Министерство образования Республики Беларусь

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №2**

по дисциплине

«Модели решения задач в интеллектуальных системах»

Вариант 8

Выполнил:

студент гр. 221701 Робилко Т. М.

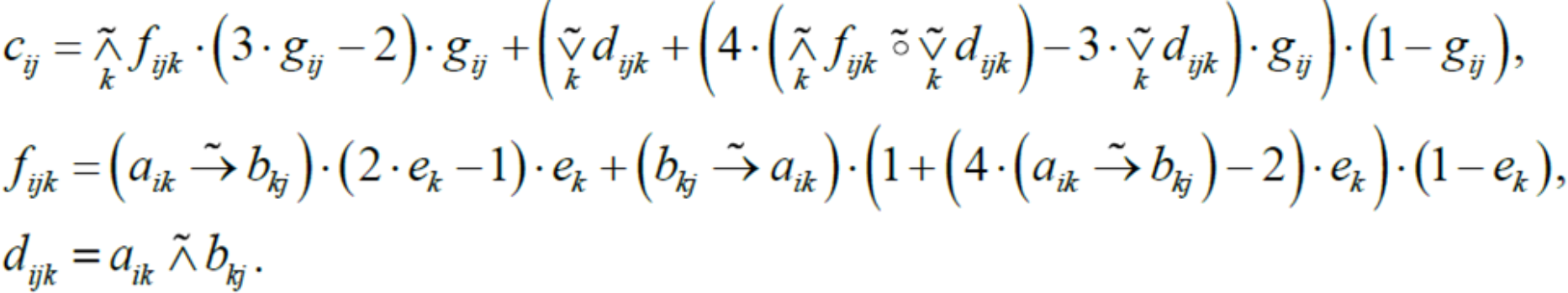
Проверил: Ивашенко В. П.

Минск, 2025

**Тема:** Реализация модели решения задач в интеллектуальных системах.

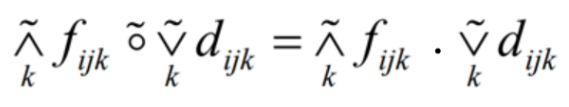
**Цель:** Реализовать и исследовать модель решения на ОКМД архитектуре задачи вычисления матрицы значений.

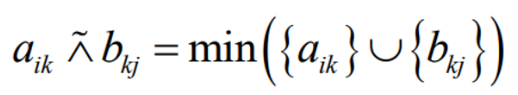
**Дано:** Сгенерированные матрицы A, B, E, G заданных размерностей pxm, mxq, 1xm, pxq соответственно со значениями в рекомендуемом диапазоне  [-1;1].

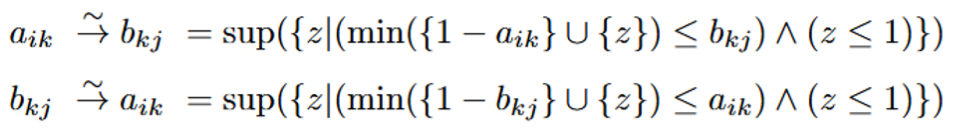




Согласно варианту лабораторной работы (вариант 8):

****

****



**Задание:** Получить матрицу значений С, соответствующую размерности *p*x*q.*

**Описание программы:**

Реализация варианта лабораторной работы производилась на ЯП Python с использованием библиотеки для отрисовки графиков matplotlib и для генерации случайных чисел random.

Программа позволяет пользователю задавать параметры размерности матриц p, q и m, а также количество процессорных элементов n.

Функционально программа разделена на блоки - блок обработки действий пользователя, блок вычислений и блок отрисовки результатов на график. Блоки и их функции приведены ниже:

1. Блок обработки действий пользователя:

* **ValidateNumericInput**(input\_str) – проверка корректности введенных данных;
* **DisplayMatrix**(matrix, title='') – вывод матриц;
* **GenerateRandomMatrix**(m, p, q) – заполнения матриц рандомными значениями в пределе [-1;1];
* **\_\_main**\_\_ – содержит цикл ввода, прекращающийся когда введенные пользователем значения окажутся корректными, инициирует заполнение матриц, нахождение матрицы С и их вывод, отвечает за подсчет большей части показателей.

1. Блок вычислений:

* **SumOperation**(x, y) – вычисление суммы двух элементов и прибавление 1 к счетчику суммы;
* **MultiplyOperation**(x, y) – вычисление произведения двух элементов и прибавление 1 к счетчику произведения;
* **DifferenceOperation**(x, y) – вычисление разности двух элементов и прибавление 1 к счетчику разности;
* **CompareMinMaxValues**(x, y) – сравнение двух элементов, определение минимального и прибавление 1 к счетчику сравнения;
* **ProcessAverageTime**();
* **Implication**(a, b) – находит импликацию;
* **ComposeOperation**(a, b) – находит композицию;
* **TNorm**(a, b) – находит t-норму
* **ProcessKfFactors**(i, j)– находит конъюнкцию из по формуле и считает изменение при нахождении и конъюнкции ;
* **ReduceKfFactors**(factor\_list) – выполняет операцию редукции для ;
* **ProcessKd**(i, j) – находит дизъюнкцию из по формуле и считает изменение при нахождении и дизъюнкции ;
* **ProcessMatrixElement**(i, j) – находит ;
* **CalculateOutputMatrix**(rows, cols, dim\_m) – находит каждый элемент матрицы ;
* **ProcessT1**() – вычисляет T1

В блоке отрисовки графиков описаны и реализованы функции для следующих графиков:

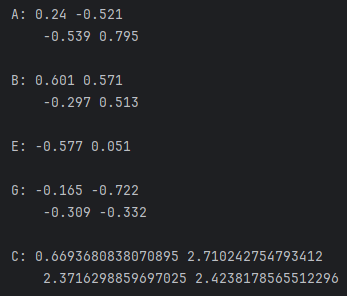
1. График зависимости коэффициента ускорения от r;
2. График зависимости Ку от n;
3. График зависимости эффективности e от r
4. График зависимости е от n
5. График зависимости коэффициента расхождения D от r
6. График зависимости коэффициента расхождения D от n

**Тестирование программы:**

Введем необходимые параметры. Время выполнения каждой операции по умолчанию равно 1.

**m=2 p=2 q=2 n=6**

Сгенерированные матрицы A, B, E, G и итоговая матрица С:

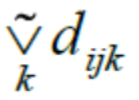


Проверка выполнения:

= min(, )

= min(, )= 0.24

= min(, )= -0.521

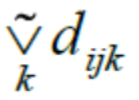
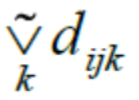
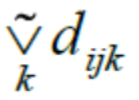
 = 1 - (1 - )(1 - ) = -0.15596

= (sup{z ∈ [0,1] ∣ min(1 − , z) ≤ }) \* (2 \* - 1) \* + (sup{z ∈ [0,1] ∣ min(1 − , z) ≤ }) \* (1 + (4 \* (sup{z ∈ [0,1] ∣ min(1 − , z) ≤ }) - 2) \* ) \* (1 - )

= (sup{z ∈ [0,1] ∣ min(1 − 0.24, z) ≤ 0.601}) \* (2 \* (-0.577) - 1) \* (-0.577) + (sup{z ∈ [0,1] ∣ min(1 − 0.601, z) ≤ 0.24}) \* (1 + (4 \* (sup{z ∈ [0,1] ∣ min(1 − 0.24, z) ≤ 0.601}) - 2) \* (-0.577)) \* (1 + 0.577) = 1.03721094

= (sup{z ∈ [0,1] ∣ min(1 + 0.521, z) ≤ -0.297}) \* (2 \* (0.051) - 1) \* (0.051) + (sup{z ∈ [0,1] ∣ min(1 + 0.297, z) ≤ -0.521}) \* (1 + (4 \* (sup{z ∈ [0,1] ∣ min(1 + 0.521, z) ≤ -0.297}) - 2) \* (0.051)) \* (1 - 0.051) = 0

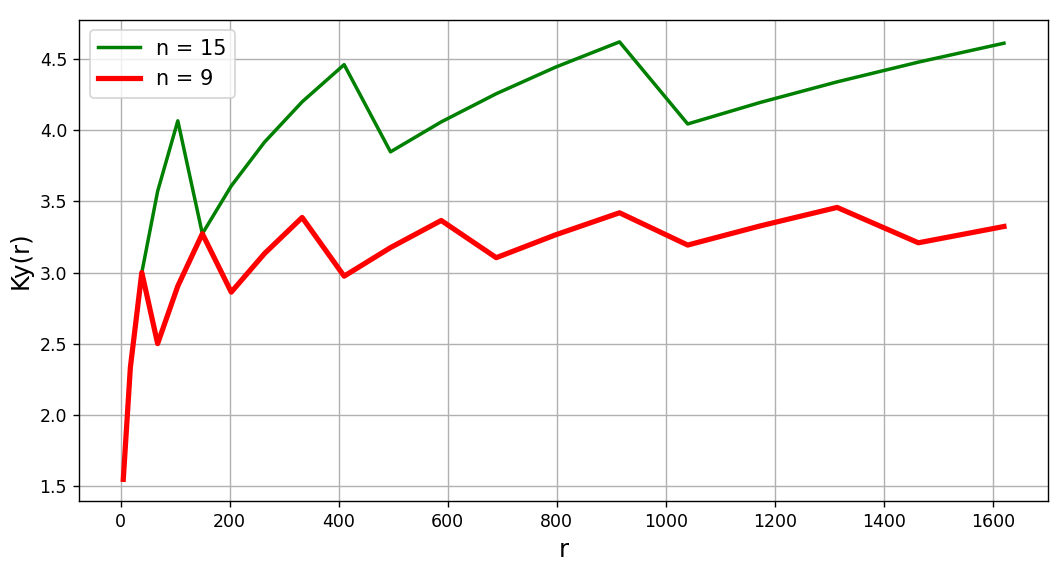
 = = 0

=  \* \* (3 - 2) + ( + \* (4 \* max((+ - 1), 0 )- 3\* ))\*(1 - )

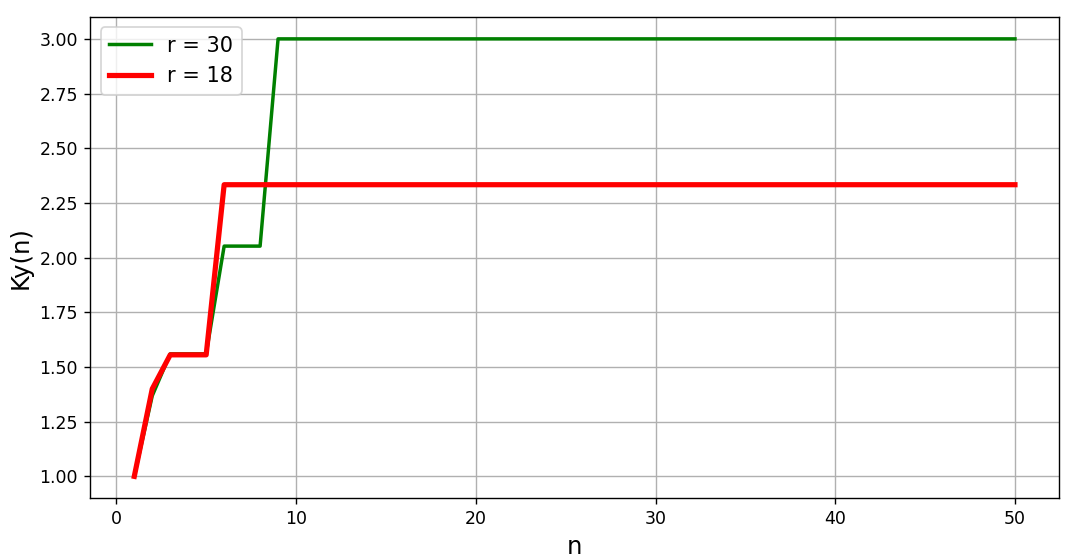
= 0 \* (-0.165) \* (3 \* (-0.165) - 2) + (-0.15596 - 0.165 \* (4 \* max(-0.165 + 0 - 1), 0) - 3 \* (-0.165))) \* (1 + 0.165) = 0.6693681

**Программа вычисляет значения правильно.**

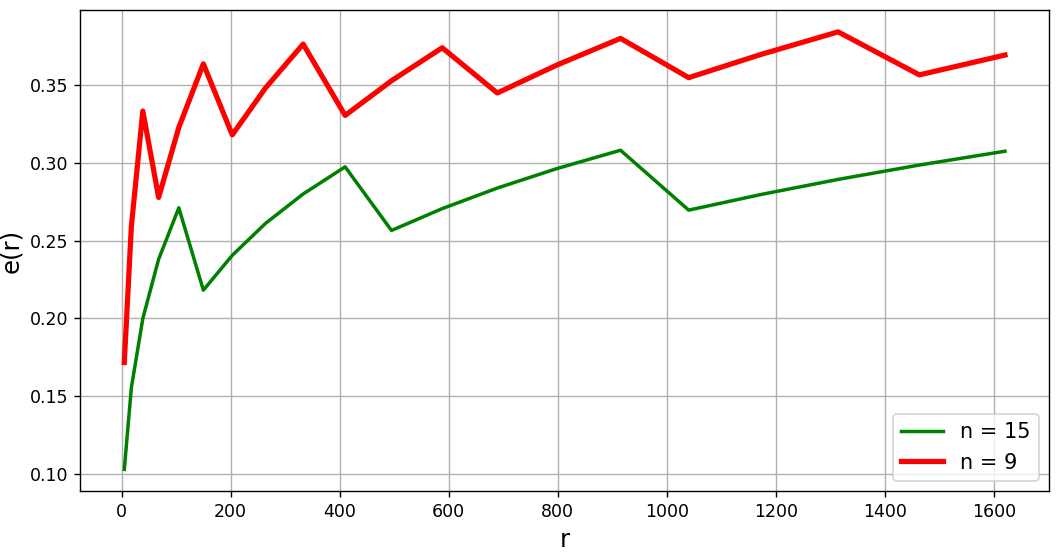
**Графики:**



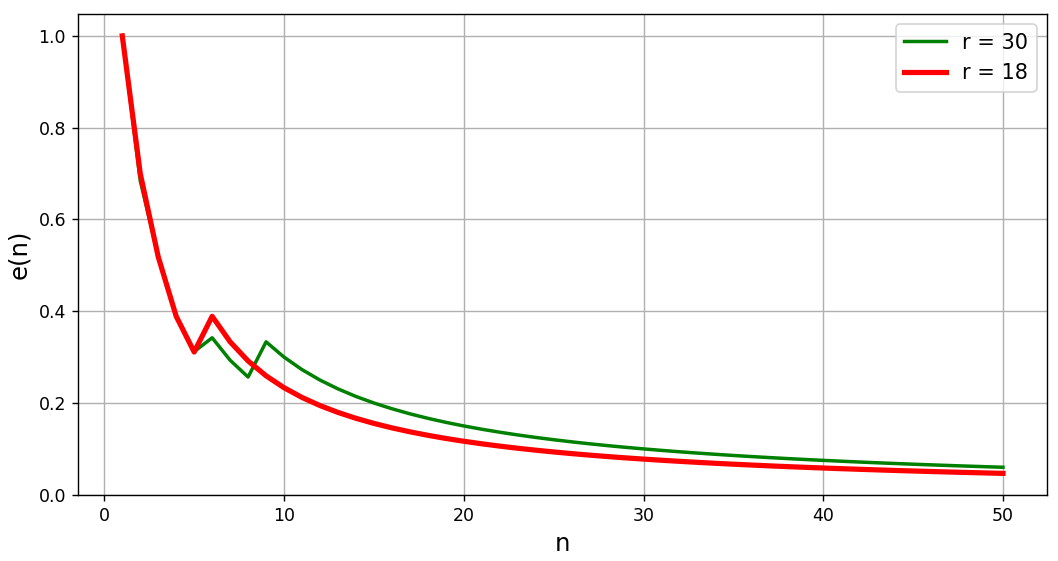
*Рисунок 1. График зависимости коэффициента ускорения от r*



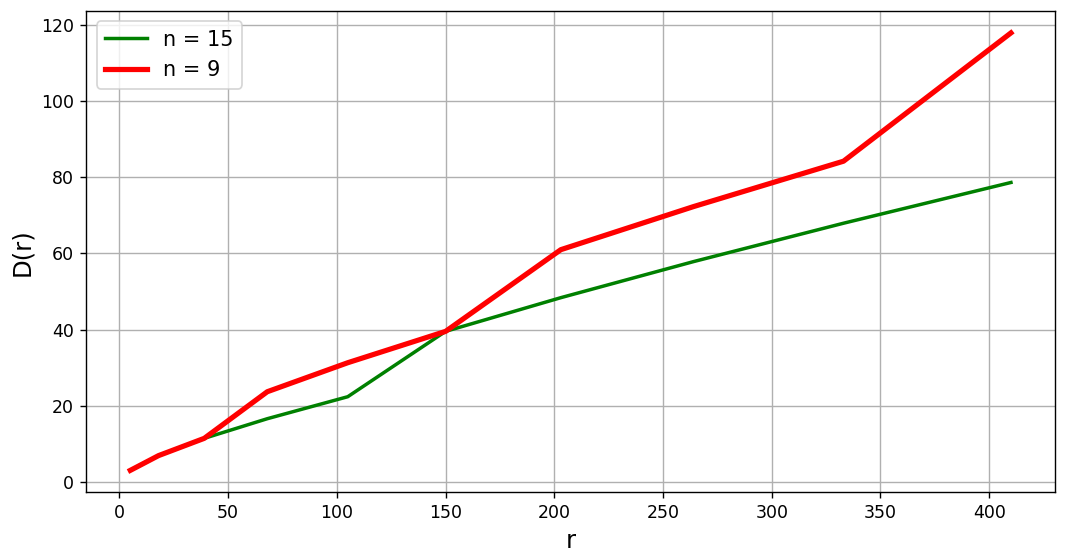
*Рисунок 2. График зависимости Ку от n*



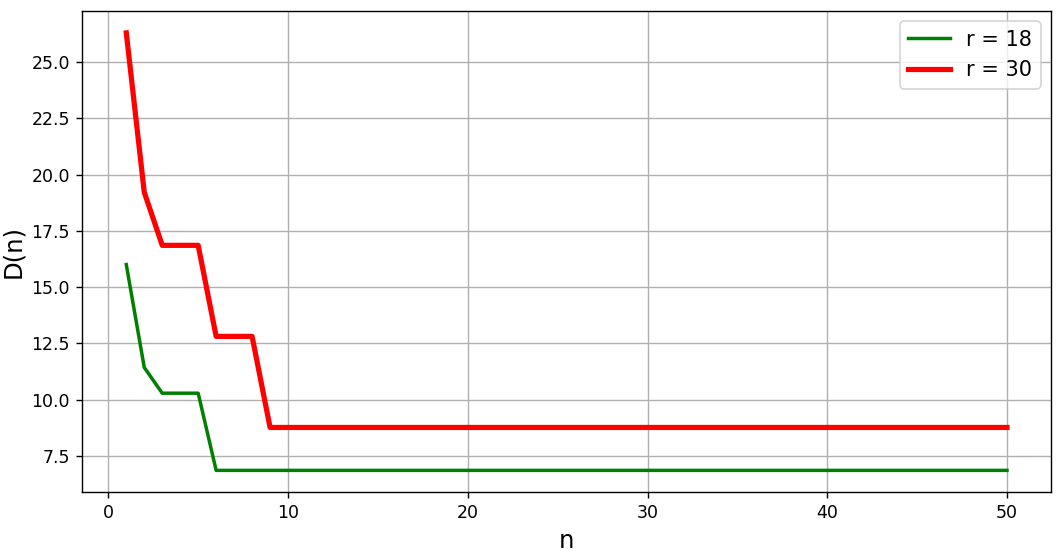
*Рисунок 3. График зависимости эффективности e от r*



*Рисунок 4. График зависимости е от n*



*Рисунок 5. График зависимости коэффициента расхождения D от r*

****

*Рисунок 6. График зависимости коэффициента расхождения D от n*

**Вопросы:**

1. **Объяснить асимптоты на графиках:**

**1.** График (Рисунок 1) не имеет чёткой асимптоты. Он имеет асимптотическое приближение. Функция возрастает на тех промежутках, где:

.

Функция убывает на тех промежутках, где:

.

**2.** На графике (Рисунок 2) нет асимптот (видно благодаря тому, что график перестаёт изменяться). достигает своего максимального значения, когда для заданного ранга r: n 3 \* m, где m – аргумент функции r(p, q, m).

**3.** Асимптотой графика (Рисунок 3) является функция y = 0.

**4.** График (Рисунок 4) не имеет чёткой асимптоты. Он имеет асимптотическое приближение. Функция возрастает на тех промежутках, где:

.

Функция убывает на тех промежутках, где:

.

**5.** График (Рисунок 5) не имеет чёткой асимптоты. Данный график имеет асимптотическое приближение.

**6.** График (Рисунок 6) не имеет асимптот. По рисунку видно, что с некоторого момента он перестаёт изменяться. достигает своего минимального значения, когда для заданного ранга r: n 3 \* m, где m – аргумент функции r(p, q, m).

1. **Спрогнозировать, как изменится вид графиков при изменении параметров модели:**

**1. График зависимости коэффициента ускорения от r (Рисунок 1).**

Если будет увеличиваться ранг задачи r, то (r) будет увеличиваться скачкообразно.

**2. График зависимости Ку от n (Рисунок 2).**

Если будет увеличиваться количество процессорных элементов n, то (n) будет увеличиваться, пока n 3 \* m, где m – аргумент функции r(p, q, m).

**3. График зависимости эффективности e от r (Рисунок 3).**

Если будет увеличиваться ранг задачи, то будет увеличиваться скачкообразно.

**4. График зависимости е от n (Рисунок 4).**

Если будет увеличиваться количество процессорных элементов n, то будет уменьшаться.

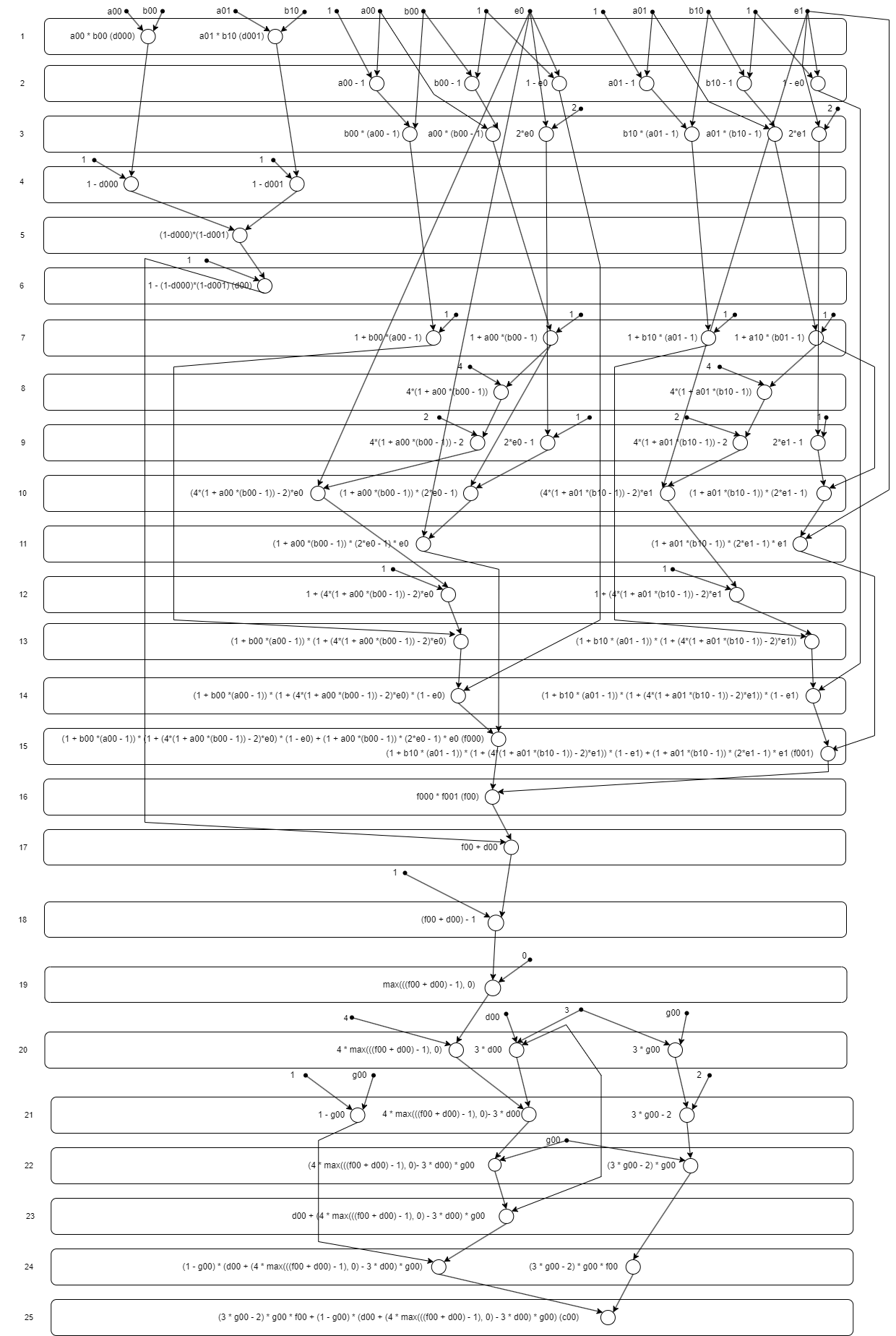
**5. График зависимости коэффициента расхождения D от r (Рисунок 5).**

Если будет увеличиваться ранг задачи, то будет увеличиваться.

**6. График зависимости коэффициента расхождения D от n (Рисунок 6).**

Если будет увеличиваться количество процессорных элементов n, будет уменьшаться, то пока n 3 \* m, где m – аргумент функции r(p, q, m).

**Информационный граф**

****

**Личный вклад**: Абушкевич А. А. - проектирование и программная реализация алгоритма, наполнение отчета, ответы на вопросы.

**Вывод:** в результате выполнения лабораторной работы была реализована и исследована модель решения на ОКМД архитектуре задачи вычисления матрицы значений. Были созданы и проанализированы графики трёх характеристик конвейерной архитектуры: коэффициенты ускорения, эффективности и расхождения.

**Использованные источники:**

1. Модели решения задач в интеллектуальных системах. В 2 ч. Ч.1: Формальные модели обработки информации и параллельные модели решения задач: учеб.-метод. пособие/ В. П. Ивашенко. – Минск : БГУИР, 2020. – 79 с.
2. https://www.cyberforum.ru/blogs/tags/simd.html

1. <https://studfile.net/preview/9991876/page:9/>

1. <https://github.com/SonyaVitkovskaya/BSUIR/tree/main/MRZvIS/sem6/lab2>