Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

Отчет по лабораторной работе №2

по дисциплине «Модели решения задач в интеллектуальных системах»
на тему «Реализация модели решения задачи на ОКМД архитектуре»

Выполнили: студенты групы 821702
Анискович А.Д.
Терехович И.Д.

Проверил: Крачковский Д.Я.

Минск 2020

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].

$$c_{ij} = \underset{k}{\tilde{\wedge}} f_{ijk} * \left(3 * g_{ij} - 2\right) * g_{ij} + \left(\underset{k}{\tilde{\vee}} d_{ijk} + \left(4 * \left(\underset{k}{\tilde{\wedge}} f_{ijk} \,\tilde{\circ}\, \underset{k}{\tilde{\vee}} d_{ijk}\right) - 3 * \underset{k}{\tilde{\vee}} d_{ijk}\right) * g_{ij}\right) * \left(1 - g_{ij}\right)$$

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

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

Вариант задания: 5

$$\underset{k}{\tilde{\wedge}} f_{ijk} = \prod_k f_{ijk}$$

$$\underset{k}{\tilde{\vee}} d_{ijk} = 1 - \prod_k \left(1 - d_{ijk}\right)$$

$$\underset{k}{\tilde{\wedge}} f_{ijk} \,\tilde{\circ}\, \underset{k}{\tilde{\vee}} d_{ijk} = \max\left(\left\{\underset{k}{\tilde{\wedge}} f_{ijk} + \underset{k}{\tilde{\vee}} d_{ijk} - 1\right\} \cup \left\{0\right\}\right)$$

$$a_{ik} \tilde{\to} b_{kj} = a_{ik} * \left(1 - b_{kj}\right) + 1$$

$$b_{kj} \tilde{\to} a_{ik} = b_{kj} * \left(1 - a_{ik}\right) + 1$$

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

Получить: ***C*** – матрицу значений соответствующей размерности ***pxq***.

Описание модели:

T_1 – время выполнения программы на одном процессорном элементе. Вычисляется путем подсчета количества вызовов той или иной операции, а затем получение значение умножается на время данной операции. Данное действие повторяется для всех операций и в конце все значения суммируются. **T_n** – время выполнения программы на n-количестве процессорных элементов. Необходимо установить зависимости между выполняемыми операциями. Вычисляется схожим путем, что и **T_1**, за исключением поиска операций, которые можно считать на различных процессорах. Время выполнения такой операции считается следующим образом, а именно находится количество вызовов данной операции и делится на количество процессорных элементов. **K_y** – коэффициент ускорения равен **T_1/T_n**. **e** – эффективность равна **K_y/n**. D - коэффициента расхождения программы, **$D = L_{sum}/L_{avg}$**. **L_{sum}** - суммарная длина программы и равна **T_n**. **L_{avg}** - средняя длина программы. Вычисляется путѐм подсчета количества вызовов операций на различных ветвях выполнения программы. Имея, количества вызовов операций, выполняющихся на ветвях программы, и их время выполнения, считаем данную величину.

Исходные данные:

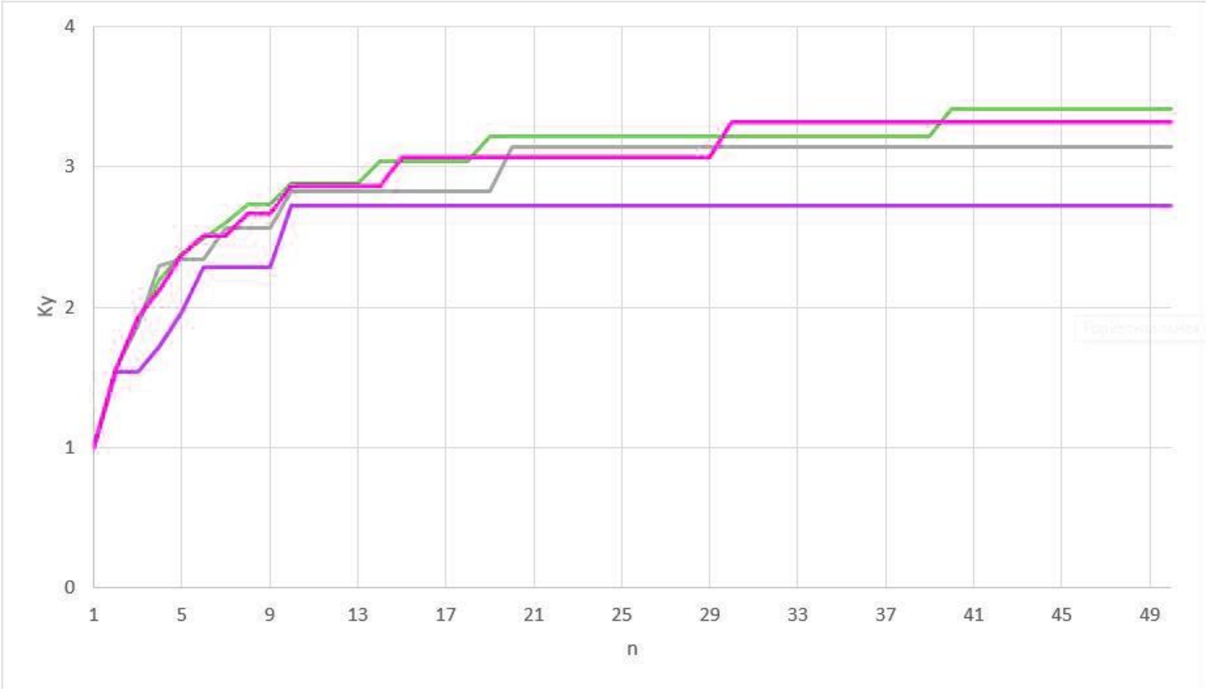
1. **p, m, q** – размерность матриц;
2. **n** – количество процессорных элементов в системе;
3. **t_i** – время (длина) выполнения операции над элементами матриц.
4. Матрицы **A**, **B**, **E**, **G** заполненные случайными числами в диапазоне [-1;1].

Вопросы:

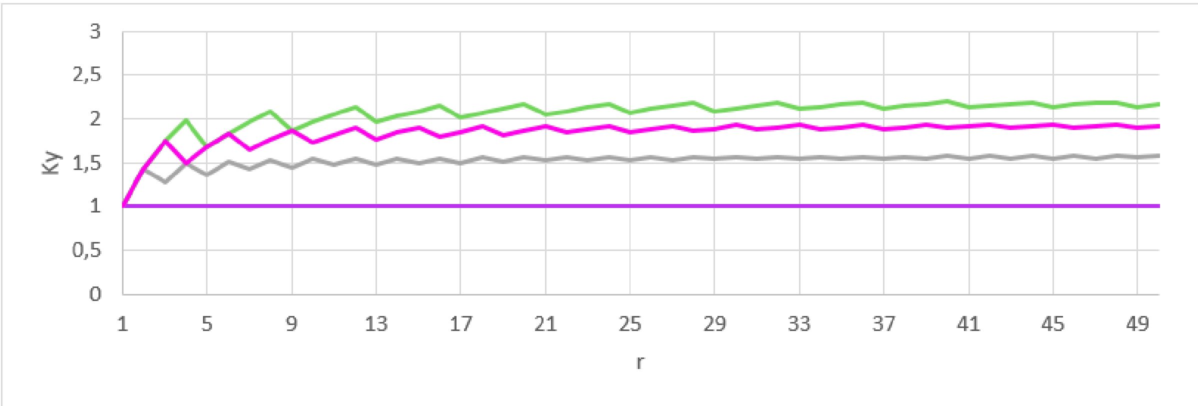
1. Проверить, что модель создана верно: программа работает правильно.

Исходные данные	Matrix A: -0.09 -0.2 0.6 0.71 Matrix B: 0.86 0.11 -0.37 -0.14 Matrix E: -1 -0.86 Matrix G: -0.6 0.42 -0.63 0.46
Результат	Matrix C: -3.14344 0.605499 3.14126 1.77107
Проверка	Matrix C[0][0] = -0.93154 – 1.212 * 1.825 = -3.14344
Вывод	Модель создана верно

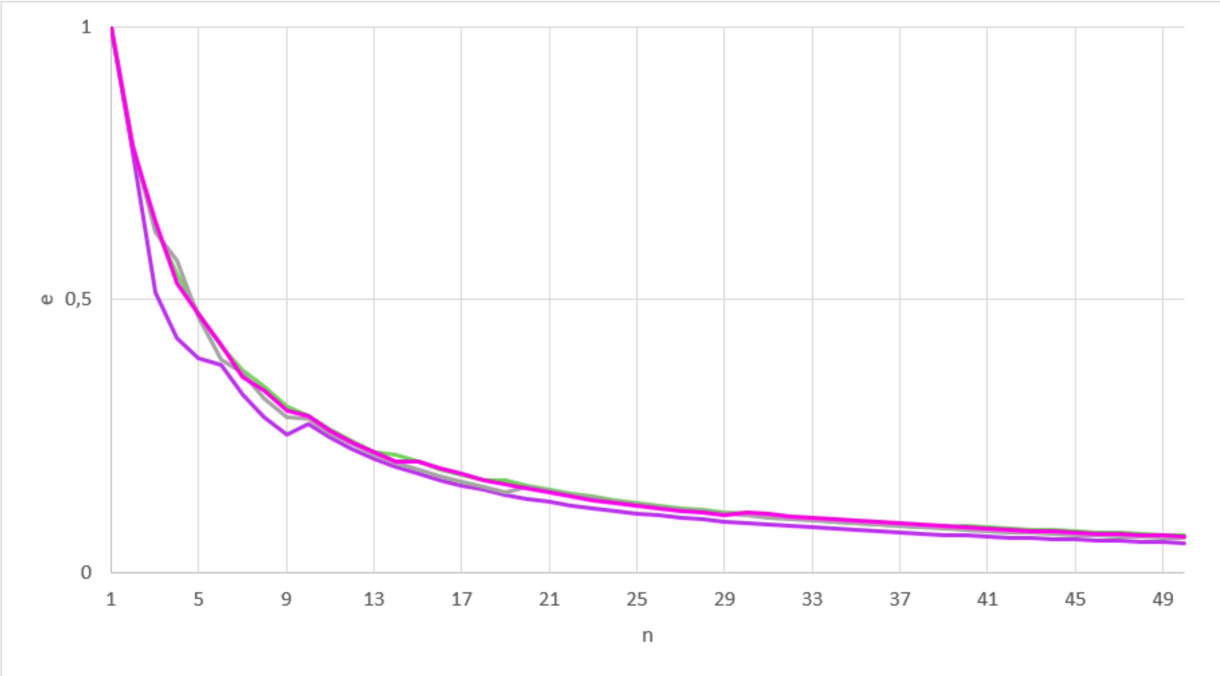
2. Построить графики и объяснить на них точки перегиба и асимптоты.



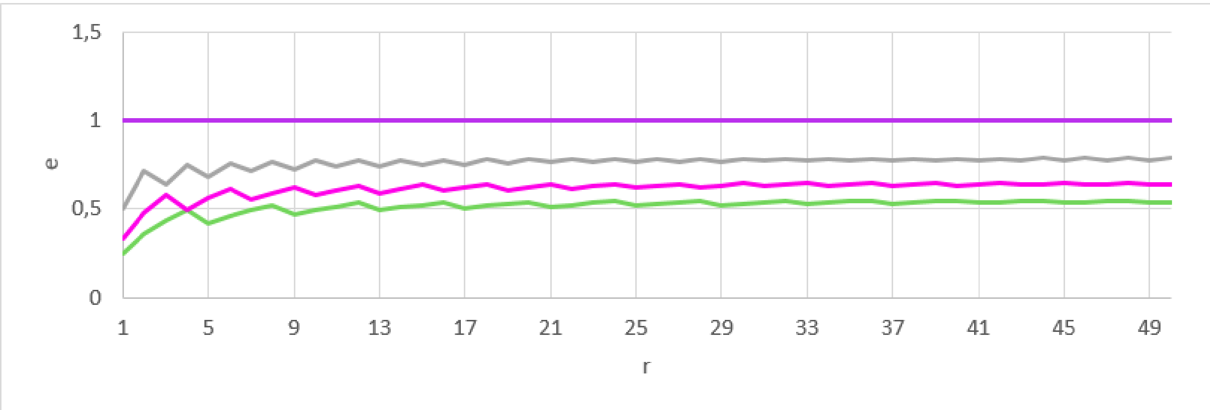
Асимптотой графика является прямая, параллельная оси абсцисс, а ордината всех точек этой прямой равна значению. Связано это с тем, что как только количество процессорных элементов становится больше ранга задачи, в вычислениях участвуют только **r** процессорных элементов, остальные никак не используются.



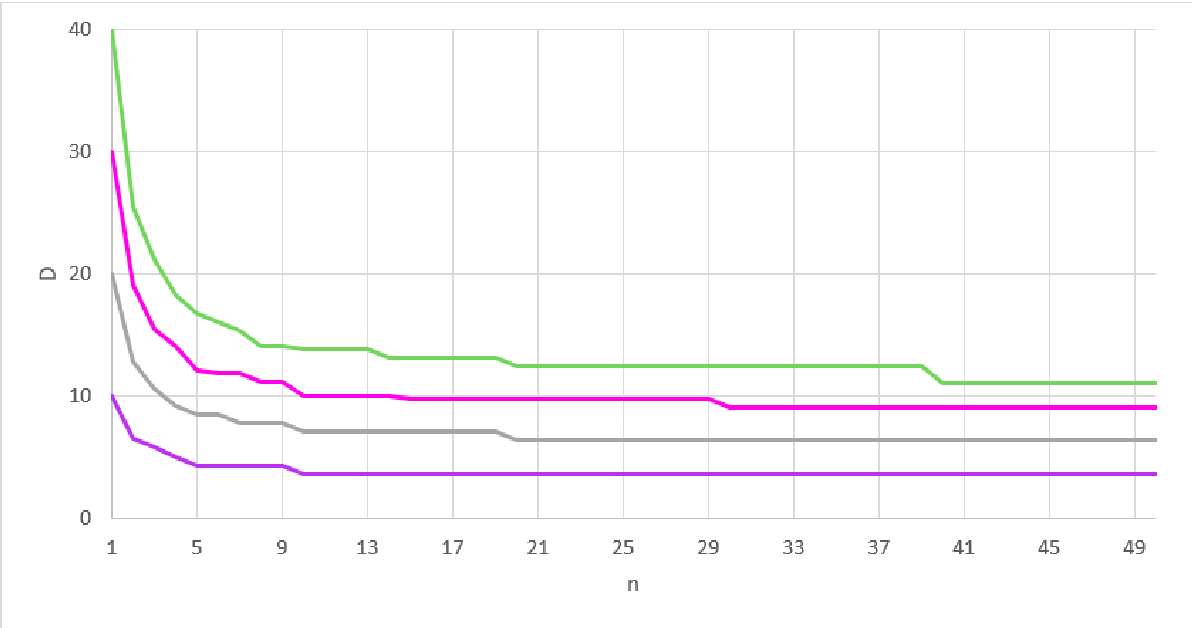
Асимптотой графика является прямая, параллельная оси абсцисс, а ордината всех точек этой прямой равна значению коэффициента ускорения при **n** = **r**. Точками перегиба являются те точки, в которых **r** кратно **n**. Связано это с тем, что при таких значениях **r**, все процессорные элементы одновременно задействованы в вычислениях.



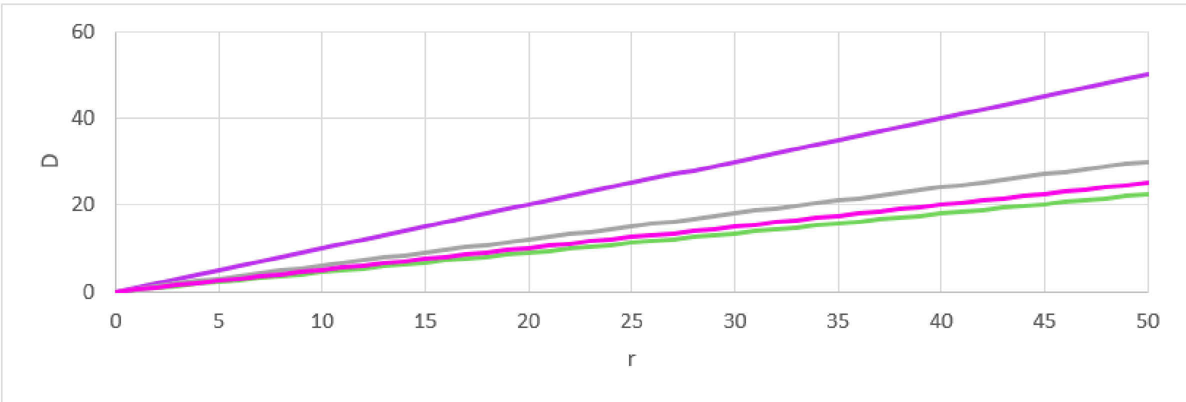
Асимптотой графика является прямая **e** = 0. Связано это с тем, что как только **n** становится равным **r**, рост коэффициента ускорения прекращается, а **n** продолжает увеличиваться.



Асимптотой графика является прямая **e** = 1. Точками перегиба являются те точки, в которых **r** кратно **n**. Связано это с тем, что при таких значениях **r**, все процессорные элементы одновременно задействованы в вычислениях.



Асимптотой графика является прямая, параллельная оси абсцисс, а ордината всех точек этой прямой равна значению коэффициенту расхождения программы при **n** = **r**. Связано это с тем, что как только количество процессорных элементов становится больше ранга задачи, в вычислениях участвуют только **r** процессорных элементов, остальные никак не используются.



Асимптотой графика является функция **D**=**k*r+b**. При n=1: k=1, b=0; при n=2: k=0.6, b=1; при n=3: k=0.5, b=1; при n=4: k=0.45, b=0.5.

3. Спрогнозировать, как изменится вид графиков при изменении параметров модели. Если модель позволяет, то проверить на ней правильность ответа.

1. Увеличивая **n**, **K_y(n)** увеличивается. Рост значения **K_y(n)** наблюдается до тех пор, пока количество процессорных элементов не становится равным рангу задачи. После этого коэффициент ускорения не изменяется. Увеличивая **r**, **K_y(r)** увеличивается скачкообразно.

2. Увеличивая **n**, **e(n)** уменьшается. Увеличивая **r**, **e(r)** растет скачкообразно.
3. Увеличивая **n**, **D(n)** уменьшается. Падение значения **D(n)** наблюдается до тех пор, пока количество процессорных элементов не становится равным рангу задачи. После этого коэффициент расхождения программы не изменяется. Увеличивая **r**, **D(r)** растет.

Вывод:

В результате выполнения лабораторной работы была реализована модель вычисления матрицы значений на ОКМД архитектуре. Данная модель была проверена на работоспособность и правильность получаемых результатов. С помощью графиков, построенных в результате выполнения лабораторной работы, были изучены зависимости коэффициента ускорения, эффективности и коэффициента расхождения программы от количества процессорных элементов и ранга задачи.