

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Национальный исследовательский Нижегородский государственный  
университет им. Н.И. Лобачевского»  
(ННГУ)**

**Институт информационных технологий, математики и механики**

Направление подготовки: «Фундаментальная информатика и  
информационные технологии»

отчёт к лабораторной работе  
на тему:  
**«Разработка Системы Поддержки Принятия Решений  
(СППР) для отыскания наилучшей стратегии переработки  
сахарной свеклы»**

**Выполнил:**

студенты группы 3823Б1ФИЗ

Кутузов И. А.

Завьялов А. А.

Поташник М. Я.

Лузан Е. А.

Гонозов Л. А.

**Научный руководитель:**

доцент каф. ДУМЧА,

Эгамов А. И.

Нижегород  
2025 г.

# Содержание

<b>Введение</b>	<b>2</b>
<b>1 Классическая задача о назначениях</b>	<b>3</b>
1.1 Постановка задачи . . . . .	3
1.2 Эквивалентная постановка задачи . . . . .	4
1.3 Алгоритм решения задачи о назначениях . . . . .	4
<b>2 Нечеткая задача о назначениях и эвристические методы ее решения</b>	<b>6</b>
2.1 Постановка модифицированной нечеткой задачи о назначениях . . . . .	6
2.2 Эвристические методы решения нечеткой задачи о назначениях . . . . .	6
<b>3 Задача о нахождении плана переработки сахарной свеклы, максимизирующего выход сахара</b>	<b>7</b>
3.1 Постановка задачи . . . . .	7
3.2 Процесс дозаривания . . . . .	7
3.3 Распределение коэффициентов на заданных отрезках . . . . .	7
3.4 Потери сахара . . . . .	8
3.5 Задача оптимизации . . . . .	8
<b>4 Описание программы для оптимизации переработки сахарной свеклы</b>	<b>9</b>
4.1 Общая информация . . . . .	9
4.2 Установка и первый запуск . . . . .	9
4.3 Пользовательский интерфейс . . . . .	10
4.3.1 Ввод данных . . . . .	10
4.3.2 Запуск и получение результатов . . . . .	13
4.3.2.1 Графическое представление . . . . .	14
4.3.2.2 Текстовое представление результатов . . . . .	16
4.4 Детали реализации . . . . .	17
4.4.1 Техническая часть . . . . .	17
4.4.2 Вычислительная часть . . . . .	17
<b>А Приложение: CTG</b>	<b>19</b>
<b>В Приложение: GitHub</b>	<b>19</b>

## Введение

В данной лабораторной работе рассматривается задача о назначениях, а именно её модификация - нечёткая задача о назначениях, возникающая при планировании переработки сахарной свеклы.

Мы сталкиваемся с типичной производственной ситуацией: у нас есть несколько партий сырья (свеклы) с разной начальной сахаристостью, которая со временем меняется — обычно уменьшается при хранении, но может также немного увеличиться в начале вследствие процесса дозаривания. Определить заранее, как точно будет меняться сахаристость каждой партии невозможно. Это обусловлено влиянием на свеклу множества факторов, большая часть из которых может стать известна только за малое время до того, как они окажут сильное влияние на сохранность свеклы.

Данная задача является «нечёткой» задачей о назначениях, где решение нужно принимать последовательно, имея неполную информацию. Точные алгоритмы (такие как венгерский) не применимы, поскольку требуют полных входных данных.

Возможным вариантом решения проблемы является применение различных эвристических стратегий. Однако невозможно заранее определить, какую именно следует применить, для достижения результата, близкого к идеальному.

**Целью** данной работы является создание СППР (Система Поддержки Принятия Решений), которая позволила бы производить оценку эффективности различных эвристических стратегий в заданных условиях.

# 1. Классическая задача о назначениях

## 1.1. Постановка задачи

Сформулируем математическую постановку задачи о назначениях: Пусть имеется  $n$  исполнителей и  $n$  работ. Задана матрица состояний  $C = (c_{ij})$  размера  $n \times n$ , где  $c_{ij} \geq 0$  — стоимость выполнения  $j$ -й работы  $i$ -м исполнителем.

Необходимо найти такое соответствие работ исполнителям (одну работу одному исполнителю), чтобы расходы на оплату труда были наименьшими. Если цель состоит в нахождении назначения с наибольшей стоимостью, то решение сводится к решению аналогичной задачи путём замены каждой стоимости  $c_{ij}$  на разность между максимальной стоимостью  $c_{ij}$ , т.е. матрица себестоимостей требует преобразования по правилу  $\bar{c} = \max(c_{ij}) - c_{ij}$ .

Введём переменные  $x_{ij}$  следующим образом:

$$x_{ij} = \begin{cases} 1, & \text{если } i\text{-й исполнитель назначен на } j\text{-ю работу,} \\ 0, & \text{иначе.} \end{cases}$$

Тогда матрица назначений  $X = (x_{ij})$  состоит из нулей и единиц.

Запишем **целевую функцию** - суммарную стоимость выполнения всех работ:

$$S(X) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \rightarrow \min,$$

которую нужно минимизировать. Также при решении задачи следует учитывать некоторые ограничения:

1. Каждый исполнитель выполняет ровно одну работу:

$$\sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n}.$$

2. Каждая работа выполняется ровно одним исполнителем:

$$\sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n}.$$

3. Условие бинарности переменных:

$$x_{ij} \in \{0, 1\}, \quad i, j = \overline{1, n}.$$

Таким образом, задача о назначениях формулируется как задача целочисленного линейного программирования:

$$\begin{cases} \min_X \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}, \\ \sum_{j=1}^n x_{ij} = 1, \quad i = \overline{1, n}, \\ \sum_{i=1}^n x_{ij} = 1, \quad j = \overline{1, n}, \\ x_{ij} \in \{0, 1\}, \quad i, j = \overline{1, n}. \end{cases}$$

## 1.2. Эквивалентная постановка задачи

Дадим эквивалентную постановку задачи о назначениях.

Пусть дана матрица  $C$  размера  $n \times n$  с неотрицательными элементами  $c_{ij}$ , где  $i = \overline{1, n}$ ,  $j = \overline{1, n}$ .

Выбор строки относительно столбца описывается перестановкой

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ \sigma(1) & \sigma(2) & \dots & \sigma(n-1) & \sigma(n) \end{pmatrix}$$

натуральных чисел от 1 до  $n$ .

Целевая функция имеет вид:

$$S(\sigma) = c_{1\sigma(1)} + c_{2\sigma(2)} + c_{3\sigma(3)} + \dots + c_{n\sigma(n)}. \quad (5)$$

Оптимизационная задача состоит в том, чтобы найти такую перестановку  $\sigma$  чисел от 1 до  $n$ , при которой значение  $S(\sigma)$ , будет минимальным (или максимальным, в зависимости от постановки).

Перестановку  $\sigma$ , при которой функция  $S(\sigma)$  принимает максимальное значение, будем называть *максимальной*, а принимающую минимальное значение — *минимальной*.

Нетрудно видеть, что перестановка  $\sigma$  гарантирует выбор элементов таким образом, что в каждой строке и в каждом столбце матрицы  $C$  выбирается ровно один элемент, входящий в сумму  $S(\sigma)$ .

## 1.3. Алгоритм решения задачи о назначениях

Задача о назначениях является частным случаем транспортной задачи. При ее решении методом полного перебора потребуется  $O(n!)$  операций. Ее можно решить любым способом, пригодным для решения транспортной задачи, в том числе симплекс-методом, однако, имеются и более подходящие алгоритмы. Рассмотрим самый известный из них - венгерский алгоритм. Доказано, что его сложность равна  $O(n^4)$ . Приведём сам алгоритм:

1. Если задача решается на максимум, то в каждой строке матрицы необходимо найти максимальный элемент, его же вычесть из каждого элемента соответствующей строки и умножить всю матрицу на  $(-1)$ . Если задача решается на минимум, то этот шаг необходимо пропустить.
2. В каждой строке матрицы состояний находят минимальный элемент и вычитают его из всех элементов строки.
3. В каждом столбце матрицы состояний находят минимальный элемент и вычитают его из всех элементов столбца.
4. Находят строку с одним 0, заключают этот 0 в квадрат и называют отмеченным. В столбце, содержащем отмеченный 0, все 0 зачёркивают и дальше не рассматривают. Продолжают шаг, пока возможно.
5. Находят столбец с одним 0 и отмечают найденный 0. В строке, содержащей отмеченный 0, все остальные 0 зачёркивают. Продолжают шаг, пока возможно.
6. Если после выполнения шагов 4-5 остались неотмеченные 0, то отмечают любой из них, а в строке и столбце, содержащих отмеченный 0, все остальные 0 зачёркивают.

7. Оптимальное решение достигнуто в том случае, если в каждой строке и в каждом столбце матрицы есть ровно один отмеченный 0. В противном случае проводят минимальное число пересекающихся вертикальных и горизонтальных прямых через все 0.
8. Среди не зачёркнутых прямыми чисел ищут минимальное, вычитая его из всех не зачёркнутых чисел и прибавляя ко всем числам, стоящим на пересечении двух (вертикальной и горизонтальной) прямых. К полученной матрице применяют алгоритм, начиная с шага 4.

## 2. Нечеткая задача о назначениях и эвристические методы ее решения

### 2.1. Постановка модифицированной нечеткой задачи о назначениях

Поставим **модифицированную задачу о назначениях**.

Для известной матрицы  $C$  найти перестановку  $\sigma$  чисел от 1 до  $n$ , такую, что значение целевой функции  $S(\sigma)$ , было бы максимальным. Кроме того, значения  $\sigma(s)$  выбираются последовательно (сначала  $\sigma(1)$ , затем  $\sigma(2)$  и т.д.). Выбор значения  $\sigma(j)$  происходит, основываясь на знании только первых  $j$  столбцов матрицы  $C$  (в момент выбора  $\sigma(j)$ , остальные  $n - j$  столбцов матрицы  $C$  неизвестны).

Поставим **нечеткую задачу о назначениях**.

Предположим, заданы константы  $c_{\min}, c_{\max}, b_{\min}, b_{\max}$ . Назовем матрицу  $C$  допустимой, если

$$c_i \in [c_{\min}, c_{\max}], \quad i = \overline{1, n}, \quad b_{ij} \in [b_{\min}, b_{\max}], \quad i = \overline{1, n}, \quad j = \overline{1, n-1},$$

а сама она строится по формуле  $c_{ij} = c_{i1} \cdot b_{i1} \cdot \dots \cdot b_{ij-1}, j = \overline{2, n}$

Если в модифицированной задаче допустимая матрица  $C$  задается описанным выше способом, элементы  $c_i, i = \overline{1, n}$ , известны, а параметры  $b_{ij-1}, i = \overline{1, n}$ , становятся известными в момент выбора  $\sigma(j)$ , то задача преобразуется в модифицированную нечеткую задачу о назначениях или, для простоты, в «нечеткую задачу о назначениях».

### 2.2. Эвристические методы решения нечеткой задачи о назначениях

Нетрудно видеть, что всевозможные алгоритмы, опирающиеся на заранее известную матрицу  $C$  в условиях неопределённости работать не будут. Следовательно, для поставленной нечеткой задачи необходимо осуществлять поиск других, эвристических методов. Ниже приведены известные эвристические методы.

1. «Жадная» (Greedy) - стратегия заключается в использовании «жадного» алгоритма. В  $j$ -ом столбце выбирается  $\max(c_{ij})$  среди ещё не выбранных  $i$
2. «Бережливая» (Thrifty) - стратегия использует противоположный по смыслу жадному – «бережливый» алгоритм: в  $j$ -ом столбце выбирается  $\min(c_{ij})$  среди ещё не выбранных  $i$
3. «Бережливая/жадная» стратегия состоит в том, что для первых  $v$  столбцов реализуется «бережливый» алгоритм, а после, начиная с  $v + 1$ -го, используется «жадный» алгоритм.
4. «Жадная/бережливая» стратегия для первых  $v$  реализуется «жадный» алгоритм, а после, начиная с  $v + 1$ -го, используется «бережливый».
5. Стратегия типа БкЖ (T(k)G) состоит в том, что в первых  $v - 1$  столбцах выбирается элемент, который в данный момент находится на  $k$ -ой позиции, считая от наименьшей (1 позиция). Далее, начиная  $v$  столбца, используется «жадный» алгоритм.
6. CTG - особый алгоритм, который является оптимальным для определённого вида матриц. Теорему можно найти в «Приложение: CTG»;
7. «Случайная» (Random) - элемент в каждом столбце выбирается случайным образом.
8. «Быстрогниющий сначала» - в первом столбце выбирает  $\max(c_{1j})$ , для остальных столбцов для каждой строки  $i$  считается среднее арифметическое по всем  $b_{ik}$ , где  $k \in \overline{0, i-1}$ . Выбирается элемент, в строке которого это среднее арифметическое наибольшее.

### 3. Задача о нахождении плана переработки сахарной свеклы, максимизирующего выход сахара

#### 3.1. Постановка задачи

Пусть имеется  $n$  партий сахарной свеклы. Каждая партия  $i$  ( $i = \overline{1, n}$ ) характеризуется начальной долей сахара  $a_i \in [a_{\min}, a_{\max}]$ .

Процесс переработки разбит на  $n$  этапов. На каждом этапе  $j$  ( $j = \overline{1, n}$ ) перерабатывается ровно одна партия массой  $M$ , где  $M$  — производственная мощность завода.

При хранении содержание сахара в партиях изменяется. Введём коэффициенты деградации свеклы  $b_{ij} \in (0, 1)$  ( $i = \overline{1, n}, j = \overline{1, n-1}$ ), где  $b_{ij}$  показывает относительное изменение доли сахара в партии  $i$  при переходе от этапа  $j$  к этапу  $j+1$ .

Матрица состояний  $C = (c_{ij})_{n \times n}$  определяет долю сахара в партии  $i$ , если она перерабатывается на этапе  $j$ :

$$c_{i1} = a_i; \quad c_{ij} = a_i \prod_{k=1}^{j-1} b_{ik}, \quad j = \overline{2, n}.$$

Нетрудно видеть, что данная задача является нечёткой задачей о назначении на нахождение максимума.

#### 3.2. Процесс дозаривания

Дозаривание свёклы - процесс, при котором после сбора урожая, при дальнейшем хранении фруктов, овощей, корнеплодов, в них какое-то время повышается доля сахара.

Для упрощения математической модели будем считать, что число этапов, во время которых происходит дозаривание одинаково для всех партий сахарной свеклы. Также считаем, что процесс дозаривания длится от первого этапа до начала  $v$  этапа,  $2 \leq v \leq [\frac{n}{2}]$ . На этапах дозаривания параметры  $b_{ij} \in (\beta_{\min}, \beta_{\max}]$ ,  $\beta_{\min} \geq 1$ ,  $i = \overline{1, n}, j = \overline{1, v-1}$ .

#### 3.3. Распределение коэффициентов на заданных отрезках

Параметры  $a_i$  и  $b_{ij}$  могут распределяться по-разному на заданных отрезках. Особо важным является распределение коэффициентов деградации  $b_{ij}$ . Будем различать два варианта:

1.  $b_{ij}$  - распределены равномерно на отрезке  $[\beta_1, \beta_2]$
2. Концентрированное распределение:
  - - У каждого этапа  $b_{ij}$  лежат внутри отрезка, длина которого минимум вчетверо меньше исходного
  - При каждом  $i$  существует константа  $\delta_i$ ,  $\delta_i \leq \left| \frac{\beta_2 - \beta_1}{4} \right|$ , такая, что  $b_{ij} \in [\beta_i^1, \beta_i^2] \subset [\beta_1, \beta_2]$ , где  $|\beta_i^1 - \beta_i^2| = \delta_i$ .
  - На отрезке  $[\beta_i^1, \beta_i^2]$  параметры  $b_{ij}$  распределены равномерно.
  - Иначе говоря, в этом случае коэффициенты деградации каждой партии лежат в маленькой окрестности некой точки, то есть деградация свеклы происходит «приблизительно одинаково» на каждом этапе.

Для дозаривания определения аналогичны.

### 3.4. Потери сахара

Рассматривается матрица  $\tilde{S} = C - \tilde{L}$ , где элемент  $l_{ij}$  - потери сахара при обработке  $i$ -ой партии свеклы на  $j$ -ом этапе. Значение  $l_{ij}$  (в %) может быть вычислено по следующей формуле:  $l_{ij} = 1.1 + 0.1541 \times (K_i + Na_i) + 0.2159 \times N_i + 0.9989 \times I_{ij} + 0.1967$ , где соответствующие значения:

- $K_i \in [4.8; 7.05]$ ;
- $Na_i \in [0.21; 0.82]$ ;
- $N_i \in [1.58; 2.8]$ ;  $I_{ij} = I_{i0} \times (1.029)^{7j-7}$ , где  $j$  - номер этапа переработки (нумерация с 1); 7 - количество дней (за один этап переработки примем неделю)
- $I_{i0} \in [0.62; 0.64]$ .

### 3.5. Задача оптимизации

Имеется матрица состояний  $\tilde{S} = C - \tilde{L}$ , где

- $C$  - Матрица коэффициентов содержания сахара для каждой партии(строки) на каждом этапе(столбцы)
- $\tilde{L}$  - матрица с неотрицательными элементами потери сахара
- $\tilde{S}$  - матрица коэффициентов содержания сахара после учёта потерь

Последовательность переработки сырья будем описывать перестановкой натуральных чисел от 1 до  $n$ :

$$\sigma = \begin{pmatrix} 1 & 2 & \dots & n-1 & n \\ \sigma(1) & \sigma(2) & \dots & \sigma(n-1) & \sigma(n) \end{pmatrix}$$

А общий выход сахара будем рассчитывать по формуле:

$$S(\sigma) = \tilde{s}_{s(1)1} + \tilde{s}_{s(2)2} + \dots + \tilde{s}_{s(n)n}$$

Масса конечного продукта  $= S * M$ . Оптимизация состоит в том, чтобы найти такую перестановку  $\sigma$  чисел от 1 до  $n$  (последовательность переработки имеющихся партий сырья), при которой  $S(\sigma)$ , будет максимальным. -  $\sigma$  - стратегия -  $\sigma$ , при которой  $S$  максимальна - максимальная стратегия

## 4. Описание программы для оптимизации переработки сахарной свеклы

### 4.1. Общая информация

В рамках данной работы была разработана система поддержки принятия решений, задача которой - моделирование процесса деградации и переработки сахарной свёклы, а также оценка возможных методов её решения в условиях неопределённости, когда входные данные представлены в виде интервалов возможных значений.

**Виртуальным экспериментом** будем называть генерацию матрицы  $C$  и процесс решения нечёткой задачи о назначениях.

**Генеральным экспериментом** будем называть проведение  $N$  виртуальных экспериментов и получение решения путём усреднения их результатов.

Программа производит генеральный эксперимент на основе заданных начальных условиях и выводит результат в графическом и текстовом представлениях. Такая система позволяет пользователю наглядно понять разницу между различными стратегиями, оценить пригодность их для своей ситуации и принять правильное решение в реальной ситуации.

Некоторые общие особенности программы:

- Для сравнения наряду с эвристическими методами применяются алгоритмы «MINIMA» и «HUNGARIAN», данные алгоритмы применяются только когда известна вся матрица состояний  $C$ , что, как было сказано выше, невозможно в условиях задачи. Однако в рамках компьютерного эксперимента это позволяет наглядно увидеть, насколько близко можно приблизиться к идеальному результату, используя эвристические алгоритмы. При составлении рекомендаций данные алгоритмы не учитываются;
- Реализована возможность изменения масштаба путём нажатия сочетаний клавиш «ctrl-» и «ctrlshift=».
- Есть возможность создания вкладок - можно параллельно иметь несколько вариантов входных данных и переключаться между результатами простой сменой вкладки;
- Также потянув за фиолетовые границы между блоками можно изменять их пропорции. Например, выделить больше места на график.

### 4.2. Установка и первый запуск

Программа предоставляется пользователю в portable-формате. Это означает, что установка для её использования не требуется. Пользователь получает .zip архив - популярный способ распространения программного обеспечения. Сначала нужно распаковать архив: для этого достаточно нажать по нему правой кнопкой мыши и выбрать соответствующий пункт.

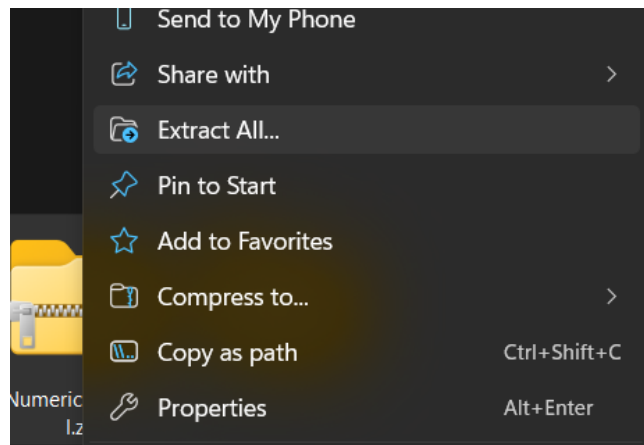


Рис. 1: Распаковка архива.

Затем нужно открыть папку с распакованным архивом, найти в ней *DecisionSupportSystem.exe* и запустить его.

### 4.3. Пользовательский интерфейс

При запуске будет открыт интерфейс программы, состоящий из нескольких блоков.

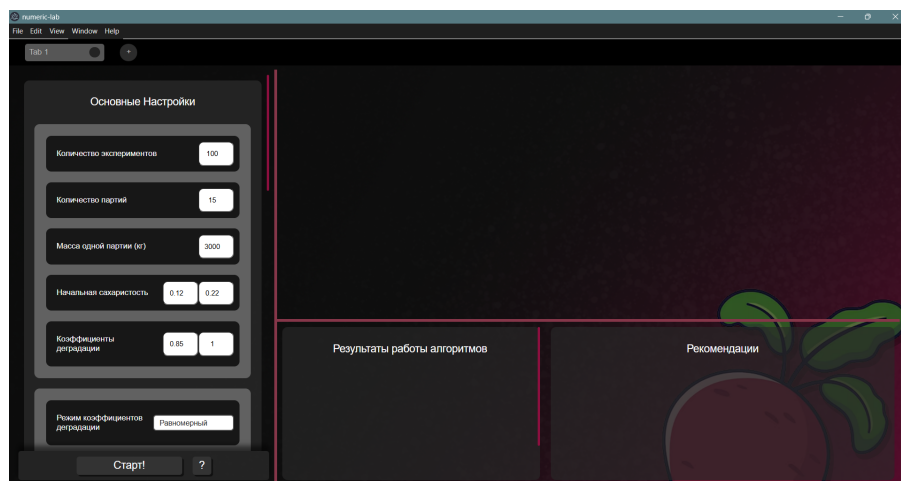


Рис. 2: Интерфейс программы.

Слева расположен блок ввода данных для проведения генерального эксперимента. Для доступа ко всем полям необходимо привести указатель мыши в область блока и с помощью колёсика выполнить прокрутку вниз.

#### 4.3.1. Ввод данных

Изначально в поля уже введёны шаблонные данные, однако пользователь сможет легко изменить их. Блок «основных настроек» включает в себя ввод следующих данных:

1. Количество экспериментов - количество виртуальных экспериментов, содержащихся в генеральном;
2. Количество партий - количество партий (также является количеством этапов переработки) свеклы, доступных заводу;

3. Масса одной партии (кг) - масса свеклы в одной партии;
4. Начальная сахаристость - диапазон значений начальной сахаристости свеклы. Для каждой партии будет выбрано число из этого отрезка.
5. Коэффициенты деградации - диапазон, в котором будут сгенерированы коэффициенты деградации для каждого этапа и партии;
6. Режим коэффициентов деградации - были описаны в секции «Распределение коэффициентов на заданных отрезках». При выборе опции «Концентрированный» генерации коэффициентов деградации будет выполнена в случайном подинтервале диапазона коэффициентов деградации из предыдущего пункта;
7. Диапазон концентрированной деградации - отношение длины подинтервала к диапазону коэффициентов деградации из предыдущего пункта или во сколько раз будет «сжат» диапазон значений коэффициентов деградации при выборе опции «Концентрированный».

Основные Настройки

Количество экспериментов 100

Количество партий 9

Масса одной партии (кг) 1

Начальная сахаристость -0.12 0.22

Коэффициенты деградации 0 1

Режим коэффициентов деградации Равномерный

Диапазон концентрированной деградации 0.25

Рис. 3: Основные настройки.

Блок дополнительных настроек позволяет задавать некоторые настройки эвристических стратегий (описания стратегий можно найти в секции «Эвристические методы решения нечеткой задачи о назначениях»):

1. Стадия Жадно-Бережливого алгоритма - этап, начиная с которого будет изменена стратегия выбора следующей партии с «жадной» на «бережливую» в Жадно-Бережливом алгоритме;

2. Стадия Бережливо-Жадного алгоритма - этап, начиная с которого будет изменена стратегия выбора следующей партии с «бережливой» на «жадную» в Бережливо-Жадном алгоритме;
3. Стадия алгоритма БКЖ - этап, начиная с которого следующей партией будет не та, которая в данный момент находится на  $k$ -ой позиции, считая от наименьшей (1 позиция), а партия с наибольшей долей оставшейся сахаристости («жадный» алгоритм);
4. Ранг алгоритма БКЖ - задание переменной  $k$  для БКЖ алгоритма;
5. Стадия алгоритма СТГ - это номер этапа переработки, на котором завершается процесс дозаривания и начинается процесс увядания партий сахарной свеклы. До этапа  $v - 1$  включительно происходит дозаривание (увеличение сахаристости), а начиная с этапа  $v$  — увядание (уменьшение сахаристости). Этот параметр является ключевым для построения оптимальной перестановки  $\omega$ , которую генерирует алгоритм СТГ.

Дополнительные Настройки	
Стадия Жадно-Бережливого алгоритма	2
Стадия Бережливо-Жадного алгоритма	2
Стадия алгоритма БКЖ	1
Ранг алгоритма БКЖ	1
Стадия алгоритма СТГ	1

Рис. 4: Дополнительные настройки.

Также имеются дополнительные блоки, которые могут быть включены по желанию пользователя, нажатием на кнопку (изначально красная, после включения - зелёная) в поле «Учитывать...»;

Список дополнительных блоков:

- Режим дозаривания - возможность включить режим дозаривания и задать число стадий, на которых будет происходить дозаривание, а также его коэффициенты. Описание процесса дозаривания дано в секции «Процесс дозаривания»;
- Неорганические вещества - учитывать влияние неорганических веществ при переработке свеклы. Можно задать несколько элементов. Более подробное описание учета влияния неорганических веществ можно найти в секции «Потери сахара»;

### Неорганические Вещества

☒ Учитывать неорганические в-ва

Диапазон Калия

07.05

Диапазон Натрия

-0.210.81

Диапазон Азота

.052.8

Диапазон I

00.64

### Дозаривание

☒ Учитывать дозаривание

Число стадий дозаривания

1

Диапазон дозаривания

11.15

Рис. 5: Дополнительные блоки.

### 4.3.2. Запуск и получение результатов

Для запуска программы необходимо нажать кнопку «Старт!» в левом нижнем углу, ниже блока задания начальных условий.

По введённым данным будет построен:

1. Графики, отражающие состояние каждого алгоритма на каждом этапе в генеральном эксперименте;
2. Список алгоритмов и усреднённый по всем виртуальным экспериментам результат, которого они достигли;
3. Рекомендации.



Рис. 6: Результат работы программы.

#### 4.3.2.1. Графическое представление

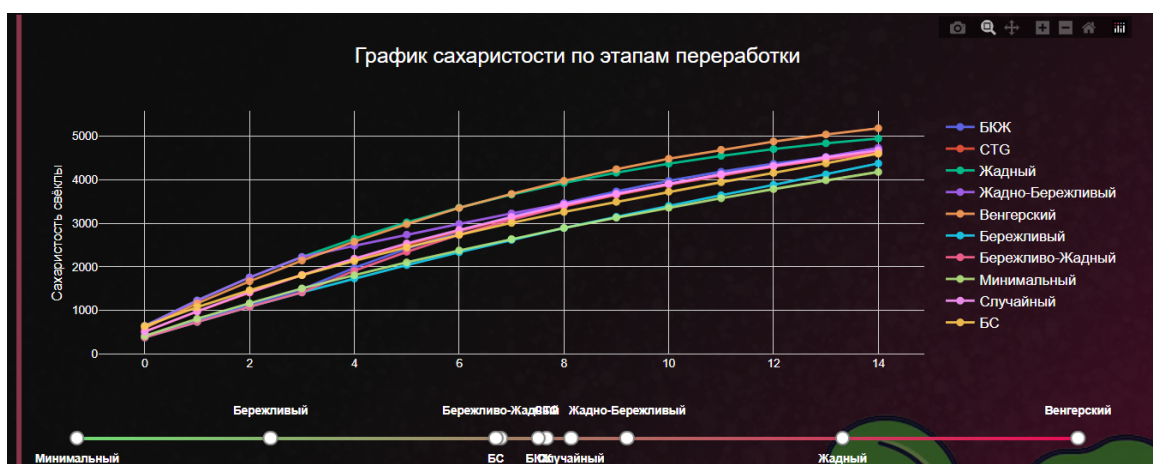


Рис. 7: Полученные графики.

Построенные графики демонстрируют результаты генерального эксперимента, а именно для каждой стратегии иллюстрируется масса полученного на конкретном этапе сахара. График обладает высокой степенью интерактивности. Например, пользователь может скрывать графики некоторых стратегий путём нажатия на название графика в легенде. Данный функционал позволяет уменьшить количество элементов на экране и производить сравнительное исследование поведения некоторых стратегий, не отвлекаясь на другие.

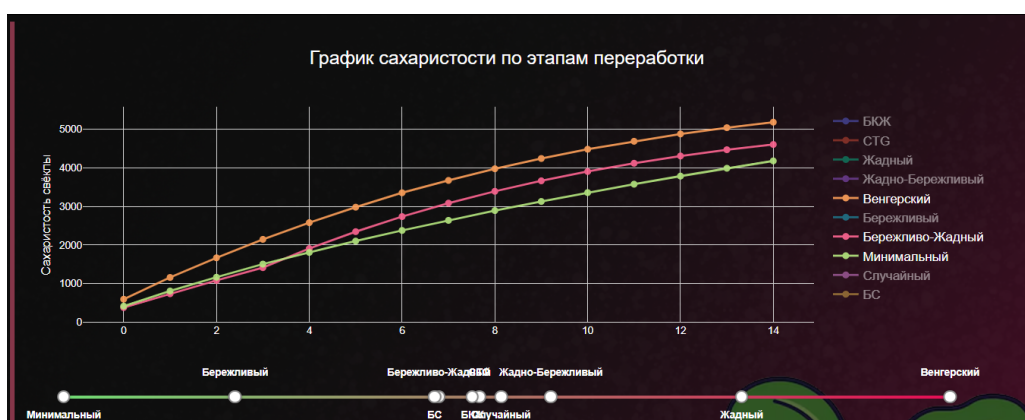


Рис. 8: Скрытие графиков.

Также при наведении на график есть возможность узнать точное его значение в этой точке.

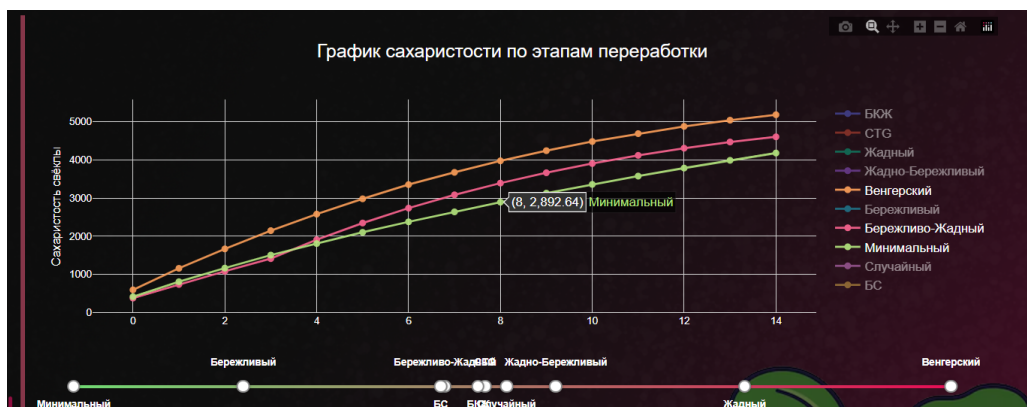


Рис. 9: Точное значение.

В правом верхнем углу графика присутствует множество инструментов, позволяющих выполнять манипуляции с графиком. при наведении на инструмент можно прочесть его название. приведём описание некоторых из них:

- **Download plot as png** - позволяет сохранить график в формате *.png*, который затем можно будет встроить, например, в отчёт;
- **Zoom** - позволяет выбрать область для приближения. Для возврата к исходному размеру можно воспользоваться инструментом **Autoscale**;

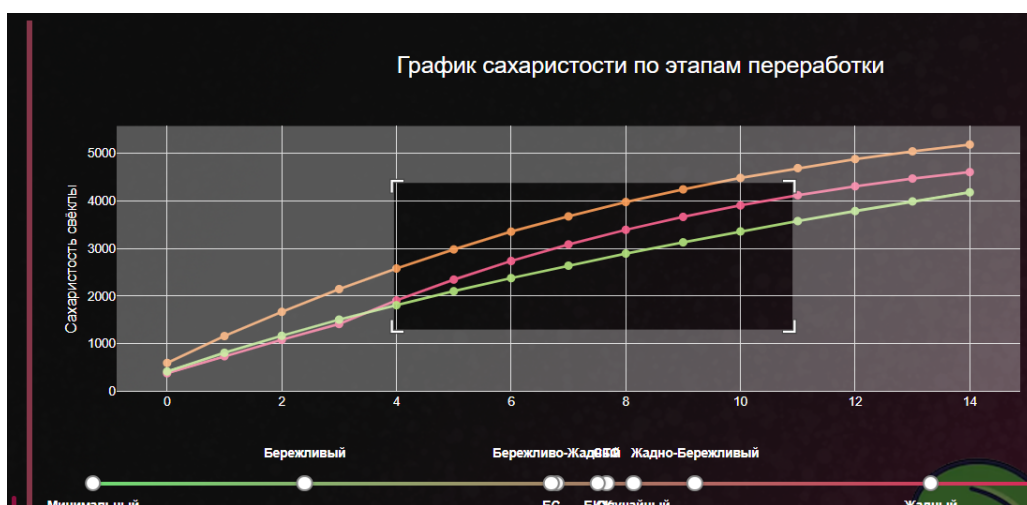


Рис. 10: Выбор области для приближения.

- **Pan** - позволяет «зажать» левую кнопку мыши и двигать график;
- **Zoom in** и **Zoom out** - регулируют масштаб графика.

Также чуть ниже графика приведен отрезок, представляющий собой удобное графическое представление для сравнения результатов различных стратегий. Он позволяет наглядно сравнить, насколько одна стратегия лучше другой и наоборот.

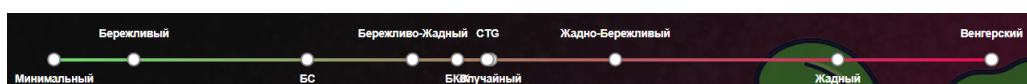


Рис. 11: Отрезок с результатами использования стратегий.

#### 4.3.2.2. Текстовое представление результатов

Ниже графика представлено текстовое представление результатов генерального эксперимента. В правом блоке приведен список всех стратегий, отсортированных от лучшей к худшей по количеству сахара (в килограммах), которое сможет принести использование данной стратегии, а также это же значение в процентах от наилучшего результата.

в левом блоке представлен вывод о лучшей и худшей стратегиях для текущих входных данных.

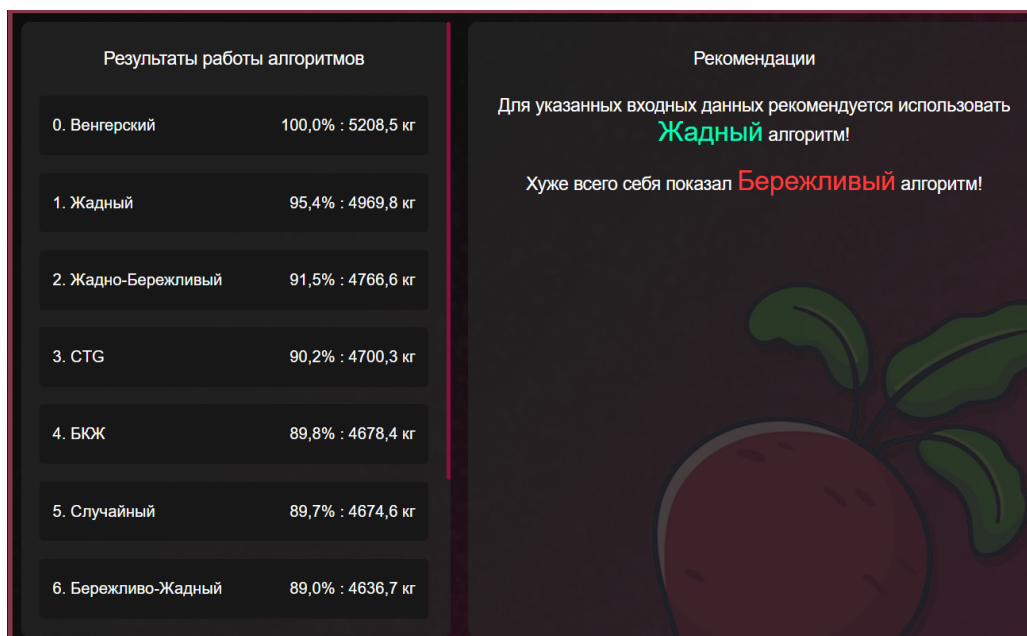


Рис. 12: Текстовое представление результатов.

При желании можно изменить входные данные и снова нажать кнопку старт для запуска программы с новыми условиями.

## 4.4. Детали реализации

Архитектура приложения была спроектирована таким образом, чтобы обеспечить расширяемость, стабильность и удобство сопровождения. Реализованные модули взаимодействуют последовательно и обеспечивают полный цикл работы - от ввода информации до получения итогового решения. Отдельное внимание уделено удобству работы пользователя с программой. Интерфейс разработан таким образом, чтобы обеспечить интуитивность, минимизировать количество лишних действий и сделать процесс использования максимально понятным даже для неподготовленных пользователей.

### 4.4.1. Техническая часть

В основе приложения лежит Electron engine, он отвечает за:

1. Создание главного окна приложения;
2. Загрузку интерфейса (HTML/CSS/JS);
3. Взаимодействие между интерфейсом и внутренней логикой;

Пользовательский интерфейс построен на веб-технологиях:

- HTML/CSS - структура страницы и её оформление;
- JavaScript - интерактивность, обработка действий пользователя, отправка запросов алгоритмическому модулю.

Данный подход обеспечивает гибкость пользовательского интерфейса за счёт возможности использовать web-технологии.

### 4.4.2. Вычислительная часть

Все вычисления внутри приложения выполнены с использованием Python.

Краткое описание основных классов программы:

- *Core* - вычислительное ядро программы, выполняет функции оболочки для вычислений и функции передачи данных в интерфейс;
- *Preprocessor* - выполняет предобработку данных - создаёт матрицу состояний  $C$ , с которой после работают алгоритмы;
- *Solver* - обработка поступивших данных доступными алгоритмами-стратегиями;
- *Postprocessor* - структурирует ответы, полученные классом *Solver*;
- Классы алгоритмов - находятся в папке */Algorithms/*. Каждый содержит реализацию некоторого алгоритма стратегии из «Эвристические методы решения нечеткой задачи о назначениях».

Также присутствуют некоторые классы-структуры, призванные упростить передачу данных между классами:

- *Input* - описывает структуру, в которой храниться вся введенная пользователем информация о параметрах генерального эксперимента;
- *Data* - содержит только те данные из *Input*, которые необходимы алгоритмам после генерации матрицы состояний  $C$ ;
- *Result* - структура, хранящая результаты вычислений, которые будут представлены пользователю.

## Заключение

Проведённый анализ показывает, что задача планирования переработки сахарной свеклы представляет собой нетривиальную оптимизационную проблему, формализуемую как **нечёткая задача о назначениях**. Её ключевая особенность заключается в необходимости принятия последовательных решений в условиях неполной информации о динамике изменения сахаристости партий сырья.

Основные выводы:

1. **Производственная значимость** - задача имеет непосредственное практическое применение в агропромышленном комплексе, где оптимизация порядка переработки партий свеклы может привести к значительному экономическому эффекту за счёт увеличения выхода сахара;
2. **Математическая сложность** - классические точные алгоритмы решения задачи о назначениях (например, венгерский алгоритм) неприменимы в условиях неопределённости, когда матрица состояний (сахаристости) не известна полностью на момент начала принятия решений;
3. **Эвристический подход** - единственным практически реализуемым путём решения является применение различных **эвристических стратегий**, способных работать с частичной информацией и принимать решения на каждом этапе переработки;
4. **Необходимость системного анализа** - поскольку эффективность эвристик сильно зависит от конкретных входных параметров, необходим инструмент для их комплексной оценки.

Таким образом, создание **СППР (Системы Поддержки Принятия Решений)** становится не только целесообразным, но и необходимым шагом. Такая система должна обеспечивать:

- Моделирование процесса переработки при различных входных данных;
- Реализацию набора эвристических алгоритмов для последовательного назначения партий;
- Сравнительный анализ эффективности алгоритмов по критерию общего выхода сахара;
- Визуализацию результатов и формирование обоснованных рекомендаций по выбору стратегии для конкретных производственных условий.

Реализация данной системы позволит перевести процесс планирования переработки сахарной свеклы из области интуитивных решений в область **доказательно-оптимизационного подхода**, что соответствует современным требованиям к управлению технологическими процессами в агропромышленном комплексе.

## А. Приложение: CTG

**Теорема А.1 (11).** Пусть для параметров матрицы  $C$  выполняются следующие условия:

1.  $c_{11} = c_{21} = \dots = c_{n1}$ ;
2.  $b_{ij} = b_i, \quad i = 1, n, \quad 2 \leq v \leq [n/2], \quad j = 1, v - 1$ ;
3.  $b_{ij} = \tilde{b}_i, \quad \tilde{b}_i < 1, \quad i = 1, n, \quad j = v, n$ ;
4.  $1 < b_1 < b_2 < \dots < b_n < \left(\frac{n-1}{n-2}\right)^{\frac{n}{n+1}}$ ;
5. изначально известен порядок 4);
6.  $\tilde{b}_k = \varphi_k b_k^{-1}, \quad \varphi_k = b_k^{-s}$ , где фиксированная константа  $s \in [0, 1/n]$ .

При данных условиях перестановка  $\omega$  – оптимальная, здесь

$$\begin{aligned}\omega(1) &= n - 2v + 3, \\ \omega(2) &= n - 2v + 5, \\ &\dots, \\ \omega(v-1) &= n - 1, \\ \omega(v) &= n, \\ \omega(v+1) &= n - 2, \\ \omega(v+2) &= n - 4, \\ &\dots, \\ \omega(2v-1) &= n - 2v + 2, \\ \omega(2v) &= n - 2v + 1, \\ &\dots, \\ \omega(n-1) &= 2, \\ \omega(n) &= 1.\end{aligned}$$

## В. Приложение: GitHub

Полный код программы выложен на ресурсе GitHub.com:  
<https://github.com/Akarfire/NumericMethodsLabWork/>.