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**

Центр прикладных информационных технологий

Направление подготовки: «Фундоментальная информатика и
информационные технологии»

ЛАБОРАТОРНАЯ РАБОТА

на тему:

«Оптимальный план переработки различных партий сахарной свеклы»

Выполнили:
студенты группы 3821Б1ФИ2:
Исаев Д.
Косарев Ег.
Казанцев Ев.
Козырева Ек.
Вьюнов Д.
Канаков Р.

Преподаватель:
Эгамов А. И.

Нижний Новгород
2023 г.

1 Введение

Пусть есть n партий сахарной свеклы равной массы, занумерованных от 1 до n. Масса одной партии свеклы – это масса, которую производственные мощности предприятия могут переработать за определенный промежуток времени (например, за одни сутки). Различные партии отличаются по производственной ценности – проценту выхода готового продукта из единицы массы (что соответствует сахаристости, проценту содержания сахара в свекле). Обозначим производственную ценность – долю содержания сахара в одном килограмме свеклы (сахаристость) i-й партии свеклы,$a_i, i = \overline{1, n}$). Таким образом, для переработки n партий сырья необходимо n этапов, занумеруем их от 1 до n. Пусть за время хранения на j-м этапе переработки i-ая партия свеклы теряет некоторую долю своей производственной ценности (свекла снижает свою сахаристость). Обозначим – коэффициент деградации, определяющий увядание, потерю влаги, снижение сахаристости и т.п., i-ой партии свеклы на j-м этапе переработки – элемент матрицы B порядка $n * (n-1)$. Для этих коэффициентов справедливы неравенства $0 < b_{ij} < 1$. Предполагается, что в течение одного этапа переработки данной партии свеклы ее производственная ценность не меняется.

Тогда у i-ой партии свеклы в течение всей переработки производственная ценность будет изменяться следующим образом: $p_{i1} = a_i$ – перед первым этапом, $p_{i2} = a_i b_{i1}$ – после первого этапа (перед вторым), $p_{i3} = a_i b_{i1} b_{i2}$ – после второго (перед третьим),. . . , $p_{in} = a_i b_{i1} b_{i2}, ..., b_{in-1}$– к началу последнего n-го этапа переработки (если, конечно, эта партия свеклы не будет переработана до этого момента). Таким образом, строится матрица P порядка $n * n$ с элементами p_{ij}.
Выход готового продукта (сахара) зависит от многих величин: от процентного содержания грязи на свекле, содержания нитратов, повреждений при транспортировке на переработку, температуры обработки и так далее, но основное влияние на выход конечного продукта оказывает процентное содержание сахарозы.
Пусть $\sigma = \sigma(i)$– перестановка натуральных чисел от 1 до n, соответствующая порядку переработки партий сырья. Тогда выход конечного продукта после завершения всех этапов будет пропорционален значению целевой функции [4], которая запишется в виде

$$S = \sum_{i=1}^{n} P_{\sigma(i)i} = \sum_{i=1}^{n} (a_{\sigma(i)} \prod_{j=1}^{i-1} b_{\sigma(i)j}), (i = 1 \prod_{j=1}^{i-1} b_{\sigma(i)j} = 1)$$

Задача поиска оптимального графика обработки сводится к нахождению такой перестановки, для которой значение функции будет максимальным, поэтому всего существует n! различных вариантов переработки.

Дозаривание – процесс доведения снятых недозрелых плодов (корнеплодов) в хранилищах, складах или специально оборудованных камерах до состояния потребительской спелости [15]. Как правило на практике в первые этапы хранения происходит процесс дозаривания, то есть на этих этапах коэффициент деградации $b_{ij} > 1$. Это условие верно на этапах от 1-го этапа до n-го этапа, где $[n/3] <= v <= [n/2] + 1$ После ν -го этапа начинается процесс увядания b_{ij}.

2 Определения

Венгерский алгоритм (также известный как метод Куна-Манкреса) - это алгоритм решения задачи о назначениях на матрице стоимостей или выгод. Он находит оптимальное соответствие между рядами и столбцами матрицы, минимизируя/максимизируя сумму значений этих соответствий.

Жадный алгоритм (также называемый эвристическим алгоритмом) - это алгоритм, который каждый шаг выбирает наилучшее доступное решение из множества возможностей на основе локальной оптимальности. В контексте матриц, жадный алгоритм может выбрать наилучшую пару значений (например, максимальное/минимальное значение) из матрицы на каждом шаге, не обязательно учитывая глобальную оптимальность.

Бережливый (также называемый скупым) алгоритм - это алгоритм, который на каждом шаге выбирает наименьшее количество ресурсов или действий, достаточное для достижения цели. В случае матриц, бережливый алгоритм может выбирать пути или сочетания, которые требуют наименьшего количества элементов для покрытия или распределения.

Бережливо-жадный алгоритм - это комбинация бережливого и жадного алгоритмов. Он стремится выбрать оптимальное решение, учитывая локальные оптимальности и минимальное использование ресурсов. С первого по θ период используется бережливый алгоритм. С периода до последнего используется жадный алгорит

Жадно-бережливый алгоритм - это также комбинация жадного и бережливого алгоритмов. Он стремится выбрать наилучшее решение на каждом шаге, учитывая глобальную оптимальность и минимальное использование ресурсов. С первого по $\theta + 1$ период используется жадный алгоритм. С периода до последнего используется бережливый алгоритм.

Последние алгоритмы используются в основном в моделях, учитывающих дозаривание, причем $\theta = \nu$. Существует пример, см. ниже, когда бережливо-жадный алгоритм проигрывает только оптимальному (и даже будет оптимальным).

3 Результаты

3.1 Первый пункт программы

```
Выберите:
1. Вручную ввести матрицу
2. Эксперименты
3. Проанализировать эксперименты из файла
4. Проанализировать относительную погрешность из файлов экспериментов
```

Осуществим стратегию переработки 4 партий для следующей матрицы

$$P_1 = \begin{pmatrix} 7 & 6 & 5.1 & 4 \\ 6 & 5.1 & 4 & 2 \\ 5 & 4 & 2 & 1 \\ 4 & 2 & 1 & 0.5 \end{pmatrix}$$

```
Венгерский алгоритм (максимум) : S = 16.0000000
выбор этапов: [4 3 2 1]
7.0000000       6.0000000       5.1000000       [4.0000000]
6.0000000       5.1000000       [4.0000000]     2.0000000
5.0000000       [4.0000000]     2.0000000       1.0000000
[4.0000000]     2.0000000       1.0000000       0.5000000
Венгерский алгоритм (минимум) : S = 13.0000000
выбор этапов: [1 4 3 2]
[7.0000000]     6.0000000       5.1000000       4.0000000
6.0000000       5.1000000       4.0000000       [2.0000000]
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       [2.0000000]     1.0000000       0.5000000
Жадный алгоритм : S = 14.6000000
выбор этапов: [1 2 3 4]
[7.0000000]     6.0000000       5.1000000       4.0000000
6.0000000       [5.1000000]     4.0000000       2.0000000
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       2.0000000       1.0000000       [0.5000000]
Бережливый алгоритм : S = 16.0000000
выбор этапов: [4 3 2 1]
7.0000000       6.0000000       5.1000000       [4.0000000]
6.0000000       5.1000000       [4.0000000]     2.0000000
5.0000000       [4.0000000]     2.0000000       1.0000000
[4.0000000]     2.0000000       1.0000000       0.5000000

Введите theta для бережливо-жадного и жадно-бережливого алгоритмов
(от 1 до n) (по умолчанию [n/3]):
Бережливо-жадный алгоритм, theta = 1 : S = 16.0000000
выбор этапов: [4 3 2 1]
7.0000000       6.0000000       5.1000000       [4.0000000]
6.0000000       5.1000000       [4.0000000]     2.0000000
5.0000000       [4.0000000]     2.0000000       1.0000000
[4.0000000]     2.0000000       1.0000000       0.5000000
Жадно-бережливый алгоритм, theta = 1 : S = 13.0000000
выбор этапов: [1 4 3 2]
[7.0000000]     6.0000000       5.1000000       4.0000000
6.0000000       5.1000000       4.0000000       [2.0000000]
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       [2.0000000]     1.0000000       0.5000000
```

Осуществляем те же операции для матрицы P_2, которая получается из матрицы P_1 заменой элемента $p_{13} = 5.1$ на 5.5.

$$P_2 = \begin{pmatrix} 7 & 6 & 5.5 & 4 \\ 6 & 5.1 & 4 & 2 \\ 5 & 4 & 2 & 1 \\ 4 & 2 & 1 & 0.5 \end{pmatrix}$$

```
Венгерский алгоритм (максимум) : S = 16.1000000
выбор этапов: [3 2 1 4]
7.0000000       6.0000000       [5.5000000]     4.0000000
6.0000000       [5.1000000]     4.0000000       2.0000000
[5.0000000]     4.0000000       2.0000000       1.0000000
4.0000000       2.0000000       1.0000000       [0.5000000]
Венгерский алгоритм (минимум) : S = 13.0000000
выбор этапов: [1 4 3 2]
[7.0000000]     6.0000000       5.5000000       4.0000000
6.0000000       5.1000000       4.0000000       [2.0000000]
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       [2.0000000]     1.0000000       0.5000000
Жадный алгоритм : S = 14.6000000
выбор этапов: [1 2 3 4]
[7.0000000]     6.0000000       5.5000000       4.0000000
6.0000000       [5.1000000]     4.0000000       2.0000000
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       2.0000000       1.0000000       [0.5000000]
Бережливый алгоритм : S = 16.0000000
выбор этапов: [4 3 2 1]
7.0000000       6.0000000       5.5000000       [4.0000000]
6.0000000       5.1000000       [4.0000000]     2.0000000
5.0000000       [4.0000000]     2.0000000       1.0000000
[4.0000000]     2.0000000       1.0000000       0.5000000

Введите theta для бережливо-жадного и жадно-бережливого алгоритмов
(от 1 до n) (по умолчанию [n/3]):
Бережливо-жадный алгоритм, theta = 1 : S = 16.0000000
выбор этапов: [4 3 2 1]
7.0000000       6.0000000       5.5000000       [4.0000000]
6.0000000       5.1000000       [4.0000000]     2.0000000
5.0000000       [4.0000000]     2.0000000       1.0000000
[4.0000000]     2.0000000       1.0000000       0.5000000
Жадно-бережливый алгоритм, theta = 1 : S = 13.0000000
выбор этапов: [1 4 3 2]
[7.0000000]     6.0000000       5.5000000       4.0000000
6.0000000       5.1000000       4.0000000       [2.0000000]
5.0000000       4.0000000       [2.0000000]     1.0000000
4.0000000       [2.0000000]     1.0000000       0.5000000
```

3.2 Второй пункт программы. Эксперимент

3.2.1 Эксперимент 1

Количество экспериментов = 100, n=20, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.2$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.9, \max b_{ij} = 0.99$

```
Средние S:
Венгерский алгоритм (максимум): 3.3508056373724666
Венгерский алгоритм (минимум): 2.8494381183762347
Жадный алгоритм: 3.192385997507915
Бережливый алгоритм: 2.9950387759514623
Бережливо-жадный алгоритм: 3.157360677062385
Жадно-бережливый алгоритм: 3.0589235471877707
Усреднённая погрешность S:
Венгерский алгоритм (максимум): 0.0
Венгерский алгоритм (минимум): 0.1496259626056315
Жадный алгоритм: 0.047278074889708062
Бережливый алгоритм: 0.10617352956943776
Бережливо-жадный алгоритм: 0.05773088064331044
Жадно-бережливый алгоритм: 0.08710803364100078
```

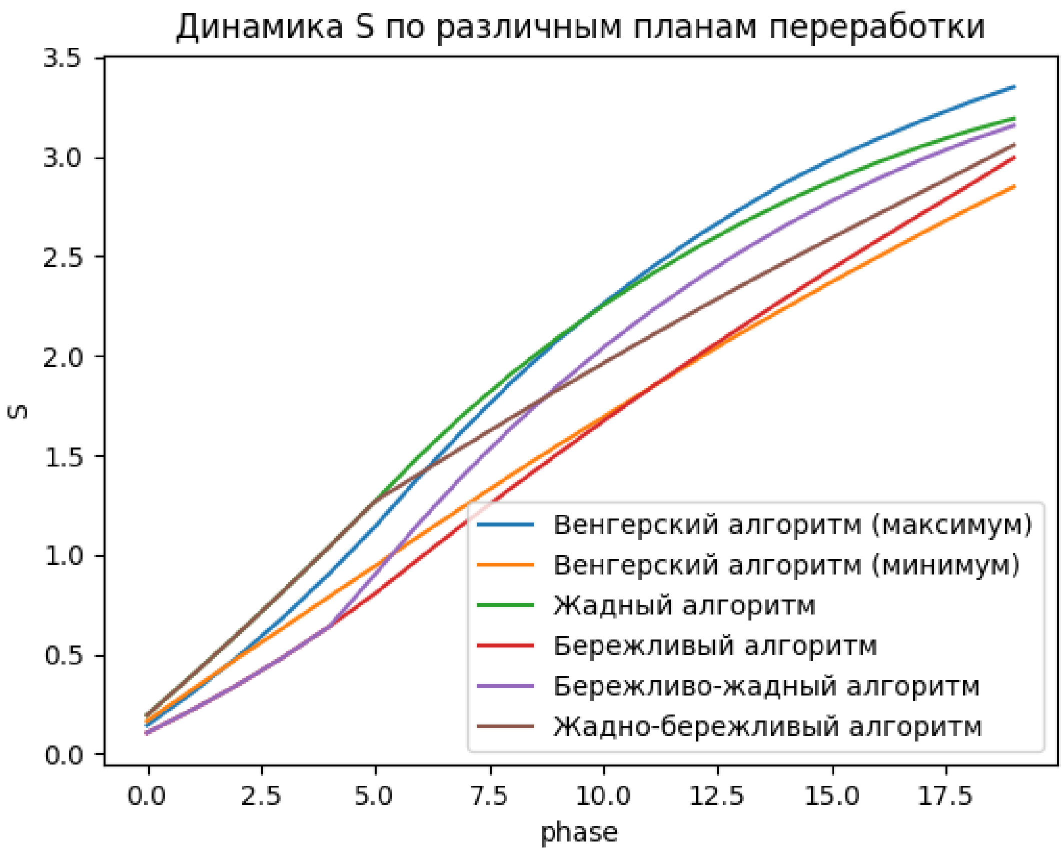

Рис. 1: Эксперимент 1, Количество экспериментов = 100, n=20, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.2$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.9, \max b_{ij} = 0.99$

3.2.2 Эксперимент 2

количество экспериментов = 100, n=20, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.2$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.7, \max b_{ij} =$

0.8

```
Средние S:
Венгерский алгоритм (максимум): 2.1853529174365276
Венгерский алгоритм (минимум): 1.5250879608195687
Жадный алгоритм: 2.109573254677118
Бережливый алгоритм: 1.5783552554973415
Бережливо-жадный алгоритм: 1.8733622066885074
Жадно-бережливый алгоритм: 1.8892681279549637
Усреднённая погрешность S:
Венгерский алгоритм (максимум): 0.0
Венгерский алгоритм (минимум): 0.30213195834358225
Жадный алгоритм: 0.03467616701850646
Бережливый алгоритм: 0.27775727073465517
Бережливо-жадный алгоритм: 0.1427644515715077
Жадно-бережливый алгоритм: 0.13548602933611226
```

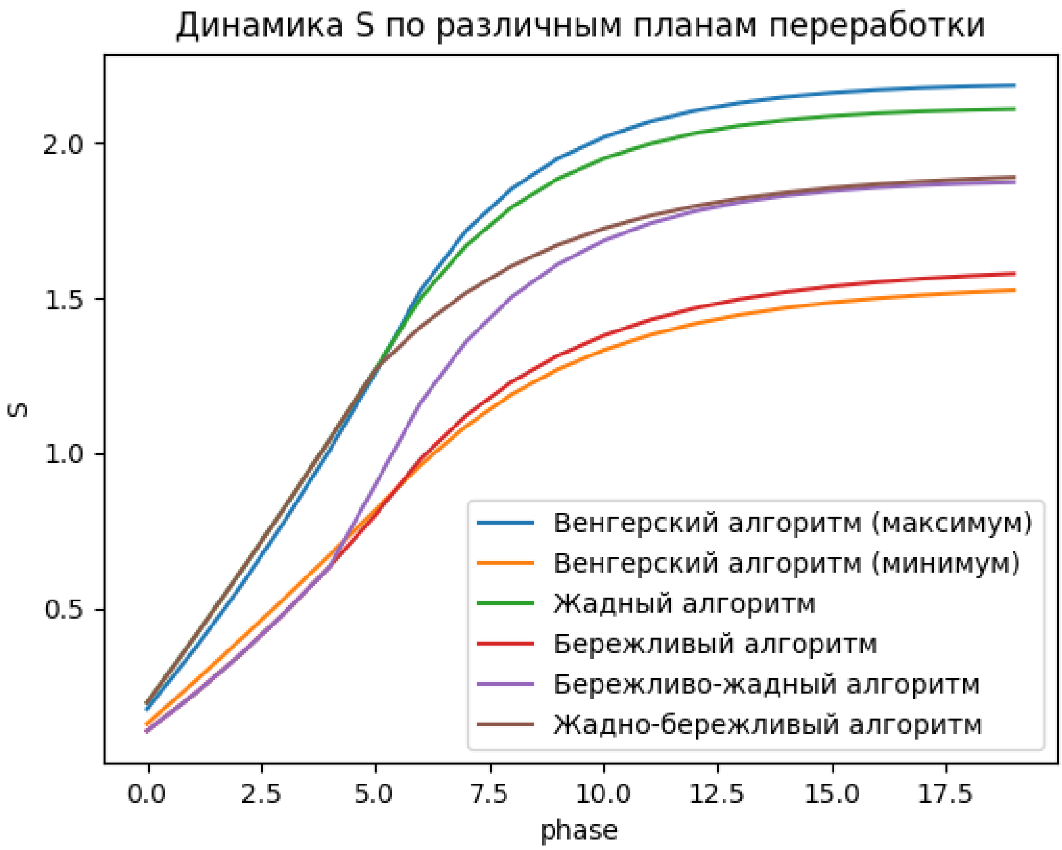

Рис. 2: Эксперимент 2, количество экспериментов = 100, n=20, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.2$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.7, \max b_{ij} = 0.8$

3.2.3 Эксперимент 3

количество экспериментов = 20, n=100, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.3$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.9, \max b_{ij} = 0.99$

```
Средние S:
Венгерский алгоритм (максимум): 4.526449181024683
Венгерский алгоритм (минимум): 3.731597417935289
Жадный алгоритм: 4.307169809653175
Бережливый алгоритм: 3.9374056252699323
Бережливо-жадный алгоритм: 4.235682387824037
Жадно-бережливый алгоритм: 4.0484432178992344
Усреднённая погрешность S:
Венгерский алгоритм (максимум): 0.0
Венгерский алгоритм (минимум): 0.17560160984939147
Жадный алгоритм: 0.048444014856225175
Бережливый алгоритм: 0.13013369469032782
Бережливо-жадный алгоритм: 0.06423728215475587
Жадно-бережливый алгоритм: 0.10560285645739623
```

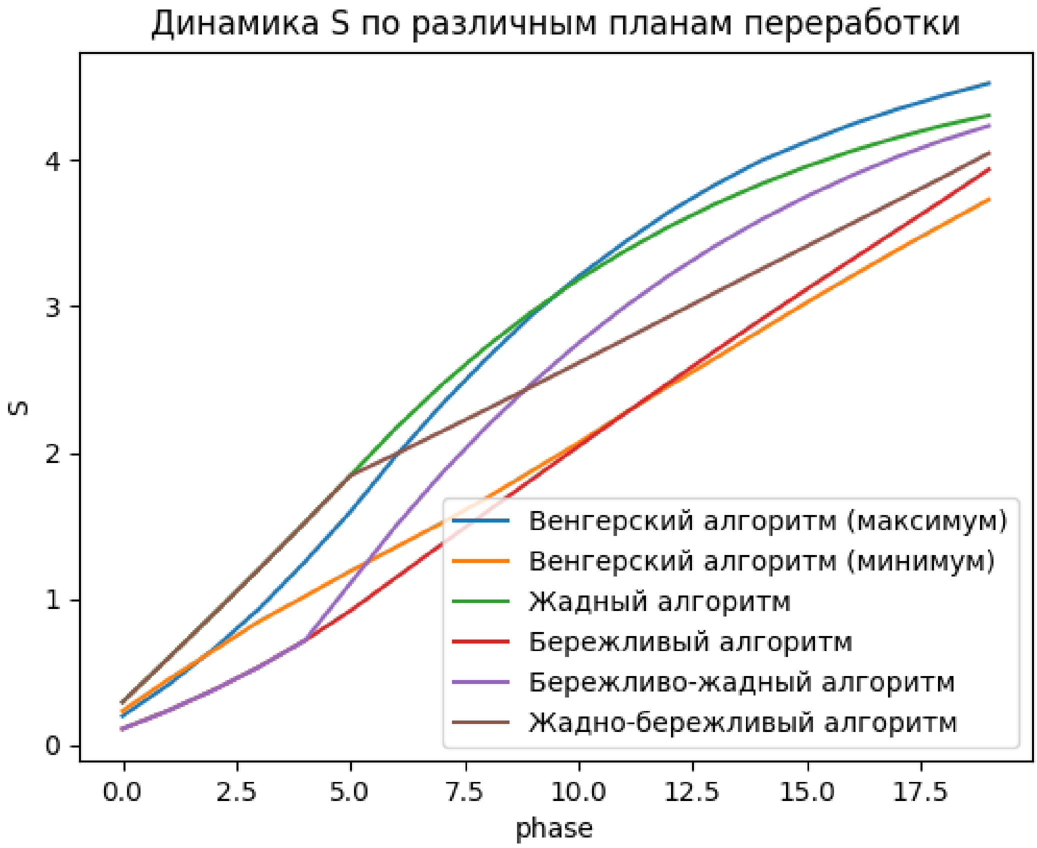

Рис. 3: Эксперимент 3, количество экспериментов = 20, n=100, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.3$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.9, \max b_{ij} = 0.99$

3.2.4 Эксперимент 4

количество экспериментов = 20, n=100, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.3$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.7, \max b_{ij} = 0.8$

```
Средние S:
Венгерский алгоритм (максимум): 3.054439864256901
Венгерский алгоритм (минимум): 1.8711931671855668
Жадный алгоритм: 2.9446713700972116
Бережливый алгоритм: 1.9476659666604146
Бережливо-жадный алгоритм: 2.486774222478785
Жадно-бережливый алгоритм: 2.5395931375077527
Усреднённая погрешность S:
Венгерский алгоритм (максимум): 0.0
Венгерский алгоритм (минимум): 0.3873858218384667
Жадный алгоритм: 0.03593735645091657
Бережливый алгоритм: 0.36234921844360746
Бережливо-жадный алгоритм: 0.18584934292566938
```

```
Жадно-бережливый алгоритм: 0.16855683844815283
```

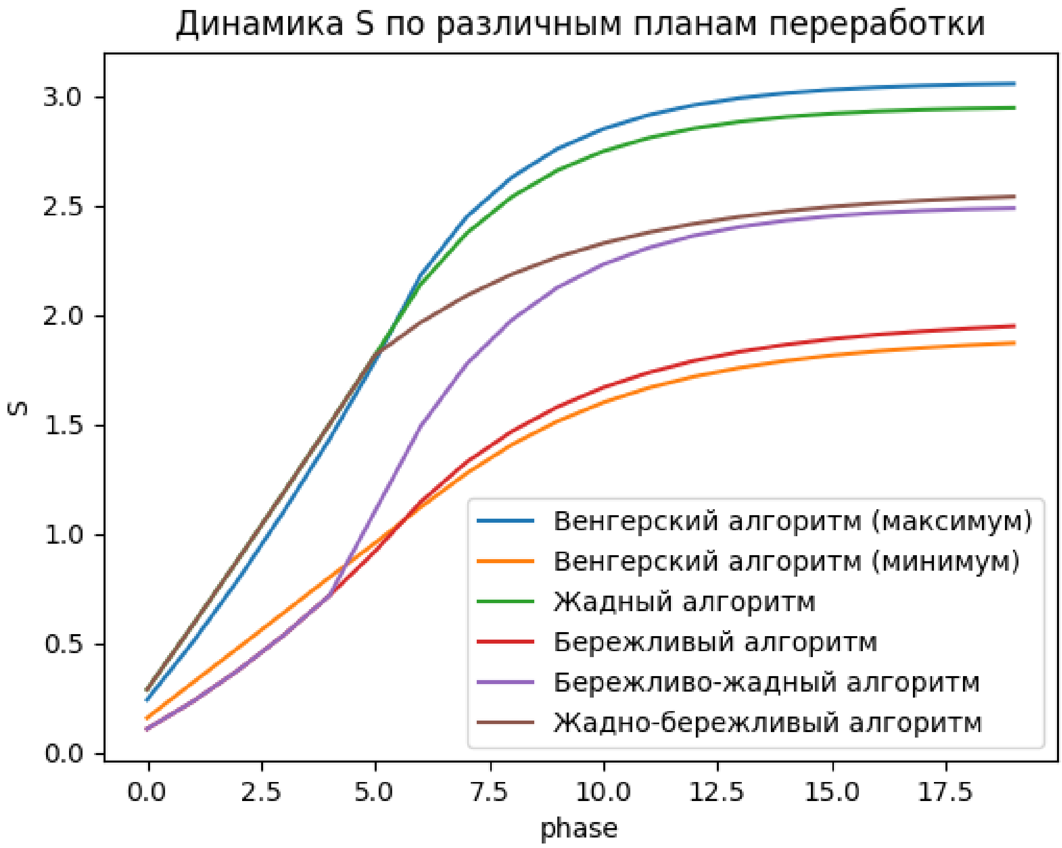

Рис. 4: Эксперимент 4, количество экспериментов = 20, n=100, $\nu = [\frac{n}{3}]$, $\min a_i = 0.1, \max a_i = 0.3$, во время дозаривания $\min b_{ij} = 1.01, \max b_{ij} = 1.1$, после дозаривания $\min b_{ij} = 0.7, \max b_{ij} = 0.8$

4 Выводы

В рамках данной лабораторной работы мы изучили различные алгоритмы для определения оптимального плана переработки различных партий сахарной свеклы. В ходе исследования были рассмотрены следующие алгоритмы: Венгерский, Жадный алгоритм, Бережливый алгоритм, Бережливо-жадный алгоритм и Жадно-бережливый алгоритм.

Безусловно лучшим алгоритмам оказался Венгерский. Но он не применим на реальном производстве, поскольку требует знать все элементы матрицы, а значит мы должны знать с какой скоростью каждая партия будет деградировать еще до старта работы.

Лучшим алгоритмом готовым к реальному применению является Жадный. Он показывает хорошие результаты с разными данными.

Так же, хорошо себя показал Бережливо-жадный. Он является эффективным при маленьком деградирование свеклы.

В ходе экспериментов по выявлени. оптимального плана переработки сахрной свеклы, выясняли, с учетом дозаривания наиболее подходящий алгоритм - жадный алгоритм.

5 Список литературы

- Л 12 Лабораторная работа «Решение прикладных задач дискретной оптимизации»: Учебно-методическое пособие / Авторы: Д.В. Баландин, О.А. Кузенков, Д.С. Малышев, О.В. Приставченко, А.И. Эгамов. – Нижний Новгород: Нижегородский госуниверситет, 2023. – 23 с.

- Баландин Д.В. и др. Стратегия переработки партий сахарной свеклы при близких параметрах ее увядания / Сборник трудов Второго всероссийского научно-практического семинара «Математическое и компьютерное моделирование и бизнес-анализ в условиях цифровизации экономики». Нижний Новгород, 22 апреля 2022. С. 10–18.