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].

$$c_{ij} = \tilde{\bigwedge_k} f_{ijk} \cdot (3 \cdot g_{ij} - 2) \cdot g_{ij} + \left(\tilde{\bigvee_k} d_{ijk} + \left(4 \cdot \left(\tilde{\bigwedge_k} f_{ijk} \,\tilde{\circ}\, \tilde{\bigvee_k} d_{ijk}\right) - 3 \cdot \tilde{\bigvee_k} d_{ijk}\right) \cdot g_{ij}\right) \cdot (1 - g_{ij}),$$

$$f_{ijk} = (a_{ik} \tilde{\to} b_{kj}) \cdot (2 \cdot e_k - 1) \cdot e_k + (b_{kj} \tilde{\to} a_{ik}) \cdot \left(1 + \left(4 \cdot (a_{ik} \tilde{\to} b_{kj}) - 2\right) \cdot e_k\right) \cdot (1 - e_k),$$

$$d_{ijk} = a_{ik} \tilde{\wedge} b_{kj}.$$

$$\tilde{\bigwedge_k} f_{ijk} = \prod_k f_{ijk}$$

$$\tilde{\bigvee_k} d_{ijk} = 1 - \prod_k (1 - d_{ijk})$$

Согласно варианту:

$$\tilde{\bigwedge_k} f_{ijk} \,\tilde{\circ}\, \tilde{\bigvee_k} d_{ijk} = \min(\{f_{ijk}\} \cup \{d_{ijk}\})$$

$$a_{ik} \tilde{\wedge} b_{kj} = a_{ik} \cdot b_{kj}$$

$$a_{ik} \tilde{\to} b_{kj} = \sup(\{z | ((1 - a_{ik})z \le b_{kj}) \wedge (z \le 1)\})$$

$$b_{kj} \tilde{\to} a_{ik} = \sup(\{z | ((1 - b_{kj})z \le a_{ik}) \wedge (z \le 1)\})$$

Задание: Получить матрицу значений C, соответствующую размерности pxq.

Дополнительные теоретические сведения:

Коэффициент ускорения: $Ky(n,r) = \frac{T_1}{T_n}$, где:

> T_1 – время решения задачи на первой(последовательной) архитектуре;
> T_n – время решения задачи на другой (параллельной) архитектуре.

Эффективность вычислительных систем: $e(n,r) = \frac{Ky(n,r)}{n}$, где:

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

Коэффициент расхождения: $D = \frac{L_{sum}}{L_{avg}}$, где:

> L_{sum}— время (суммарная длина программы) решения задачи на (параллельной) архитектуре.

L_{avg} — 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 $(\frac{b}{1-a})$, ограничивая результат единицей.
- find_compose(a, b) вычисляет min($(\frac{b}{1-a})$,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 f_{ijk} по заданной формуле, при этом обновляя глобальное время Tn.
- find_kd(i, j) аналогично find_kf, но вычисляет дизъюнкцию d_{ijk} и накапливает время Tn.
- find_cij(i, j) с помощью find_kf и find_kd получает f и d, затем получая элемент c_{ij} 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 C и их вывод, а также отвечает за подсчет большей части показателей.

Демонстрация работы:

Входные данные:

```
m = 2
p = 2
q = 2
n = 1
```

Сгенерированные матрицы A, B, E, G:

```
A:
   0.705  0.152
   -0.11  -0.352

B:
   -0.464  -0.979
   0.146  0.39

E:
   -0.44  0.017

G:
   0.079  -0.034
   0.742  -0.497
```

Полученная матрица C:

```
C:
   -0.34085023334732056  -0.5687543254722608
   0.0427916942120238  0.9712923776955825
```

Параметры:

```
Parametrs:
T1 = 224
Tn = 224
r = 18
Ky = 1.0
e = 1.0
Lsum = 224
Lavg = 12.444444444444445
D = 18.0
```

Проверка выполнения:

$d_{ijk} = a_{ik} * b_{kj}$

$d_{000} = a_{00} * b_{00}$= -0.32712

$d_{001} = a_{01} * b_{10}$= 0.022191996

$\tilde{\vee}_k d_{ijk}$ = 1 - (1 - d_{000})(1 - d_{001}) = -0.29766855296

$$f_{ijk} = \min\left(\left(\frac{b_{kj}}{1-a_{ik}}\right), 1\right) * \left(1 + \left(4 * \min\left(\left(\frac{b_{kj}}{1-a_{ik}}\right), 1\right) - 2\right) * e_k\right) * (1 - e_k) + \min\left(\left(\frac{b_{kj}}{1-a_{ik}}\right), 1\right) * (2 * e_k - 1) * e_k$$

$$f_{000} = \min\left(\left(\frac{b_{00}}{1-a_{00}}\right), 1\right) * \left(1 + \left(4 * \min\left(\left(\frac{b_{00}}{1-a_{00}}\right), 1\right) - 2\right) * e_0\right) * (1 - e_0) + \min\left(\left(\frac{b_{00}}{1-a_{00}}\right), 1\right) * (2 * e_0 - 1) * e_0 = 1.922221906085$$

$$f_{001} = \min\left(\left(\frac{b_{10}}{1-a_{01}}\right), 1\right) * \left(1 + \left(4 * \min\left(\left(\frac{b_{10}}{1-a_{01}}\right), 1\right) - 2\right) * e_1\right) * (1 - e_1) + \min\left(\left(\frac{b_{10}}{1-a_{01}}\right), 1\right) * (2 * e_1 - 1) * e_1 = 0.168232522988$$

$\overset{\sim}{\bigwedge_k} f_{ijk} = f_{000} * f_{001} = 0.3233802410042$

$c_{ij} = \overset{\sim}{\bigwedge_k} f_{ijk} * g_{ij} * (3g_{ij} - 2) + (\overset{\sim}{\bigvee_k} d_{ijk} + g_{ij} * (4 * \overset{\sim}{\bigvee_k} d_{ijk} * \overset{\sim}{\bigwedge_k} f_{ijk} - 3 * \overset{\sim}{\bigvee_k} d_{ijk})) * (1 - g_{ij})$

$c_{00} = \overset{\sim}{\bigwedge_k} f_{ijk} * g_{00} * (3g_{00} - 2) + (\overset{\sim}{\bigvee_k} d_{ijk} + g_{00} * (4 * \overset{\sim}{\bigvee_k} d_{ijk} * \overset{\sim}{\bigwedge_k} f_{ijk} - 3 * \overset{\sim}{\bigvee_k} d_{ijk})) * (1 - g_{00})$ = 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:

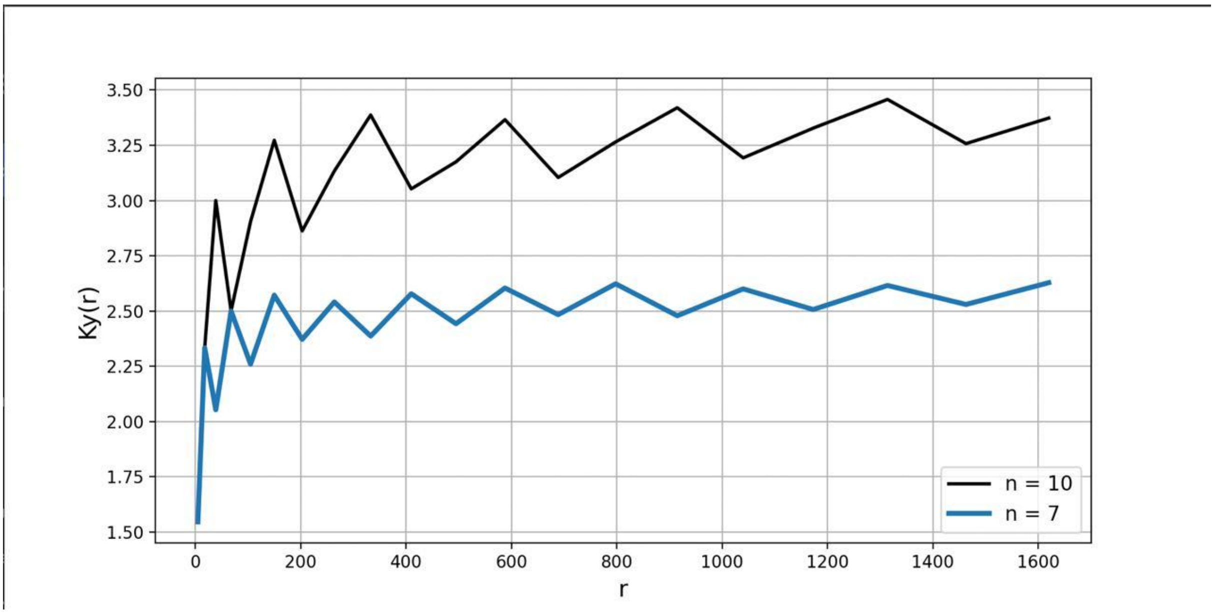


График зависимости коэффициента ускорения Ky от r

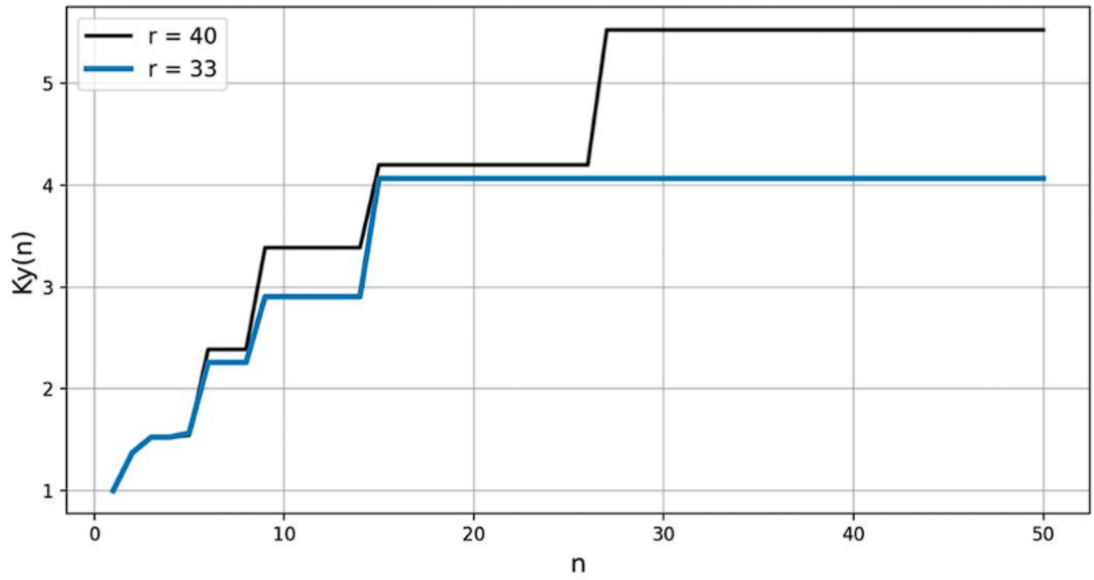


График зависимости Ку от n

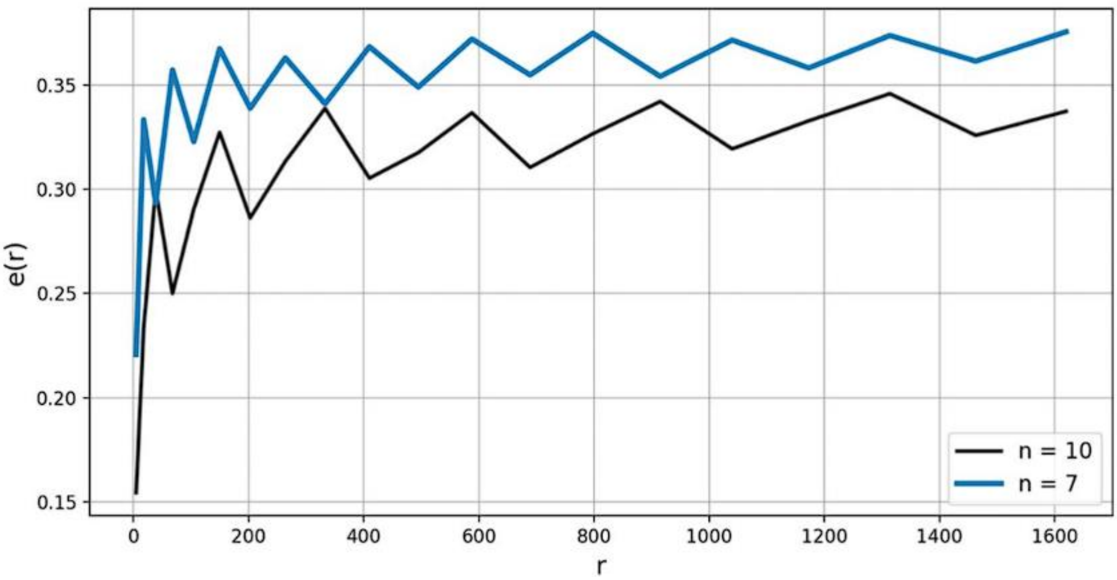


График зависимости эффективности е от r

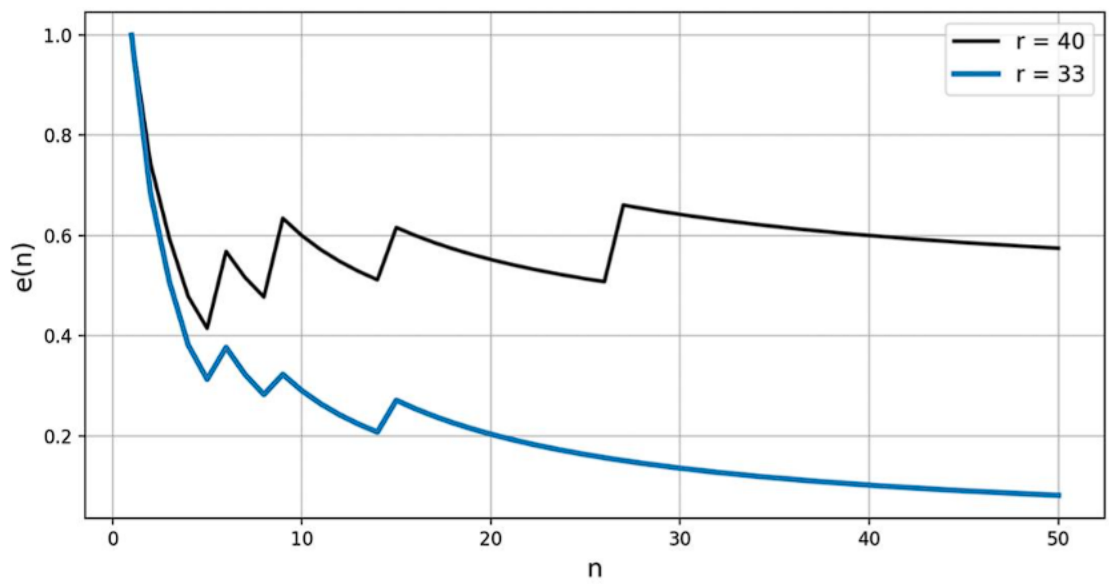


График зависимости e от n

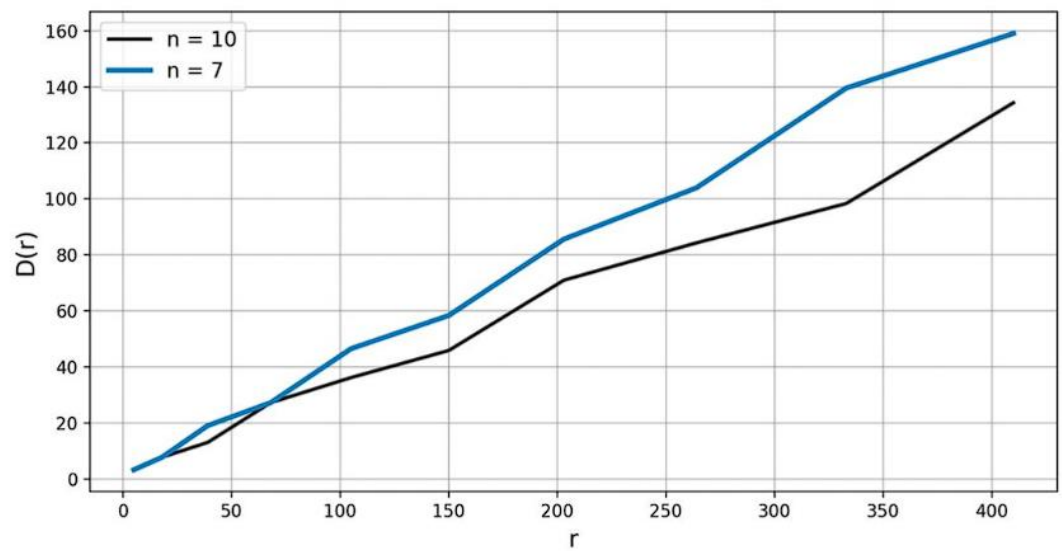


График зависимости коэффициента расхождения D от r

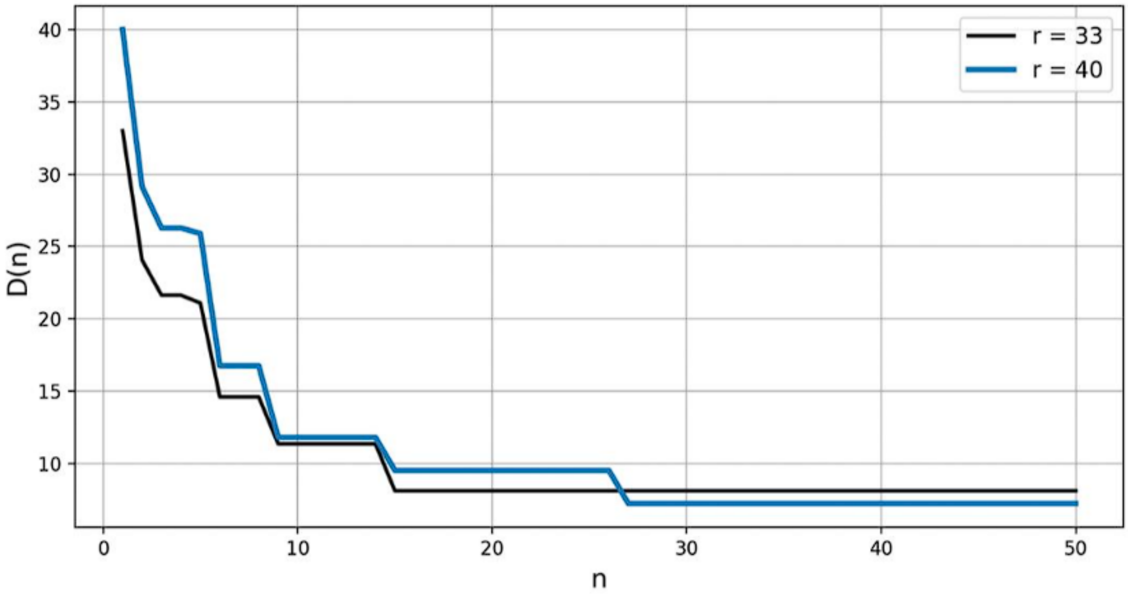


График зависимости коэффициента расхождения D от n

Контрольные вопросы:

1. *Объяснить асимптоты на графиках:*

График $K_y(r)$ не имеет чёткой асимптоты, но имеет асимптотическое приближение.

Функция $K_y(r)$ возрастает на тех промежутках, где:

$(m_{i+1} \times 3)\ \%\ n - (m_i \times 3)\ \%\ n\ > 0.$

Функция $K_y(r)$ убывает на тех промежутках, где:

$(m_{i+1} \times 3)\ \%\ n - (m_i \times 3)\ \%\ n < 0.$

На графике $K_y(n)$ нет асимптот, т.к. в определённый момент график перестает изменяться вовсе.

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

Асимптотой графика $e(n)$ является функция y = 0.

График e(r) не имеет чёткой асимптоты, но имеет асимптотическое приближение.

Функция e(r) возрастает на тех промежутках, где:

$(m_{i+1} \times 3)\ \%\ n - (m_i \times 3)\ \%\ n\ > 0.$

Функция e(r) убывает на тех промежутках, где:

$(m_{i+1} \times 3)\ \%\ n - (m_i \times 3)\ \%\ n < 0.$

График $D(r)$ не имеет чёткой асимптоты, но имеет асимптотическое приближение.

График $D(n)$ не имеет асимптот, т.к. в определённый момент график перестает изменяться вовсе.

$D(n)$ достигает своего минимального значения, когда для заданного ранга r: n ≥ 3 * m, где m – аргумент функции r(p, q, m).

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

Увеличивая r, K_y(r) будет увеличиваться скачкообразно.

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

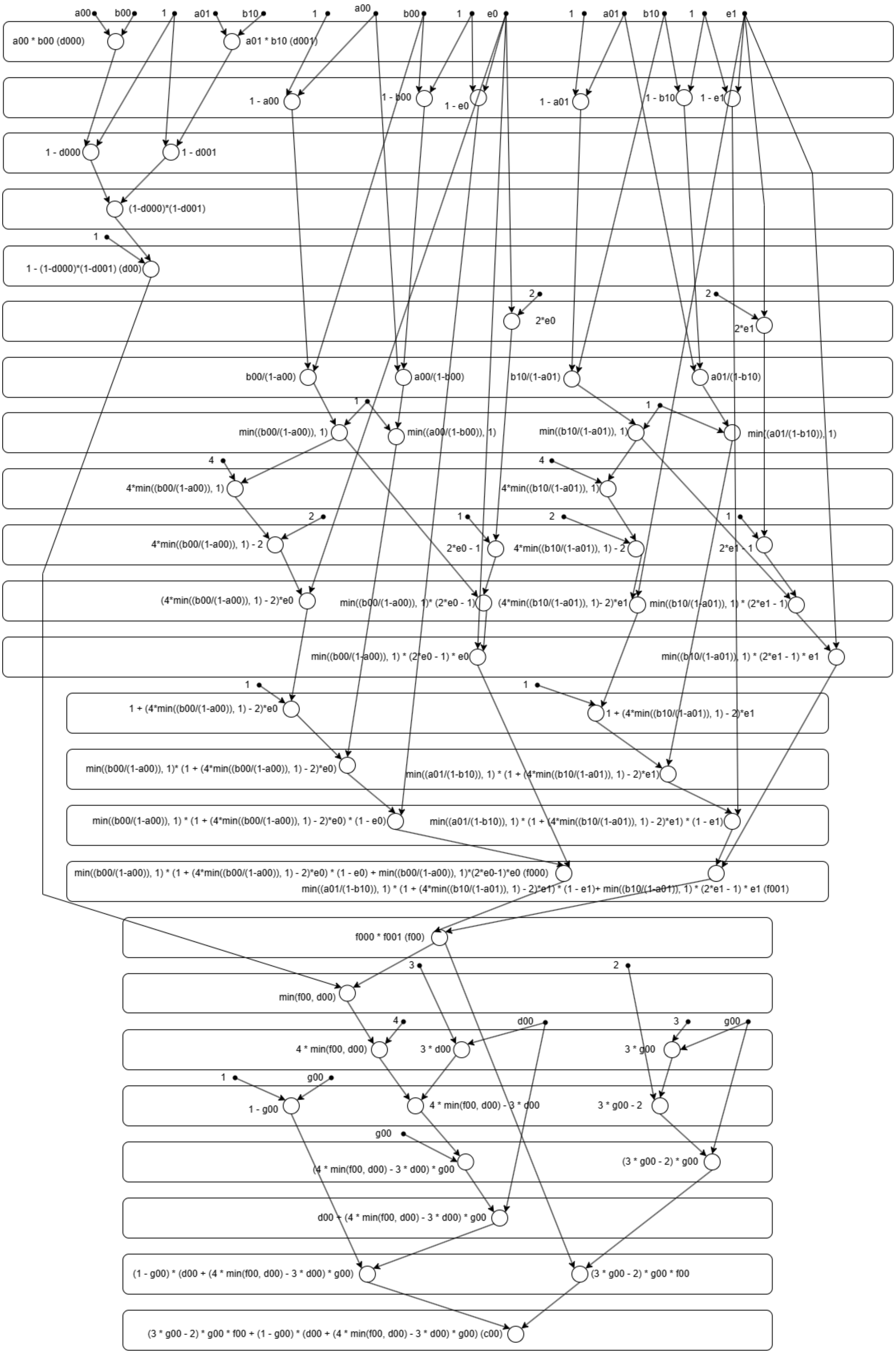
Увеличивая r, e(r) будет увеличиваться скачкообразно.

Увеличивая n, $e(n)$ будет уменьшаться.

Увеличивая r, $D(r)$ будет увеличиваться.

Увеличивая n, $D(n)$ 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.