

Тема диплома:
**«Моделирование типовых складских процессов.
Алгоритмы расчета параметров зоны отбора склада».**

Содержание

Обозначения.....	3
1. Постановка задачи.....	4
1.1. Определение зоны отбора склада	4
1.2. Разработка алгоритма пополнения запасов	4
1.3. Основные гипотезы для построения зоны отбора. Простые модели.....	5
1.3.1. Определение количества каждого товара для размещения в зоне отбора.....	5
1.3.2. Определение видов товаров, представленных в зоне отбора.....	6
1.4. Усложнение модели зоны отбора	7
1.4.1. Учет динамики рынка	9
1.4.2. Перерасчет зоны отбора	10
1.4.3. Суммарные характеристики при многократном перерасчете зоны отбора...	10
2. Математические методы для решения дискретных задач оптимизации	13
2.1. Метод вектора спада.....	13
2.1.1. Основные положения и определения.....	13
2.1.2. Описание общей схемы метода.....	14
2.1.3. Замечания.....	15
2.1.4. Луч-метод.....	17
2.1.5. Алгоритмы приближенного решения задач выпуклого целочисленного программирования луч-методом	19
2.1.6. Направляющие лучи и способы их выбора.....	20
2.1.7. Способы определения целочисленных точек со взаимно простыми координатами (точек окрестности $O(x^0)$)	21
2.1.8. Луч-метод для решения целочисленного вогнутого программирования	23
3. Результаты моделирования операций отбора/пополнения на складе продуктов питания.....	27
3.1. Описание входных данных	27
3.2. Сценарии моделирования.....	27
3.2.1. Рассмотрение существующей зоны отбора	27
3.2.2. Варьирование размера зоны отбора	30
3.3. Выводы	36
4. Экономическая часть	37
4.1. Введение	37
4.2. Расчет себестоимости программного продукта	37
4.2.1. Материальные затраты	38
4.2.2. Затраты на оплату труда	38
4.2.3. Единый социальный налог	39
4.2.4. Расходы на амортизацию.....	39
4.2.5. Итоговые результаты	40
4.3. Расчет цены программного продукта	40
4.4. Экономическая эффективность разработки	40
5. Охрана труда и окружающей среды	42
5.1. Введение	42
5.2. Причины опасности работы за компьютером	42
5.2.1. Устройство вывода информации.....	43
5.2.2. Устройства ввода информации	45
5.3. Выводы	45
Приложение 1. Графическое представление оптимального решения для части товаров...	47
Список литературы	48

Обозначения

- c_1 - стоимость отбора из зоны отбора [у.е.] (коробочный отбор),
- c_2 - стоимость отбора из зоны хранения [у.е.] (палетный отбор),
- c_r - стоимость перемещения товара из зоны хранения в зону отбора [у.е.],
- m_1 - стоимость перемещения из зоны хранения в зону отбора (при реорганизации зоны отбора) [у.е./палета],
- m_2 - стоимость перемещения из зоны отбора в зону хранения (при реорганизации зоны отбора) [у.е./палета],
- d_i - количество коробочных отборов i -го товара [безразмерная],
- D_i - количество палетных отборов i -го товара [безразмерная] ,
- p_i - объем палеты i -го товара [м^3],
- b_i - объем коробки i -го товара [м^3],
- f_i - поток i -го товара за период (учитывает поток i -го товара предназначенного для штучного отбора, т.е. число коробок i -го товара) [коробка],
- F_i - поток i -го товара за период [м^3],
- v_i - количество палето-мест, выделенных под i -й товар в зоне отбора [палета],
- x_i - **переменная, определяющая размещение i -го товара в зоне отбора,**
- y_i - **переменная, определяющая присутствие заказов (среди заказов на рассматриваемом горизонте) на штучный отбор,**
- s_i - количество палето-мест, выделенных под i -й товар в зоне отбора, в результате прошлого расчета,
- I_1 - затраты на отбор/пополнение,
- I_2 - затраты на реорганизацию зоны отбора,
- I - суммарные затраты на отбор/пополнение и реорганизацию,
- $R_{\text{попол}}$ - суммарное число пополнений товаров,
- $W_{\text{деф}}$ - суммарный относительный дефицит товаров,
- $F_{\text{пал}}$ - объем зоны отбора,
- $P_{\text{зак}}$ - общее количество заказов,
- $G_{\text{тов}}$ - количество товаров, присутствующих в рассматриваемых заказах,
- $L_{\text{стр}}$ - суммарное количество строк в заказах,
- $P_{\text{зак}}^0$ - число заказов, используемое для первоначального расчета зоны отбора,
- $K_{\text{цикл.зак}}$ - количество заказов, при накоплении которых производится перерасчет зоны отбора.

1. Постановка задачи

1.1. Определение зоны отбора склада

Одной из основных задач управления складом является выбор оптимальной топологии, т.е. структуры хранения товаров, с целью их эффективного учета и контроля. Как правило, всю рабочую площадь склада разделяют на две зоны [1]. Первая зона – зона хранения товара, в которой товар размещается и хранится при поступлении на склад. Вторая зона – зона отбора товара (активная зона) предназначена для комплектации поступающих заказов на отгрузку товара со склада.

Зона отбора, представляет собой «склад внутри склада». Здесь могут располагаться все товары в небольших количествах, поэтому комплектовщики заказов тратят меньше времени на поиск нужных товаров, так как им не приходится искать товар по всему складу. Использование зоны отбора связано с издержками необходимыми на пополнение товара в зоне отбора из зоны хранения. Основными параметрами зоны отбора являются:

- размещаемые виды товаров;
- пространство, выделяемое под каждый товар.

Отсюда возникает задача поиска оптимальных параметров зоны отбора склада с целью минимизации издержек, связанных с отбором и пополнением товаров.

1.2. Разработка алгоритма пополнения запасов

Рассмотрение зоны отбора связано и с последующим пополнением товаров в ней. Для этого рассмотрим стоимость пополнения, которая зависит от:

- числа необходимых пополнений товара за период;
- количества пополняемого товара;
- времени, в которое происходит пополнение (во время отбора или в какое-то менее критичное время)

Предполагается, что стоимость пополнения зависит от числа необходимых пополнений товара за период. Сразу становится ясно, что число пополнений зависит, от типа хранения товара. Например, товар отбирается коробками, которые размещены на палете (подставка для хранения товаров в коробках). В зоне хранения товар размещен на палетах, поэтому при его перемещении в зону отбора пополнение происходит целыми палетами, т.е. стоимость одного пополнения – стоимость перемещения одной палеты. Здесь и далее стоимость пополнения будет измеряться в условных единицах (стоимость пополнения рассчитывается, исходя из общего числа пополнений и заработной платы персонала, выполняющего данную работу).

Вначале мы рассмотрим модель, в которой и отбор и пополнение измеряются коробками, т.е. стоимость одного пополнения – стоимость перемещения одной коробки и отбор одной единицы товара – отбор одной коробки.

Пусть под i -й товар выделен объем v_i в зоне отбора, тогда возникает вопрос, как часто необходимо пополнять i -й товар. Это зависит от потока f_i этого товара через склад за период $K_{\text{цикл.зак}}$. Поток, обычно измеряют в кубических метрах за единицу времени, и может быть определен следующим образом:

$$\text{поток_товара} \left[\frac{\text{м}^3}{\text{период}} \right] = \frac{\text{объем_товара} \left[\frac{\text{шт}}{\text{период}} \right]}{\text{объем_товара} \left[\frac{\text{шт}}{\text{коробка}} \right]} \text{объем_коробки} \left[\frac{\text{м}^3}{\text{коробка}} \right]$$

В сделанных выше предположениях, число пополнений i -го товара будет равно $\frac{f_i}{v_i}$.

1.3. Основные гипотезы для построения зоны отбора. Простые модели

1.3.1. Определение количества каждого товара для размещения в зоне отбора

Предположим, что мы определили какие товары размещать в зоне отбора, теперь необходимо определить пространство под каждый товар.

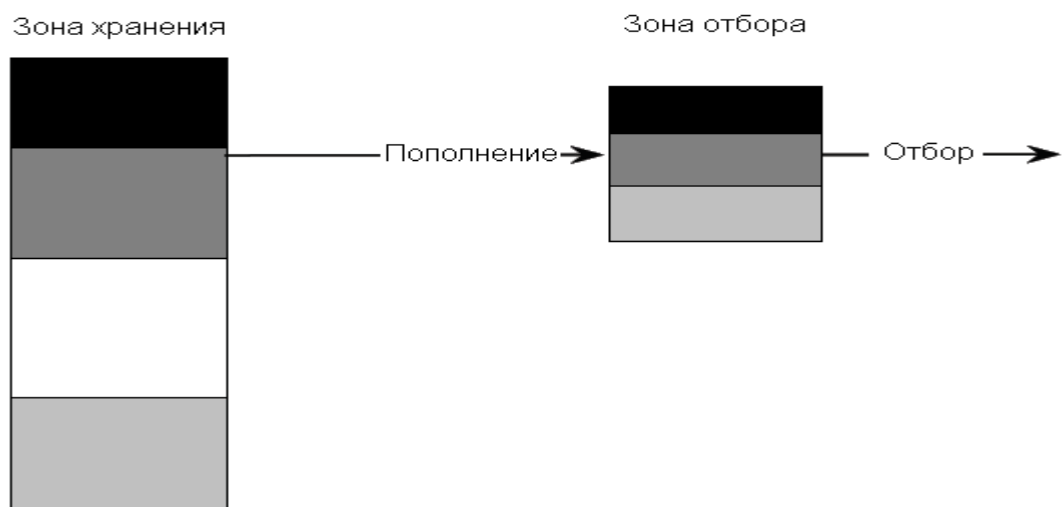


Рис. 1. Определение количества каждого товара для размещения в зоне отбора

Стоимость хранения объема v_i (в кубических метрах) i -го товара зависит от стоимости его последующего пополнения, чем больше запас товара в зоне отбора, тем реже мы его пополняем и тем меньше пространства под остальные товары.

Рассмотрим следующие предположения:

- пополнение i -го товара происходит только тогда, когда его объем v_i исчерпан (стратегия пополнения);
- стоимость каждого пополнения не зависит от пополняемого количества

В этих предположениях, стоимость пополнения i -го товара за период равна $c_r \frac{f_i}{v_i}$, где

c_r - стоимость одного пополнения.

Тогда вопрос об определении пространства под каждый товар в зоне отбора объема $F_{\text{нал}}$, может быть сведен к решению следующей задачи оптимизации (минимизация суммарных затрат на пополнение, при заданном размере зоны отбора):

$$\begin{aligned} \min_{v_i} \sum_{i=1}^N c_r \frac{f_i}{v_i} \\ \sum_{i=1}^N v_i \leq F_{нал} \\ v_i \geq 0 \end{aligned} \quad (1.1)$$

где N – количество видов товара на складе.

1.3.2. Определение видов товаров, представленных в зоне отбора

На каждом складе есть товары, на которые спрос меньше чем на остальные. Эти товары следует размещать в зоне хранения и от туда же производить их отбор, чтобы избежать дополнительных затрат на пополнение. Таким образом такая стратегия снизит суммарные издержки связанные с пополнением товара. Основной нашей задачей является расчет параметров зоны отбора, поэтому будем предполагать, что объем зоны хранения достаточно большой.

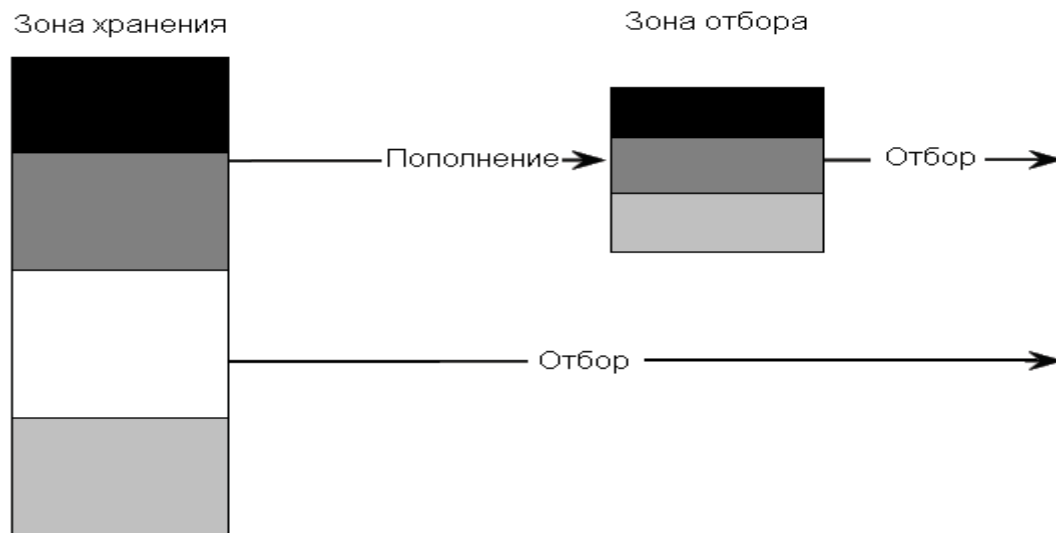


Рис. 2. Определение видов товаров, представленных в зоне отбора

Введем следующие обозначения:

c_1 - стоимость отбора одной единицы товара из зоны отбора;

c_2 - стоимость отбора одной единицы товара из зоны хранения;

x_i - переменная, определяющая размещение i -го товара в зоне отбора;

$$x_i = \begin{cases} 1, & v_i \neq 0 \\ 0, & v_i = 0 \end{cases}.$$

Тогда суммарная стоимость обработки (хранение, отбор, пополнение) i -го товара равна $c_1 + c_r$, если товар расположен в зоне отбора.

Пусть p_i - число отборов i -го товара в планируемом периоде. Тогда задача о расчете параметров зоны отбора может быть записана в следующем виде

$$\begin{aligned}
& \min_{v_i, x_i} \sum_{i=1}^N \left(c_1 p_i + c_r \frac{f_i}{v_i} \right) x_i + c_2 p_i (1 - x_i) \\
& \sum_{i=1}^N v_i \leq F_{\text{нал}} \\
& v_i \geq 0 \\
& x_i = \{0; 1\}
\end{aligned} \tag{1.2}$$

Минимизация суммарных затрат на отбор/пополнение, при фиксированном размере зоны отбора.

1.4. Усложнение модели зоны отбора

Основные моменты, которые не были учтены в описанной выше постановке:

- *разнородность заказов* (для комплектации необходим либо палетный отбор, либо штучный (коробочный), либо и тот и другой);
- *дискретность хранения товара* (как правило в зоне отбора под товар выделяются палето-места). Зона отбора разбита на палето-места, размеры палеты фиксированы;
- *затраты на организацию зоны отбора*.

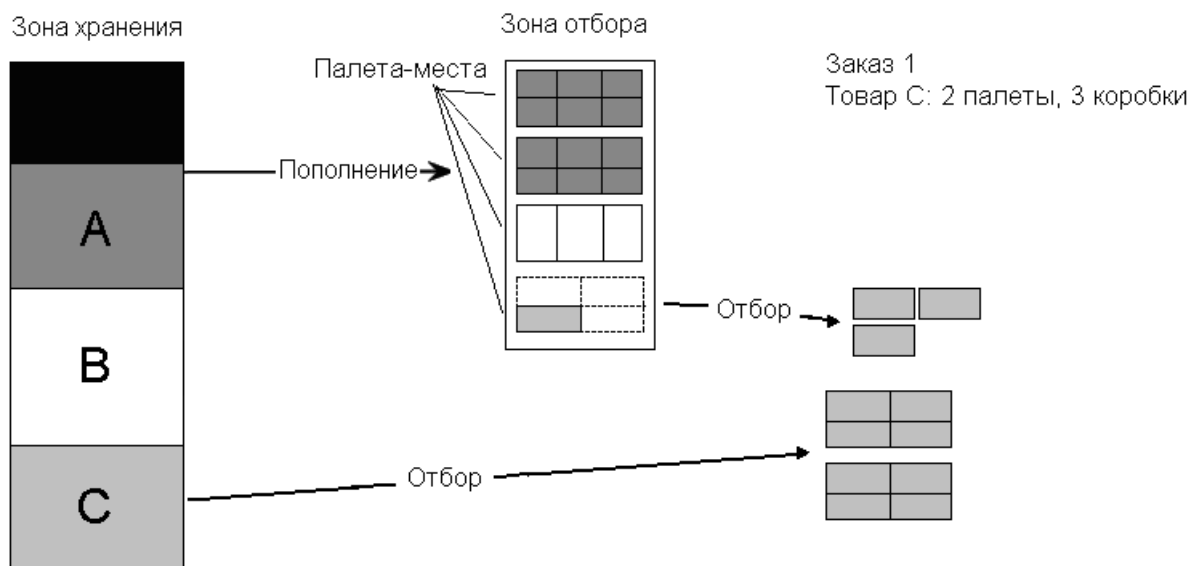


Рис. 3. Модель зоны отбора (разнородные заказы, дискретность хранения товара)

Учет разнородности заказов требует введения новых определений и обозначений, связанных с картонизацией товара (параметры упаковки).

В общем случае под картонизацией товара понимается описание всех физических параметров упаковки, в которой может храниться товар. Здесь мы не рассматриваем много аспектов, связанных с детальными параметрами упаковки товара, поэтому ограничимся следующими обозначениями:

p_i - объем палеты i -го товара [м^3];

b_i - объем коробки i -го товара [м^3].

Будем рассматривать следующую стратегию отбора: целые палеты для формирования заказа отбираются из зоны хранения, штучный отбор производится из зоны отбора. Если палетный отбор производить из зоны отбора, то требуются дополнительные затраты на пополнение.

Введем следующие обозначения:

c_1 - стоимость отбора из зоны отбора [у.е.];

c_2 - стоимость отбора из зоны хранения [у.е.];

c_3 - затраты, связанные с дефицитом товара (присутствуют заказы на коробочный отбор, но товар не представлен в зоне отбора, т.е. отбор не может быть произведен) [у.е./м³];

c_r - стоимость перемещения товара из зоны хранения в зону отбора [у.е.];

d_i - количество коробочных отборов i -го товара [безразмерная];

D_i - количество палетных отборов i -го товара [безразмерная];

f_i - поток i -го товара за период (учитывает поток i -го товара предназначенного для штучного отбора, т.е. число коробок i -го товара) [коробка];

F_i - поток i -го товара за период [м³], $F_i = f_i b_i$;

v_i - количество палето-мест, выделенных под i -й товар в зоне отбора [палета];

x_i - переменная, определяющая размещение i -го товара в зоне отбора;

$$x_i = \begin{cases} 1, v_i \neq 0 \\ 0, v_i = 0 \end{cases};$$

y_i - переменная, определяющая присутствие заказов (среди заказов на рассматриваемом горизонте) на штучный отбор,

$$y_i = \begin{cases} 1, d_i \neq 0 \\ 0, d_i = 0 \end{cases};$$

m_1 - стоимость перемещения из зоны хранения в зону отбора [у.е./палета];

$m_1 x_i$ - затраты на первоначальное размещение i -го товара в зоне отбора.

С учетом указанных предположений задача расчета параметров зоны отбора может быть формализована следующим образом:

$$\begin{aligned} \min_{x_i, v_i} \sum_{i=1}^N \left[\left(c_1 d_i + c_r \frac{F_i}{v_i p_i} \right) x_i + c_3 F_i (1 - x_i) y_i + D_i c_2 + m_1 x_i \right] \\ \sum_{i=1}^N v_i \leq F_{\text{max}} \\ x_i \in \{0; 1\} \\ v_i \geq 0, v_i - \text{целое} \end{aligned} \tag{1.3}$$

где N – количество товаров на складе.

Здесь производится минимизация суммарных затрат на отбор/пополнение, с учетом картонизации товара, а также затрат на первоначальную организацию зоны отбора, при фиксированном ее объеме.

Замечания:

- 1) Мы минимизируем суммарные издержки на отбор/пополнение, и вместе с этим пытаемся, чтобы поступающие заказы были полностью скомплектованы (зона отбора содержит как можно больше товаров, на которые присутствует спрос на рассматриваемом горизонте планирования). Последнее, достигается за счет слагаемого, содержащего c_3 (затраты, связанные с дефицитом товара).
- 2) На практике, оказывается невозможным оценить издержки c_3 , так как они зависят от очень многих параметров, которые сложно оценить количественно (репутация компании, удовлетворенность клиентов и т.п.). Поэтому, вместо слагаемого, содержащего c_3 , введем новое слагаемое – $c_1 d_i + c_r \frac{F_i}{v_i} M$ ($M \gg 1$), представляющее собой штраф в том случае, когда i -й товар не представлен в зоне отбора.

$$\min_{x_i, v_i} \sum_{i=1}^N \left[\left(c_1 d_i + c_r \frac{F_i}{v_i p_i} \right) x_i + \left(c_1 d_i + c_r \frac{F_i}{p_i} M \right) (1 - x_i) y_i + D_i c_2 + m_1 x_i \right]$$

$$\sum_{i=1}^N v_i \leq K$$

$$x_i \in \{0; 1\}$$

$$v_i \geq 0, v_i - \text{целое}$$
(1.4)

Слагаемое $\left(c_1 d_i + c_r \frac{F_i}{p_i} M \right) (1 - x_i) y_i$ имеет следующий физический смысл, если мы не размещаем i -й товар в зоне отбора, то затраты будут больше по сравнению со случаем, когда i -й товар представлен в зоне отбора.

1.4.1. Учет динамики рынка

Так как спрос на товары меняется с течением времени, возникает необходимость перерасчета параметров зоны отбора с целью минимизировать издержки, связанные с комплектацией заказов на отгрузку. То есть необходимо рассчитать какие товары являются более популярными, какие менее, изучить структуру планируемых заказов (палетный, коробочный отбор), и с учетом этого реорганизовывать зону отбора. Зона отбора не может оставаться статичной.

В качестве критерия, сигнализирующего о начале перерасчета зоны отбора, будем рассматривать некоторый цикл поступающих заказов.

Входными данными для расчета зоны отбора будет являться некоторое множество, поступивших заказов на отгрузку. После того как данные заказы выполнены, производится перерасчет зоны отбора с учетом следующей группы заказов и т. д.

1.4.2. Перерасчет зоны отбора

После того как склад запущен в работу, зона отбора и зона хранения заполнены некоторым образом. Поэтому необходимо учитывать издержки связанные с реорганизацией зоны отбора.

Введем дополнительные обозначения:

s_i - количество палето-мест, выделенных под i -й товар в зоне отбора, в результате прошлого расчета,

m_1 - стоимость перемещения из зоны хранения в зону отбора [у.е./палета],

m_2 - стоимость перемещения из зоны отбора в зону хранения [у.е./палета],

$g_i(v_i)$ - функция издержек реорганизации зоны отбора для i -го товара [у.е.],

$$g_i(v_i) = \begin{cases} m_1(v_i - s_i), v_i \geq s_i \\ m_2(s_i - v_i), v_i < s_i \end{cases}.$$

Тогда задача расчета оптимальных параметров зоны отбора с учетом реорганизации, запишется в следующем виде:

$$\begin{aligned} \min_{x_i, v_i} \sum_{i=1}^N & \left[\left(c_1 d_i + c_r \frac{F_i}{v_i p_i} \right) x_i + \left(c_1 d_i + c_r \frac{F_i}{p_i} M \right) (1 - x_i) y_i + g_i(v_i) + D_i c_2 \right] \\ \sum_{i=1}^N v_i & \leq K \\ x_i & \in \{0; 1\} \\ v_i & \geq 0, v_i - \text{целое} \end{aligned} \quad (1.5)$$

В данной постановке учитываются не только затраты на отбор/пополнение, но и затраты на реорганизацию зоны отбора.

1.4.3. Суммарные характеристики при многократном перерасчете зоны отбора

Для поддержания мобильности зоны отбора, необходимо регулярно производить ее реорганизацию. Определим суммарные характеристики данного процесса. Для этого введем следующие определения и обозначения:

Определение 1. Цикл заказов - количество заказов, при накоплении которого производится перерасчет зоны отбора.

k - количество циклов заказов,

N_j - множество (индексы) товаров, присутствующих в j -м цикле заказов

d_{ij} - количество штучных отборов i -го товара в j -м цикле заказов [безразмерная]

D_{ij} - количество палетных отборов i -го товара в j -м цикле заказов [безразмерная]

f_{ij} - поток i -го товара за период (учитывает поток i -го товара предназначенного для штучного отбора в j -м цикле заказов, т.е. число коробок i -го товара) [коробка]

F_{ij} - поток i -го товара в j -м цикле заказов [м^3], $F_{ij} = f_{ij}b_i$

v_{ij} - количество палето-мест, выделенных под i -й товар в зоне отбора в j -м цикле заказов [палета]

x_{ij} - переменная, определяющая размещение i -го товара в зоне отбора в j -м цикле заказов

$$x_{ij} = \begin{cases} 1, v_{ij} \neq 0 \\ 0, v_{ij} = 0 \end{cases}$$

y_{ij} - переменная, определяющая присутствие заказов i -го товара на штучный отбор в j -м цикле заказов

$$y_{ij} = \begin{cases} 1, d_{ij} \neq 0 \\ 0, d_{ij} = 0 \end{cases}$$

$g_{ij}(v_{ij})$ - функция издержек реорганизации зоны отбора для i -го товара в j -м цикле заказов [y.e.]

$$g_{ij}(v_{ij}) = \begin{cases} m_1(v_{ij} - s_i), v_{ij} \geq s_i \\ m_2(s_i - v_{ij}), v_{ij} < s_i \end{cases}$$

В каждом цикле заказов решается задача

$$\begin{aligned} \min_{x_{ij}, v_{ij}} \sum_{i=1}^{N_j} & \left[\left(c_1 d_{ij} + c_r \frac{F_{ij}}{v_{ij} p_i} \right) x_{ij} + \left(c_1 d_{ij} + c_r \frac{F_{ij}}{p_i} M \right) (1 - x_{ij}) y_{ij} + g_{ij}(v_{ij}) + D_{ij} c_2 \right] \\ \sum_{i=1}^{N_j} v_{ij} & \leq F_{нал} \\ x_{ij} & \in \{0; 1\} \\ v_{ij} & \geq 0, v_{ij} - \text{целое} \end{aligned} \quad (1.6)$$

$\{v_{ij}^*, x_{ij}^*\}$ - оптимальное решение задачи (1.6).

Рассмотрим следующие суммарные характеристики, которые будем рассчитывать на этапе моделирования

I_1 - затраты на отбор/пополнение

$$I_1 = \sum_{j=1}^k \sum_{i \in N_j} \left[\left(c_1 d_{ij} + c_r \frac{F_{ij}}{v_{ij}^* p_i} \right) x_{ij}^* + D_{ij} c_2 \right] \quad (1.7)$$

I_2 - затраты на реорганизацию зоны отбора

$$I_2 = \sum_{j=1}^k \left(\sum_{i \in N_j} g_{ij}(v_{ij}^*) + \sum_{i \in N_{j-1} \setminus N_j} m_2 s_i \right) \quad (1.8)$$

слагаемое $\sum_{i \in N_{j-1} \setminus N_j} m_2 s_i$ - есть затраты связанные с удалением товаров (из зоны отбора), на которые нет коробочного спроса в j-й период

I - суммарные затраты

$$I = I_1 + I_2 \quad (1.9)$$

$R_{\text{попол}}$ - суммарное число пополнений товаров

$$R_{\text{попол}} = \sum_{j=1}^k \sum_{i \in N_j} \frac{F_{ij}}{v_{ij}^* p_i} x_{ij}^* \quad (1.10)$$

Определение 2. Суммарный относительный дефицит товаров - суммарное число раз, когда невозможно осуществить коробочный отбор товаров, из-за их отсутствия в зоне отбора.

$W_{\text{деф}}$ - суммарный относительный дефицит товаров

$$W_{\text{деф}} = \sum_{j=1}^k \sum_{i \in N_j} w_{ij}(x_{ij}^*, d_{ij}) \quad (1.11)$$

$$w_{ij}(x_{ij}^*, d_{ij}) = \begin{cases} 1, x_{ij}^* = 0, d_{ij} > 0 \\ 0, x_{ij}^* = 1, d_{ij} > 0 \end{cases} \quad (1.12)$$

По величине $W_{\text{деф}}$ можно судить о том, на сколько полно комплектуются заказы, т.е. $W_{\text{деф}}$ может служить показателем удовлетворенности клиентов.

2. Математические методы для решения дискретных задач оптимизации

2.1. Метод вектора спада

2.1.1. Основные положения и определения

Рассмотрим дискретное топологическое пространство X , т.е. пространство состоящее исключительно из изолированных точек. Условие изолированности точки x в топологическом пространстве означает, что в этом **пространстве** подмножество $\{x\}$ является открытым множеством.

Пусть топология пространства X порождена некоторой введенной в множестве его точек метрикой $\rho(x, y)$. Метрика представляет собой однозначную, неотрицательную, действительную функцию, определенную для любых двух точек x и y из X и удовлетворяющую следующим условиям:

- 1) $\rho(x, y) = 0$ тогда и только тогда, когда $x = y$
- 2) $\rho(x, y) = \rho(y, x)$
- 3) $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$

Рассмотрим дискретное пространство X как метрическое пространство с метрикой $\rho(x, y)$. Пусть $G \subseteq X$.

Определение 1. Окрестностью $O_G(x, r)$ с центром в точке $x \in G$ и радиусом $r > 0$ назовем множество всех точек $y \in G$, удовлетворяющих неравенству $\rho(x, y) \leq r$.

Если $x \in G$, то очевидно, что $O_x(x, r) \cap G = O_G(x, r)$.

Окрестность $O_G(x, r)$ с центром в точке $x \in G$ назовем вырожденной, если $O_G(x, r) \setminus \{x\} = \emptyset$.

Пусть $0 < r_1 < \dots < r_m = r = f(x)$ - произвольная действительная функция, определенная на G .

Определение 2. Точку $x \in G$ назовем точкой минимума функции f относительно окрестности радиуса r , если окрестность $O_G(x, r)$ не вырождена и для всех точек $y \in O_G(x, r)$ выполняется условие $f(x) \leq f(y)$.

Отметим следующий **очевидный** факт: если **некоторая** точка $x \in G$ не является точкой минимума **функции f относительно** окрестности радиуса $r_1 > 0$, то она не является таковой и относительно окрестности любого радиуса r_2 , удовлетворяющего неравенству $r_1 \leq r_2$.

Определение 3. Точку $x \in G$ назовем точкой локального минимума функции f , если найдется такой радиус $r > 0$, для которого x является точкой минимума функции f относительно окрестности радиуса r .

Определение 4. Функцию $\Delta^r(x)$, определенную на $G \subseteq X$, назовем вектором спада функции $f(x)$ относительно окрестности радиуса r , если выполняются условия:

- 1) Для каждой точки $x \in G$ значение функции $\Delta^r(x)$ является $n(x)$ -мерным действительным вектором с компонентами $\Delta_1(x), \dots, \Delta_n(x)$;

- 2) Точка x является точкой минимума функции f относительно окрестности радиуса r тогда и только тогда, когда $\Delta_i(x) \geq 0$ при $i = 1, \dots, n(x)$.

Если x не является точкой минимума функции f относительно $O_G(x, r)$, то с помощью вектора спада $\Delta^r(x)$ можно определить точку $x' \in O_G(x, r)$ такую, что $f(x') < f(x)$.

Из данного определения следует, что вектор $\Delta^r(x)$ позволяет для каждой точки $x \in G$ определить направления уменьшения («спада») значений **функции** f в окрестности $O_G(x, r)$. Идея использования вектора спада $\Delta^r(x)$ для отыскания такого направления и лежит в основе рассматриваемого метода.

Необходимо отметить, что часто координаты $\Delta_1(x), \dots, \Delta_n(x)$ вектора спада можно вычислить намного проще, чем сами значения **функции** f в точках окрестности $O_G(x, r)$. Кроме того, в большинстве случаев для выявления в окрестности $O_G(x, r)$ точки x' такой, что $f(x') < f(x)$, или для установления факта локальной минимальности значения $f(x)$ достаточно ограничиться рассмотрением лишь части координат вектора $\Delta^r(x)$.

Итак, речь идет о решении задачи дискретного программирования, состоящей в нахождении локального **минимума (точнее: минимума относительно окрестности радиуса $r > 0$)** функции $f(x)$, определенной на дискретном множестве G .

Перейдем к непосредственному изложению схемы метода вектора спада, которую можно представить в виде следующих двух алгоритмов.

2.1.2. Описание общей схемы метода

I алгоритм

1. Выбираем некоторое начальное приближение $x_0 \in G$ (случайно или исходя из особенностей рассматриваемой задачи) и задаем величину радиуса $r > 0$.
2. Полагаем $h = 0$.
3. На $(h + 1)$ -м шаге алгоритма выполняем следующие **действия**:
 - 3.1. Рассматриваем окрестность $O_G(x^h, r)$. Если она вырождена, то для применения алгоритма необходимо или увеличить радиус r , или выбрать другое начальное **приближение** x^0 . Если окрестность $O_G(x^h, r)$ не вырождена, переходим к следующему пункту алгоритма.
 - 3.2. По значениям координат вектора спада $\Delta^r(x^h)$ определяем, является ли x^h точкой минимума функции $f(x)$ относительно окрестности радиуса r . Если да, переходим к п. 4, в противном случае к п. 3.3
 - 3.3. По значениям координат вектора спада находим точку $x^{h+1} \in O_G(x^h, r)$, для которой $f(x^{h+1}) < f(x^h)$. Заменяя h на $h + 1$, переходим к п. 3.
4. Работа по алгоритму заканчивается, так как x^h - искомая точка локального минимума функции $f(x)$.

II алгоритм

1. Выбираем начальное приближение $x_0 \in G$ и задаем радиус $r > 0$.
2. Задаем некоторую последовательность радиусов $\{r_1, \dots, r_m\}$, удовлетворяющую соотношениям $0 < r_1 < \dots < r_m = r$.

3. Полагаем $h = 0$.
4. На $(h + 1)$ -м шаге алгоритма выполняем следующие действия.
 - 4.1. Полагаем $j = 1$.
 - 4.2. Рассматриваем окрестность $O_G(x^h, r_j)$. Если она вырождена, то при $j = m$ можно сделать вывод о том, что для применения данного алгоритма необходимо увеличить радиус r или выбрать другое начальное приближение. При $j < m$ заменяем j на $j + 1$ и снова выполняем данный пункт алгоритма.
 - 4.3. Если окрестность $O_G(x^h, r_j)$ не вырождена, то по значениям координат вектора $\Delta^{r_j}(x^h)$ выясняем, является ли $f(x^h)$ минимумом функции $f(x)$ относительно окрестности радиуса r_j . Если да, то при $j < m$, заменив j на $j + 1$, переходим к началу п. 4.3, а при $j = m$ – к п. 5; в противном случае – к п. 4.4.
 - 4.4. По значениям координат вектора спада $\Delta^{r_j}(x^h)$ определяем в множестве $O_G(x^h, r_j)$ точку x^{h+1} , для которой $f(x^{h+1}) < f(x^h)$. Заменяя h на $h + 1$, переходим к п. 4.
5. Работа по алгоритму заканчивается, так как x^h является искомой точкой минимума функции $f(x)$ относительно окрестности радиуса r и, следовательно, точкой локального минимума этой функции.

2.1.3. Замечания

Рациональное сочетание первого и второго алгоритмов при организации поиска локального минимума функции $f(x)$ и составляет сущность метода вектора спада.

Сделаем три замечания, касающиеся различий двух алгоритмов.

Замечание 1. При выполнении **II алгоритма** необходимость в рассмотрении окрестностей радиуса r_m возникает лишь для незначительной части его шагов. Для большинства шагов достаточно ограничиться вычислением координат вектора спада функции $f(x)$ в окрестности $O_G(x^h, r_1)$. Это во многих случаях намного проще вычисления координат вектора спада $\Delta^r(x^h)$, которым приходится пользоваться на каждом шаге **I алгоритма** при $r = r_m > 1$.

Замечание 2. При выполнении **II алгоритма** в тех случаях, когда на некотором $(h + 1)$ -м шаге необходимо рассмотреть окрестности с радиусами, отличными от r_1 , вместо любой такой окрестности $O_G(x^h, r_j)$ достаточно рассмотреть множество $O_G(x^h, r_j) \setminus O_G(x^h, r_{j-1})$. При этом нужно исследовать лишь компоненты вектора спада $\Delta^{r_j}(x^h)$, соответствующие точкам указанного множества.

Замечание 3. Указанное в замечании 1 преимущество **II алгоритма** над **I** полностью реализуется в том случае, когда в **I алгоритме** на переход от точки x^h к точке x^{h+1} налагается дополнительное условие $f(x^{h+1}) = \min_{x \in O_G(x^h, r)} f(x)$. Очевидно, что **схема II**

алгоритма не допускает наложения такого условия на переход от x^h к x^{h+1} в качестве обязательного.

Вопрос о сходимости рассмотренных выше вычислительных схем метода вектора спада может быть выяснен на основании следующей теоремы.

Теорема 1. Пусть функция $f(x)$, определенная на G , удовлетворяет двум условиям:

- 1) Она ограничена снизу на G , т.е. существует такое постоянное число c , что $f(x) \geq c$ для любой точки $x \in G$;
- 2) Существует такое число $\delta > 0$, что для любых двух точек $x', x'' \in G$, для которых $f(x') \neq f(x'')$ выполняется неравенство $|f(x') - f(x'')| \geq \delta$.

Тогда при любом выборе начального приближения $x^0 \in G$ и радиуса $r > 0$ в обоих алгоритмах и последовательности $\{r_1, \dots, r_m\}$ во **II алгоритме** процессы вычислений по обоим алгоритмам сходятся за конечное число шагов, не превышающее величины $[f(x^0) - c]\delta^{-1}$.

Справедливость теоремы следует непосредственно из правил построения I и II алгоритмов. Действительно, элементы последовательности $\{f(x^h) | h = 0, 1, \dots\}$, получаемой в результате работы по каждому из двух алгоритмов, удовлетворяют соотношениям $f(x^0) > f(x^1) > \dots > f(x^h) > f(x^{h+1}) > \dots$

Из этих неравенств и **ксловий** теоремы следует, что $f(x^0) - c \geq k\delta$, где k – число шагов определенного алгоритма. Таким образом, справедливо неравенство $k \leq \frac{f(x^0) - c}{\delta}$, что и доказывает теорему.

Отметим, что для функции $f(x)$, определенной на некотором конечном множестве, выполняются оба условия теоремы и, следовательно, в этом случае вычислительный процесс по методу вектора спада заканчивается за конечное число шагов.

В тех случаях, когда по методу вектора спада решается оптимизационная задача, целевая функция которой не удовлетворяет условиям теоремы, в алгоритм метода можно вводить тот или иной признак окончания вычислительного процесса. Например, работу по алгоритму (I или II) можно остановить, когда число шагов превысит некоторую наперед заданную величину. При этом параллельно можно оценивать, в какой мере получаемое значение $f(x^h)$ на каждом h -м шаге удовлетворяет практическим требованиям. При такой постановке вопроса вычислительный процесс может продолжаться до того шага работы алгоритма, на котором будут удовлетворены практические требования, предъявляемые к решению задачи. Работу алгоритма можно также окончить по истечении некоторого отведенного промежутка времени t , если соответствующая экстремальная задача решается в АСУ, функционирующей в реальном времени. В последнем случае, естественно, достигаемая точность решения задачи существенно зависит от величины t . Если время решения экстремальной задачи не лимитируется, то остановку вычислительного процесса можно **осуществить и по следующему** правилу.

Пусть функция $f(x)$ удовлетворяет лишь первому условию теоремы. В этом случае при реализации метода вектора спада возможна ситуация: $\lim_{h \rightarrow \infty} f(x^h) = const$. Тогда для достаточно больших значений индекса h все значения

$$f(x^h), f(x^{h+1}), f(x^{h+2}), \dots \quad (1)$$

будут незначительно отличаться друг от друга. Поэтому значения функции $f(x)$ в точках x^h, x^{h+1}, \dots в определенном смысле можно считать «приближенными» локальными

минимумами. В этом случае, вычисления по алгоритму продолжаются до того шага, когда $|f(x^{k+1}) - f(x^k)| < \varepsilon, k = h, h+1, \dots$, где ε - заданная точность решения.

Для полноты описания общей схемы метода вектора спада остановимся на таком возможном случае, когда полученное в результате применения одного из алгоритмов локальное решение задачи не удовлетворяет некоторому заданному критерию (например, критерию, устанавливающему максимально допустимое отклонение локального решения от глобального оптимума). В этом случае для нахождения других приближенных решений задачи предлагаются перечисленные ниже пути.

Во-первых, можно увеличить величину радиуса r и продолжить вычислительный процесс по алгоритму, исходя из полученной точки x^* как нового начального приближения. Эту процедуру можно использовать несколько раз. Очевидно, что допустимые решения задачи, получаемые в результате таких процедур, будут «ближе» к глобальному оптимуму, чем x^* . Однако для больших значений радиуса r вычислительный процесс по методу вектора спада может значительно усложниться и поэтому на данном пути повышения точности решения задачи целесообразно воспользоваться II алгоритмом метода.

Во-вторых, можно, увеличив величину радиуса r , осуществить лишь один шаг применяемого алгоритма для получения точки x' такой, что $f(x') < f(x^*)$. Затем рекомендуется продолжить вычисленный процесс по алгоритму, вернувшись, однако, к первоначальной величине радиуса r .

В-третьих, можно выбрать некоторое другое начальное приближение $x^{01} \in G$ и повторить вычисления, исходя из этого нового приближения, но не изменяя величины r . В общем случае в результате работы по алгоритму получаем другое локальное решение x^{*1} задачи, т.е. $x^{*1} \neq x^*$. Очевидно, что при повторном отыскании локального решения при фиксированном радиусе r , но исходя из разных начальных приближений, получаем несколько различные локальные решения задачи, из которых в качестве решения следует выбрать то, которое доставляет минимальное значение функции $f(x)$.

В-четвертых, можно использовать такой способ: если в окрестности $O_G(x^*, r)$ имеется некоторая точка x' , для которой $f(x') \neq f(x^*)$, то работу по алгоритму можно продолжить, осуществив переход на следующий его шаг на основе точки x' . Такое дополнение к основному алгоритму обычно не влечет за собой большой дополнительной вычислительной работы и в то же время в ряде случаев может привести к более точному решению задачи.

2.1.4. Луч-метод

2.1.4.1. Основные положения

Рассмотрим задачу определения минимума выпуклой функции

$$\min f(x) \quad (2)$$

$$x \in D \subset R^n \quad (3)$$

$$x_i - \text{целые} \quad (4)$$

D – выпуклое, ограниченное множество. Точку $x \in R^n$, координаты которой целые, будем называть целочисленной.

Определение 1. Окрестностью $O(x^0)$ целочисленной точки $x^0 \in R^n$ назовем множество всех целочисленных точек $x' \in R^n$ таких, что совокупность чисел $|x'_1 - x_1^0|, \dots, |x'_n - x_n^0|$ взаимно проста, т.е. их наибольший общий делитель равен 1.

Из этого определения вытекают следующие свойства точек окрестности $O(x^0)$.

Свойство 1. $x^0 \notin O(x^0)$

Свойство 2. Если $x' \in O(x^0)$, то в интервале (x^0, x') не имеется ни одной целочисленной точки, т.е. не существует ни одной целочисленной точки, представимой в виде $x = x^0 + \lambda(x' - x^0), 0 < \lambda < 1$.

Доказательство. Докажем это свойство **от противного**. Пусть имеется целочисленная точка x'' , лежащая между x^0 и x' , т.е. существует значение $\bar{\lambda} = \bar{\lambda}$ такое, что $x'' = x^0 + \bar{\lambda}(x' - x^0), 0 < \bar{\lambda} < 1$.

Так как координаты точек x'', x', x^0 целые, то $\bar{\lambda}$ может быть представима в виде $\bar{\lambda} = p/q$, p, q – взаимно простые числа, $q > 1$ и $0 < p < q$. Следовательно, для всех $i = 1, 2, \dots, n$

$$x''_i - x_i^0 = \frac{p(x'_i - x_i^0)}{q} \quad \text{целые.}$$

Так как p и q взаимно просты, то $(x'_i - x_i^0)$ кратны $q > 1$, что противоречит условию $x' \in O(x^0)$.

Теорема 1. Каждая целочисленная точка лежит на одном из лучей, проведенном из целочисленной точки x^0 через одну из точек ее окрестности $x' \in O(x^0)$.

Доказательство. Докажем, что для любой целочисленной точки $x'' \in D$ существуют такие $x' \in O(x^0)$ и $\bar{\lambda} > 0$, что

$$x'' = x^0 + \bar{\lambda}(x' - x^0).$$

Рассмотрим точки, лежащие в полуинтервале $(x^0, x'']$:

$$x = x^0 + \mu(x'' - x^0), 0 < \mu \leq 1$$

Пусть наибольший общий делитель чисел $|x''_1 - x_1^0|, \dots, |x''_n - x_n^0|$ равен $d \geq 1$. Тогда при

$\mu = \frac{1}{d}$ получим точку $x' = x^0 + \frac{1}{d}(x'' - x^0)$, лежащую в полуинтервале $(x^0, x'']$, для которой

числа $x'_i - x_i^0 = \frac{x''_i - x_i^0}{d}$, для всех $i = 1..n$, взаимно просты. Следовательно, $x' \in O(x^0)$.

Таким образом, x'' лежит на луче, проведенном из x^0 через точку $x' \in O(x^0)$, и при $\bar{\lambda} = d$, $x'' = x^0 + d(x' - x^0)$.

Обозначим $D(x^0) = \{x \in D \mid f(x) < f(x^0)\}$. Условие оптимальности точки x^0 формулируется в виде следующей теоремы.

Теорема 2. Для того чтобы точка $x^0 \in D$ была оптимальной, необходимо и достаточно, чтобы $O(x^0) \cap D(x^0) = \emptyset$.

Доказательство. Необходимость. Пусть $x^0 \in D$ - оптимальная точка, т.е. $f(x^0) < f(x)$ для всех $x \in D$, x - целые. Значит, множество $D(x^0)$ не содержит ни одной целочисленной точки и, следовательно, $O(x^0) \cap D(x^0) = \emptyset$.

Достаточность. Пусть $O(x^0) \cap D(x^0) = \emptyset$. Тогда $D(x^0)$ не содержит ни одной из точек окрестности $O(x^0)$. Для доказательства оптимальности x^0 нужно показать, что $D(x^0)$ вообще не содержит целых точек. Заметим, что $D(x^0)$ - выпуклое множество, так как функция $f(x)$ - выпуклая. Поэтому, если предположить, что в $D(x^0)$ содержится какая-либо целочисленная точка x'' , то в силу выпуклости $D(x^0)$, в $D(x^0)$ вместе с x'' содержится и точка x' из окрестности $O(x^0)$, лежащая на полуинтервале $(x^0, x'']$, что противоречит условию $O(x^0) \cap D(x^0) = \emptyset$. Теорема доказана.

Опишем общую схему луч-метода для решения задачи (1)-(3). Пусть задана допустимая точка $x^0 \in D$. Будем искать такую точку $x' \in O(x^0)$, что $x' \in O(x^0) \cap D(x^0)$. Если $O(x^0) \cap D(x^0) = \emptyset$, то x^0 есть решение задачи (2)-(4).

Если x найдется, то, решая задачу минимизации

$$f(\lambda) = f(x^0 + \lambda(x' - x^0))$$

При условии $\lambda \geq 0, x^0 + \lambda(x' - x^0) \in D$, найдем $\lambda = \lambda_{\min}$.

Затем определим целочисленную точку x'' :

$$x'' = \begin{cases} x^0 + [\lambda_{\min}](x' - x^0), & \text{если } x^0 + ([\lambda_{\min}] + 1)(x' - x^0) \notin D \text{ или} \\ f(x^0 + ([\lambda_{\min}] + 1)(x' - x^0)) \geq f(x^0 + [\lambda_{\min}](x' - x^0)), \\ x^0 + ([\lambda_{\min}] + 1)(x' - x^0) & \text{в противном случае} \end{cases}$$

Здесь и далее под знаком $[\lambda]$ будем понимать целую часть числа λ .

Иными словами, ищем минимум функционала $f(\lambda)$ вдоль луча, выходящего из точки x^0 и проходящего через точку x' , а затем находим на этом луче целочисленную точку $x'' \in D$, ближайшую по функционалу к $f(\lambda_{\min})$. Обозначим полученную точку x'' через x^0 . При этом новая область $D(x^0)$ получается из старой добавлением условия $f(x) < f(x^0) = f(x'')$. Так как функция $f(x)$ и область D выпуклые, то область $D(x^0)$ также выпуклая. Получили задачу, аналогичную исходной. Далее процесс повторяется. Конечность метода вытекает из конечности числа целочисленных точек в D .

Из описанной общей схемы ясно, что основная трудность реализации луч-метода заключается в поиске допустимой целочисленной точки и в проверке критерия оптимальности. Как известно, задача отыскания целочисленного **решения системы линейных** является в общем случае NP трудной, а выяснение выполнимости критерия оптимальности путем просмотра всех точек окрестности $O(x^0)$ может оказаться практически невозможным из-за большого числа таких точек. Поэтому в алгоритмах реализуются приближенные варианты луч-метода, в которых поиск очередной целочисленной точки из $O(x^0) \cap D(x^0)$ ведется путем просмотра не всех точек окрестности, а лишь ее некоторой части.

2.1.5. Алгоритмы приближенного решения задач выпуклого целочисленного программирования луч-методом

Опишем алгоритмы приближенного решения задачи (2)-(4) луч-методом, в которых поиск точек из $O(x^0) \cap D(x^0)$ ведется путем просмотра определенной части точек окрестности $O(x^0)$. В частности, предлагается просматривать точки окрестности $O(x^0)$, расположенные вблизи некоторого **направляющего** луча. Таким образом, алгоритмы приближенного решения характеризуются двумя этапами: способом выбора направляющих лучей и способом выбора точек окрестности $O(x^0)$ вдоль направляющих лучей.

2.1.6. Направляющие лучи и способы их выбора

Пусть задана некоторая допустимая точка x^0 . Луч $L = x^0 + \lambda l, \lambda \geq 0$, где $l = (l_1, \dots, l_n)$ будем называть направляющим лучом, если для некоторого $\lambda' > 0$ $f(x^0 + \lambda' l) < f(x^0)$.

Изложим несколько способов выбора направляющего луча L .

- 1) Пусть x^* есть решение задачи без условия целочисленности. В качестве луча L рассмотрим

$$L = x^0 + \lambda(x^* - x^0), \lambda \geq 0$$

- 2) Пусть область D задана системой следующих неравенств:

$$g_i(x) \leq b_i, i = 1..m,$$

где $g_i(x)$ - выпуклые, гладкие функции. Вычислив градиенты функций $g_i(x)$ в точке x^0 , определим направляющие лучи

$$L_i = x^0 + \lambda \text{grad}[g_i(x^0)], \lambda \geq 0, \text{ если } \exists \lambda' > 0 \text{ такое, что } f(x^0 + \lambda' \text{grad}[g_i(x^0)]) < f(x^0) \text{ и}$$

$$L_i = x^0 - \lambda \text{grad}[g_i(x^0)], \lambda \geq 0, \text{ если } \exists \lambda' > 0 \text{ такое, что } f(x^0 - \lambda' \text{grad}[g_i(x^0)]) < f(x^0)$$

- 3) В качестве направляющего луча выбирается

$$L = x^0 - \lambda \text{grad}[g_i(x^0)], \lambda \geq 0$$

- 4) Определим некоторое множество целочисленных точек

$$M = \{x \in R^n \mid x_i - \text{целые}, f(x) < f(x^0)\}.$$

В качестве направляющих лучей будем выбирать лучи, исходящие из точки x^0 и проходящие через точки множества M . Рассмотрим один из следующих возможных способов определения M :

$$M_k = \{x^k \in R^n \mid x^k = (x_1^k, \dots, x_n^k), k = 1, \dots, n, \text{ где}$$

$$x_k \in [x_k^* - 1, x_k^* + 1], x_i = [x_i^*], i = 1, \dots, n; i \neq k,$$

$$f(x^k) < f(x^0)\}$$

где x^* есть решение задачи (2)-(3).

В качестве множества M можно использовать множество целочисленных точек, получаемых в процессе реализации луч-метода. Пусть получены точки x^0, x^1, \dots, x^{q_0} . Зафиксировав некоторое $x^i, 0 \leq i \leq q_0$, рассмотрим его в качестве x^0 , а M определим как множество всех $x^j, i < j \leq q_0$.

2.1.7. Способы определения целочисленных точек со взаимно простыми координатами (точек окрестности $O(x^0)$)

Пусть выбран направляющий луч $L = x^0 + \lambda l, \lambda \geq 0$ одним из описанных выше способов. Изложим следующие способы выбора точек окрестности $O(x^0)$ вдоль выбранного направляющего луча L . Решим задачу $\max_{x^0 + \lambda l \in D, \lambda \geq 0} \lambda$. Пусть решение - $\lambda = \lambda_{\max}$. Определим

$$|l_{i_0}| = \min_{1 \leq i \leq n, l_i \neq 0} |l_i|.$$

Обозначим $x^{p,q} = x^0 + \frac{p}{q} \lambda_{\max} l$, где $0 < q = \max\{1, [|l_{i_0}|]\}$ - целое, а $p = 1, 2, \dots, q$. Будем просматривать следующие целочисленные точки $\bar{x} = (\bar{x}_1, \dots, \bar{x}_n)$:

а) $\bar{x}_i = [x_i^{p,q}]$

б) $\bar{x}_i = \begin{cases} [x_i^{p,q}], & \text{если } -\frac{\partial f(x)}{\partial x_i} \Big|_{x^{p,q}} \geq 0 \\ [x_i^{p,q}] + 1, & \text{если } -\frac{\partial f(x)}{\partial x_i} \Big|_{x^{p,q}} < 0 \end{cases}$

в) $\bar{x}_i = \begin{cases} [x_i^{p,q}], & \text{если } x_i^{p,q} - [x_i^{p,q}] \leq 0.5 \\ [x_i^{p,q}] + 1, & \text{если } x_i^{p,q} - [x_i^{p,q}] > 0.5 \end{cases}$

г) Пусть $J = \{1, \dots, n\}$, возьмем некоторое разбиение множества J : $J_1, J_2, J_3 \subset J, J_1 \cup J_2 \cup J_3 = J, J_i \cap J_j = \emptyset, i \neq j, i, j = 1, 2, 3$

Определим \bar{x}_i по способу а), если $i \in J_1$; по способу б), если $i \in J_2$; по способу в), если $i \in J_3$. Если среди таких \bar{x} найдется точка, лежащая в $D(x^0)$, то, определив наибольший общий делитель d чисел $|\bar{x}_i - x_i^0|$, вычислив точку $x' = x^0 + \frac{\bar{x} - x^0}{d} \in O(x^0)$.

Для определения точек окрестности $O(x^0)$ описанным выше способом необходимо определить наибольший общий делитель n чисел. Известно, что приближительной оценкой сложности его определения является $n \log_2 N$, где $N = \max_{1 \leq i \leq n} |\bar{x}_i - x_i^0|$. Поэтому при больших n и N определение наибольшего общего делителя чисел $|\bar{x}_1 - x_1^0|, \dots, |\bar{x}_n - x_n^0|$ может оказаться трудоемким.

Изложим несколько способов получения точек окрестности $O(x^0)$ вблизи направляющего луча, не вычисляя наибольшего общего делителя n чисел.

1) В этом способе точки из $O(x^0)$ определяются так, чтобы они не выходили за пределы единичных кубов, содержащих точки $\bar{x} \in L' \cap D$, где $L' = \{x \in R^n \mid x = x^0 + \lambda l, \lambda \geq 0\}$. Заметим, что не всякий такой единичный куб имеет вершину, принадлежащую $O(x^0)$. Поэтому будем определять те точки на луче L , для которых это условие заведомо выполняется. Можно воспользоваться таблицей простых чисел в диапазоне $[1, M]$, величина которого зависит от конкретных характеристик ЭВМ (объем памяти, быстродействие и др.), на которой производятся расчеты.

Для каждой координаты $1 \leq i \leq n$ найдем в таблице простых чисел те, которые лежат в интервале $[1, \lambda_{\max} l_i]$. Получим множества $S_i = \{s_1, \dots, s_{r_i}\}, i = 1, \dots, n$ простых чисел. Возьмем $s_k \in S_i$ и определим $\lambda_k = \frac{s_k}{l_i}$. Так как $s_k \leq \lambda_{\max} l_i$, то $\lambda_k \leq \lambda_{\max}$, поэтому $x^k = x^0 + \lambda_k l \in L' \cap D$, причем i -я компонента вектора $x^k - x^0$ является простой: $x_i^k - x_i^0 = s_k$. Рассмотрим целочисленную точку $\bar{x}^k = (\bar{x}_1^k, \dots, \bar{x}_n^k)$, полученную из x^k следующим образом: $\bar{x}_i^k = x_i^k = x_i^0 + s_k$, а $\bar{x}_j^k, j \neq i, j = 1, \dots, n$ образуются из x_j^k по правилам а)-г), описанным выше. Если для $j \neq i$ одна из компонент вектора $(\bar{x}^k - x^0)$ не кратна s_k , то $x^k \in O(x^0)$. Если все компоненты полученного вектора $(\bar{x}^k - x^0)$ кратны s_k , то, разделив их на s_k , получим точку $\tilde{x}^k = x^0 + \frac{1}{s_k}(\bar{x}^k - x^0)$, которая является точкой окрестности $O(x^0)$, так как $\tilde{x}_i^k - x_i^0 = \frac{1}{s_k}(x_i^0 + s_k - x_i^0) = 1$.

Таким образом, для каждого $k, 1 \leq k \leq r_i$, получаем точки окрестности $O(x^0)$ вдоль луча L , которые являются вершинами единичного куба, содержащего x^k . Прделаем это для всех множеств $S_i, i = 1, \dots, n$. Каждую из формируемых точек $\bar{x}^k \in O(x^0)$ необходимо проверить на принадлежность к множеству $D(x^0)$.

- 2) Известно, что два последовательных целых числа взаимно просты. Этот факт используется при построении еще одного способа получения точек окрестности вдоль луча $L = x^0 + \lambda l, \lambda > 0$. Пусть $l = (l_1, \dots, l_n)$. Рассмотрим две координаты $l: l_{i_0} \neq l_{j_0}$. Пусть для определенности $l_{i_0} < l_{j_0}$. Если $\lambda_{\max}(l_{j_0} - l_{i_0}) > 1$, то определим вектор $\bar{l} = \frac{l}{l_{j_0} - l_{i_0}}$.

$$\text{Следовательно, } \bar{l}_{i_0} = \frac{l_{i_0}}{l_{j_0} - l_{i_0}}, \quad \bar{l}_{j_0} = \frac{l_{j_0}}{l_{j_0} - l_{i_0}} = 1 + \frac{l_{i_0}}{l_{j_0} - l_{i_0}} = 1 + \bar{l}_{i_0}.$$

Округлим \bar{l}_{i_0} до ближайшего целого, получив \tilde{l}_{i_0} и положим $\tilde{l}_{j_0} = \tilde{l}_{i_0} + 1$. Учитывая, что $\bar{l}_{j_0} = \bar{l}_{i_0} + 1$, имеем $|\tilde{l}_{j_0} - [\bar{l}_{j_0}]| \leq 1$. Остальные координаты l округлим до ближайших целых, получим вектор $\tilde{l} = (\tilde{l}_1, \dots, \tilde{l}_n)$, являющийся вершиной единичного куба, содержащего \bar{l} . Так как $\tilde{l}_{j_0} = \tilde{l}_{i_0} + 1$, то числа $\tilde{l}_{j_0}, \tilde{l}_{i_0}$ взаимно простые, следовательно, $\tilde{x} = x^0 + \tilde{l} = x^0 + \bar{l} + (\tilde{l} - \bar{l}) \in O(x^0)$. $x^0 + \bar{l}$ лежит на луче L , так как $\bar{l} = \frac{l}{l_{j_0} - l_{i_0}}$ и $x^0 + \bar{l} \in D$, поэтому \tilde{x} является вершиной единичного куба, содержащего $x^0 + \bar{l}$ и отклонена от L на $\tilde{l} - \bar{l}$. Пусть $\lambda_{\max}(l_{j_0} - l_{i_0}) \leq 1$. Возможны следующие случаи:

I. $[\lambda_{\max} l_{j_0}] = [\lambda_{\max} l_{i_0}] + 1$, в этом случае определим $\tilde{l}_{i_0} = [\lambda_{\max} l_{i_0}], \tilde{l}_{j_0} = [\lambda_{\max} l_{j_0}] = [\lambda_{\max} l_{i_0}] + 1$

II. $[\lambda_{\max} l_{j_0}] = [\lambda_{\max} l_{i_0}]$, определим $\tilde{l}_{i_0} = [\lambda_{\max} l_{i_0}], \tilde{l}_{j_0} = [\lambda_{\max} l_{i_0}] + 1$

Остальные координаты l получим из координат $\lambda_{\max} l$ округлением до ближайшего целого. Ясно, что

$$x = x^0 + l \in O(x^0).$$

Перебирая все пары i_0, j_0 координат l , будем иметь C_n^2 точек $O(x^0)$ в направлении L . Получая точки окрестности $O(x^0)$, необходимо проверять их на допустимость, так как отклонение $\tilde{l} - \bar{l}$ может привести к нарушению какого-нибудь ограничения области D .

2.1.8. Луч-метод для решения целочисленного вогнутого программирования

Рассмотрим задачу

$$\min f(x) \quad (5)$$

$$x \in D \subset R^n \quad (6)$$

$$x_i - \text{целые} \quad (7)$$

D – выпуклое, ограниченное множество. $f(x)$ – вогнутая функция, т.е. для $\forall x_1, x_2 \in D, 0 \leq \lambda \leq 1$ выполняется неравенство $f(\lambda x_1 + (1 - \lambda)x_2) \geq \lambda f(x_1) + (1 - \lambda)f(x_2)$.

Определим $D(x^0)$ аналогично тому, как это было сделано в п. 1.2.1

$$D(x^0) = \{x \in D \mid f(x) < f(x^0)\}.$$

Из-за того, что $f(x)$ – вогнутая функция, $D(x^0)$ не будет выпуклым множеством, и потому соотношение $D(x^0) \cap O(x^0) = \emptyset$ может являться лишь необходимым условием оптимальности точки x^0 , так как из него не будет следовать, что $D(x^0)$ не содержит целочисленных точек.

Опишем модификацию луч-метода для решения этой задачи. Смысл модификации заключается в том, что наряду с множеством $D(x^0)$ формируется и аппроксимирующее множество $K(x^0) \subset R^n$, обладающее двумя свойствами:

- $K(x^0)$ - выпуклое множество
- $K(x^0) \supset D(x^0)$

Это позволяет, как будет показано ниже, пользоваться соотношением $K(x^0) \cap O(x^0) = \emptyset$ в качестве оптимальности условия оптимальности точки x^0 .

Теорема 3. Если $x^0 \in D$ и $K(x^0) \cap O(x^0) = \emptyset$, то x^0 - есть решение задачи (4)-(6).

Доказательство. Так как $K(x^0)$ - выпуклое множество, условие $K(x^0) \cap O(x^0) = \emptyset$ означает, что $K(x^0)$ не содержит ни одной целочисленной за исключением, быть может, точки x^0 . А так как $D(x^0) \subset K(x^0)$ и $x^0 \notin D(x^0)$, то $D(x^0)$ не содержит ни одной целочисленной точки. Следовательно, x^0 есть решение (5)-(7).

Теорема 3 является достаточным условием оптимальности x^0 для задачи (5)-(7), в то время как теорема 2 является необходимым и достаточным условием оптимальности x^0 в задаче (2)-(4).

Опишем общую схему луч-метода. Прежде всего заметим, что если $f(x)$ – вогнутая функция и λ_{\max} есть решение задачи $\max \lambda$, при условии $x^0 + \lambda(x' - x^0) \in D, \lambda \geq 0, x', x^0 \in D$, то минимум $f(\lambda) = f(x^0 + \lambda(x' - x^0))$ при тех же

условиях $x^0 + \lambda(x' - x^0) \in D, \lambda \geq 0$ достигается либо при $\lambda = 0$, либо при $\lambda = \lambda_{\max}$, т.е. $\min f(\lambda) = \min\{f(0), f(\lambda_{\max})\} = \min\{f(x^0), f(x'')\}$, где $x'' = x^0 + \lambda_{\max}(x' - x^0)$.

Пусть известна точка $x^0 \in D$. Будем искать такую точку $x' \in O(x^0)$, что $x' \in O(x^0) \cap K(x^0)$. Если такой точки нет, то x^0 - решение задачи. Если x' найдется, то определим

$$x'' = x^0 + [\lambda_{\max}](x' - x^0).$$

Если $f(x'') < f(x^0)$, то обозначим x'' через x^0 и повторим процесс, при этом множество $D(x^0)$ сузится, что даст возможность построить новое множество $K(x^0)$, входящее в $K(x^0)$, полученное на предыдущем шаге. Если же $f(x'') \geq f(x^0)$, то просматриваем другие точки $x' \in O(x^0) \cap K(x^0)$ до тех пор, пока не найдется точка x'' такая, что $f(x'') < f(x^0)$, либо выяснится, что такая точка не существует, и в этом случае x^0 будет искомым решением.

Эффективность метода зависит от точности аппроксимации множества $D(x^0)$, поэтому желательно использовать такие способы аппроксимаций, чтобы в процессе расчетов образуемые множества $K(x^0)$ получались последовательно входящими друг в друга. Для некоторых классов функций такие аппроксимации удается построить. Рассмотрим некоторые из них.

2.1.8.1. Примеры некоторых функций

1) Функции, монотонно убывающие по норме.

Рассмотрим монотонно **убывающую** по норме функцию $f(x)$, определенную на выпуклой области $D \subset R_+^n$. Определим

$$\|x\| = \left(\sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}.$$

Определение 1. Функция $f: R^n \rightarrow R^1$ называется монотонно убывающей по норме, если для $x_1, x_2 \in R^n$ таких, что $\|x_1\| \geq \|x_2\|$, имеет место неравенство $f(x_1) \leq f(x_2)$.

Пусть $x^0 \in D$ и выполняется требование: если $f(x_1) \neq f(x_2)$, то $\|x_1\| \neq \|x_2\|$. Рассмотрим функционал

$$L(x) = \sum_{i=1}^n x_i, p(x^0) = \|x^0\|.$$

Определим множество $K(x^0) = \{x \in D, L(x) > p(x^0)\}$. Покажем, что $K(x^0) \supset D(x^0) = \{x \in D, f(x) < f(x^0)\}$. Пусть $x' \in D(x^0)$. Это означает, что $\|x'\| > \|x^0\|$. Нетрудно убедиться, что

$$\max_{\left\{ \begin{array}{l} \sum_{i=1}^n x_i \leq \|x^0\| \\ x_i \geq 0 \end{array} \right\}} \sum_{i=1}^n x_i^2 = \|x^0\|^2,$$

поэтому $x' \in \left\{ x \in D \mid \sum_{i=1}^n x_i > \|x^0\| \right\} = K(x^0)$, что означает $D(x^0) \subset K(x^0)$. Если $\{x \in D, L(x) = \|x^0\|\} \neq \emptyset$,

то последовательно получаемые в процессе реализации луч-метода множества $K(x^0)$ при изменении x^0 окажутся вложенными друг в друга.

2) Сепарабельные функции.

Рассмотрим задачу

$$f(x) = \min \sum_{i=1}^n g_i(x_i) \quad (8)$$

при условии

$$x \in D \quad (9)$$

$$a_i \leq x_i \leq b_i \quad (10)$$

$$x_i - \text{целые} \quad (11)$$

D – ограниченное выпуклое множество, $g_i(x_i)$ – вогнутые функции. Величины a_i, b_i могут быть уточнены решением $2n$ оптимизационных задач типа $\min_{x \in D} x_i, \max_{x \in D} x_i$. Будем строить

разбиение множества $D(x^0) = \bigcup_{i=1}^q D_i(x^0)$ и для него рассматривать аппроксимирующие множества $K_i(x^0) \supset D_i(x^0)$. Получение очередной допустимой точки x^1 сведется к применению луч-метода на каждом из множеств D_i . При разбиении D будем пользоваться схемой ветвления метода ветвей и границ. Выбор множества ветвления D_i произведем по наименьшей оценке функционала на этом множестве. Пусть $l_j(x_j)$ есть прямая, проходящая через $g_j(\bar{a}_j), g_j(\bar{b}_j)$, где \bar{a}_j, \bar{b}_j – минимальное и максимальное значение j -й координаты элементов выбранного множества разбиения. Выбор компоненты j_0 переменной ветвления осуществим из условия максимального отклонения функции $g_i(x_i)$ в известной целочисленной допустимой точке от значения линейной функции $l_i(x_i)$ в этой же точке. Проиллюстрируем способ построения множеств $K_i(x^0)$. Рассмотрим прямые $l_i(x_i) = p_i x_i + d_i$, проходящие через точки $g_i(a_i), g_i(b_i), i = 1, \dots, n$. Так как $g_i(x_i)$ вогнутые, то $l_i(x_i) \leq g_i(x_i), a_i \leq x_i \leq b_i$. Определим $j_0: g_{j_0}(x_{j_0}^0) - l_{j_0}(x_{j_0}^0) = \max_{1 \leq j \leq n} (g_j(x_j^0) - l_j(x_j^0))$.

Разобьем D на два непересекающихся множества

$$D_1 = \{x \in D \mid x_{j_0} < x_{j_0}^0\},$$

$$D_2 = \{x \in D \mid x_{j_0} \geq x_{j_0}^0\}.$$

Определим кусочно-линейную функцию $\bar{l}_{j_0}(x_{j_0})$, состоящую из двух линейных частей таких, что одна из них является отрезком, соединяющим $g_{j_0}(a_{j_0})$ с $g_{j_0}(x_{j_0}^0)$, а

другая часть является отрезком, соединяющим $g_{j_0}(x_{j_0}^0)$ с $g_{j_0}(b_{j_0})$. Ввиду вогнутости $g_{j_0}(x_{j_0})\bar{l}_{j_0}(x_{j_0}) \geq l_{j_0}(x_{j_0})$. Определим функционал

$$L^0(x) = \sum_{i=1, i \neq j}^n l_i(x_i) + \bar{l}_{j_0}(x_{j_0}).$$

Оценкой $f(x)$ на множестве $D_i(x^0)$ будет $\underline{L}_i = \min_{x \in D_i(x^0)} L^0(x)$. Из определения следует линейность $L^0(x)$ на каждом множестве D_1, D_2 . Обозначим

$$K_1(x^0) = \{x \in D_1 \mid L^0(x) < f(x^0)\},$$

$$K_2(x^0) = \{x \in D_2 \mid L^0(x) < f(x^0)\}.$$

Множества $K_1(x^0), K_2(x^0)$ выпуклые и $K_1(x^0) \supset D_1(x^0), K_2(x^0) \supset D_2(x^0)$. Применим луч-метод на каждом из множеств $D_1(x^0), D_2(x^0)$. Минимум функционала будет минимум значений $f(x)$ на полученных допустимых целочисленных точках из $D_1(x^0), D_2(x^0)$. Далее процесс повторяется, при этом производится дальнейшее ветвление множества D и построение аппроксимирующих множеств описанным выше способом. Легко убедиться, что объединения множеств $K_1(x^0)$, получаемых на каждом шаге, будут образовывать последовательность входящих друг в друга множеств. Заметим, что оценки \underline{L}_i , определяемые путем решения оптимизационных задач с линейным функционалом, можно использовать не только для определения множества ветвления, но и для отбраковки множеств D . Если для некоторого $1 \leq i \leq q, \underline{L}_i > f(x^{j_0})$, где $f(x^{j_0})$ - значение, на котором достигается минимум функционала, то множество D_i можно исключить из дальнейшего рассмотрения.

3. Результаты моделирования операций отбора/пополнения на складе продуктов питания

3.1. Описание входных данных

Рассматривались данные реального склада. На складе хранятся продукты питания. Из-за специфики склада и хранимой продукции, заказы на отгрузку редко содержат товары в объеме больше одной палеты. В основном, заказы включают много товарных позиций, но в небольшом объеме.

- Параметры склада

- площадь склада – 40000 м²,
- $F_{пал} = 1200$ палето-мест (потенциально объем зоны отбора может быть расширен до 1500 палето-мест).

- Данные по заказам на отгрузку

- временной интервал 16.10.2007 – 27.10.2007,
- $P_{зак} = 2483$ (заказы),
- $G_{тов} = 2354$ (товары),
- $L_{стр} = 52205$ (число строк в заказах).

- Затраты

Затраты на отбор/пополнение (у.е.)

- $c_1 = 0.2$ (стоимость отбора из зоны отбора, штучный отбор),
- $c_2 = 1$ (стоимость отбора из зоны хранения, палетный отбор),
- $c_r = 0.8$ (стоимость перемещения товара из зоны хранения в зону отбора).

Затраты на реорганизацию зоны отбора (у.е./палета)

- $m_1 = 0.8$ (стоимость перемещения из зоны хранения в зону отбора),
- $m_2 = 1$ (стоимость перемещения из зоны отбора в зону хранения).

Данные по затратам отличаются незначительно для разных складов [2].

3.2. Сценарии моделирования

Далее исследуются зависимости суммарных ($I, I_1, I_2, R_{попол}, W_{деф}$) характеристик от величины цикла заказов ($K_{цикл.зак}$) и размера зоны отбора ($F_{пал}$). В приложении 1 приведены графики зависимости оптимального решения от величины цикла заказов, для части товаров.

3.2.1. Рассмотрение существующей зоны отбора

В этом сценарии моделирования была рассмотрена существующая зона отбора ($F_{пал} = 1200$ палето-мест). На данном шаге варьировались следующие параметры алгоритма расчета зоны отбора:

- Начальное число заказов, используемое для первого расчета ($P_{зак}^0 = 500, 1000$)
- Количество заказов, при накоплении которых производится перерасчет зоны отбора ($K_{цикл.зак} = 200, 300, 400, 500$)

Далее приведены графики зависимости суммарных характеристик от величины цикла заказов.

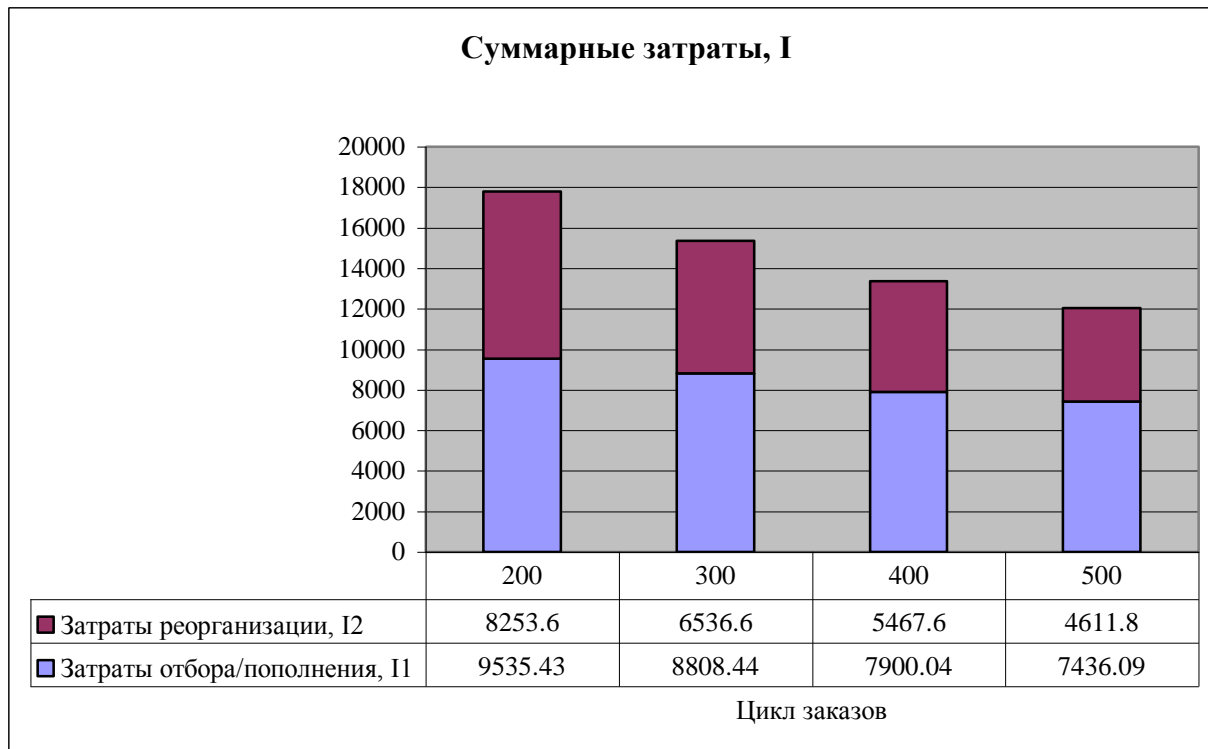


Рис. 4. Суммарные затраты, I. Зона отбора, $F_{нал} = 1200$ палето-мест. Первоначальная статистика, $P_{зак}^0 = 500$ заказов.

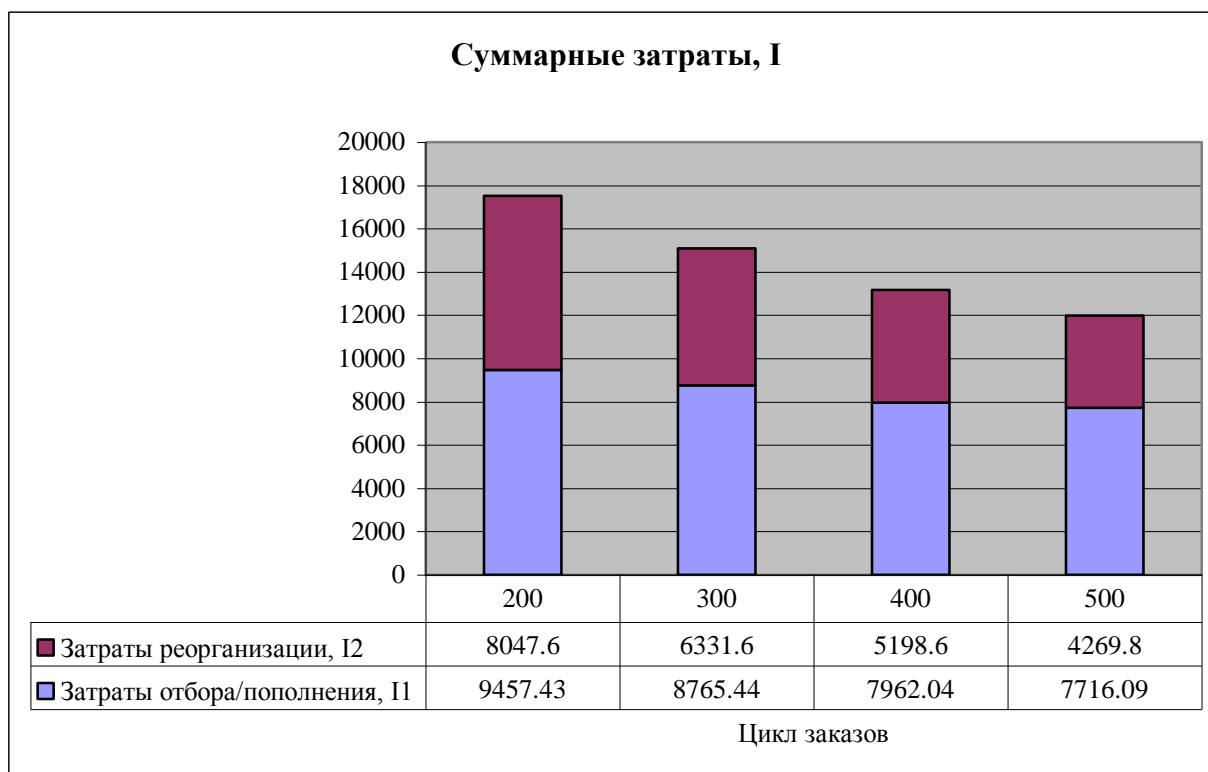


Рис. 5. Суммарные затраты, I .Зона отбора, $F_{нал} = 1200$ палето-мест. Первоначальная статистика, $P_{зак}^0 = 1000$ заказов.

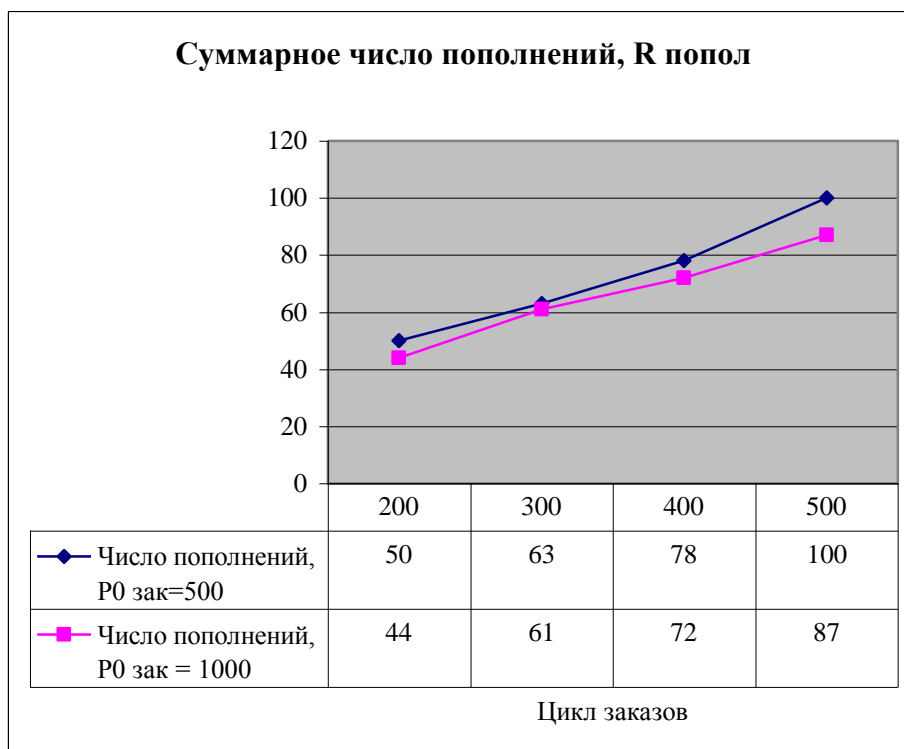


Рис. 6. Суммарное число пополнений, $R_{попол}$. Зона отбора, $F_{нал} = 1200$ палето-мест. Первоначальная статистика, $P_{зак}^0 = 500, 1000$ заказов.

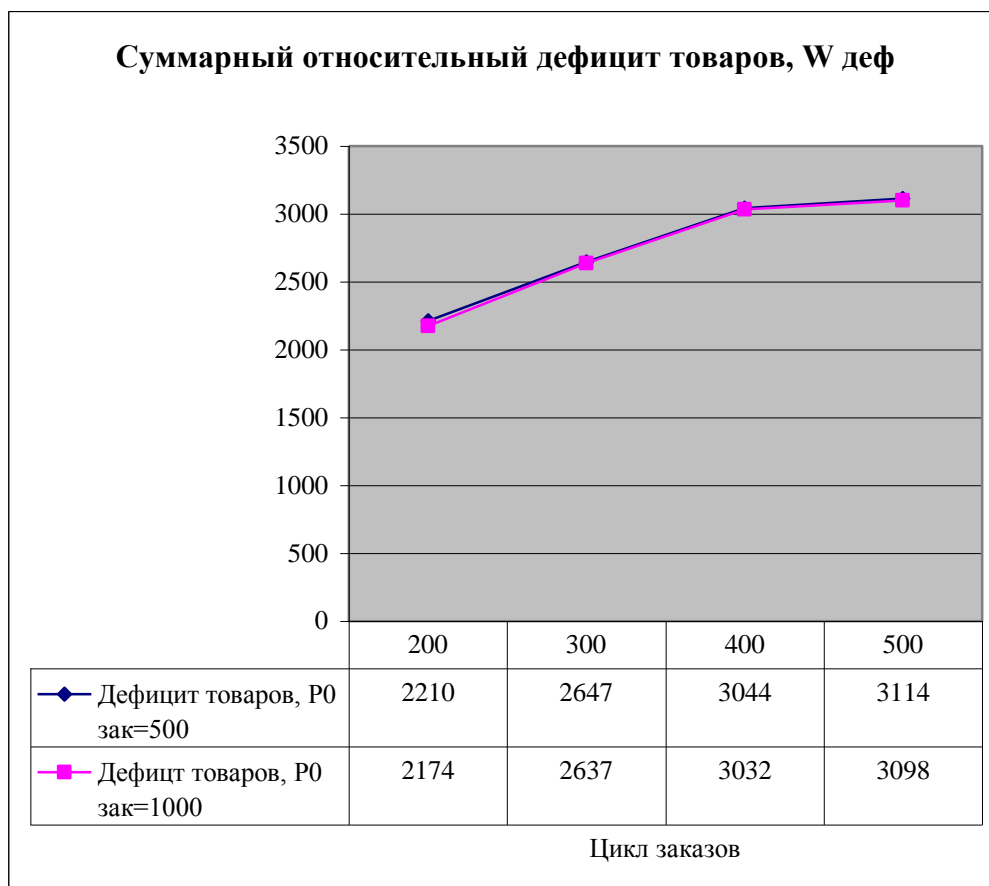


Рис. 7. Относительный дефицит товаров, $W_{\text{деф}}$. Зона отбора, $F_{\text{пал}} = 1200$ палето-мест. Первоначальная статистика, $P_{\text{зак}}^0 = 500, 1000$ заказов.

Чем меньше величина цикла заказов, тем чаще перерасчитывается зона отбора, т.е. зона отбора становится более «мобильной» к удовлетворению поступающих заказов. Как следствие этой «мобильности», снижается число пополнений и возрастают затраты на реорганизацию и суммарные затраты.

В случае большой величины цикла заказов $K_{\text{цикл.зак}} = 500$, зона отбора менее всего эффективна. Возрастает относительный дефицит товаров, хотя затраты как на отбор/пополнение, так и на реорганизацию минимальны.

Зависимость между изменениями затрат, пополнением, дефицитом и величиной цикла заказов сохраняется при различном размере начальной статистики заказов ($P_{\text{зак}}^0$). При рассмотрении большей исходной статистики заказов, первоначальный расчет зоны отбора производится более точно, и как следствие этого, все показатели лучше.

3.2.2. Варьирование размера зоны отбора

В данном сценарии варьировался размер зоны отбора ($F_{\text{пал}} = 1300, 1400, 1500$ палето-мест). Параметры алгоритма расчета зоны отбора, рассматривались аналогичными первому сценарию, для дальнейшего сравнения:

- Начальное число заказов, используемое для первого расчета ($P_{\text{зак}}^0 = 500$)
- Количество заказов, при накоплении которых производится перерасчет зоны отбора ($K_{\text{цикл.зак}} = 200, 300, 400, 500$)

Далее приведены графики зависимости суммарных характеристик от величины цикла заказов (при различных размерах зоны отбора).

Суммарные затраты

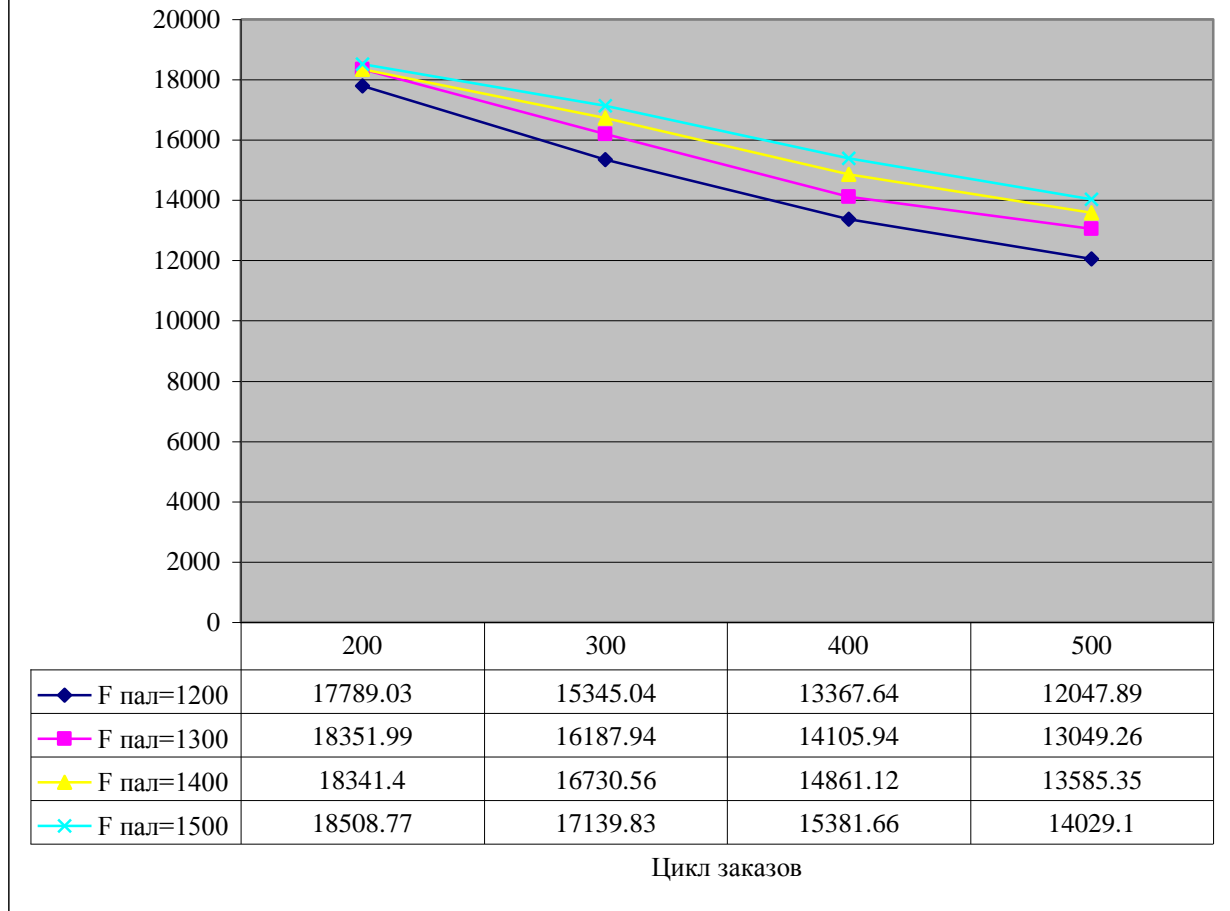


Рис. 8. Суммарные затраты, I . Варьирование размера зоны отбора

Затраты отбора/пополнения, I_1

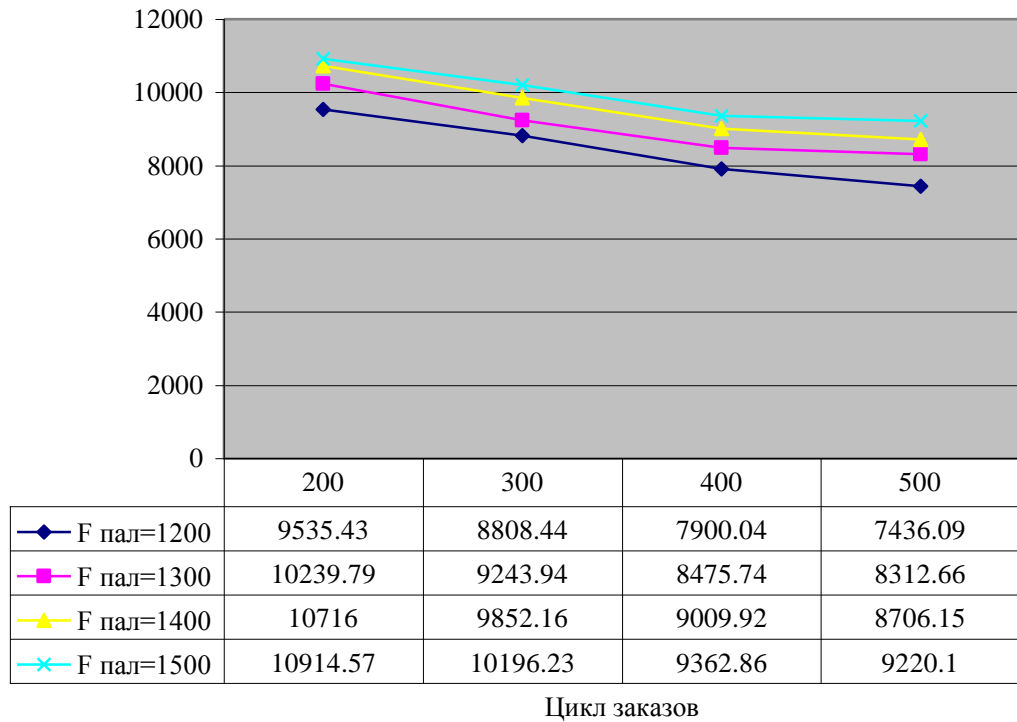


Рис. 9. Затраты отбора/пополнения, I_1 . Варьирование размера зоны отбора

Затраты реорганизации зоны отбора, I_2

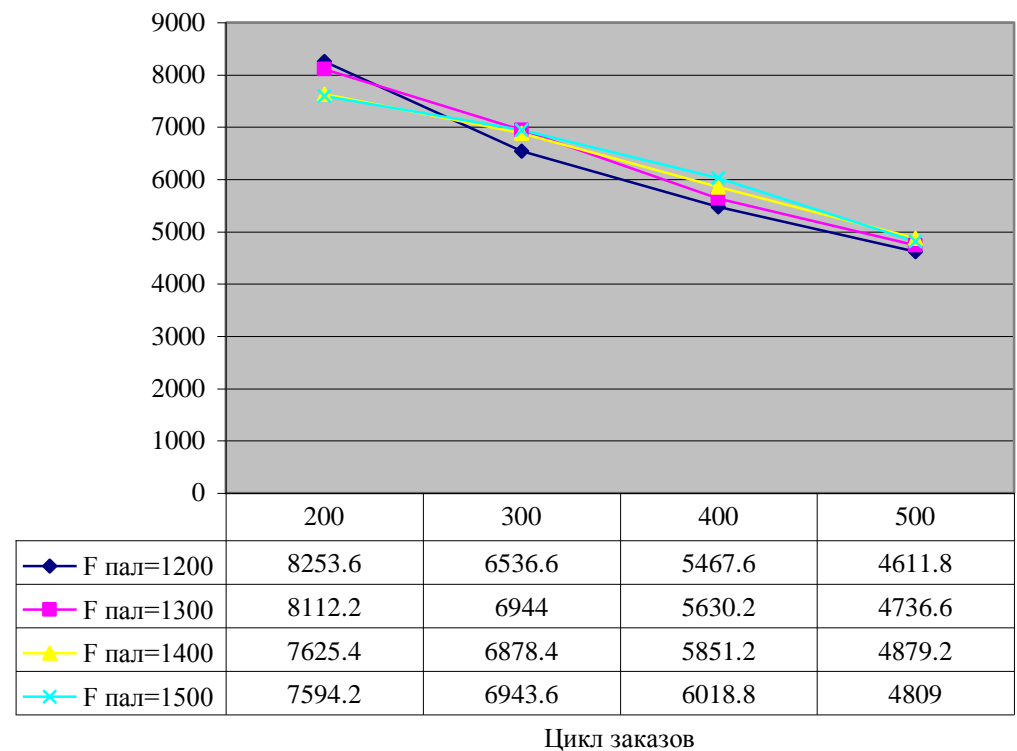


Рис. 10. Затраты реорганизации зоны отбора, I_2 . Варьирование размера зоны отбора

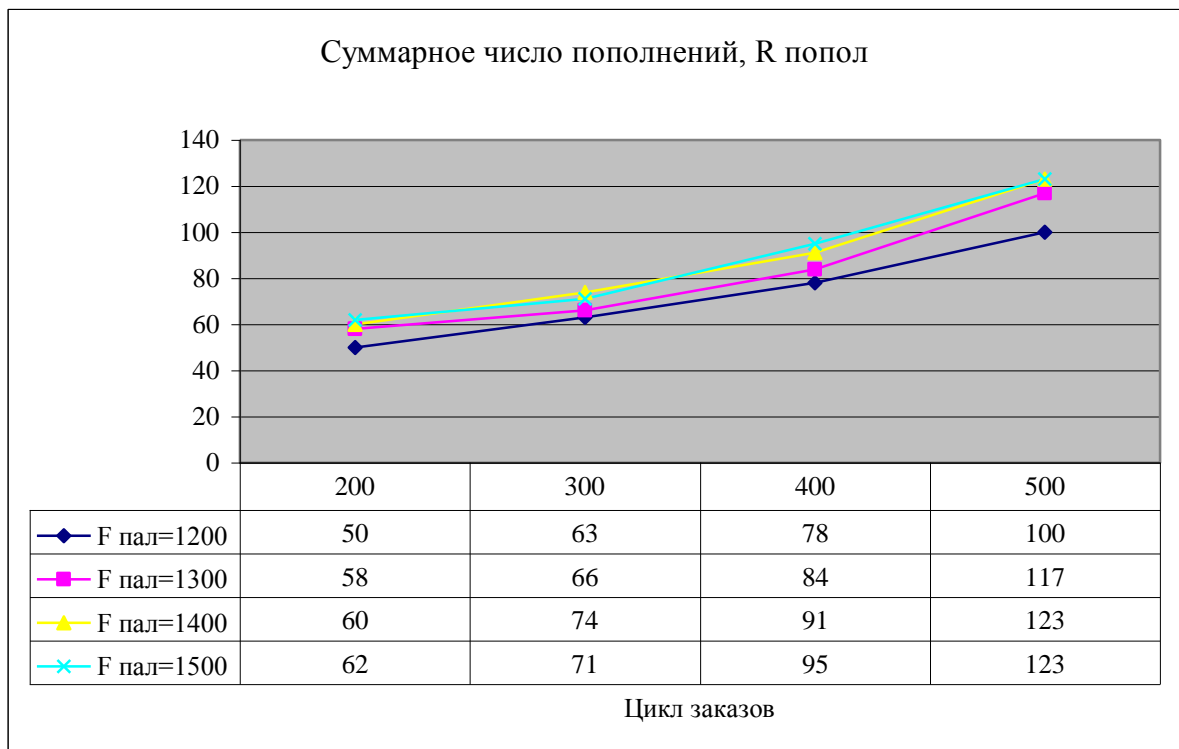


Рис. 11. Суммарное число пополнений, $R_{\text{попол}}$. Варьирование размера зоны отбора

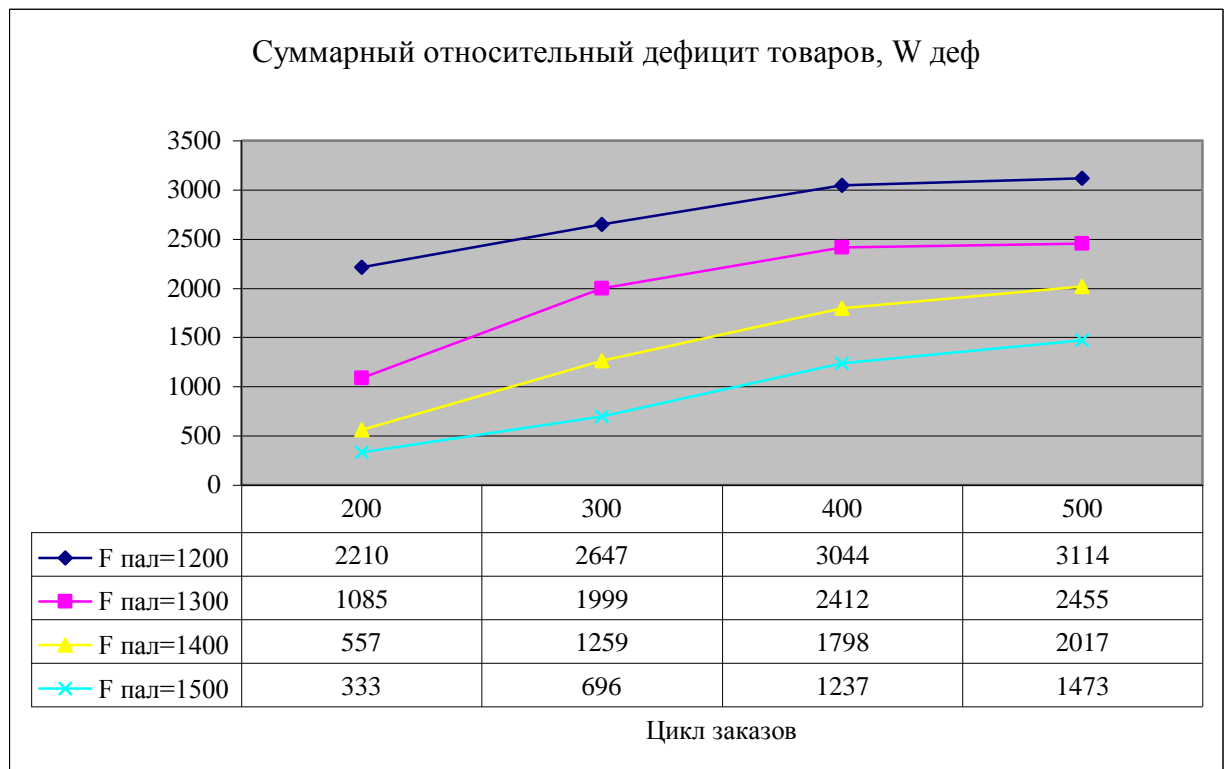


Рис. 12. Относительный дефицит товаров, $W_{\text{деф}}$. Варьирование размера зоны отбора

В случае, когда размер зоны отбора $F_{пал} = 1500$ палето-мест, то суммарные затраты на реорганизацию минимален.

Табл. 1. Затраты реорганизации (I_2) при различных значениях величины цикла заказов ($K_{цикл.зак}$). Размер зоны отбора $F_{пал} = 1500$ палето-мест.

$K_{цикл.зак}$	200	300	400	500
I_2	7594.2	6943.6	6018.8	4809

Также, при данном размере зоны отбора и цикле заказов $K_{цикл.зак} = 200$, относительный дефицит товаров минимален, $W_{деф} = 333$.

Это можно проинтерпретировать следующим образом: количество различных товаров в заказах сопоставимо с размером зоны отбора, т.е. заказы максимально удовлетворяются (максимально полно комплектуются), и требуются небольшие изменения зоны отбора при ее перерасчете.

Общие затраты при $F_{пал} = 1500$, так как производится большее число отборов, из-за низкого относительного дефицита товаров. По этой же причине увеличивается число пополнений.

Табл. 2. Суммарные затраты (I) при различных значениях величины цикла заказов ($K_{цикл.зак}$). Размер зоны отбора $F_{пал} = 1500$ палето-мест.

$K_{цикл.зак}$	200	300	400	500
I	18508.77	17139.83	15381.66	14029.1

Рассмотрим более подробно случай, $K_{цикл.зак} = 200$

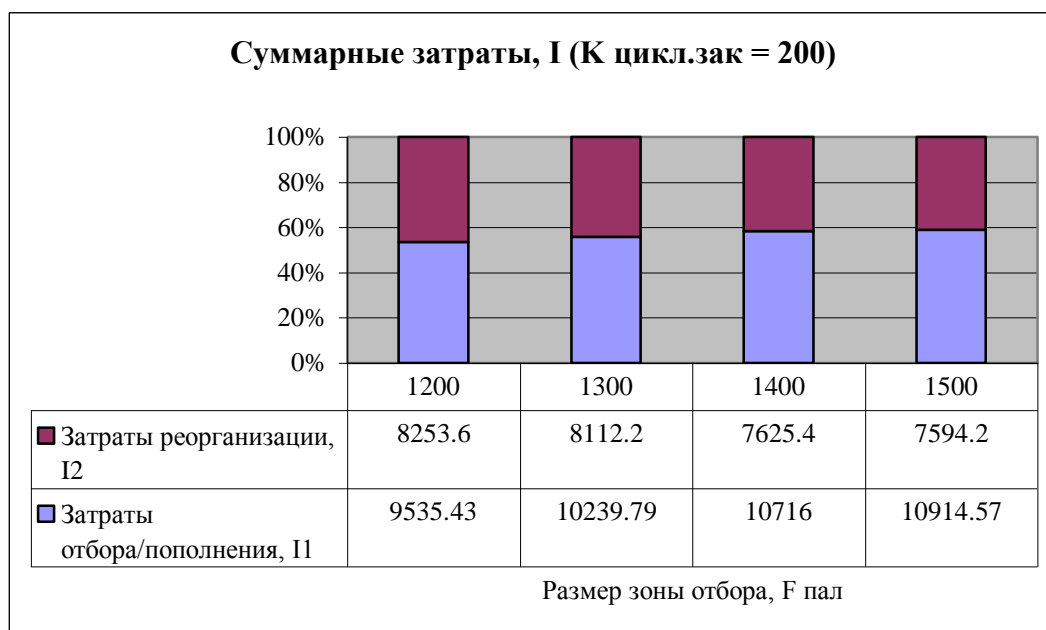


Рис. 13. Суммарные затраты, I . Цикл заказов $K_{цикл.зак} = 200$.

Несмотря на то, что затраты на реорганизацию зоны отбора (I_2) сокращаются, затраты отбора/пополнения (I_1) увеличиваются значительно. По этой причине, суммарные затраты увеличиваются с увеличением размера зоны отбора.

Табл. 3. Суммарные затраты (I) при различных размерах зоны отбора ($F_{пал}$). Цикл заказов $K_{цикл.зак} = 200$.

$F_{пал}$	1200	1300	1400	1500
I	17789,03	18351,99	18341,4	18508,77

Относительный дефицит товаров ($W_{деф}$) значительно сокращается с увеличением размера зоны отбора.

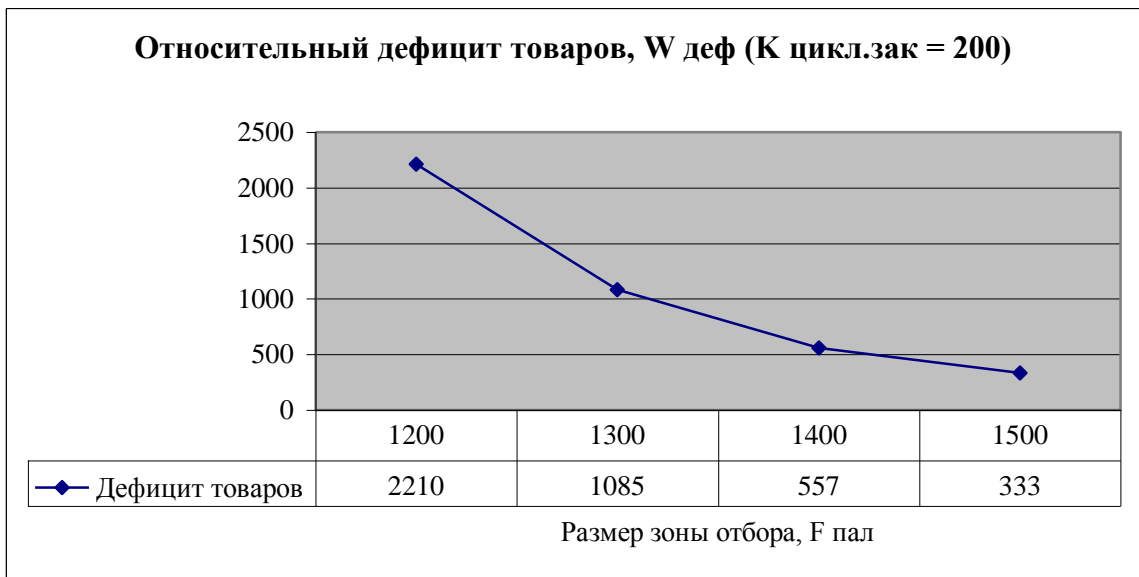


Рис. 14. Относительный дефицит товаров, $W_{деф}$. Цикл заказов $K_{цикл.зак} = 200$.

При дальнейшем увеличении размера зоны отбора, относительный дефицит товаров будет стремиться к нулю. Так как, зона отбора рассчитывается таким образом, чтобы в ней были представлены все товары для каждого цикла заказов, то когда размер зоны отбора будет больше номенклатуры товаров для каждого цикла заказов, тогда все заказы будут комплектоваться полностью.

Из-за специфики рассматриваемого склада (в заказах редко присутствуют товары в объеме больше одной палеты), с увеличением размера зоны отбора число пополнений меняется незначительно.

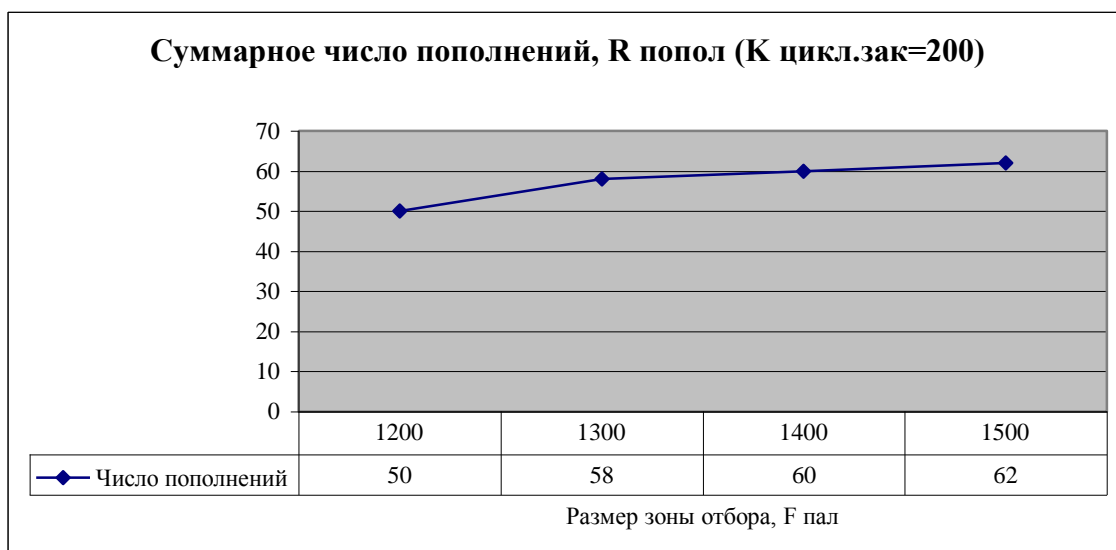


Рис. 15. Суммарное число пополнений, $R_{\text{попол}}$. Цикл заказов $K_{\text{цикл.зак}} = 200$.

3.3. Выводы

Из числа допустимых размеров зоны отбора, рациональным является - $F_{\text{пал}}^* = 1500$ палето-мест, для рассматриваемого склада, в том случае если ключевым показателем эффективности работы склада является, как можно более полная комплектация заказов, и как следствие больший оборот товаров.

Максимальная скомплектованность заказов напрямую зависит от относительного дефицита товаров. При размере зоны отбора $F_{\text{пал}}^* = 1500$ палето-мест, относительный дефицит товаров минимален, по сравнению с другими размерами, и равен $W_{\text{деф}} = 333$.

В случае, когда ключевым показателем эффективности работы склада являются суммарные затраты, то наилучший размер зоны отбора, из числа возможных, $F_{\text{пал}}^* = 1200$ палето-мест. В этом же случае достигается и минимальное число пополнений.

Что касается параметров алгоритма расчета, то увеличение первоначальной статистики ($P_{\text{зак}}^0$) приведет к улучшению суммарных характеристик, но незначительно. Варьирование величины цикла заказов ($K_{\text{цикл.зак}}$) специфично для каждого склада и требует дополнительного исследования.

4. Экономическая часть

4.1. Введение

В дипломном проекте рассматривается задача моделирования типовых складских процессов, а именно расчет параметров зоны отбора склада с целью минимизации общих затрат. Данная задача относится к классу задач дискретной нелинейной оптимизации.

Решение такого рода задач оказывается довольно сложным без привлечения ЭВМ и соответствующего программного обеспечения. Разработка программного обеспечения является достаточно трудоемким процессом и включает в себя несколько стадий:

1) разработка технического задания (ТЗ)

- постановка задачи
- определение необходимых прикладных программ, информационной базы
- выбор языков программирования
- выбор методов выполнения работы
- разработка календарного плана выполнения работ

2) проектирование разработки программного продукта

- разработка структуры входных и выходных данных
- разработка и (или) описание алгоритмов реализации решения данной задачи
- разработка структуры программы
- разработка пояснительной записки
- подготовка программной документации

3) технико-экономическое обоснование разработки программного продукта

- расчет себестоимости программного продукта
- обоснование экономической эффективности разработки

4.2. Расчет себестоимости программного продукта

Производство продукции (работ и услуг) связано с определенными затратами или издержками. Все затраты предприятия на производство и реализацию продукции, выраженные в денежной форме, образуют себестоимость продукции. Себестоимость - важнейший показатель эффективного использования производственных ресурсов. В нашей стране состав себестоимости продукции регламентируется государством. Основные принципы формирования этого состава определены в Законе РФ «О налоге на прибыль предприятий и организаций» и конкретизированы в Положении о составе затрат. Регламентирующая роль государства по отношению к себестоимости продукции проявляется также в установлении норм амортизации основных средств, тарифов отчислений на социальные нужды, в некоторые виды фондов и т.п.

Затраты, формирующие себестоимость продукции (работ, услуг) группируются по следующим элементам:

- 1) материальные затраты
- 2) затраты на оплату труда
- 3) единый социальный налог
- 4) амортизация основных фондов

5) прочие затраты (налоги, сборы, отчисления в фонды и т.п.)

4.2.1. Материальные затраты

К материальным затратам относятся затраты на предметы труда (используемые в производстве), запасные части и материалы для ремонта основных средств, износ инвентаря, инструментов, приборов и других средств труда, не относимых к основным фондам, затраты на сырье и материалы подсобных промышленных производств, затраты на работы и услуги производственного характера, выполняемые сторонними организациями или производствами предприятия, не относящимися к основному виду деятельности. К работам и услугам производственного характера относятся: транспортные работы по обслуживанию производства, оплата работ по всем видам ремонта техники сторонними организациями и др.

В таблице 1 приведены статьи материальных затрат на производство данного программного продукта, учитывающие затраты на оборудование рабочего места и на организацию процесса работы:

Табл. 4. Материальные затраты

Наименование товара	Ед. измерения	Кол-во	Цена за ед. в руб.	Общая стоимость, руб.
Компьютер ASUS M6000	Шт.	1	26423	26423
Принтер HP LaserJet 1018 Rus Black LE (CC563A)	Шт.	1	4199	4199
Картридж для принтера HP Q2612A	Шт.	1	2023	2023
Бумага для принтера	Пачка 500 листов.	1	150	150
Итого				32795

4.2.2. Затраты на оплату труда

Согласно статье 255 НК РФ в состав расходов на оплату труда включаются предусмотренные нормами законодательства либо трудовыми договорами (контрактами) и (или) коллективным договором:

- любые начисления работникам в денежной и (или) натуральной формах;
- стимулирующие начисления и надбавки, компенсационные начисления, связанные с режимом работы или условиями труда;
- премии и единовременные поощрительные начисления;
- расходы, связанные с содержанием работников.

Будем предполагать, что разработка данного программного продукта продолжительностью 3 мес. осуществляется

- 1) мной, студентом Московского Авиационного Института, получающим ежемесячно стипендию 1200 руб. Следовательно, за 3 месяца моя заработная плата составит $C_1 = 3\ 600$ руб.
- 2) научным руководителем, получающим ежемесячно (за 168 часов) заработную плату 8 000 руб. Время, потраченное на консультации в течение 3х месяцев,

составляет 35 часов. Следовательно, заработная плата консультанта по основной части диплома составит $C_2 = 8\,000 \cdot 35 / 168 = 1667$ руб.

- 3) консультантом по основной части диплома, получающим ежемесячно (за 168 часов) заработную плату 10 000 руб. Время, потраченное на консультации в течение 3х месяцев, составляет 20 часов. Следовательно, заработная плата консультанта по основной части диплома составит $C_3 = 10\,000 \cdot 20 / 168 = 1190$ руб.
- 4) консультантом по экономической части диплома, получающим ежемесячно (за 168 часов) заработную плату 10 000 руб. Время, потраченное на консультации в течение 3х месяцев, составляет 2 часа. Следовательно, заработная плата консультанта по экономической части диплома составит $C_4 = 10\,000 \cdot 2 / 168 = 119$ руб.
- 5) консультантом по разделу диплома «Охрана окружающей среды и БЖД», получающим ежемесячно (за 168 часов) заработную плату 10 000 руб. Время, потраченное на консультации в течение 3х месяцев, составляет 2 часа. Следовательно, заработная плата консультанта по разделу диплома «Охрана окружающей среды и БЖД» составит $C_5 = 10\,000 \cdot 2 / 168 = 119$ руб.

Итак, общие расходы на оплату труда составят

$$C = C_1 + C_2 + C_3 + C_4 + C_5 = 3\,600 + 1667 + 1190 + 119 + 119 = 6695 \text{ руб.}$$

4.2.3. Единый социальный налог

Согласно главе 24 НК РФ ЕСН составляет 26% от фонда заработной платы. Следовательно, единый социальный налог за 3 месяца составит $6695 \cdot 0.26 = 1740$ руб.

4.2.4. Расходы на амортизацию

Амортизация в бухгалтерском учёте — процесс перенесения по частям стоимости основных средств и нематериальных активов по мере их физического или морального износа на стоимость производимого продукта. Амортизируемым признается имущество со сроком полезного использования более 12 месяцев и первоначальной стоимостью более 10 000 рублей. Об этом говорится в пункте 1 статьи 256 Налогового кодекса.

В данном случае необходимо учесть амортизацию компьютера и программного обеспечения. Ежемесячная сумма амортизации зависит от срока полезного использования объекта и метода начисления амортизации. Компьютеры относятся ко второй амортизационной группе (постановление правительства от 1 января 2002 г. № 1). В нее входит имущество со сроком полезного использования свыше двух и до трех лет включительно. При разработке используется 1 ПК стоимостью 26423 руб. Пусть срок его полезного использования составляет 5 лет.

Программное обеспечение относится к нематериальным активам. Согласно п.14 ПБУ 14/2000 стоимость нематериальных активов также погашается посредством амортизации. Сумма амортизационных отчислений по нематериальным активам в целях исчисления налога на прибыль определяется в соответствии с пунктом 2 статьи 258 Налогового кодекса: «Определение срока полезного использования объекта нематериальных активов производится исходя из срока действия патента, свидетельства и (или) из других ограничений сроков использования объектов интеллектуальной собственности в соответствии с законодательством Российской Федерации или применимым законодательством иностранного государства, а также исходя из полезного срока использования нематериальных активов, обусловленного соответствующими договорами. По нематериальным активам, по которым невозможно определить срок полезного использования объекта нематериальных активов, нормы амортизации устанавливаются в

расчете на десять лет (но не более срока деятельности налогоплательщика)». Положим срок полезного использования программного обеспечения (в данном случае GAMS IDE, MS Excel) 5 лет. Стоимость лицензионных версий GAMS IDE составляет примерно 17 000 руб., MS Excel составляет примерно 6000 руб.

Будем рассчитывать амортизацию линейным способом. При применении линейного метода норма амортизации по каждому объекту амортизируемого имущества определяется по формуле:

$$A = \frac{P}{k},$$

где A - норма амортизации, P - текущая стоимость объекта основных средств, k - установленный срок полезного использования.

Таблица 5. Амортизация

Наименование основного средства	P , руб.	k , мес.	A , руб. в мес.
Компьютер	26423	60	440
GAMS IDE	17 000	60	284
MS Excel	6000	60	100
Итого			824

За 3 месяца расходы на амортизацию составят $824 * 3 = 2472$ руб.

4.2.5. Итоговые результаты

Таблица 6. Расчет себестоимости

Статьи расходов	Сумма, руб.
Материальные затраты	32795
Расходы на оплату труда	6695
Расходы на амортизацию	2472
Единый социальный налог	1740
Итого	43702

Следовательно, себестоимость готового программного продукта составляет 43702 руб.

4.3. Расчет цены программного продукта

Цена продукта = себестоимость продукта + прибыль + НДС.

Пусть прибыль составляет 20% от себестоимости программного продукта. Тогда прибыль равна $43702 * 0.2 = 8740$ руб.

Возьмем НДС равным 18%, $(43702 + 8740) * 0.18 = 9440$ руб.

Цена продукта = $43702 + 8740 + 9440 = 61882$ руб.

4.4. Экономическая эффективность разработки

Целесообразность внедрения новой разработки, программного продукта основывается на оценках общей эффективности капитальных вложений.

Эффектом может быть снижение себестоимости продукции, повышение качества продукции, прирост объема производства продукции и т.д. В конечном счете, все проектные решения направлены на получение дополнительной прибыли.

При расчете экономической эффективности проектных решений необходимо определить источник финансирования капитальных вложений. Возможны следующие источники: собственные средства предприятия и заемные (кредиты банка).

Если проведение мероприятий предполагается за счет кредита, то должны быть определены сроки возврата суммы кредита и платы за кредит. Использование кредита для финансирования капитальных вложений приведет к удлинению срока окупаемости, так как часть прибыли будет направлена на погашение процентов за пользование кредитом.

В данном случае мы используем собственные средства.

Прирост прибыли определяется по формуле:

$$\Delta\Pi = \Pi_c + \Pi_k + \Pi_n,$$

где

Π_c - прирост прибыли за счет снижения себестоимости;

Π_k – прирост прибыли за счет повышения качества продукции;

Π_n – прирост прибыли за счет роста объемов производства.

Коэффициент экономической эффективности инвестиций δ показывает, сколько прибыли будет получено на каждый рубль капитальных вложений:

$$\delta = \frac{\Delta\Pi}{K},$$

где капитальные вложения в данный программный продукт, или его цена ($K = 61882$ руб.).

Пусть использование программного продукта в течение 1 месяца привело к следующим результатам:

$$\Pi_c = 4\,000 \text{ руб.},$$

$$\Pi_k = 3\,500 \text{ руб.},$$

$$\Pi_n = 3\,000 \text{ руб.}$$

Тогда $\Delta\Pi = 4\,000 + 3\,500 + 3\,000 = 10\,500$ руб. и коэффициент экономической эффективности

$$\delta = \frac{10500}{61882} = 0.17$$

Данный программный продукт окупится за

$$\eta = \frac{1}{\delta} = \frac{61882}{10500} \approx 6 \text{ мес.}$$

5. Охрана труда и окружающей среды

5.1. Введение

Ежегодно в мире собирается огромное количество компьютеров. Следствием этого является огромный рост числа новых программных продуктов. Дома, на работе, в институте: уже практически не осталось областей, как в науке, так и в обыденной жизни, в которые не вторглись бы ЭВМ. Все больше времени мы проводим, глядя на экран монитора.

В своей дипломной работе я формализовал задачу перерасчета параметров зоны отбора склада и рассмотрел алгоритм, позволяющий решить данную задачу. Данный алгоритм может быть использован для принятия решения о реорганизации зоны отбора склада, но при этом необходимо производить сбор определенной аналитики для расчетов и последующее сценарное моделирование. Все это выполняется при помощи компьютера, а значит возникает новая проблема: проблема безопасности. Ведь неправильно подобранные и расположенные устройства могут оказать существенное влияние на самочувствие пользователя программного комплекса. Ниже я приведу рекомендации, которые необходимо соблюдать, чтобы максимально обезопасить себя и свести утомляемость к минимуму.

Обычно безопасность работы на ЭВМ рассматривают в двух аспектах. Первый определяется физиологическими особенностями работы человека за компьютером. Второй – техническими параметрами средств компьютеризации. Эти составляющие – "человеческая" и "техническая" – тесно переплетены и взаимозависимы. Науку, занимающуюся исследованием подобных проблем назвали эргономикой. Эргономика – это наука о взаимодействии человека, основной целью которой является создание совершенной и безопасной техники, максимально ориентированной на человека, организация рабочего места, профилактика труда.

5.2. Причины опасности работы за компьютером

Основными повреждающими здоровье при работе за компьютером, являются следующие неспецифические (т.е. не связанные именно с работой за компьютером) факторы:

Длительная гиподинамия. Любая поза при длительной фиксации вредна для опорно-двигательного аппарата, кроме того, ведет к застою крови во внутренних органах и капиллярах.

Нефизиологическое положение различных частей тела.

Длительно повторяющиеся однообразные движения. Они могут вести к физическому повреждению суставов и сухожилий.

Долгое пребывание в замкнутом, а еще хуже - душном помещении.

Также существуют специфические повреждающие факторы при работе с компьютером. Это, в основном, световое, электромагнитное и прочее излучение, в основном монитора.

Для борьбы с неспецифическими повреждающими факторами рекомендации просты - надо хотя бы раз в час устраивать перерывы, походить, размяться. Еще лучше выполнять пару физических упражнений. Совсем хорошо составить для себя комплекс упражнений для самостоятельной разблокировки позвоночника. А если какие-то проблемы уже возникли, лучше обратиться к специалисту.

На специфических повреждающих факторах я бы хотел остановиться подробнее. Как правило, наибольший вред здоровью пользователя компьютера наносят устройства ввода-вывода: монитор, клавиатура, мышь.

5.2.1. Устройство вывода информации

5.2.1.1. Стандарты безопасности

В наше время, когда проблемы безопасности работы за компьютером стоят как нельзя остро, появляется множество различных стандартов на экологическую безопасность оборудования персонального компьютера. Считается, что современный монитор должен соответствовать по крайней мере четырем общепринятым стандартам безопасности и эргономике:

FCC Class B - этот стандарт разработан канадской федеральной комиссией по коммуникациям для обеспечения приемлемой защиты окружающей среды от влияния радиопомех в замкнутом пространстве. Оборудование, соответствующее требованиям FCC Class B, не должно мешать работе теле- и радио аппаратуры.

MPR-II - этот стандарт был выпущен в Шведском национальном департаментом. MPR-II налагает ограничения на излучения от компьютерных мониторов и промышленной техники, используемой в офисе.

TCO'95 (а также современный TCO'99) - рекомендация, разработанная Шведской конференцией профсоюзов и Национальным советом индустриального и технического развития Швеции (NUTEK), регламентирует взаимодействие с окружающей средой. Она требует уменьшения электрического и магнитного полей до технически возможного уровня с целью защиты пользователя. Для того, чтобы получить сертификат TCO'95 (TCO'99), монитор должен отвечать стандартам низкого излучения (Low Radiation), т.е. иметь низкий уровень электромагнитного поля, обеспечивать автоматическое снижение энергопотребления при долгом не использовании, отвечать европейским стандартам пожарной и электрической безопасности

EPA Energy Star VESA DPMS - согласно этому стандарту монитор должен поддерживать три энергосберегающих режима - ожидание (stand-by), приостановку (suspend) и "сон" (off). Такой монитор при долгом простое компьютера переводится в соответствующий режим, с низким энергопотреблением.

Но это не всё. Во-первых, стандарты безопасности/эргономики (в частности TCO), есть лишь некоторый уровень параметров, который считается безопасным на сегодняшний момент и который можно реализовать при определенных условиях. Во-вторых, безопасность не ограничивается только низкими интенсивностями электромагнитных полей, более того, в понятии эргономическая безопасность значения переменных электромагнитные (э/м) полей стоят на последнем месте.

5.2.1.2. Вредоносное влияние монитора

Итак, какие основные параметры наносят вред здоровью? Перечислим их в порядке убывания степени влияния на здоровье.

Картинка на экране монитора по своим характеристикам не соответствует тому, к чему со времен появления человека приспособлены наши глаза, и, соответственно, зрение портится.

Постоянный электростатический потенциал, то есть постоянное электрическое поле, которое создает экран монитора, вредно сказывается на здоровье.

Переменные электромагнитные поля, которые создают электрические внутренности монитора, также влияют на самочувствие.

5.2.1.3. Влияние на зрение

Плохая освещенность (как слабая, так и слишком большая) мгновенно сказывается на зрении и усталости. Нормы освещенности экрана жестко прописаны в ГОСТе и нормах СанПин на рабочее место. Они составляют около 250 лк. Ясно, что простому пользователю эта цифра мало что скажет. К сожалению самим измерить освещенность сложно. Проверить это могут только люди с соответствующим оборудованием и квалификацией. Поэтому приведу пример: Если ваша настольная лампа (60-100 Вт) с абажуром светит на стол и клавиатуру (но не напрямую на экран монитора) и еще включено верхнее освещение (одна две лампы на потолке), то приблизительно можно считать, что это допустимый стандартами уровень освещенности. Ни в коем случае нельзя ставить яркие лампы у себя за спиной. Это приведет не только к увеличению интенсивности освещения, но и к наличию бликов на экране, которые отвлекают внимание, снижают качество восприятия информации на экране.

Теперь обратимся к визуальным параметрам, к самому изображению. Посмотрим на изображение не с точки зрения монитора, а с точки зрения человека. Для человека нужно, чтобы глядя на изображение на мониторе его глаза не уставали. Т.е., чтобы картинка воспринималась нами также, как и другие предметы, попадающие в поле зрения. А вот это как раз и невозможно. Дело в том, что наши глаза устроены для восприятия предметов в отраженном свете. Мы не можем смотреть на солнце или на лампу. А монитор является лампой. Кроме того, сам принцип формирования изображения из дискретных частей (точек) и быстрая поочередная подсветка каждой точки для формирования стабильного изображения тоже противоречит “понятиям” принятым природой. Так вот для того, чтобы максимально приблизить параметры картинки на экране монитора к привычным нашему глазу значениям и вводятся в стандарты эргономики требования к частоте обновления экрана, яркости, контрастности, цветотемпературы, наличию антибликового покрытия. Итак, изображение на экране, чтобы глаза не уставали должно удовлетворять следующим требованиям:

Частота регенерации экрана (кадровой развертки) не менее 85 Гц;

Яркость и контрастность изображения на экране должна соответствовать яркости и контрастности окружающих предметов. Диапазоны изменения этих параметров должны позволять настраивать таким образом монитор в широких диапазонах освещенностей. При этом на экране желательно, чтобы было различимо максимальное количество различных градаций поля серого.

Отсутствие бликов на экране. Конечно, все современные кинескопы имеют антибликовое покрытие, которое заметно снижает отражения от экрана. Однако ни одно современное покрытие не способно защитить от блика на экране от лампы, расположенной у вас за спиной.

Необходимо выбрать тот видеорежим, в котором вы можете с расстояния 50 см от экрана монитора различать символы как в центре, так и на углах изображения без напряжения глаз. Иначе говоря, изображение должно быть четким. Этот параметр не прописан ни в одном из стандартов, так как зависит от многих причин и зачастую резко варьируется от одного монитора к другому. На четкость изображения влияют фокусировка, контрастность и т.д. Поэтому надо выбирать экземпляр монитора с наилучшей четкостью изображения.

5.2.1.4. Влияние электростатического потенциала и э/м полей

Современная наука не знает точно как влияют на организм человека э/м поля. Есть отдельные работы в этой области, но точно проверенных данных нет. Более того, человек постоянно находится под воздействием э/м полей. Любой электрический бытовой прибор создает переменные э/м поля. Поэтому, те цифры, которые написаны в нормах ТСО,

ГОСТ указаны из расчета – чем меньше, тем лучше. И на сегодняшний момент развития технологий изготовления дисплеев эти нормы реально достижимы и, поэтому, считаются безопасными.

Влияние электростатического потенциала, стоящего в нашем списке вторым, изучено немного лучше и доказано, что долгое воздействие электростатического поля вызывает кожные заболевания. Поэтому если клавиатура расположена близко от монитора с большим статическим потенциалом экрана, то есть риск возникновения кожных (раковых) заболеваний. Что же можно сделать, чтобы снизить их влияние на организм?

Во-первых, надо помнить, что монитор не будет удовлетворять требованиям на э/м излучение и электростатический потенциал если он не заземлен. Во всяком случае ни один производитель этого не гарантирует. Розеток с заземлением в наших домах нет. Нужна розетка европейского стандарта, чтобы если не заземлить, то хотя бы “занулить” компьютер и монитор.

Во-вторых, как можно меньше используйте удлинителей и разветвителей. Именно длинные кабели питания и сигнальные кабели зачастую становятся источником э/м излучения.

Приняв эти меры предосторожности, вы максимально снизите уровни полей создаваемых монитором и компьютером. Можно также прибегнуть к защитным фильтрам, однако не всякий фильтр вам поможет, а хорошему современному монитору достаточно и его внутренних систем защиты.

Надо отметить, что в случае с оборудованием компьютерных классов и офисов ситуация еще сложнее. Если каждый монитор удовлетворяет стандартам ТСО, ГОСТ, то весь комплекс в целом этим стандартам может и не удовлетворять. Вредные поля от каждого монитора и компьютера могут как складываться так и взаимно уничтожаться и заранее знать какая ситуация на том или ином рабочем месте. Плохо проведенная общая “земля” может наоборот стать основным источником э/м полей. И проверить это можно только измерив эти поля. Поэтому лучше, заранее, в процессе обустройства класса или офиса проконсультироваться у профессионала.

5.2.2. Устройства ввода информации

В отличие от мониторов для компьютерных устройств ввода (клавиатура и мышь) в настоящее время не имеется общепринятых и широко распространенных стандартов. В тоже время многие производители данного оборудования рекламируя свою продукцию, описывают различные конструктивные решения, повышающие эргономичность ее использования: клавиатура с возможностью регулирования расположение клавиш, мышь с формой, уменьшающей усталость кисти при длительной работе. Хотя некоторые из них стоит рассматривать только как броскую рекламу, многие модели действительно являются своеобразным технологическим скачком вперед с точки зрения безопасности работы за компьютером.

Однако есть и некоторые общие правила. Мышь должна соответствовать размеру руки. Сейчас многие новые мыши оснащены колесиком, и это удобно в работе. Держать такую мышь стоит за края большим пальцем и мизинцем, чтобы указательный лежал на левой кнопке, средний на колесике, а безымянный на правой кнопке. При этом запястье должно лежать на столе постоянно, а катать мышь по столу надо только движениями пальцев.

5.3. Выводы

Как уже было сказано ранее, при организации рабочего места, лучше всего обратиться за помощью к профессионалу. Но если такой возможности нет, то соблюдение вышеперечисленных требований, накладываемых на выбор, расположение и

освещенность монитора, а также на выбор устройств ввода информации, поможет существенно снизить вредное воздействие на организм специфических повреждающих факторов. Это особенно важно для людей, которые вынуждены проводить за компьютером большую часть своего времени.

Использование данных рекомендаций поможет пользователям разработанного программного комплекса намного дольше сохранить свое здоровье в норме и избежать быстрого переутомления.

Приложение 1. Графическое представление оптимального решения для части товаров.

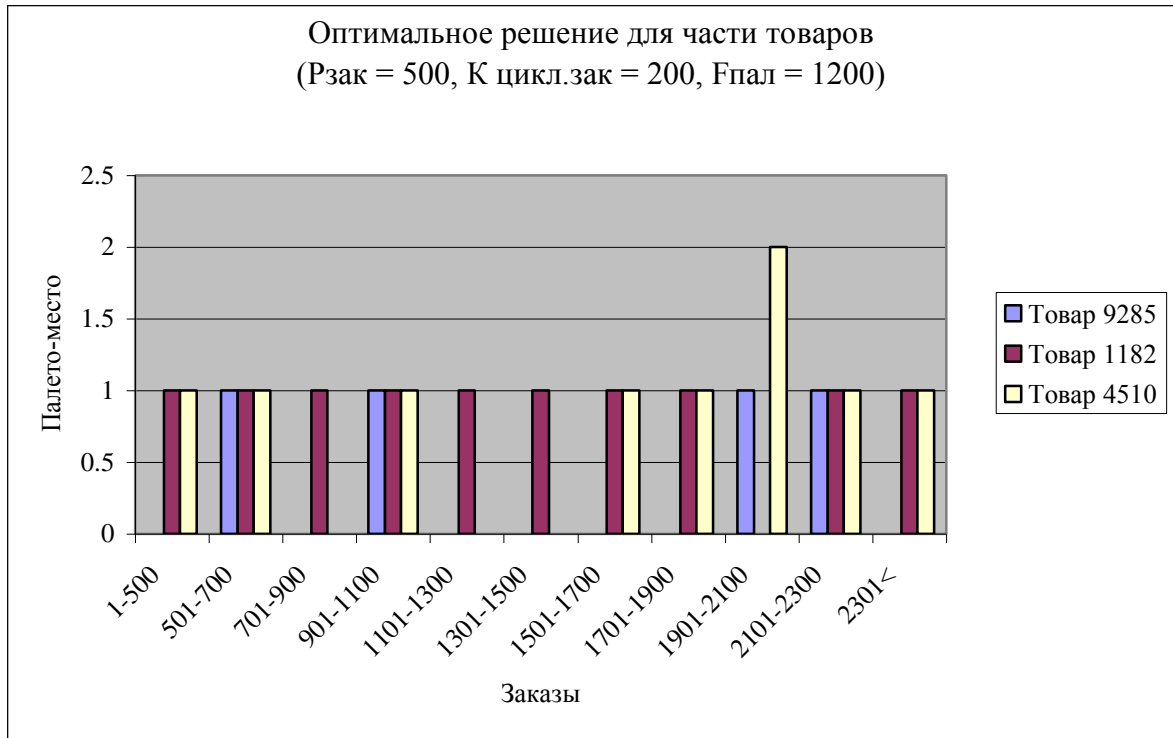


Рис. 16. Оптимальное решение для части товаров ($P_{зак} = 500$ заказов, $K_{цикл.зак} = 200$ заказов, $F_{пал} = 1200$ палето-мест)

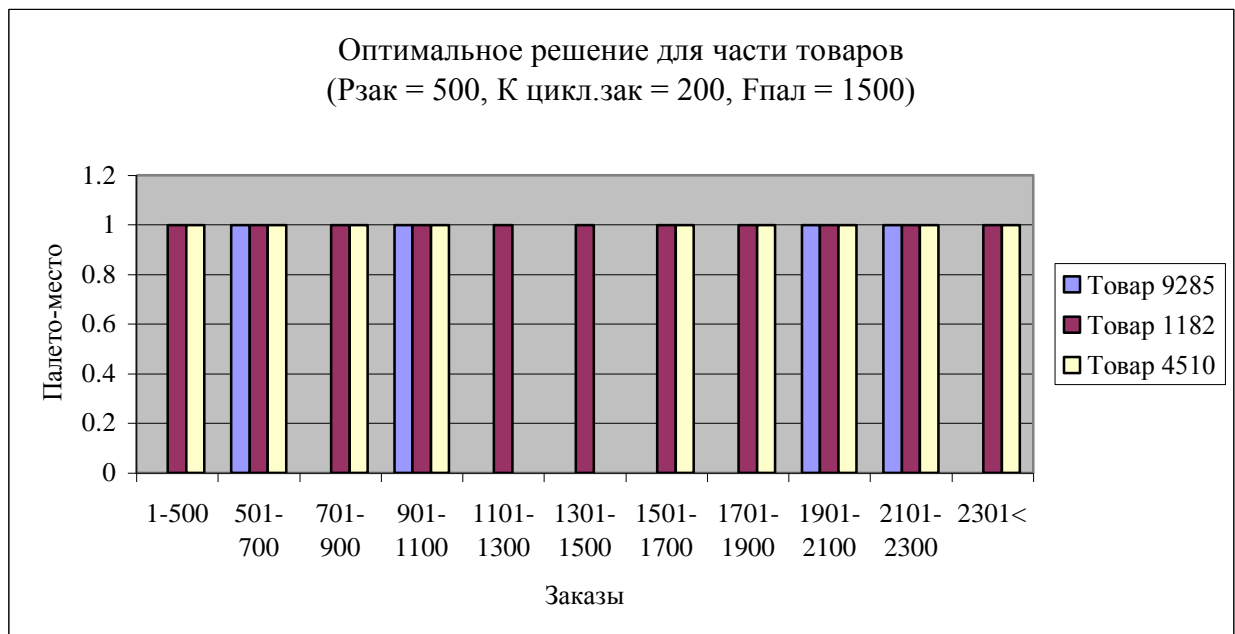


Рис. 17. Оптимальное решение для части товаров ($P_{зак} = 500$ заказов, $K_{цикл.зак} = 200$ заказов, $F_{пал} = 1500$ палето-мест)

Список литературы

- 1) **Warehouse and distribution science, John J. Bartholdi, Steven T. Hackman**
- 2) Информация о тарифах на стандартные складские операции,
http://www.loginfo.ru/arhiv/2007/log06_07s3.php
- 3) «Введение в вычислительный анализ и целочисленное программирование», Белоусов Е.Г. 1977
- 4) «Приближенные методы решения дискретных задач оптимизации», И.В. Сергиенко, Т.Т. Лебедева, В.А. Рощин, Киев «Наукова думка» 1980
- 5) «Математические модели и методы решения задач дискретной оптимизации», И.В. Сергиенко, Киев «Наукова думка» 1988
- 6) Арсеньев В.В., Сажин Ю.Б. Методические указания к выполнению организационно-экономической части дипломных проектов по созданию программной продукции. М.: изд. МГТУ им. Баумана, 1994. 52 с.
- 7) «Налоговый кодекс Российской Федерации. Части первая и вторая». – М.: ТК Велби, Изд-во Проспект, 2006г.
- 8) Методические указания к организационно-экономической части дипломных проектов и работ – М.: МАИ, 1996г.
- 9) <http://www.gams.com/sales/academicp.pdf>
- 10) Охрана труда. Под ред. Б.А. Князевского. - "Высшая школа" 1982.
- 11) Дайнов М.И., Метечко Л.Б.,Толоконникова В.В. Методические указания к дипломному проектированию «Экологические платежи за загрязнение окружающей природной среды»-М.:Изд-во МАИ, 2001г
- 12) ГОСТ 27954 – 88 «Видеомониторы персональных ЭВМ. Типы, основные параметры, общие технические требования»
- 13) ГОСТ 12.4.124-83 «Средства защиты от статического электричества. Общие требования»