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 4

Выполнила:

студент гр. 221703 Демидовец Д.В.

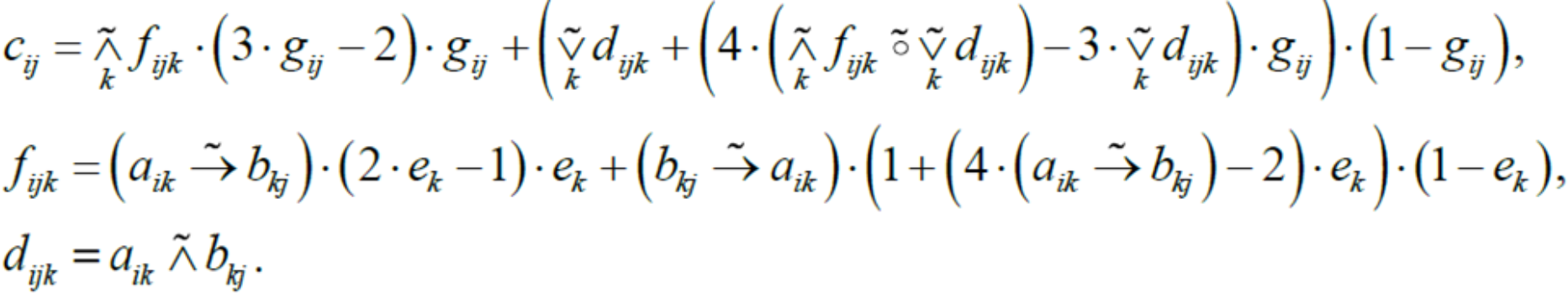
Проверил: Ивашенко В. П.

Минск, 2025

**Тема:** Программирование параллельного решения задач на параллельной архитектуре.

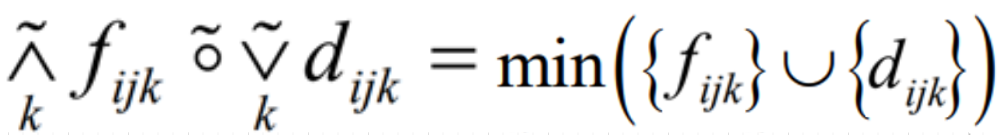
**Цель:** Реализовать и исследовать модель решения на ОКМД архитектуре задачи вычисления матрицы значений.

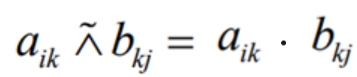
**Дано:** Сгенерированные матрицы A, B, E, G заданных размерностей pxm, mxq, 1xm, pxq соответственно со значениями в рекомендуемом диапазоне  [-1;1].

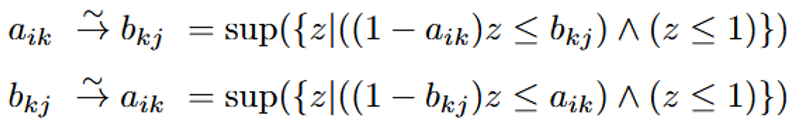




Согласно варианту:

****

****

****

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

**Дополнительные теоретические сведения:**

*Коэффициент ускорения:* , где:

– время решения задачи на первой(последовательной) архитектуре;

– время решения задачи на другой (параллельной) архитектуре.

*Эффективность вычислительных систем:*  , где:

– количество процессорных элементов в системе (совпадает с количеством этапов конвейера).

*Коэффициент расхождения:* , где:

— время (суммарная длина программы) решения задачи на (параллельной) архитектуре.

— среднее время (средняя длина программы) решения задачи на той же архитектуре.

**Описание программы:**

Программа позволяет пользователю задавать следующие параметры:

1. Параметры матриц – p, q, m;
2. Количество процессорных элементов n.

Программа включает в себя следующие методы:

Арифметические операции:

* sum(a, b) выполняет сложение двух чисел; при каждом вызове увеличивает счётчик sum\_call на 1.
* mult(a, b) выполняет умножение двух чисел; при каждом вызове увеличивает счётчик mult\_call на 1.
* div(a, b) выполняет деление двух чисел; при каждом вызове увеличивает счётчик div\_call на 1.
* diff(a, b) выполняет вычитание двух чисел; при каждом вызове увеличивает счётчик diff\_call на 1.
* compare(a, b, max\_or\_min) возвращает максимум или минимум из двух чисел в зависимости от флага max\_or\_min; при каждом вызове увеличивает счётчик compare\_call на 1.

Утилитарные функции:

* check\_input(str) проверяет, что все символы строки — цифры (из допустимого алфавита ALFABET).
* print\_matrix(matr, name='') выводит на экран матрицу matr с заголовком name в удобочитаемом формате.
* fill\_matrix(m, p, q) генерирует случайные матрицы: A размерности p×m, B размерности m×q, E размерности 1×m, G размерности p×q.

Значения элементов находятся в диапазоне [-1; 1] с округлением до трёх знаков.

Методы расчёта элементов искомой матрицы C:

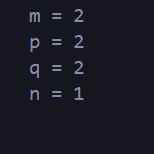
* find\_impl(a, b) вычисляет импликацию по формуле () , ограничивая результат единицей.
* find\_compose(a, b) вычисляет min((),1) согласно варианту.
* find\_tnorm(a, b) вычисляет t‑норму как произведение двух аргументов (mult(a, b)) согласно варианту.
* find\_kf(i, j) для фиксированных i, j последовательно по k считает конъюнкцию по заданной формуле, при этом обновляя глобальное время Tn.
* find\_kd(i, j) аналогично find\_kf, но вычисляет дизъюнкцию и накапливает время Tn.​
* find\_cij(i, j) с помощью find\_kf и find\_kd получает f и d, затем получая элемент по заданной формуле.

Все эти функции объединены в find\_C(x, y, m), которая по заданным размерам x=p, y=q и глубине m последовательно заполняет матрицу C[p][q] вызовами find\_cij(i, j).

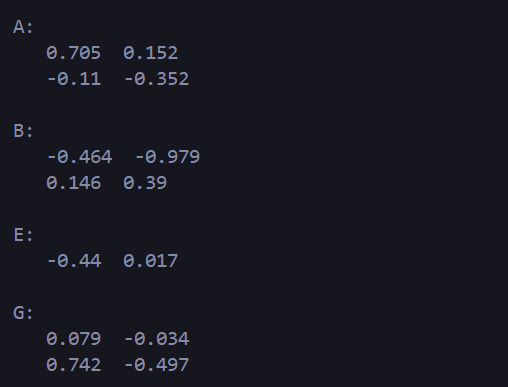
Метод main() содержит цикл для ввода пользователем значений и инициирует заполнение матриц, нахождение матрицы С и их вывод, а также отвечает за подсчет большей части показателей.

**Демонстрация работы:**

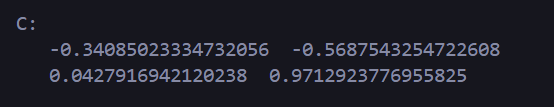
Входные данные:

****

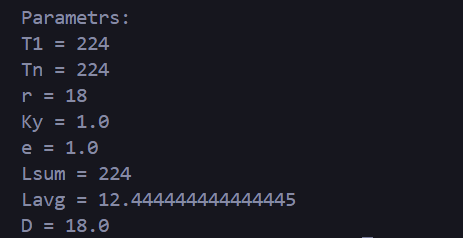
Сгенерированные матрицы A, B, E, G:



Полученная матрица С:



Параметры:



Проверка выполнения:

= \*

= \* = -0.32712

= \* = 0.022191996

 = 1 - (1 - )(1 - ) = -0.29766855296

=

= 1.922221906085

0.168232522988

 = \* = 0.3233802410042

=  \* \* (3 - 2) + ( + \* (4 \* \*  - 3\* ))\*(1 - )

=  \* \* (3 - 2) + ( + \* (4 \* \*  - 3\*))\*(1 - ) = 0.3233802410042 \* 0.079\*(3\*0.079 - 2) + (-0.29766855296 + 0.079\*(4\* (-0.09626012839) - 3\*(-0.29766855296)))\*(1-0.079) = -0.340850233

На наглядном примере можно увидеть, что значения в программе вычисляются корректно.

**Графики:**

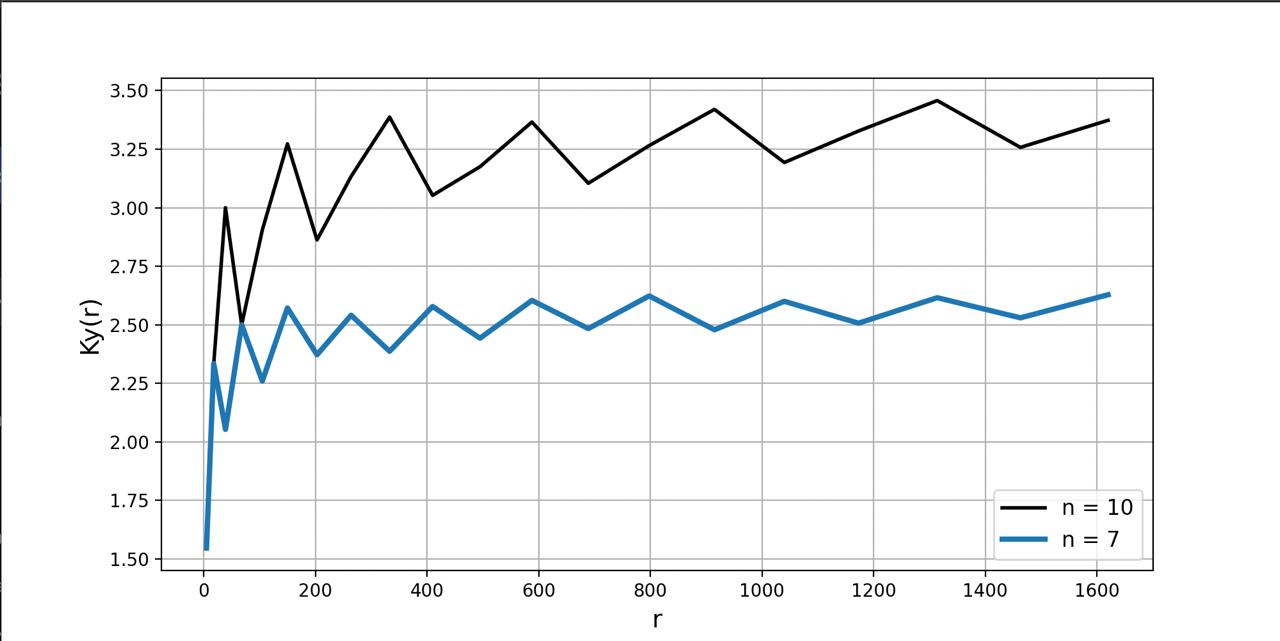


График зависимости коэффициента ускорения от r

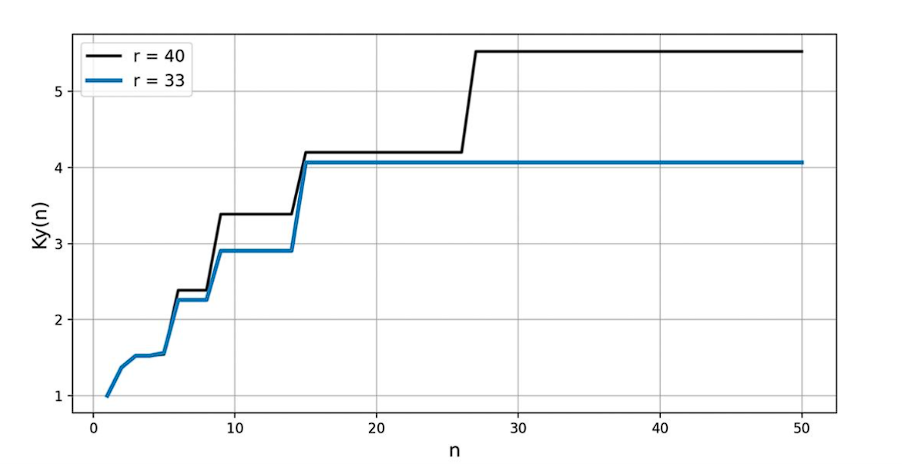


График зависимости Ку от n

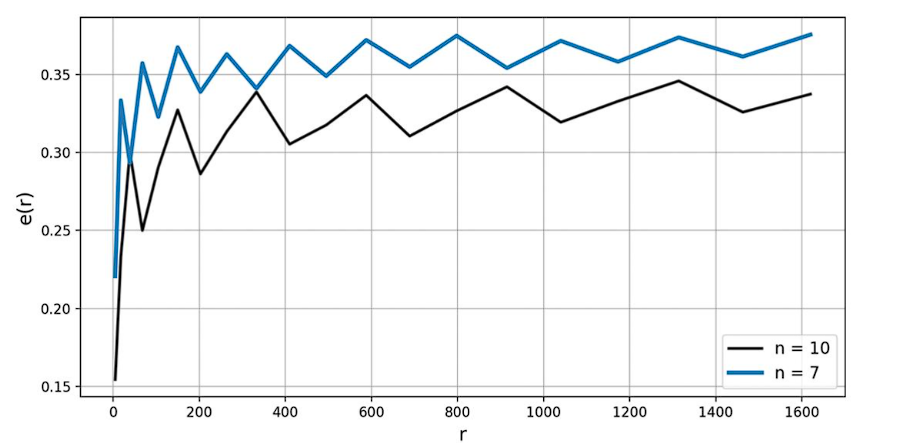


График зависимости эффективности e от r

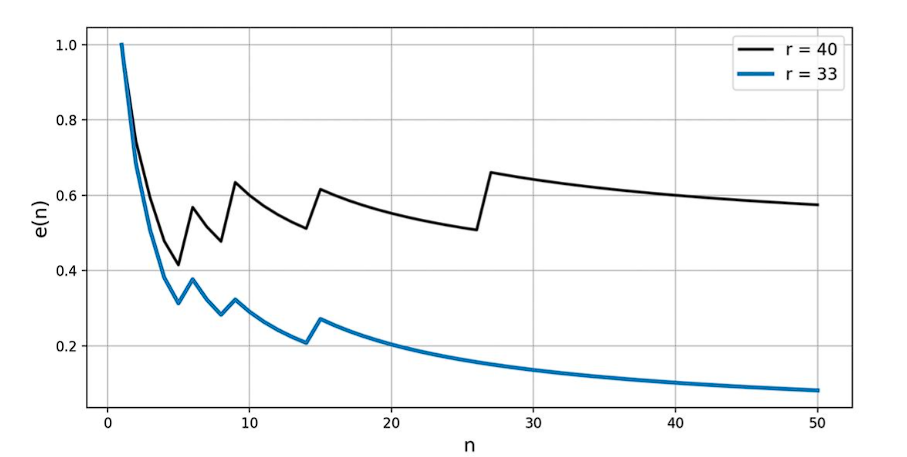
****

График зависимости е от n

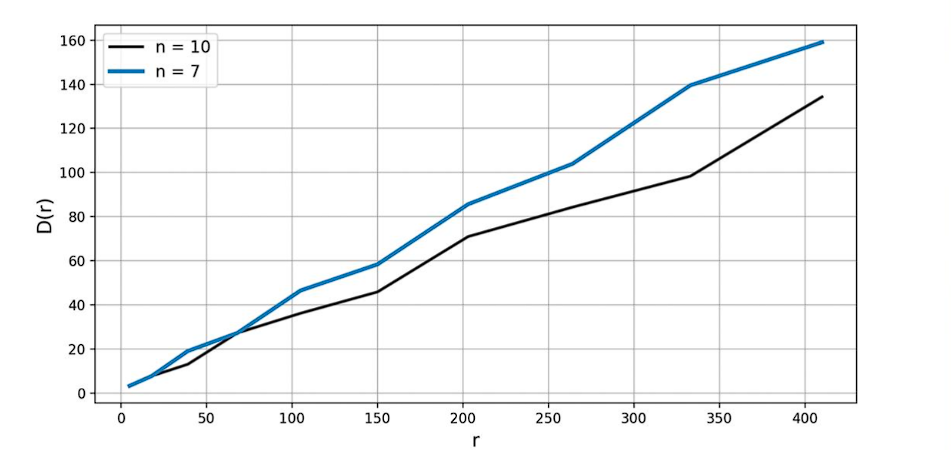


График зависимости коэффициента расхождения D от r

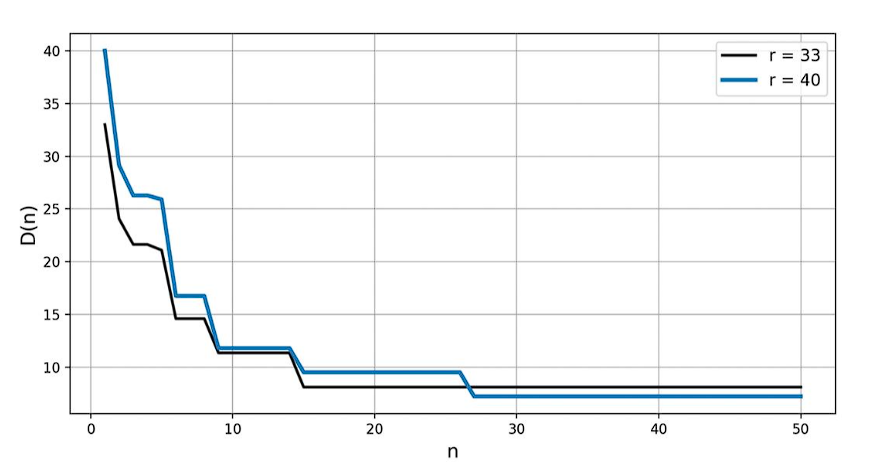
****

График зависимости коэффициента расхождения D от n

**Контрольные вопросы:**

1. *Объяснить асимптоты на графиках:*

График не имеет чёткой асимптоты, но имеет асимптотическое приближение.

Функция возрастает на тех промежутках, где:

.

Функция убывает на тех промежутках, где:

.

На графике  нет асимптот, т.к. в определённый момент график перестает изменяться вовсе.

достигает своего максимального значения, когда для заданного ранга r: n 3 \* m, где m – аргумент функции r(p, q, m).

Асимптотой графика является функция y = 0.

График не имеет чёткой асимптоты, но имеет асимптотическое приближение.

Функция возрастает на тех промежутках, где:

.

Функция убывает на тех промежутках, где:

.

График не имеет чёткой асимптоты, но имеет асимптотическое приближение.

График не имеет асимптот, т.к. в определённый момент график перестает изменяться вовсе.

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*:

Увеличивая r, (r) будет увеличиваться скачкообразно.

Увеличивая n, (n) будет увеличиваться, пока n 3 \* m, где m – аргумент функции r(p, q, m).

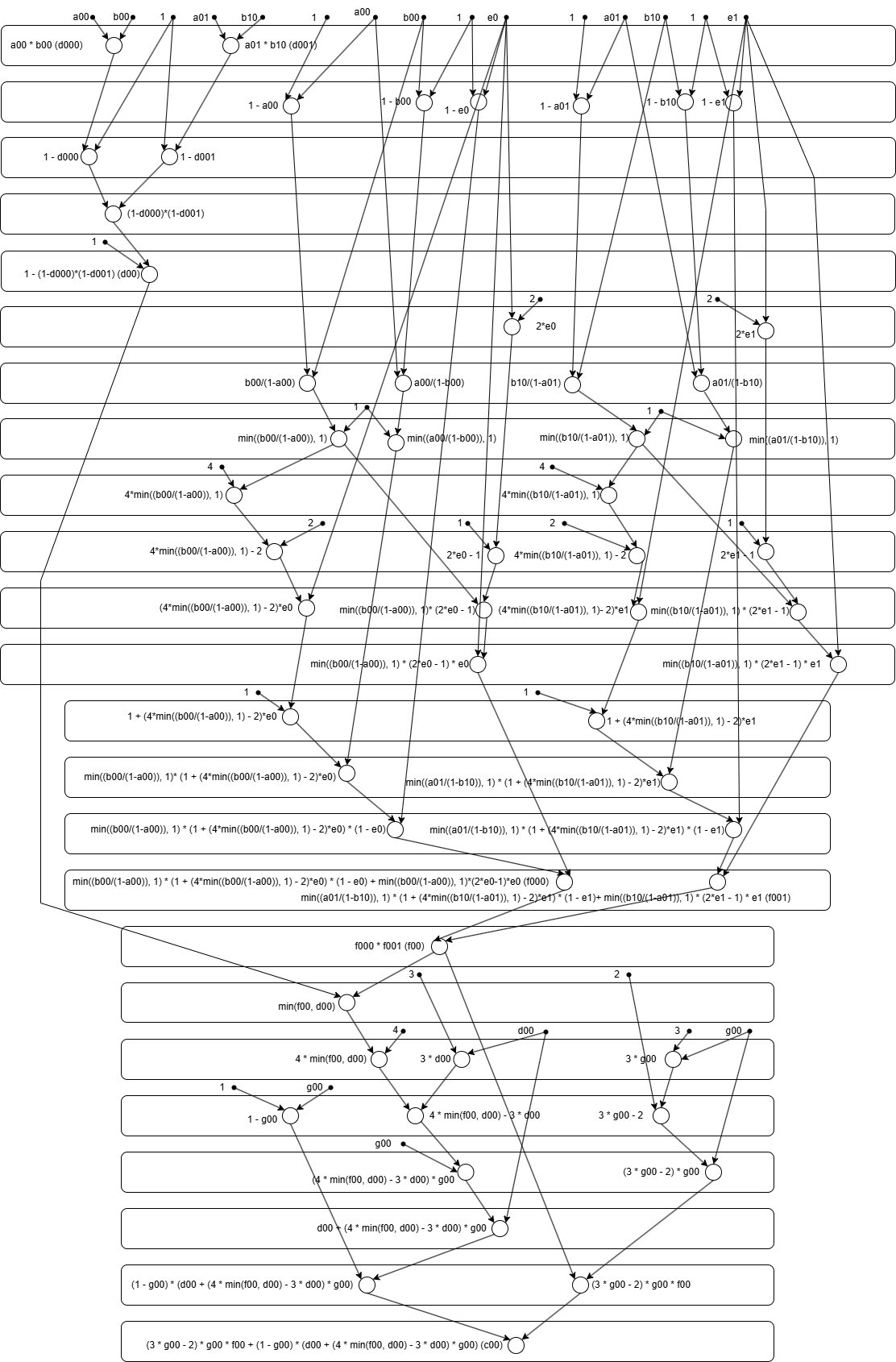
Увеличивая r, будет увеличиваться скачкообразно.

Увеличивая n, будет уменьшаться.

Увеличивая r, будет увеличиваться.

Увеличивая n, будет уменьшаться, пока n 3 \* m, где m – аргумент функции r(p, q, m).

**Информационный граф**

****

**Вывод:** в ходе лабораторной работы была разработана и изучена модель решения задачи вычисления матрицы значений на архитектуре с однородной конвейерной многопроцессорной структурой (ОКМД). Также были проведены отладка и тестирование модели, построен информационный граф. Были сформированы и проанализированы графики ключевых характеристик конвейерной архитектуры, а именно коэффициента 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.