен бесплатно каждому.

4. *Широкий выбор виджетов:* PyQt предоставляет большое количество готовых виджетов, которые можно использовать в своих приложениях: кнопки, поля ввода, графические элементы, таблицы и многое другое.
5. *Простота использования:* PyQt имеет простой и понятный API, благодаря которому создание приложений становится проще и быстрее.
6. *Актуальные возможности:* PyQt постоянно обновляется и улучшается, что позволяет использовать новые возможности Qt и Python.
7. *Большое сообщество разработчиков:* PyQt имеет большое сообщество разработчиков, которые поддерживают его и разрабатывают новые библиотеки и расширения, а также имеется значительное количество форумов в интернете для обсуждения возникших проблем.
8. *Обширная документация:* PyQt имеет обширную и понятную документацию, содержащую примеры кода и руководства для различных задач.
9. *Легкая интеграция с другими инструментами:* PyQt может легко интегрироваться с другими библиотеками и инструментами Python, что делает его еще более мощным и удобным в использовании.

PyQT является Python-оберткой QT, поэтому обладает большинством его функций. За счет своих возможностей и преимуществ PyQt позволяет быстро создавать сложные приложения с элегантным пользовательским интерфейсом, обладающие хорошей производительностью и широкими возможностями настройки.

- Математический пакет - SymPy (Версия 1.11.1)

SymPy — это библиотека для символьных математических вычислений на языке Python. Она предоставляет широкий функционал для математических

операций, таких как алгебраические упрощения, дифференцирование, интегрирование, решение уравнений, построение графиков, численное интегрирование, и многое другое.

Преимущества SymPy:

1. *Символьные вычисления*: SymPy может выполнять все операции обычной математики, но используя символьные переменные вместо чисел. Это позволяет удобно работать с выражениями, которые нельзя вычислить в явном виде, или если значения переменных неизвестны.
2. *Широкий набор функций*: SymPy содержит более 100 функций для работы с символьными выражениями, включая алгебру, геометрию, теорию чисел и многое другое.
3. *Легко использовать*: SymPy поставляется с документацией и множеством примеров использования, которые помогут быстро освоить библиотеку и начать ей пользоваться.
4. *Можно интегрировать с другими библиотеками*: SymPy легко интегрируется с другими библиотеками для научных вычислений, например, NumPy и SciPy, что позволяет использовать их функции в выражениях SymPy.
5. *Open source*: SymPy является бесплатной и открытой для использования библиотекой, что делает ее доступной для всех.
6. *Кроссплатформенность*: SymPy работает на различных операционных системах, таких как Windows, Linux и macOS.

SymPy – это мощная библиотека для символьных вычислений в Python, которая может быть использована как профессионалами, так и новичками. Она имеет множество преимуществ, которые делают ее предпочтительным выбором для математических вычислений в Python.

3.6. Инструкция к запуску программы

Шаг 1: установите Python на свой компьютер

Перед тем, как можно будет запустить программу, нам нужно убедиться, что интерпретатор Python установлен и готов к работе на вашем компьютере. Можно скачать последнюю версию Python с официального сайта <https://www.python.org/downloads/>

Шаг 2: скачайте программу с Github / Склонируйте репозиторий

- Для скачивания программы с Github нужно открыть URL https://github.com/VladislavChepusov/supply_chains в веб-браузере. Далее необходимо выбрать зеленую кнопку "Code" и выбрать "Download ZIP", чтобы скачать папку с программой в ZIP-архиве на свой компьютер.
- Для клонирования сначала запустите командную строку и перейдите в каталог, в который вы хотите клонировать репозиторий. Затем выполните следующую команду:

```
git clone https://github.com/VladislavChepusov/supply\_chains
```

Шаг 3: перейдите в папку с программой

После этого, нужно открыть терминал или командную строку (в зависимости от операционной системы) и перейти в папку с программой с помощью команды "cd" (например, "cd Desktop/supply_chains-master").

Шаг 4: установите все зависимости

Перейдите в каталог репозитория, установите все необходимые зависимости, выполнив следующие команды:

```
cd supply_chains  
pip install -r requirements.txt
```

Шаг 5: запустите программу

После установки всех зависимостей, мы можем запустить программу. В папке с программой нужно открыть терминал или командную строку и запустить файл "main.py" с помощью Python с помощью команды:

python main.py

Программа должна начать работать.

3.7.Пример работы ПО со скриншотами интерфейса

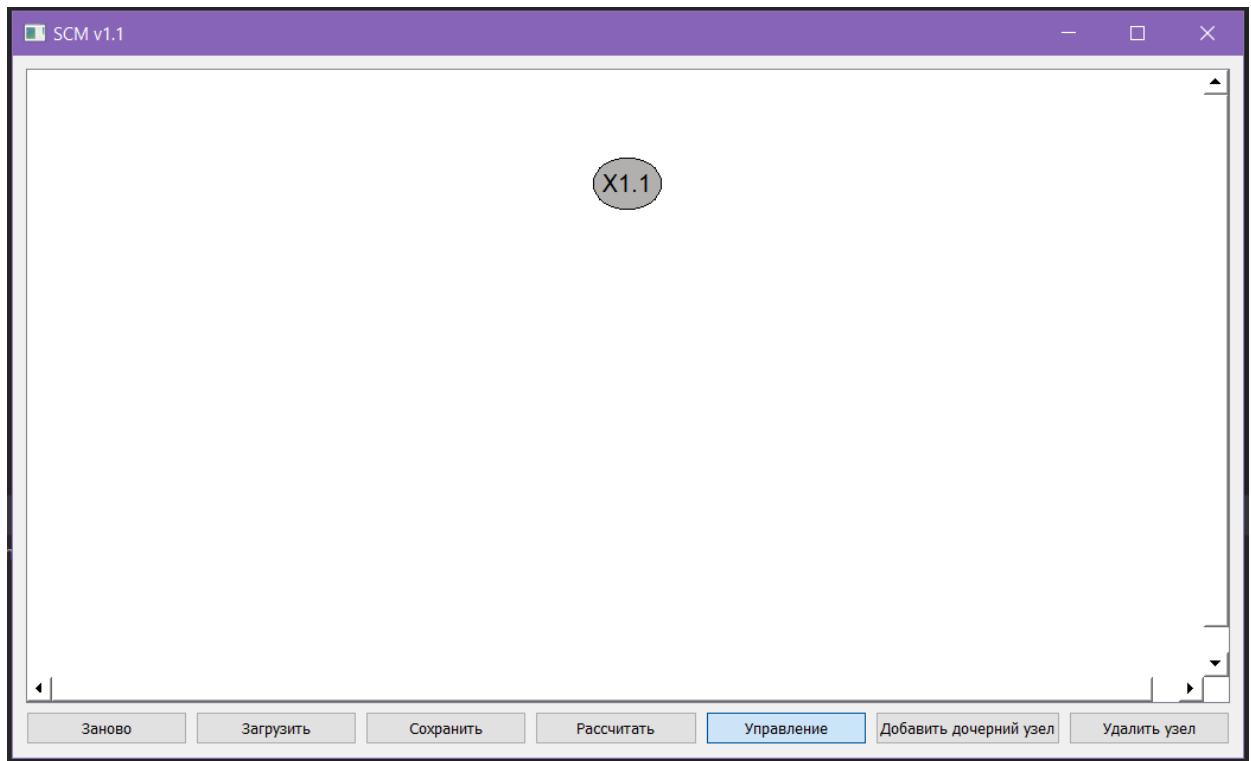


Рисунок 7 – Начальное состояние системы

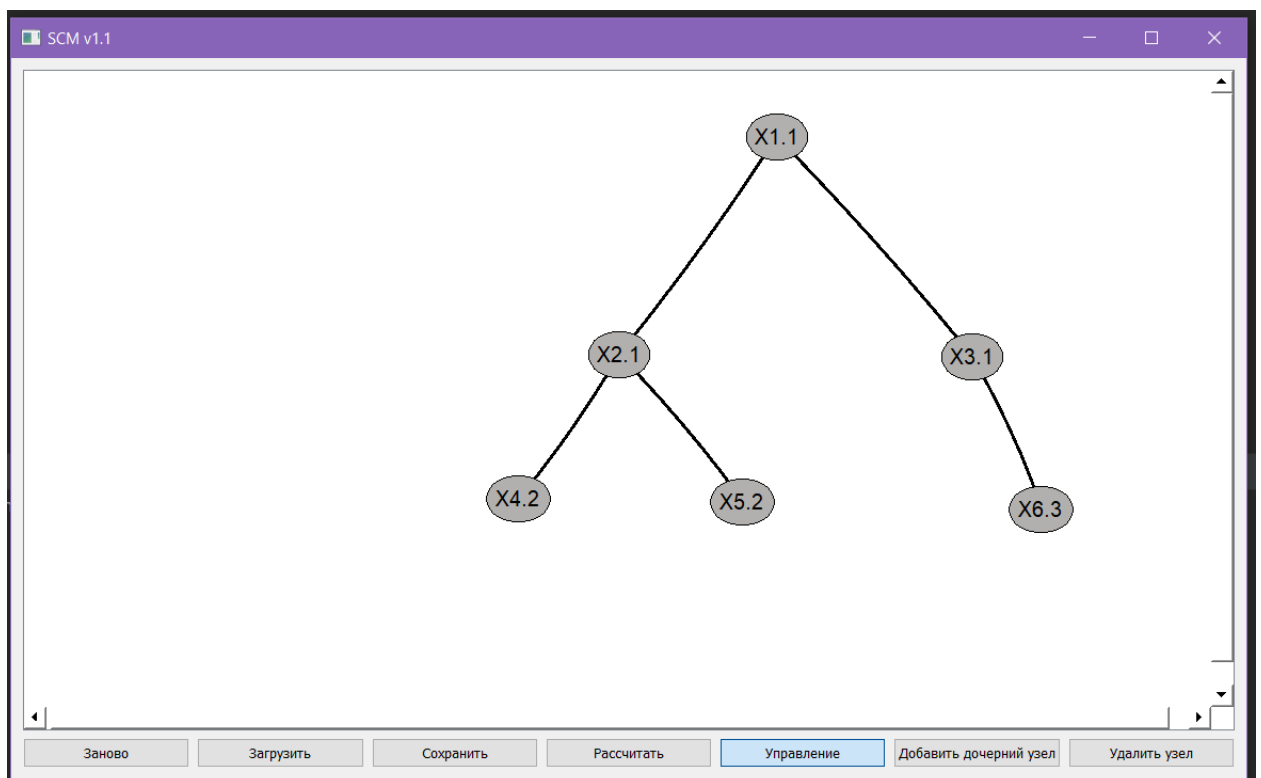


Рисунок 8 – Добавили несколько узлов в цепь

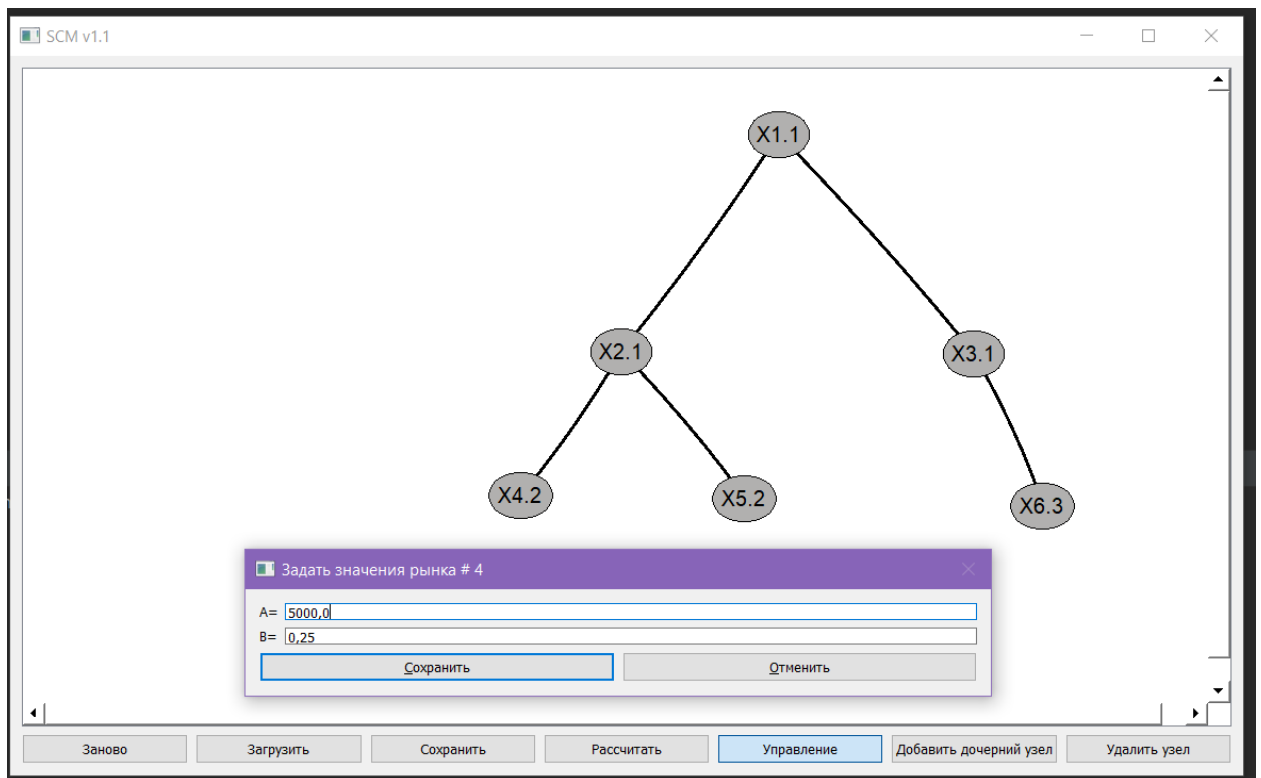


Рисунок 9 – Задаем значения коэффициентов (а,в) линейный функций цены для терминального рынка №4

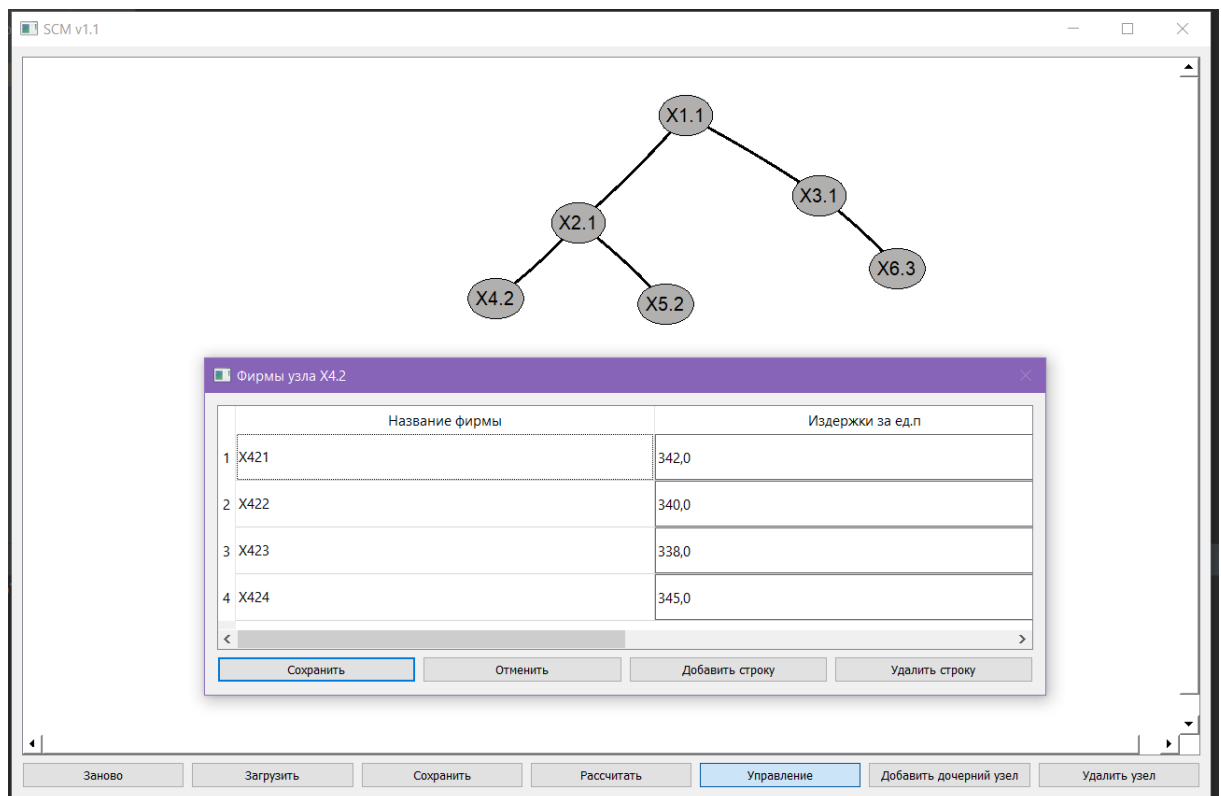


Рисунок 10 – Заполняем данные об узле 4(сколько содержится фирм в узле, их название и издержки каждой компании)

Итоговый расчет				
	Фирма	Издержки	Объем	Прибыль
1	Узел №1			
2	Цена на узле №1 = 2583.06102362204			
3	X111	1500.0	8151.03703705105	8828070.51692969
4	X112	1505.0	8113.40740742136	8746748.29470733
5	Узел №2			
6	Цена на узле №2 = 4396.97045594510			
7	X211	700.0	11815.6911635925	13161609.8365420
8	Узел №3			
9	Цена на узле №3 = 3965.37401574803			
10	X311	720.0	2207.70997375196	1462194.99846205
11	X312	710.0	2241.04330708527	1506682.53127042
12	Узел №4			
13	Цена на узле №4 = 4790.57636475608			
14	X421	342.0	206.423635243917	10652.6792968282
15	X422	340.0	214.423635243917	11494.3738378040
16	X423	338.0	222.423635243917	12368.0683787796
17	X424	345.0	194.423635243917	9450.13748536468
Ок		Экспорт в Excel		

Рисунок 11 – Получения итоговых значений после заполнения всей доступной информации о цепи

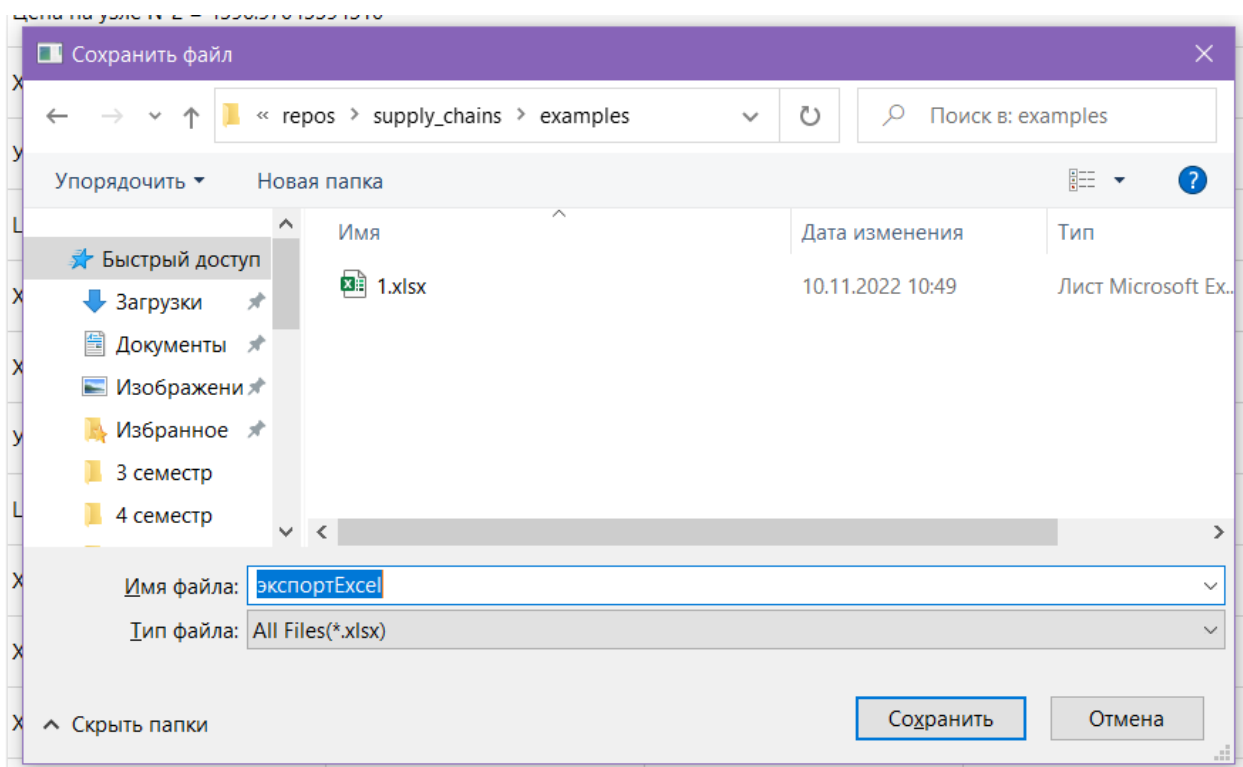


Рисунок 12 – Сохранение полученных результатов в формате .xlsx

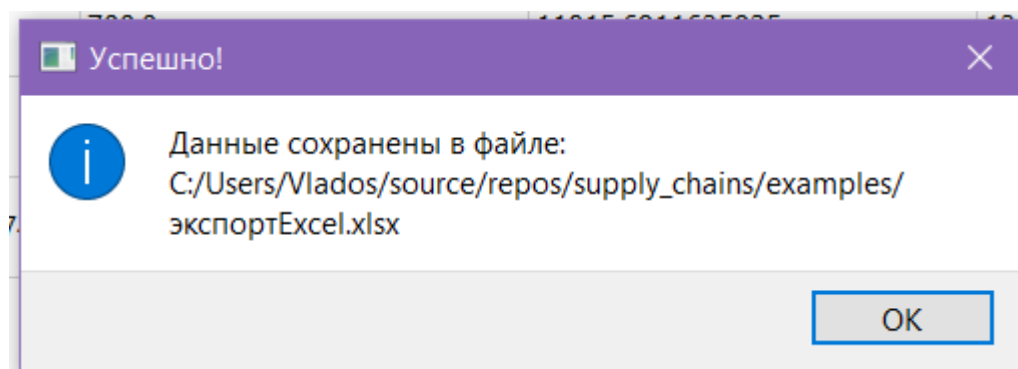


Рисунок 13 – Сообщение об успехе + полное имя файла

экспортExcel.xlsx - Excel Владислав Чепусов

Файл Главная Вставка Разметка страниц Формулы Данные Рецензирование Вид Справка Помощь Поделиться

Вставить Шрифт Выравнивание Число Условное форматирование Форматировать как таблицу Ячейки Редактирование

Буфер обмена Стили

G12

	A	B	C	D	E	F	G
1	Фирма	Издержки	Объем	Прибыль			
2	Узел №1						
3	Цена на узле №1 = 2583.06102362204						
4	X111	1500.0	8151.03703705105	8828070.51692969			
5	X112	1505.0	8113.40740742136	8746748.29470733			
6	Узел №2						
7	Цена на узле №2 = 4396.97045594510						
8	X211	700.0	11815.6911635925	13161609.8365420			
9	Узел №3						
10	Цена на узле №3 = 3965.37401574803						
11	X311	720.0	2207.70997375196	1462194.99846205			
12	X312	710.0	2241.04330708527	1506682.53127042			
13	Узел №4						
14	Цена на узле №4 = 4790.57636475608						
15	X421	342.0	206.423635243917	10652.6792968282			
16	X422	340.0	214.423635243917	11494.3738378040			
17	X423	338.0	222.423635243917	12368.0683787796			
18	X424	345.0	194.423635243917	9450.13748536468			
19	Узел №5						
20	Цена на узле №5 = 5011.98030396340						
21	X521	120.0	5500.10942242554	2722608.32927888			
22	X522	122.0	5477.88720020332	2700652.33603363			
23	Узел №6						
24	Цена на узле №6 = 4832.68700787402						
25	X631	200.0	4448.75328083990	2968710.86306756			
26							
27							
28							
29							
30							
31							

Sheet1

100 %

Рисунок 14 – Экспорт данных в Excel. Отображение данных

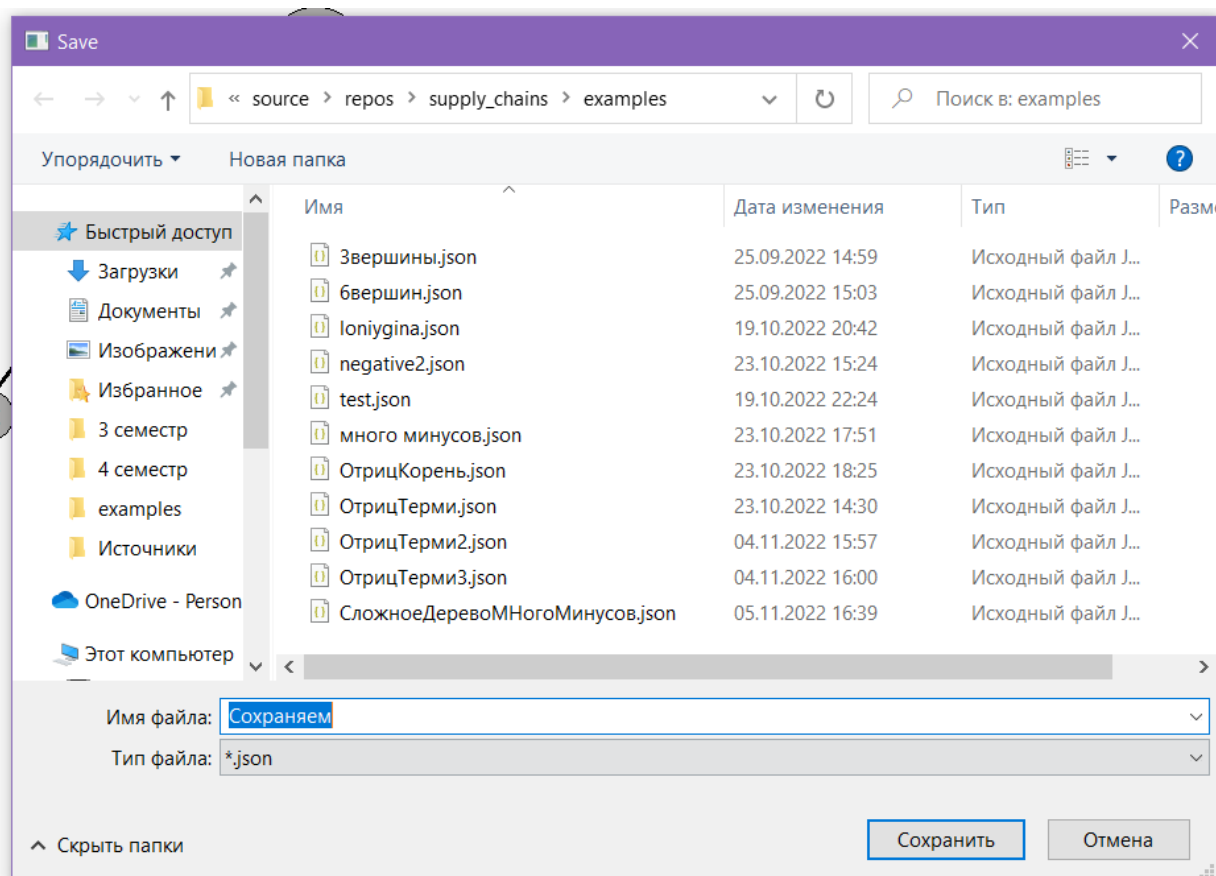


Рисунок 15 – Сохранение структуры цепи поставок и информации о фирмах, содержащихся в ней в формат json

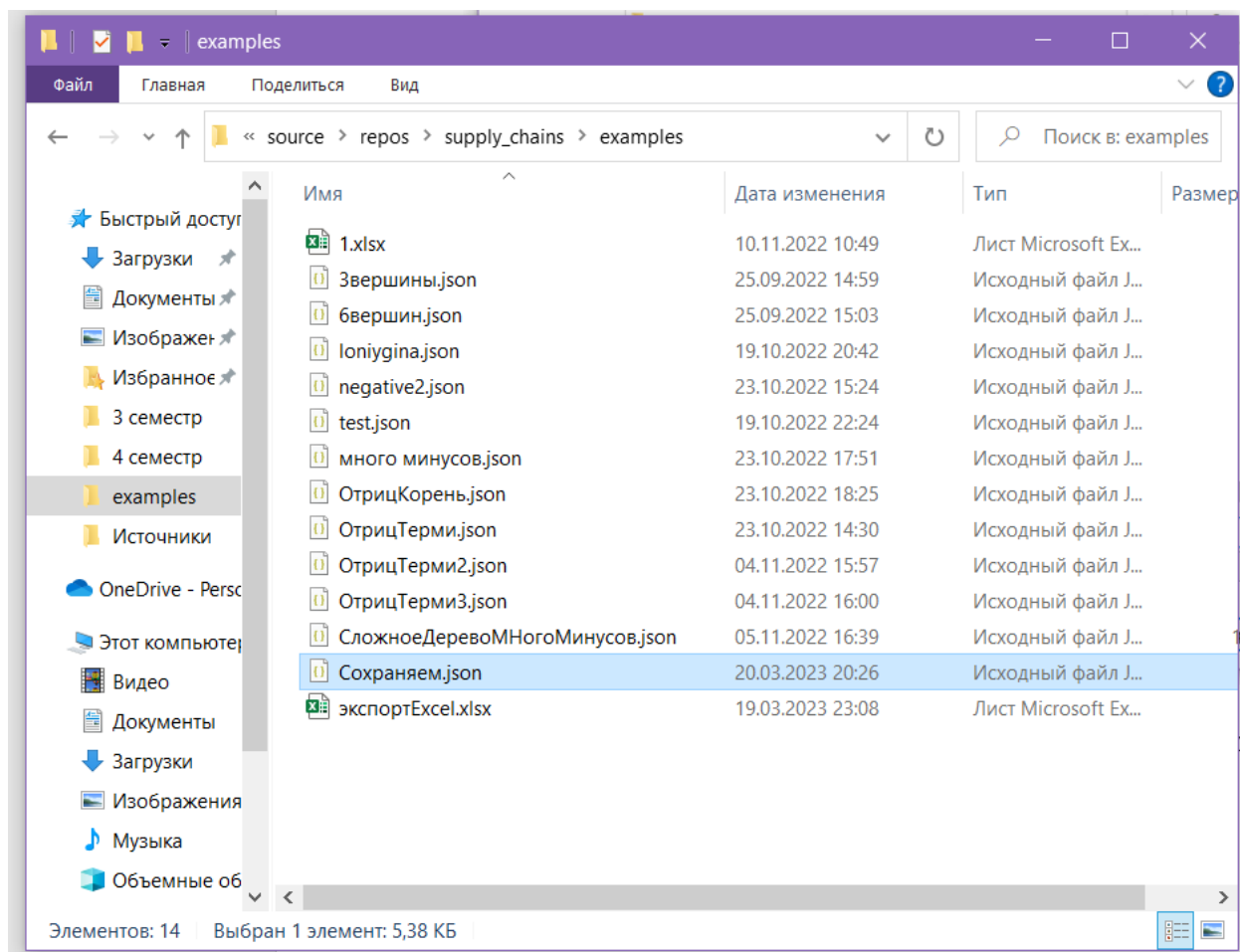


Рисунок 16 – Скриншот директории куда сохранили структуру с демонстрацией самого файла

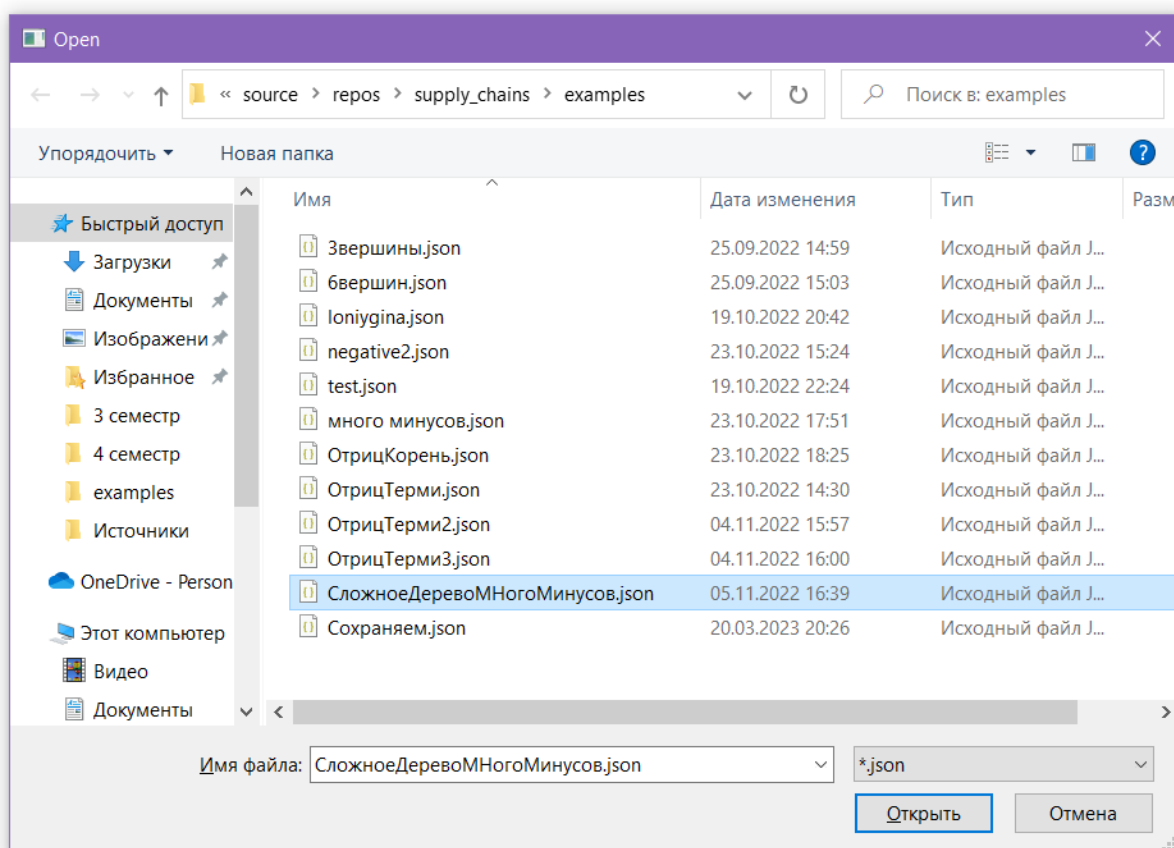


Рисунок 17– загрузка структуры цепи поставок и информации о фирмах, содержащихся в ней из формата json

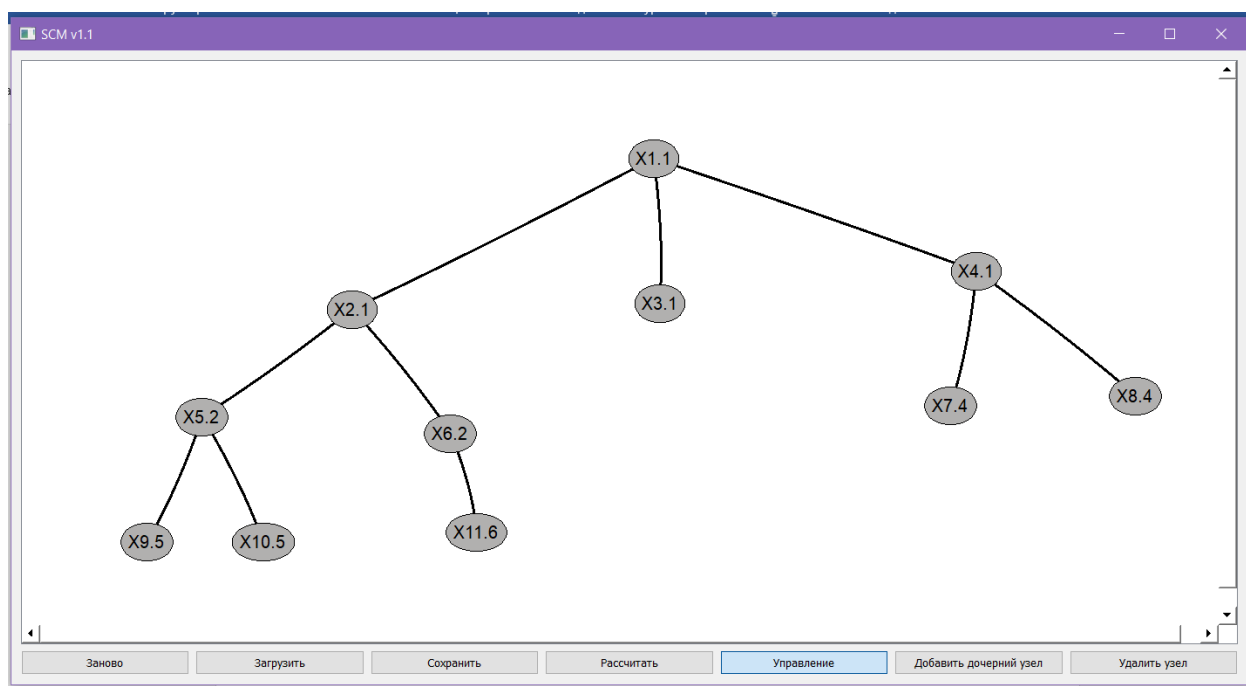


Рисунок 18 – визуальный результат загрузки данных

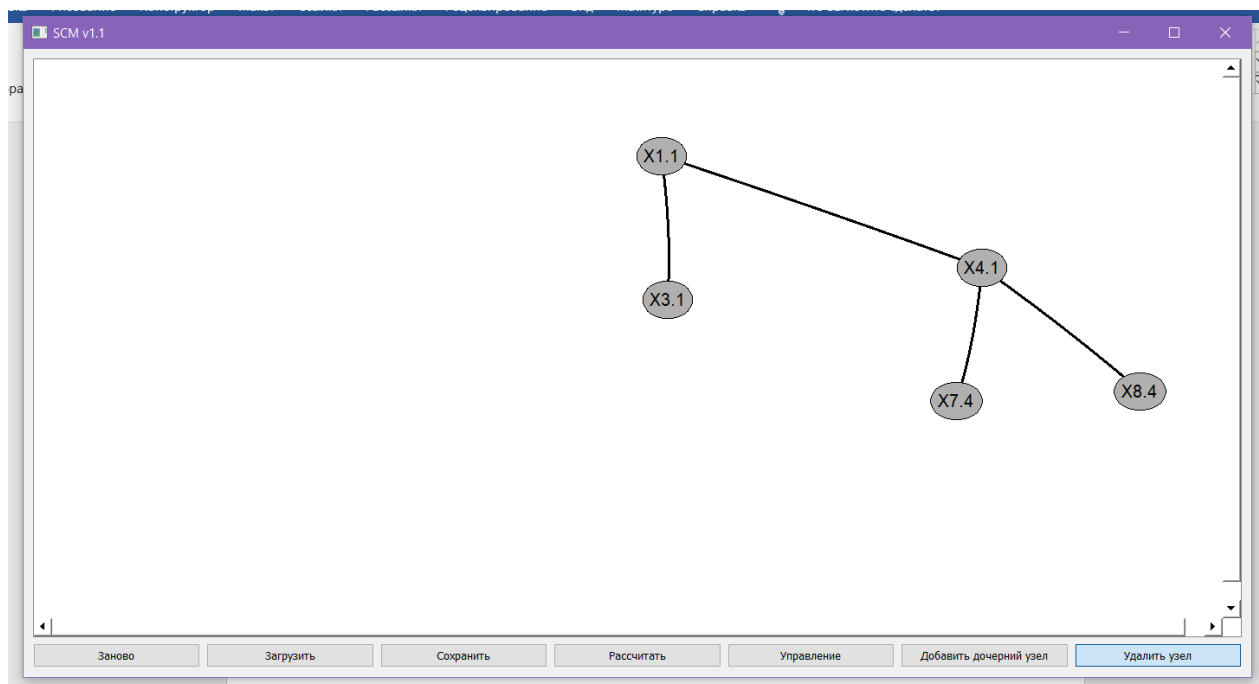


Рисунок 19 – функция удаление узла на примере X2.1

ЗАКЛЮЧЕНИЕ

В данном исследовании была создана математическая модель для древовидной цепи поставок, а также был разработан способ поиска оптимального решения относительно прибыли участников, конкурирующих по экономической модели Курно и реализовано программное обеспечение по данному алгоритму.

Программный инструментарий прошел проверку на основе тестовых данных.

Подводя итоги, можно сказать, что в результате успешного выполнения всех поставленных в данной работе задач, цель дипломного проектирования была достигнута.

Выделим несколько потенциальных направлений, которые могут быть исследованы в дальнейшем:

1. Расширение цепи поставок благодаря добавлению нового типа узлов-поставщиков.
2. Использование централизованного подхода для улучшения эффективности цепи поставок.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Дегтерев Д.А. Зарубежные работы по теории игр / Д.А. Дегтерев // Международные процессы. - 2009. - № 2 (20), май-август.
2. Петросян Л.А., Зенкевич Н.А., Шевкопляс Е.В., 2014. Теория игр / 2-ое издание, Санкт-Петербург
3. Н. А. Зенкевич, Л. А. Петросян, Д. В. К. Янг. ДИНАМИЧЕСКИЕ ИГРЫ И ИХ ПРИЛОЖЕНИЯ В МЕНЕДЖМЕНТЕ /Издательство: «Высшая школа менеджмента», 2009 – 417 С.
4. Крылатков П. П., Прилуцкая М. А., 2018. Управление цепью поставок (SCM) / 2-ое издание, Екатеринбург
5. Christopher, M. Logistics and Supply Chain Management: Strategies for Reducing Cost and Improving Service / M. Christopher –London, 2011.
6. Deming Zhou Uday S. Karmarkar. Competition in MultiEchelon Distributive Supply Chains with Linear Demand // International journal of production research, London, 2015. 31 с
7. Парфёнов Михаил Александрович Эволюция управления цепями поставок: от интегрированной логистики к сетевым структурам // Вестник РГЭУ РИНХ. 2010. №32.
8. Щербаков В. В., Логистика и управление цепями поставок : учебник для академического бакалавриата / В. В. Щербаков [и др.] ; под редакцией В. В. Щербакова. — Москва / Издательство: Юрайт, 2019. — 582 с.
9. Tyagi R.K. On the effect of downstream entry // Management science, 1999. № 45. P. 59-73
10. Ziss S. Vertical separation and horizontal mergers // Journal of industrial economics, 1995. №43. P. 63-75
11. Cachon G.P. Supply chain coordination with contracts // Handbooks in Operations Research & Management Science, 2003. Vol. 11. P. 227-339

- 12.Kaya M., Ozer O. Pricing in business-to-business contracts: sharing risk, profit and information // The Oxford Handbook of Pricing Management. Oxford: Oxford University Press, 2012. P. 738-783.
- 13.Adida E., DeMiguel V. Supply Chain competition with multiple manufacturers and retailers // Operation Research, 2011. Vol. 59, №1. P.156-172.
- 14.Lee H. Decentralized multi-echelon supply chains: incentives and information / H. Lee, S. Whang // Management Science – 1999. – Vol. 54, №5. – P.633-640.
- 15.Corbett, C.J., & Karmarkar, U.S. Competition and Structure in Serial Supply Chains with Deterministic Demand / C.J. Corbett, U.S. Karmarkar // Manag. Sci. - 2001. - T. 47. - С. 966-978.
- 16.Carr, S.M. & Karmarkar, U.S. Competition in Multiechelon Assembly Supply Chains / S.M. Carr, U.S. Karmarkar // Manag. Sci. - 2005. - T. 51. - С. 45-59.
- 17.Ю.Е. Лонягина, Н. А. Зенкевич. Модели конкуренции и кооперации многоуровневых сетей поставок / Издательство: «Санкт-Петербургский Государственный Университет», 2018 – 90 с.
- 18.Захаров, А. В. Теория игр в общественных науках: учебник для вузов / А. В. Захаров; Нац. исслед. ун-т «Высшая школа экономики». — М.: Изд. дом Высшей школы экономики, 2015. — (Учебники Высшей школы экономики). — 304 с. — 1500 экз.
- 19.Mas-Collel A., Whinston M.D., Green J.R. Microeconomic theory. — N.-Y.: Oxford Univ. Press, 1995. — 981 p.
- 20.The Theory of Industrial Organization / Jean Tirole. – Cambridge, Massachusetts: The MIT Press, 1988. – 496 с. – ISBN 978-0-262-20071-4.
- 21.Васин А. А., Морозов В. В. Теория игр и модели математической экономики. — М.: МГУ, 2005, 272 с

- 22.Лонягина Ю., Никольченко Н., Зенкевич Н. Конкурентное и кооперативное поведение в распределительных сетях // Вклад в теорию игр и менеджмент. 2018. Т. 11. С. 73–102.
- 23.Дюсуше О. М. Статичное равновесие Курно–Нэша и рефлексивные игры олигополии: случай линейных функций спроса и издержек // Экономический журнал ВШЭ. 2006. №1.
- 24.Компания GMDH [Электронный ресурс] – Нью-Йорк, NY 10006 Соединенные Штаты. Автор: Кошулько А. – URL: <https://gmdhsoftware.com/ru/scm-software/>? (дата обращения: 20.03.2023).
- 25.Вигерс Карл, Битти Джой Разработка требований к программному обеспечению. 3-е изд., дополненное/ Пер. с англ. – М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. -736стр.: ил.
- 26.830-1998 - IEEE Recommended Practice for Software Requirements Specifications. - NY: The Institute of Electrical and Electronics Engineers, 2009. - 31 с
- 27.Writing Software Requirements Specifications (SRS) [Электронный ресурс]. — INKtopia Limited, 2020 — URL: <https://techwhirl.com/writing-software-requirements-specifications/> (дата обращения: 20.03.2023)
- 28.SimbirSoft. Нефункциональные требования: как не пустить систему ко дну [Электронный ресурс]. Блог компании SimbirSoft. Автор: Елена Иванова, ведущий аналитик ИТ-компании SimbirSoft. URL: <https://habr.com/ru/company/simbirsoft/blog/688428/> (дата обращения: 20.03.2023).
- 29.JetBrains. PyCharm Documentation [Электронный ресурс]: Quick Start Guide. URL: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html> (дата обращения: 21.03.2023).
- 30.Мовсесян А.А., Стародубцева О.А. ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ МАРКЕТИНГА НА РЫНКАХ B2B И B2C //

Инфраструктурные отрасли экономики: проблемы и перспективы развития. 2016.

- 31.Воронцова, Е. А., Хильдебранд, Р. Ф., Гасников, А. В., & Стонякин, Ф. С. Выпуклая оптимизация: учебное пособие / Е. А. Воронцова, Р. Ф. Хильдебранд, А. В. Гасников, Ф. С. Стонякин. - Москва: МФТИ, 2021. - 364 с. -