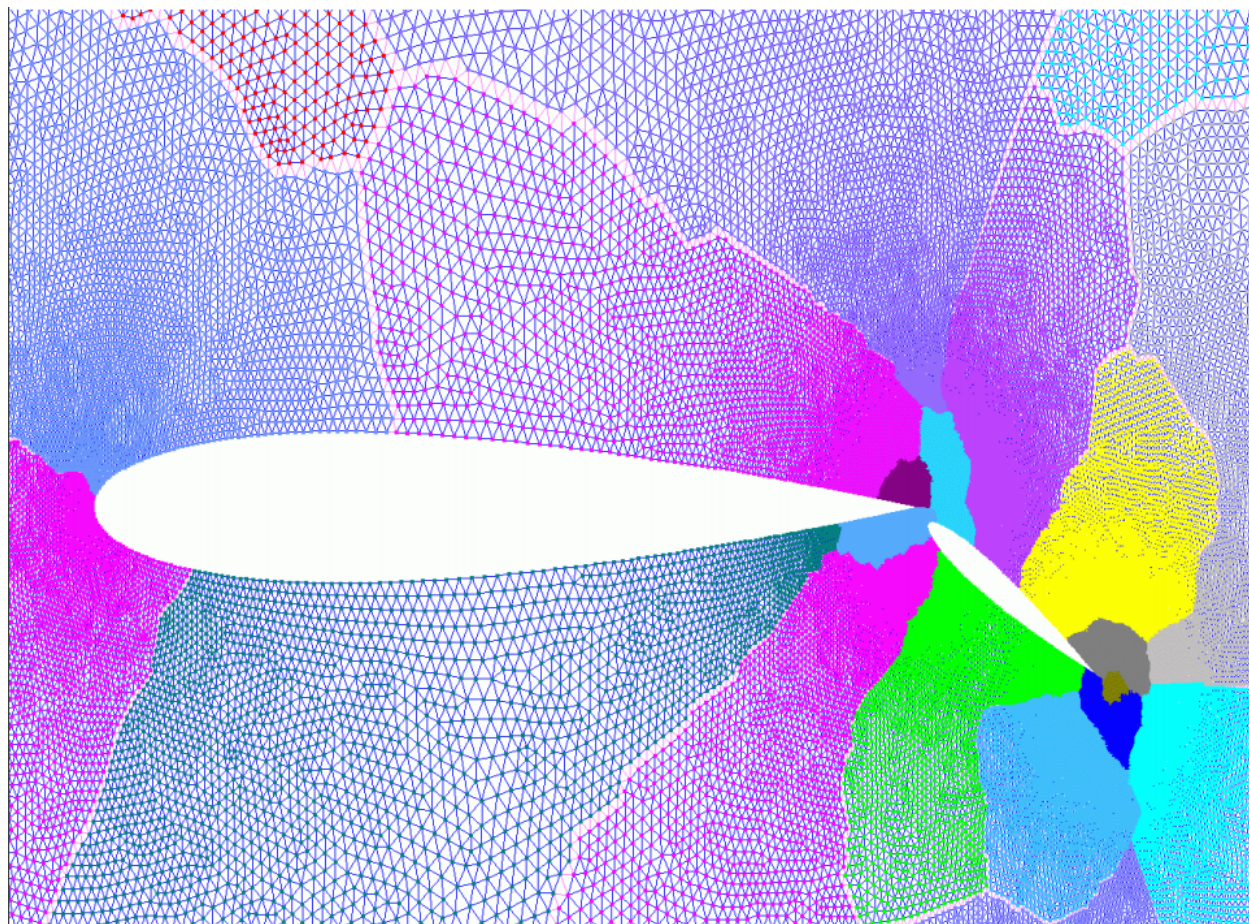


# ОТЧЕТ

Задача распределения узлов между процессорами.



**Кузьменко Илья**

Последняя дата редактирования: 23.04.2023

Группа 304

МГУ ВМК ВМ

[ilyexakuzmenko@gmail.com](mailto:ilyexakuzmenko@gmail.com)

Научный руководитель: Якововский Михаил Владимирович

## ~~~~~СОДЕРЖАНИЕ~~~~~

Постановка задачи.....	2
Программная реализация.....	3
Введение.....	3
Наименования файлов.....	3
Компиляция.....	3
Вспомогательные функции.....	4
Ход программы. Файл <i>redistribution.cpp</i> .....	5
Результаты выполнения.....	6
Примеры.....	7
Литература.....	21

## ~~~~~ПОСТАНОВКА ЗАДАЧИ~~~~~

-> Задача:

Реализовать программу, которая разбивает квадратную матрицу вычислительных узлов размерности *m*х*m* между *N* процессорами так, чтобы трудоемкости на каждом процессоре приблизительно были одинаковыми.

-> Дополнительное условие:

Изначально трудоемкость каждого узла равна 1. Из левого нижнего узла матрицы начинает двигаться круг, внутри которого трудоемкость умножается на 10.

Программа должна эффективно разделить узлы между процессорами.

-> Входные данные:

```
N = 5; // Число процессоров
m = 10; // Количество строк матрицы
n = m; // Количество столбцов матрицы
nnodes = n * m; // Общее количество узлов
ncon = 1; // Количество весов каждого узла (1 - трудоемкость)
float k = 0.5; // Коэффициент трудоемкости при передаче информации
Radius = 4; // Радиус круга, задаваемый количеством узлов (шт).
Speed = 1; // Количество узлов, которое пройдет центр круга по диагонали за
            секунду (шт/сек).
Time = 2; // Время, которое двигался круг по диагонали (сек).
```

## ~~~~~ПРОГРАММНАЯ РЕАЛИЗАЦИЯ~~~~~

### 1. Введение.

Для реализации решения поставленной задачей была использована дополнительная внешняя библиотека *METIS*.

– *METIS (Matrix Elementary Transformation on Irregular Structures)* - это пакет программ для разбиения графов на части и решения различных задач, связанных с ними, таких как распределение задач на процессоры в распределенных вычислениях или уменьшение размерности больших графов для более эффективного анализа. *METIS* основан на алгоритмах разбиения графов на части, которые оптимизируют множество параметров, таких как размер каждой части, количество связей между частями и распределение веса на процессорах.

*METIS* написан на языке C и доступен как открытое программное обеспечение под лицензией Apache.

### 2. Наименования файлов.

Полноценная программа написана на 2 языках программирования: *C++* и *Python*, и состоит из 4 файлов:

- a) *MAIN.cpp* - главный файл, который запускает все остальные файлы;
- b) *redistributor.cpp* - файл, который содержит работу с матрицей.
- c) *graph.py* - файл, который строит графики с распределением вычислительных узлов.
- d) *laboriousness.cpp* - файл, который рассчитывает трудоемкости для каждого процессора.

В ходе выполнения программа создает 2 дополнительных файла в текущей директории:

- a) *matrix.csv* - файл, содержащий матрицу с разбиением узлов между процессорами, где элемент матрицы - номер процессора, который его обрабатывает.
- b) *matrixCirlce.csv* - файл, содержащий матрицу с маской круга.

### 3. Компиляция.

Для того, чтобы протестировать работу программы, необходимо:

- 1) Установить *METIS*.
- 2) Поместить все файлы в одну директорию.

3) Один раз скомпилировать файл redistributor.cpp:

```
$ g++ redistributor.cpp -lmetis
```

4) Установить в файле graph.py пути к создавшимся в ходе компиляции файлам matrix.csv и matrixCircle.csv.

5) Скомпилировать файл MAIN.cpp и запустить получившийся файл:

```
$ g++ MAIN.cpp
```

```
$ ./a.out
```

## 4. Вспомогательные функции.

Файл redistributor.cpp содержит несколько вспомогательных функций:

a) Функция:

```
void getMAINvalues(idx_t N1, idx_t m1, idx_t n1, idx_t nnodes1, idx_t  
ncon1, idx_t Radius1, idx_t Speed1, idx_t Time1){ ... }
```

– функция, связывающая файл redistributor.cpp и MAIN.cpp.

b) Функция:

```
idx_t* movingCircle (idx_t R, idx_t V, idx_t T){ ... }
```

– функция возвращает матрицу трудоемкостей, где внутри круга трудоемкость 10, а вне круга 1. Круг, центр которого располагался в начальный момент времени в НИЖНЕМ ЛЕВОМ узле, двигается по диагонали, стартуя из НИЖНЕГО ЛЕВОГО узла.

$R$  - Радиус круга, задаваемый количеством узлов,

$V$  - количество узлов, проходимых центром круга по диагонали за секунду,

$T$  - количество секунд - время, которое двигался круг по диагонали.

c) Функция:

```
void printAdjncy(idx_t* adjncy, idx_t* xadj){ ... }
```

– Функция выводит массив, содержащий список всех соседних вершин для каждой вершины графа. Используется для отладки программы.

$adjncy$  - массив, содержащий список всех соседних вершин для каждой вершины графа.

$xadj$  - массив, содержащий индексы начала каждой строки в массиве  $adjncy$ .

Файл laboriousness.cpp также содержит несколько вспомогательных функций:

a) Функция:

```
void readMatrixSize(const string& file, idx_t& numRows, idx_t& numCols) { ... }
```

– Функция считывает размерность матрицы из файла *file*. Работает с файлами *matrix.csv* и *matrixCirlce.csv*.

*file* - имя файла;

*numRows* - количество строк матрицы;

*numCols* - количество столбцов матрицы.

b) Функция:

```
void readMatrix(const string& file, idx_t* matrix, idx_t& numRows, idx_t& numCols){ ... }
```

– Функция считывает матрицу из файла *file* в *matrix*. Работает с файлами *matrix.csv* и *matrixCirlce.csv*.

*file* - имя файла;

*matrix* - адрес пространства, куда будет записана матрица;

*numRows* - количество строк матрицы;

*numCols* - количество столбцов матрицы.

## 5. Ход программы. Файл *redistribution.cpp*.

Для того, чтобы программа находила необходимое разбиение, было пройдено несколько этапов:

1) Выделение памяти.

2) Заполнение *xadj* и *adjncy*.

3) Вычисление маски трудоемкостей *circle*.

4) Заполнение *vwgt*.

*vwgt* - массив, содержащий веса для каждого узла.

5) Вызов функции *METIS\_PartGraphRecursive()* для разбиения графа на части:

```
METIS_PartGraphRecursive(&nnodes, &ncon, xadj, adjncy, vwgt, adjwgt, NULL, &N, NULL, NULL, options, &objval, part);
```

6) Изменение значения элемента в матрице *T*.

7) Вывод маски *circle* в файл *matrixCircle.csv*.

8) Вывод матрицы *T* в файл *matrix.csv*.

9) Очистка памяти.

## ~~~~~РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ~~~~~

Для наглядности разбиения был добавлен файл `graph.py`. Результаты его работы будут прикреплены ниже.

Для демонстрации работы программы будем рассматривать входные значения и результат работы программы.

### **Введем обозначения:**

$N$  - число процессоров;

$m$  - размерность квадратной матрицы узлов;

$R$  - радиус круга, задаваемый количеством узлов (считая центр, то есть при  $R=1$  в маске круга должна быть только 1 ячейка - центр) (*шт*) ;

$V$  - скорость - количество узлов, которые прошел центр круга по диагонали за 1 секунду (*шт/сек*)

$T$  - время, которое двигался круг по диагонали (*сек*).

$k$  - Коэффициент трудоемкости при передачи информации от одного процессора к другому

## Примеры:

Для всех следующих случаев рассмотрим  $k = 0.5$ .

**NumProc** – номер процессора

**laborsGeneral** – основная трудоемкость узла, задается положением круга;

**laborsTrans** – количество ребер графа одного процессора для вычисления трудоемкости узла, вызванной передачей информации другому процессору;

**laborsMain** – итоговая трудоемкость на процессоре;

**Average labor intensity** – среднее значение трудоемкости на процессоре;

**deviation** – отклонение итоговой трудоемкости от средней;

**Max deviation** – максимальное отклонение среди всех процессоров.

1. **Ввод:**  $N = 3$ ,  $m = 5$ ,  $R = 1$ ,  $V = 0$ ,  $T = 1$ :

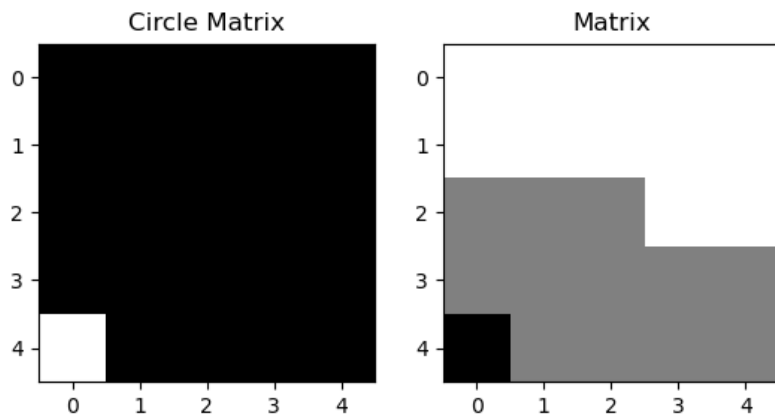
Вывод:

Маска круга:

```
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
0 0 0 0 0
2 0 0 0 0
```

Матрица распределения:

```
[[ 3, 3, 3, 3, 3],
 [ 3, 3, 3, 3, 3],
 [ 2, 2, 2, 3, 3],
 [ 2, 2, 2, 2, 2],
 [ 1, 2, 2, 2, 2]]
```



Average labor intensity: 11.3333

Max deviation: 4.66667

NumProc: 1 laborsGeneral: 10 laborsTrans: 2 k: 0.5 laborsMain: 11 deviation: -0.333333

NumProc: 2 laborsGeneral: 12 laborsTrans: 8 k: 0.5 laborsMain: 16 deviation: 4.66667

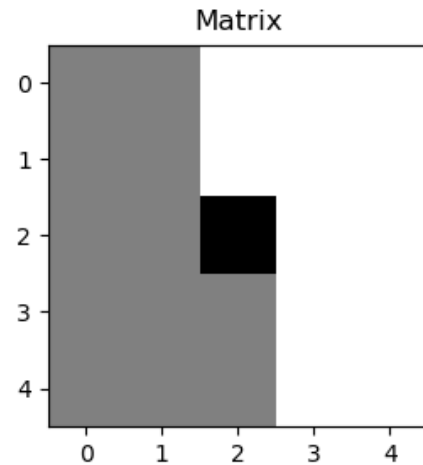
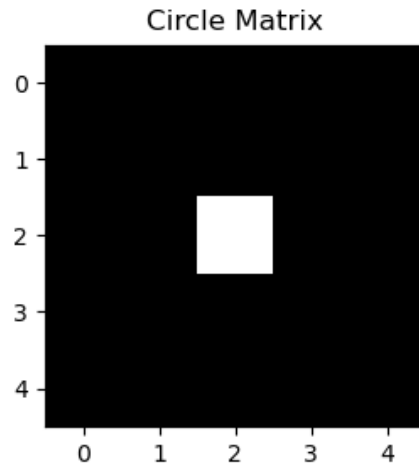
NumProc: 3 laborsGeneral: 12 laborsTrans: 6 k: 0.5 laborsMain: 15 deviation: 3.66667

2. **Ввод:**  $N = 3$ ,  $m = 5$ ,  $R = 1$ ,  $V = 2$ ,  $T = 1$ :

Вывод:

Маска круга:

```
0 0 0 0 0
0 0 0 0 0
0 0 2 0 0
0 0 0 0 0
0 0 0 0 0
```



Матрица распределения:

```
[[ 2, 2, 3, 3, 3],
 [ 2, 2, 3, 3, 3],
 [ 2, 2, 1, 3, 3],
 [ 2, 2, 2, 3, 3],
 [ 2, 2, 2, 3, 3]]
```

Average labor intensity: 11.3333

Max deviation: 3.66667

NumProc: 1 laborsGeneral: 10 laborsTrans: 4 k: 0.5 laborsMain: 12 deviation: 0.666667

NumProc: 2 laborsGeneral: 12 laborsTrans: 6 k: 0.5 laborsMain: 15 deviation: 3.66667

NumProc: 3 laborsGeneral: 12 laborsTrans: 6 k: 0.5 laborsMain: 15 deviation: 3.66667



3. Ввод:  $N = 3$ ,  $m = 10$ ,  $R = 3$ ,  $V = 5$ ,  $T = 1$ :

Вывод:

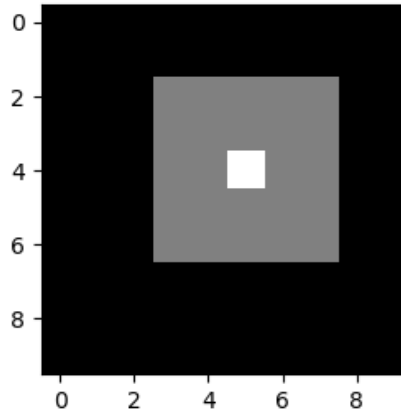
Маска круга:

```

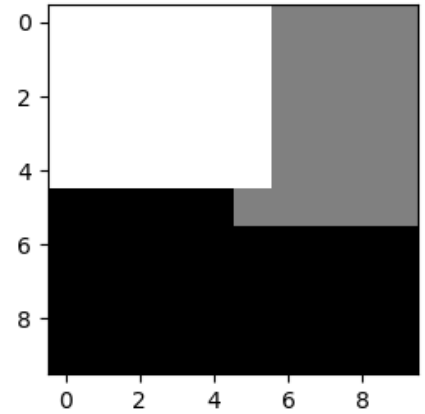
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 1 1 2 1 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

Circle Matrix



Matrix



Матрица распределения:

```

[[ 3, 3, 3, 3, 3, 3, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 3, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 3, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 3, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 3, 2, 2, 2, 2],
[ 1, 1, 1, 1, 1, 2, 2, 2, 2, 2],
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],
[ 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]]

```

Average labor intensity: 108.333

Max deviation: 8.16666

NumProc: 1 laborsGeneral: 108 laborsTrans: 11 k: 0.5 laborsMain: 113.5 deviation: 5.16666

NumProc: 2 laborsGeneral: 106 laborsTrans: 12 k: 0.5 laborsMain: 112 deviation: 3.66666

NumProc: 3 laborsGeneral: 111 laborsTrans: 11 k: 0.5 laborsMain: 116.5 deviation: 8.16666

4. Ввод:  $N = 3$ ,  $m = 10$ ,  $R = 5$ ,  $V = 5$ ,  $T = 1$ :

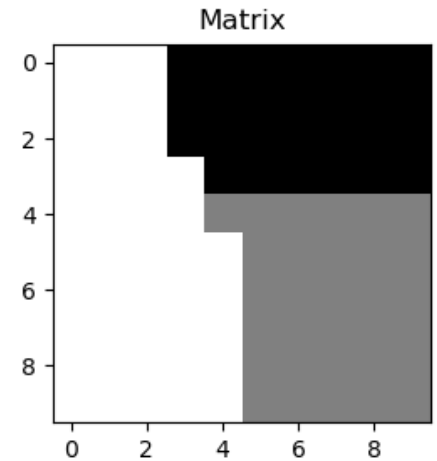
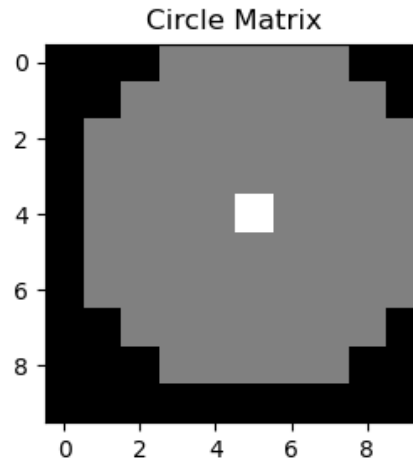
Вывод:

Маска круга:

```

0 0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 2 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 0
0 0 0 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0

```



Матрица распределения:

```

[[ 3, 3, 3, 1, 1, 1, 1, 1, 1, 1],
[ 3, 3, 3, 1, 1, 1, 1, 1, 1, 1],
[ 3, 3, 3, 1, 1, 1, 1, 1, 1, 1],
[ 3, 3, 3, 3, 1, 1, 1, 1, 1, 1],
[ 3, 3, 3, 3, 2, 2, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 2, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 2, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 2, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 2, 2, 2, 2, 2],
[ 3, 3, 3, 3, 3, 2, 2, 2, 2, 2]]

```

Average labor intensity: 240.333

Max deviation: 8.16667

NumProc: 1 laborsGeneral: 243 laborsTrans: 11 k: 0.5 laborsMain: 248.5 deviation: 8.16667

NumProc: 2 laborsGeneral: 238 laborsTrans: 13 k: 0.5 laborsMain: 244.5 deviation: 4.16667

NumProc: 3 laborsGeneral: 240 laborsTrans: 12 k: 0.5 laborsMain: 246 deviation: 5.66667

5. **Ввод:**  $N=5$ ,  $m=10$ ,  $R=5$ ,  $V=5$ ,  $T=1$ :

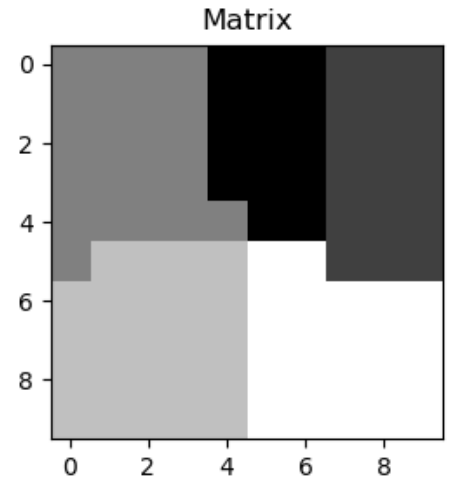
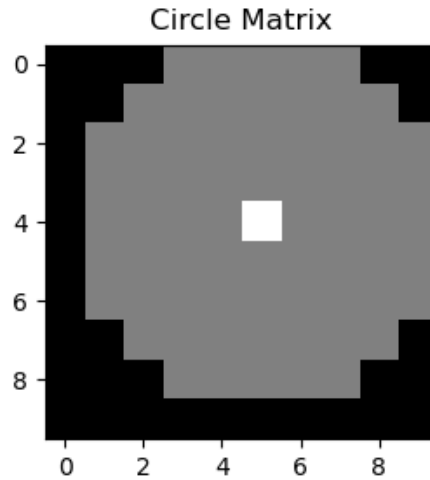
Вывод:

Маска круга:

```

0 0 0 1 1 1 1 1 0 0
0 0 1 1 1 1 1 1 1 0
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 2 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 1 1 1 1 1 1 1 1 1
0 0 1 1 1 1 1 1 1 0
0 0 0 1 1 1 1 1 0 0
0 0 0 0 0 0 0 0 0 0

```



Матрица распределения:

```

[[ 3, 3, 3, 3, 1, 1, 1, 2, 2, 2],
[ 3, 3, 3, 3, 1, 1, 1, 2, 2, 2],
[ 3, 3, 3, 3, 1, 1, 1, 2, 2, 2],
[ 3, 3, 3, 3, 1, 1, 1, 2, 2, 2],
[ 3, 3, 3, 3, 3, 1, 1, 2, 2, 2],
[ 3, 4, 4, 4, 4, 5, 5, 2, 2, 2],
[ 4, 4, 4, 4, 4, 5, 5, 5, 5, 5],
[ 4, 4, 4, 4, 4, 5, 5, 5, 5, 5],
[ 4, 4, 4, 4, 4, 5, 5, 5, 5, 5],
[ 4, 4, 4, 4, 4, 5, 5, 5, 5, 5]]

```

Average labor intensity: 144.2

Max deviation: 13.3

```

NumProc: 1  laborsGeneral: 140  laborsTrans: 13  k: 0.5  laborsMain: 146.5  deviation: 2.3
NumProc: 2  laborsGeneral: 153  laborsTrans: 9   k: 0.5  laborsMain: 157.5  deviation: 13.3
NumProc: 3  laborsGeneral: 139  laborsTrans: 12 k: 0.5  laborsMain: 145   deviation: 0.800003
NumProc: 4  laborsGeneral: 141  laborsTrans: 11 k: 0.5  laborsMain: 146.5  deviation: 2.3
NumProc: 5  laborsGeneral: 148  laborsTrans: 11 k: 0.5  laborsMain: 153.5  deviation: 9.3

```

6. Ввод:  $N = 7$ ,  $m = 100$ ,  $R = 20$ ,  $V = 40$ ,  $T = 1$ :

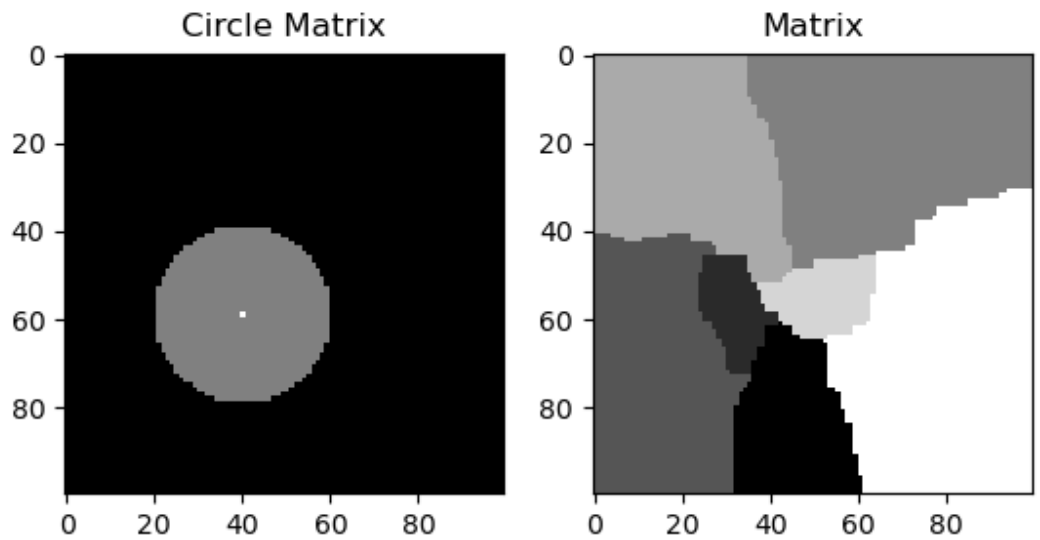
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 3029.29

Max deviation: 65.7144

NumProc: 1	laborsGeneral: 3029	laborsTrans: 107	k: 0.5	laborsMain: 3082.5	deviation: 53.2144
NumProc: 2	laborsGeneral: 3030	laborsTrans: 90	k: 0.5	laborsMain: 3075	deviation: 45.7144
NumProc: 3	laborsGeneral: 3028	laborsTrans: 111	k: 0.5	laborsMain: 3083.5	deviation: 54.2144
NumProc: 4	laborsGeneral: 3029	laborsTrans: 132	k: 0.5	laborsMain: 3095	deviation: 65.7144
NumProc: 5	laborsGeneral: 3030	laborsTrans: 122	k: 0.5	laborsMain: 3091	deviation: 61.7144
NumProc: 6	laborsGeneral: 3029	laborsTrans: 92	k: 0.5	laborsMain: 3075	deviation: 45.7144
NumProc: 7	laborsGeneral: 3030	laborsTrans: 126	k: 0.5	laborsMain: 3093	deviation: 63.7144

7. Ввод:  $N = 10$ ,  $m = 100$ ,  $R = 40$ ,  $V = 40$ ,  $T = 1$ :

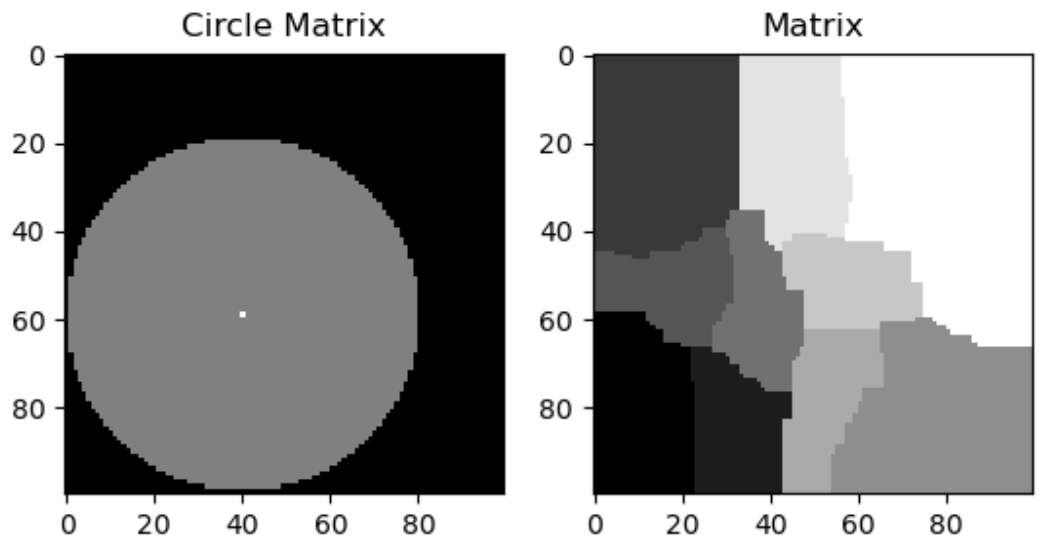
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 5511.7

Max deviation: 62.2998

NumProc: 1	laborsGeneral: 5514	laborsTrans: 64	k: 0.5	laborsMain: 5546	deviation: 34.2998
NumProc: 2	laborsGeneral: 5506	laborsTrans: 92	k: 0.5	laborsMain: 5552	deviation: 40.2998
NumProc: 3	laborsGeneral: 5514	laborsTrans: 82	k: 0.5	laborsMain: 5555	deviation: 43.2998
NumProc: 4	laborsGeneral: 5512	laborsTrans: 108	k: 0.5	laborsMain: 5566	deviation: 54.2998
NumProc: 5	laborsGeneral: 5510	laborsTrans: 128	k: 0.5	laborsMain: 5574	deviation: 62.2998
NumProc: 6	laborsGeneral: 5507	laborsTrans: 95	k: 0.5	laborsMain: 5554.5	deviation: 42.7998
NumProc: 7	laborsGeneral: 5517	laborsTrans: 109	k: 0.5	laborsMain: 5571.5	deviation: 59.7998
NumProc: 8	laborsGeneral: 5510	laborsTrans: 108	k: 0.5	laborsMain: 5564	deviation: 52.2998
NumProc: 9	laborsGeneral: 5513	laborsTrans: 121	k: 0.5	laborsMain: 5573.5	deviation: 61.7998
NumProc: 10	laborsGeneral: 5514	laborsTrans: 115	k: 0.5	laborsMain: 5571.5	deviation: 59.7998

8. **Ввод:**  $N = 20$ ,  $m = 1000$ ,  $R = 300$ ,  $V = 600$ ,  $T = 1$ :

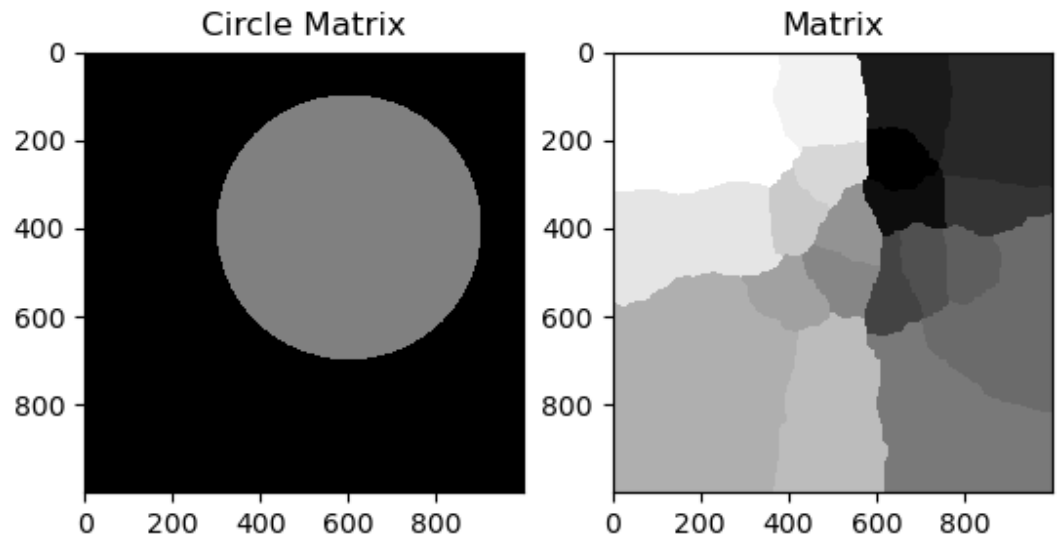
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 177205

Max deviation: 574.344

NumProc: 1	laborsGeneral: 177200	laborsTrans: 650	k: 0.5	laborsMain: 177525	deviation: 320.344
NumProc: 2	laborsGeneral: 177210	laborsTrans: 778	k: 0.5	laborsMain: 177599	deviation: 394.344
NumProc: 3	laborsGeneral: 177204	laborsTrans: 764	k: 0.5	laborsMain: 177586	deviation: 381.344
NumProc: 4	laborsGeneral: 177204	laborsTrans: 618	k: 0.5	laborsMain: 177513	deviation: 308.344
NumProc: 5	laborsGeneral: 177204	laborsTrans: 753	k: 0.5	laborsMain: 177580	deviation: 375.844
NumProc: 6	laborsGeneral: 177210	laborsTrans: 754	k: 0.5	laborsMain: 177587	deviation: 382.344
NumProc: 7	laborsGeneral: 177200	laborsTrans: 668	k: 0.5	laborsMain: 177534	deviation: 329.344
NumProc: 8	laborsGeneral: 177204	laborsTrans: 640	k: 0.5	laborsMain: 177524	deviation: 319.344
NumProc: 9	laborsGeneral: 177204	laborsTrans: 1116	k: 0.5	laborsMain: 177762	deviation: 557.344
NumProc: 10	laborsGeneral: 177206	laborsTrans: 1124	k: 0.5	laborsMain: 177768	deviation: 563.344
NumProc: 11	laborsGeneral: 177200	laborsTrans: 706	k: 0.5	laborsMain: 177553	deviation: 348.344
NumProc: 12	laborsGeneral: 177210	laborsTrans: 684	k: 0.5	laborsMain: 177552	deviation: 347.344
NumProc: 13	laborsGeneral: 177204	laborsTrans: 764	k: 0.5	laborsMain: 177586	deviation: 381.344
NumProc: 14	laborsGeneral: 177204	laborsTrans: 1075	k: 0.5	laborsMain: 177742	deviation: 536.844
NumProc: 15	laborsGeneral: 177205	laborsTrans: 1135	k: 0.5	laborsMain: 177772	deviation: 567.844
NumProc: 16	laborsGeneral: 177200	laborsTrans: 692	k: 0.5	laborsMain: 177546	deviation: 341.344
NumProc: 17	laborsGeneral: 177210	laborsTrans: 680	k: 0.5	laborsMain: 177550	deviation: 345.344
NumProc: 18	laborsGeneral: 177204	laborsTrans: 1150	k: 0.5	laborsMain: 177779	deviation: 574.344
NumProc: 19	laborsGeneral: 177205	laborsTrans: 757	k: 0.5	laborsMain: 177584	deviation: 378.844
NumProc: 20	laborsGeneral: 177205	laborsTrans: 888	k: 0.5	laborsMain: 177649	deviation: 444.344

9. Ввод:  $N = 20$ ,  $m = 1000$ ,  $R = 300$ ,  $V=0$ ,  $T = 1$ :

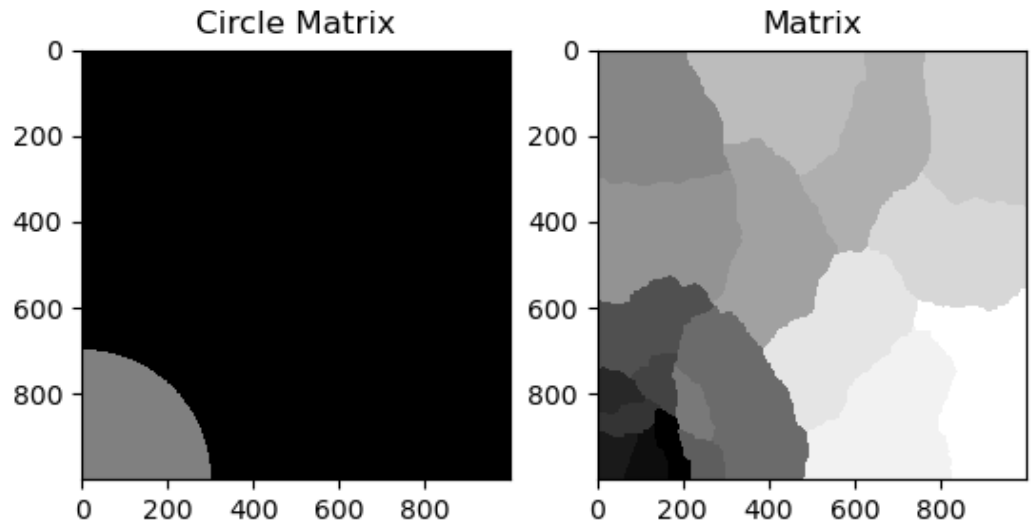
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 81936

Max deviation: 835.953

NumProc: 1	laborsGeneral: 81940	laborsTrans: 461	k: 0.5	laborsMain: 82170.5	deviation: 234.453
NumProc: 2	laborsGeneral: 81930	laborsTrans: 350	k: 0.5	laborsMain: 82105	deviation: 168.953
NumProc: 3	laborsGeneral: 81940	laborsTrans: 259	k: 0.5	laborsMain: 82069.5	deviation: 133.453
NumProc: 4	laborsGeneral: 81940	laborsTrans: 342	k: 0.5	laborsMain: 82111	deviation: 174.953
NumProc: 5	laborsGeneral: 81930	laborsTrans: 444	k: 0.5	laborsMain: 82152	deviation: 215.953
NumProc: 6	laborsGeneral: 81935	laborsTrans: 540	k: 0.5	laborsMain: 82205	deviation: 268.953
NumProc: 7	laborsGeneral: 81936	laborsTrans: 906	k: 0.5	laborsMain: 82389	deviation: 452.953
NumProc: 8	laborsGeneral: 81940	laborsTrans: 371	k: 0.5	laborsMain: 82125.5	deviation: 189.453
NumProc: 9	laborsGeneral: 81934	laborsTrans: 1275	k: 0.5	laborsMain: 82571.5	deviation: 635.453
NumProc: 10	laborsGeneral: 81935	laborsTrans: 566	k: 0.5	laborsMain: 82218	deviation: 281.953
NumProc: 11	laborsGeneral: 81936	laborsTrans: 777	k: 0.5	laborsMain: 82324.5	deviation: 388.453
NumProc: 12	laborsGeneral: 81936	laborsTrans: 1218	k: 0.5	laborsMain: 82545	deviation: 608.953
NumProc: 13	laborsGeneral: 81936	laborsTrans: 1672	k: 0.5	laborsMain: 82772	deviation: 835.953
NumProc: 14	laborsGeneral: 81936	laborsTrans: 1465	k: 0.5	laborsMain: 82668.5	deviation: 732.453
NumProc: 15	laborsGeneral: 81936	laborsTrans: 1063	k: 0.5	laborsMain: 82467.5	deviation: 531.453
NumProc: 16	laborsGeneral: 81936	laborsTrans: 704	k: 0.5	laborsMain: 82288	deviation: 351.953
NumProc: 17	laborsGeneral: 81936	laborsTrans: 1263	k: 0.5	laborsMain: 82567.5	deviation: 631.453
NumProc: 18	laborsGeneral: 81936	laborsTrans: 1584	k: 0.5	laborsMain: 82728	deviation: 791.953
NumProc: 19	laborsGeneral: 81936	laborsTrans: 1165	k: 0.5	laborsMain: 82518.5	deviation: 582.453
NumProc: 20	laborsGeneral: 81937	laborsTrans: 975	k: 0.5	laborsMain: 82424.5	deviation: 488.453

10. Ввод:  $N = 50$ ,  $m = 1000$ ,  $R = 300$ ,  $V = 400$ ,  $T = 1$ :

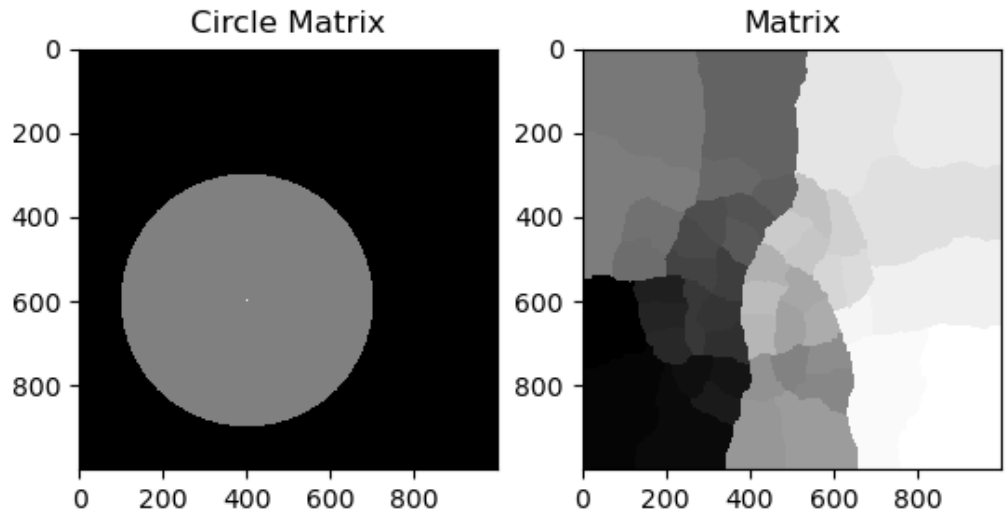
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 70881.9

Max deviation: 591.141

NumProc: 1	laborsGeneral: 70881	laborsTrans: 555	k: 0.5	laborsMain: 71158.5	deviation: 276.641
NumProc: 2	laborsGeneral: 70882	laborsTrans: 650	k: 0.5	laborsMain: 71207	deviation: 325.141
NumProc: 3	laborsGeneral: 70882	laborsTrans: 788	k: 0.5	laborsMain: 71276	deviation: 394.141
NumProc: 4	laborsGeneral: 70880	laborsTrans: 410	k: 0.5	laborsMain: 71085	deviation: 203.141
NumProc: 5	laborsGeneral: 70880	laborsTrans: 398	k: 0.5	laborsMain: 71079	deviation: 197.141
NumProc: 6	laborsGeneral: 70886	laborsTrans: 456	k: 0.5	laborsMain: 71114	deviation: 232.141
NumProc: 7	laborsGeneral: 70880	laborsTrans: 410	k: 0.5	laborsMain: 71085	deviation: 203.141
NumProc: 8	laborsGeneral: 70880	laborsTrans: 406	k: 0.5	laborsMain: 71083	deviation: 201.141
NumProc: 9	laborsGeneral: 70890	laborsTrans: 402	k: 0.5	laborsMain: 71091	deviation: 209.141
NumProc: 10	laborsGeneral: 70880	laborsTrans: 426	k: 0.5	laborsMain: 71093	deviation: 211.141
NumProc: 11	laborsGeneral: 70880	laborsTrans: 414	k: 0.5	laborsMain: 71087	deviation: 205.141
NumProc: 12	laborsGeneral: 70880	laborsTrans: 558	k: 0.5	laborsMain: 71159	deviation: 277.141
NumProc: 13	laborsGeneral: 70880	laborsTrans: 442	k: 0.5	laborsMain: 71101	deviation: 219.141
NumProc: 14	laborsGeneral: 70880	laborsTrans: 458	k: 0.5	laborsMain: 71109	deviation: 227.141
NumProc: 15	laborsGeneral: 70880	laborsTrans: 434	k: 0.5	laborsMain: 71097	deviation: 215.141
NumProc: 16	laborsGeneral: 70880	laborsTrans: 438	k: 0.5	laborsMain: 71099	deviation: 217.141
NumProc: 17	laborsGeneral: 70880	laborsTrans: 484	k: 0.5	laborsMain: 71122	deviation: 240.141
NumProc: 18	laborsGeneral: 70890	laborsTrans: 416	k: 0.5	laborsMain: 71098	deviation: 216.141
NumProc: 19	laborsGeneral: 70882	laborsTrans: 440	k: 0.5	laborsMain: 71102	deviation: 220.141
NumProc: 20	laborsGeneral: 70882	laborsTrans: 970	k: 0.5	laborsMain: 71367	deviation: 485.141
NumProc: 21	laborsGeneral: 70882	laborsTrans: 546	k: 0.5	laborsMain: 71155	deviation: 273.141
NumProc: 22	laborsGeneral: 70882	laborsTrans: 484	k: 0.5	laborsMain: 71124	deviation: 242.141
NumProc: 23	laborsGeneral: 70882	laborsTrans: 596	k: 0.5	laborsMain: 71180	deviation: 298.141
NumProc: 24	laborsGeneral: 70882	laborsTrans: 718	k: 0.5	laborsMain: 71241	deviation: 359.141
NumProc: 25	laborsGeneral: 70883	laborsTrans: 989	k: 0.5	laborsMain: 71377.5	deviation: 495.641
NumProc: 26	laborsGeneral: 70880	laborsTrans: 414	k: 0.5	laborsMain: 71087	deviation: 205.141



NumProc: 27	laborsGeneral: 70886	laborsTrans: 524	k: 0.5	laborsMain: 71148	deviation: 266.141
NumProc: 28	laborsGeneral: 70880	laborsTrans: 422	k: 0.5	laborsMain: 71091	deviation: 209.141
NumProc: 29	laborsGeneral: 70880	laborsTrans: 460	k: 0.5	laborsMain: 71110	deviation: 228.141
NumProc: 30	laborsGeneral: 70883	laborsTrans: 486	k: 0.5	laborsMain: 71126	deviation: 244.141
NumProc: 31	laborsGeneral: 70883	laborsTrans: 616	k: 0.5	laborsMain: 71191	deviation: 309.141
NumProc: 32	laborsGeneral: 70880	laborsTrans: 412	k: 0.5	laborsMain: 71086	deviation: 204.141
NumProc: 33	laborsGeneral: 70880	laborsTrans: 468	k: 0.5	laborsMain: 71114	deviation: 232.141
NumProc: 34	laborsGeneral: 70890	laborsTrans: 434	k: 0.5	laborsMain: 71107	deviation: 225.141
NumProc: 35	laborsGeneral: 70880	laborsTrans: 452	k: 0.5	laborsMain: 71106	deviation: 224.141
NumProc: 36	laborsGeneral: 70880	laborsTrans: 394	k: 0.5	laborsMain: 71077	deviation: 195.141
NumProc: 37	laborsGeneral: 70880	laborsTrans: 396	k: 0.5	laborsMain: 71078	deviation: 196.141
NumProc: 38	laborsGeneral: 70887	laborsTrans: 532	k: 0.5	laborsMain: 71153	deviation: 271.141
NumProc: 39	laborsGeneral: 70880	laborsTrans: 462	k: 0.5	laborsMain: 71111	deviation: 229.141
NumProc: 40	laborsGeneral: 70880	laborsTrans: 458	k: 0.5	laborsMain: 71109	deviation: 227.141
NumProc: 41	laborsGeneral: 70880	laborsTrans: 538	k: 0.5	laborsMain: 71149	deviation: 267.141
NumProc: 42	laborsGeneral: 70880	laborsTrans: 454	k: 0.5	laborsMain: 71107	deviation: 225.141
NumProc: 43	laborsGeneral: 70884	laborsTrans: 440	k: 0.5	laborsMain: 71104	deviation: 222.141
NumProc: 44	laborsGeneral: 70882	laborsTrans: 1182	k: 0.5	laborsMain: 71473	deviation: 591.141
NumProc: 45	laborsGeneral: 70882	laborsTrans: 1118	k: 0.5	laborsMain: 71441	deviation: 559.141
NumProc: 46	laborsGeneral: 70882	laborsTrans: 675	k: 0.5	laborsMain: 71219.5	deviation: 337.641
NumProc: 47	laborsGeneral: 70882	laborsTrans: 978	k: 0.5	laborsMain: 71371	deviation: 489.141
NumProc: 48	laborsGeneral: 70882	laborsTrans: 426	k: 0.5	laborsMain: 71095	deviation: 213.141
NumProc: 49	laborsGeneral: 70882	laborsTrans: 1059	k: 0.5	laborsMain: 71411.5	deviation: 529.641
NumProc: 50	laborsGeneral: 70882	laborsTrans: 662	k: 0.5	laborsMain: 71213	deviation: 331.141

11. Ввод:  $N = 100$ ,  $m = 1000$ ,  $R = 300$ ,  $V = 400$ ,  $T = 1$ :

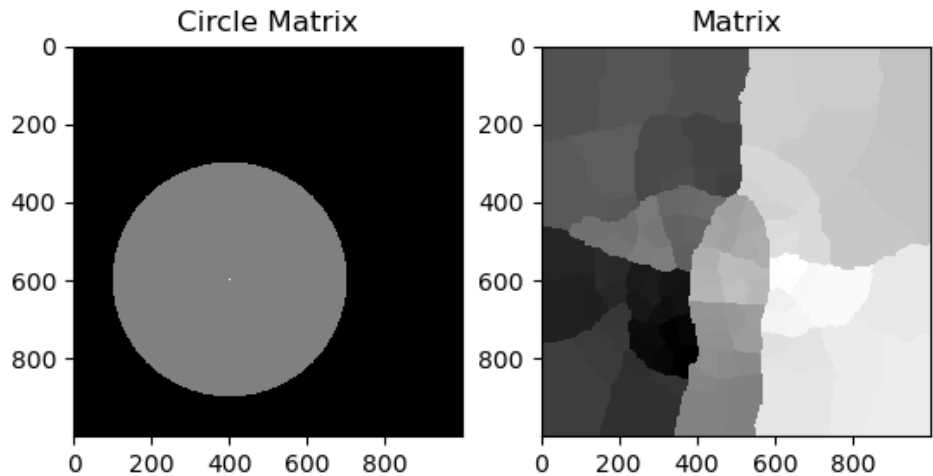
Вывод:

Маска круга:

Слишком большая.

Матрица распределения:

Слишком большая.



Average labor intensity: 35440.9

Max deviation: 529.07

NumProc: 1	laborsGeneral: 35440	laborsTrans: 320	k: 0.5	laborsMain: 35600	deviation: 159.07
NumProc: 2	laborsGeneral: 35440	laborsTrans: 344	k: 0.5	laborsMain: 35612	deviation: 171.07
NumProc: 3	laborsGeneral: 35440	laborsTrans: 298	k: 0.5	laborsMain: 35589	deviation: 148.07
NumProc: 4	laborsGeneral: 35440	laborsTrans: 280	k: 0.5	laborsMain: 35580	deviation: 139.07
NumProc: 5	laborsGeneral: 35440	laborsTrans: 324	k: 0.5	laborsMain: 35602	deviation: 161.07
NumProc: 6	laborsGeneral: 35440	laborsTrans: 352	k: 0.5	laborsMain: 35616	deviation: 175.07
NumProc: 7	laborsGeneral: 35440	laborsTrans: 324	k: 0.5	laborsMain: 35602	deviation: 161.07
NumProc: 8	laborsGeneral: 35440	laborsTrans: 322	k: 0.5	laborsMain: 35601	deviation: 160.07
NumProc: 9	laborsGeneral: 35440	laborsTrans: 310	k: 0.5	laborsMain: 35595	deviation: 154.07
NumProc: 10	laborsGeneral: 35440	laborsTrans: 278	k: 0.5	laborsMain: 35579	deviation: 138.07
NumProc: 11	laborsGeneral: 35440	laborsTrans: 292	k: 0.5	laborsMain: 35586	deviation: 145.07
NumProc: 12	laborsGeneral: 35450	laborsTrans: 348	k: 0.5	laborsMain: 35624	deviation: 183.07
NumProc: 13	laborsGeneral: 35441	laborsTrans: 320	k: 0.5	laborsMain: 35601	deviation: 160.07
NumProc: 14	laborsGeneral: 35442	laborsTrans: 514	k: 0.5	laborsMain: 35699	deviation: 258.07
NumProc: 15	laborsGeneral: 35442	laborsTrans: 425	k: 0.5	laborsMain: 35654.5	deviation: 213.57
NumProc: 16	laborsGeneral: 35440	laborsTrans: 298	k: 0.5	laborsMain: 35589	deviation: 148.07
NumProc: 17	laborsGeneral: 35440	laborsTrans: 282	k: 0.5	laborsMain: 35581	deviation: 140.07
NumProc: 18	laborsGeneral: 35440	laborsTrans: 308	k: 0.5	laborsMain: 35594	deviation: 153.07
NumProc: 19	laborsGeneral: 35440	laborsTrans: 318	k: 0.5	laborsMain: 35599	deviation: 158.07
NumProc: 20	laborsGeneral: 35441	laborsTrans: 571	k: 0.5	laborsMain: 35726.5	deviation: 285.57
NumProc: 21	laborsGeneral: 35441	laborsTrans: 360	k: 0.5	laborsMain: 35621	deviation: 180.07
NumProc: 22	laborsGeneral: 35440	laborsTrans: 290	k: 0.5	laborsMain: 35585	deviation: 144.07
NumProc: 23	laborsGeneral: 35440	laborsTrans: 356	k: 0.5	laborsMain: 35618	deviation: 177.07
NumProc: 24	laborsGeneral: 35442	laborsTrans: 550	k: 0.5	laborsMain: 35717	deviation: 276.07
NumProc: 25	laborsGeneral: 35443	laborsTrans: 468	k: 0.5	laborsMain: 35677	deviation: 236.07
NumProc: 26	laborsGeneral: 35440	laborsTrans: 274	k: 0.5	laborsMain: 35577	deviation: 136.07

NumProc: 27	laborsGeneral: 35442	laborsTrans: 680	k: 0.5	laborsMain: 35782	deviation: 341.07
NumProc: 28	laborsGeneral: 35440	laborsTrans: 278	k: 0.5	laborsMain: 35579	deviation: 138.07
NumProc: 29	laborsGeneral: 35441	laborsTrans: 374	k: 0.5	laborsMain: 35628	deviation: 187.07
NumProc: 30	laborsGeneral: 35442	laborsTrans: 634	k: 0.5	laborsMain: 35759	deviation: 318.07
NumProc: 31	laborsGeneral: 35440	laborsTrans: 332	k: 0.5	laborsMain: 35606	deviation: 165.07
NumProc: 32	laborsGeneral: 35441	laborsTrans: 682	k: 0.5	laborsMain: 35782	deviation: 341.07
NumProc: 33	laborsGeneral: 35441	laborsTrans: 425	k: 0.5	laborsMain: 35653.5	deviation: 212.57
NumProc: 34	laborsGeneral: 35441	laborsTrans: 708	k: 0.5	laborsMain: 35795	deviation: 354.07
NumProc: 35	laborsGeneral: 35440	laborsTrans: 320	k: 0.5	laborsMain: 35600	deviation: 159.07
NumProc: 36	laborsGeneral: 35441	laborsTrans: 670	k: 0.5	laborsMain: 35776	deviation: 335.07
NumProc: 37	laborsGeneral: 35442	laborsTrans: 786	k: 0.5	laborsMain: 35835	deviation: 394.07
NumProc: 38	laborsGeneral: 35440	laborsTrans: 298	k: 0.5	laborsMain: 35589	deviation: 148.07
NumProc: 39	laborsGeneral: 35440	laborsTrans: 268	k: 0.5	laborsMain: 35574	deviation: 133.07
NumProc: 40	laborsGeneral: 35450	laborsTrans: 290	k: 0.5	laborsMain: 35595	deviation: 154.07
NumProc: 41	laborsGeneral: 35440	laborsTrans: 280	k: 0.5	laborsMain: 35580	deviation: 139.07
NumProc: 42	laborsGeneral: 35440	laborsTrans: 288	k: 0.5	laborsMain: 35584	deviation: 143.07
NumProc: 43	laborsGeneral: 35440	laborsTrans: 312	k: 0.5	laborsMain: 35596	deviation: 155.07
NumProc: 44	laborsGeneral: 35440	laborsTrans: 310	k: 0.5	laborsMain: 35595	deviation: 154.07
NumProc: 45	laborsGeneral: 35442	laborsTrans: 422	k: 0.5	laborsMain: 35653	deviation: 212.07
NumProc: 46	laborsGeneral: 35440	laborsTrans: 314	k: 0.5	laborsMain: 35597	deviation: 156.07
NumProc: 47	laborsGeneral: 35440	laborsTrans: 298	k: 0.5	laborsMain: 35589	deviation: 148.07
NumProc: 48	laborsGeneral: 35440	laborsTrans: 326	k: 0.5	laborsMain: 35603	deviation: 162.07
NumProc: 49	laborsGeneral: 35440	laborsTrans: 314	k: 0.5	laborsMain: 35597	deviation: 156.07
NumProc: 50	laborsGeneral: 35440	laborsTrans: 274	k: 0.5	laborsMain: 35577	deviation: 136.07
NumProc: 51	laborsGeneral: 35441	laborsTrans: 454	k: 0.5	laborsMain: 35668	deviation: 227.07
NumProc: 52	laborsGeneral: 35442	laborsTrans: 462	k: 0.5	laborsMain: 35673	deviation: 232.07
NumProc: 53	laborsGeneral: 35440	laborsTrans: 308	k: 0.5	laborsMain: 35594	deviation: 153.07
NumProc: 54	laborsGeneral: 35440	laborsTrans: 286	k: 0.5	laborsMain: 35583	deviation: 142.07
NumProc: 55	laborsGeneral: 35440	laborsTrans: 262	k: 0.5	laborsMain: 35571	deviation: 130.07
NumProc: 56	laborsGeneral: 35440	laborsTrans: 276	k: 0.5	laborsMain: 35578	deviation: 137.07
NumProc: 57	laborsGeneral: 35440	laborsTrans: 294	k: 0.5	laborsMain: 35587	deviation: 146.07
NumProc: 58	laborsGeneral: 35440	laborsTrans: 306	k: 0.5	laborsMain: 35593	deviation: 152.07
NumProc: 59	laborsGeneral: 35440	laborsTrans: 322	k: 0.5	laborsMain: 35601	deviation: 160.07
NumProc: 60	laborsGeneral: 35440	laborsTrans: 306	k: 0.5	laborsMain: 35593	deviation: 152.07
NumProc: 61	laborsGeneral: 35440	laborsTrans: 306	k: 0.5	laborsMain: 35593	deviation: 152.07
NumProc: 62	laborsGeneral: 35450	laborsTrans: 304	k: 0.5	laborsMain: 35602	deviation: 161.07
NumProc: 63	laborsGeneral: 35440	laborsTrans: 314	k: 0.5	laborsMain: 35597	deviation: 156.07
NumProc: 64	laborsGeneral: 35440	laborsTrans: 320	k: 0.5	laborsMain: 35600	deviation: 159.07
NumProc: 65	laborsGeneral: 35440	laborsTrans: 308	k: 0.5	laborsMain: 35594	deviation: 153.07
NumProc: 66	laborsGeneral: 35440	laborsTrans: 302	k: 0.5	laborsMain: 35591	deviation: 150.07
NumProc: 67	laborsGeneral: 35440	laborsTrans: 280	k: 0.5	laborsMain: 35580	deviation: 139.07
NumProc: 68	laborsGeneral: 35440	laborsTrans: 336	k: 0.5	laborsMain: 35608	deviation: 167.07
NumProc: 69	laborsGeneral: 35440	laborsTrans: 272	k: 0.5	laborsMain: 35576	deviation: 135.07
NumProc: 70	laborsGeneral: 35440	laborsTrans: 278	k: 0.5	laborsMain: 35579	deviation: 138.07
NumProc: 71	laborsGeneral: 35440	laborsTrans: 290	k: 0.5	laborsMain: 35585	deviation: 144.07

NumProc: 72	laborsGeneral: 35440	laborsTrans: 268	k: 0.5	laborsMain: 35574	deviation: 133.07
NumProc: 73	laborsGeneral: 35440	laborsTrans: 302	k: 0.5	laborsMain: 35591	deviation: 150.07
NumProc: 74	laborsGeneral: 35450	laborsTrans: 300	k: 0.5	laborsMain: 35600	deviation: 159.07
NumProc: 75	laborsGeneral: 35440	laborsTrans: 280	k: 0.5	laborsMain: 35580	deviation: 139.07
NumProc: 76	laborsGeneral: 35440	laborsTrans: 513	k: 0.5	laborsMain: 35696.5	deviation: 255.57
NumProc: 77	laborsGeneral: 35441	laborsTrans: 846	k: 0.5	laborsMain: 35864	deviation: 423.07
NumProc: 78	laborsGeneral: 35441	laborsTrans: 980	k: 0.5	laborsMain: 35931	deviation: 490.07
NumProc: 79	laborsGeneral: 35441	laborsTrans: 1058	k: 0.5	laborsMain: 35970	deviation: 529.07
NumProc: 80	laborsGeneral: 35441	laborsTrans: 622	k: 0.5	laborsMain: 35752	deviation: 311.07
NumProc: 81	laborsGeneral: 35441	laborsTrans: 838	k: 0.5	laborsMain: 35860	deviation: 419.07
NumProc: 82	laborsGeneral: 35441	laborsTrans: 314	k: 0.5	laborsMain: 35598	deviation: 157.07
NumProc: 83	laborsGeneral: 35441	laborsTrans: 296	k: 0.5	laborsMain: 35589	deviation: 148.07
NumProc: 84	laborsGeneral: 35441	laborsTrans: 732	k: 0.5	laborsMain: 35807	deviation: 366.07
NumProc: 85	laborsGeneral: 35443	laborsTrans: 574	k: 0.5	laborsMain: 35730	deviation: 289.07
NumProc: 86	laborsGeneral: 35440	laborsTrans: 310	k: 0.5	laborsMain: 35595	deviation: 154.07
NumProc: 87	laborsGeneral: 35440	laborsTrans: 296	k: 0.5	laborsMain: 35588	deviation: 147.07
NumProc: 88	laborsGeneral: 35440	laborsTrans: 286	k: 0.5	laborsMain: 35583	deviation: 142.07
NumProc: 89	laborsGeneral: 35440	laborsTrans: 616	k: 0.5	laborsMain: 35748	deviation: 307.07
NumProc: 90	laborsGeneral: 35442	laborsTrans: 641	k: 0.5	laborsMain: 35762.5	deviation: 321.57
NumProc: 91	laborsGeneral: 35441	laborsTrans: 680	k: 0.5	laborsMain: 35781	deviation: 340.07
NumProc: 92	laborsGeneral: 35441	laborsTrans: 924	k: 0.5	laborsMain: 35903	deviation: 462.07
NumProc: 93	laborsGeneral: 35442	laborsTrans: 499	k: 0.5	laborsMain: 35691.5	deviation: 250.57
NumProc: 94	laborsGeneral: 35440	laborsTrans: 292	k: 0.5	laborsMain: 35586	deviation: 145.07
NumProc: 95	laborsGeneral: 35440	laborsTrans: 264	k: 0.5	laborsMain: 35572	deviation: 131.07
NumProc: 96	laborsGeneral: 35443	laborsTrans: 302	k: 0.5	laborsMain: 35594	deviation: 153.07
NumProc: 97	laborsGeneral: 35442	laborsTrans: 364	k: 0.5	laborsMain: 35624	deviation: 183.07
NumProc: 98	laborsGeneral: 35443	laborsTrans: 756	k: 0.5	laborsMain: 35821	deviation: 380.07
NumProc: 99	laborsGeneral: 35440	laborsTrans: 318	k: 0.5	laborsMain: 35599	deviation: 158.07
NumProc: 100	laborsGeneral: 35440	laborsTrans: 310	k: 0.5	laborsMain: 35595	deviation: 154.07

## СПИСОК ЛИТЕРАТУРЫ

1. *Якобовский М.В.* Параллельные вычисления: рациональная декомпозиция сеточных графов
2. *Корнилина М.А., Якобовский М.В.* Оценка накладных расходов при выполнении расчётов на локально измельчаемых сетках // Препринты ИПМ им. М.В.Келдыша. 2022. № 102. 36 с.
3. *George Karypis* METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices // URL: <https://usermanual.wiki/Pdf/manual.588322308/html#pf1a>