

## **GeNoC – программная реализация метода эволюционных вычислений для синтеза квазиоптимальных топологий сетей на кристалле**

GeNoC – реализация метода эволюционных вычислений в виде генетического алгоритма. Применяется для ускоренного синтеза квазиоптимальных топологий сетей на кристалле с количеством узлов достигающим сотен. Реализация алгоритма проводилась в среде математического проектирования Matlab7.13.0.564 (R2011b).

### **Принципы работы GeNoC**

В основе работы алгоритма лежит описание топологии сети на кристалле в виде матрицы смежности. Такие матрицы смежности являются симметричными, и содержат на главной диагонали нулевые элементы. Это дает возможность использовать верхнюю правую значимую часть матрицы, представляющую из себя последовательность  $N*(N-1)/2$  бит в виде хромосомы и описывает определенный экземпляр-образец топологии возможных соединений СтнК с N узлами.

### **Принцип работы GeNoC**

В начале алгоритма случайным образом генерируется начальная популяция хромосом-топологий. Для улучшения результата в начальную популяцию, кроме случайных хромосом, можно включать известные реализации регулярных топологий для заданного количества узлов.

Каждая хромосома-топология оценивается на приспособленность с помощью целевой функции. Хромосомы, которые описывают несвязные графы, исключаются сразу. Другие хромосомы-топологии оцениваются с помощью интегральной целевой функции, которая учитывает характеристики диаметра, среднего расстояния, максимальной степени вершин и количества соединений в зависимости от их значимости в соответствии с критерием оптимальности.

Отобранная часть хромосом-топологий участвует в создании новой популяции хромосом-топологий путем применения операций мутации, кроссовера (скрещивания) и селекции.

Полученная новая популяция участвует в следующем цикле генетического алгоритма. Условием остановки может быть исчерпание заданного количества поколений или отсутствие улучшения результата в течение заданного количества поколений.

## **Структура проекта GeNoC**

GeNoC состоит из m-файлов:

**GA\_startEvolution.m** – точка входа в проект, данная функция запускает выполнение эволюционных вычислений и содержит следующие настройки:

**LOAD\_FILE** – возможность указать название ранее сохраненного файла для продолжения обучения.

**SAVE\_EVERY** – указывает частоту сохранения результатов работы алгоритма (задается количеством эволюций).

**GR\_CreateGraphStruct.m** – создание структуры, которая описывает граф топологии;

Поля структуры имеют следующий смысл:

struct.Matrix – матрица связей;

struct.DG – матрица связей с весами (directedgraph);

struct.Vector – вектор, представляющий верхнюю часть матрицы связей;

struct.EdgeCount – количество ребер;

struct.MedDist – среднее минимальное расстояние;

struct.Diameter – диаметр графа;

struct.Power – максимальная степень вершины графа;

struct.Size – количество вершин графа;

struct.Goal – целевая функция, характеризующая качество графа.

GR\_CreateSparse.m – функция создания матрицы смежности;

GR\_Diameter.m – нахождение диаметра графа;

GR\_EdgesCount.m – нахождение количества ребер графа;

GR\_MaxGraphPower.m – нахождение максимальной степени вершин графа;

GR\_Size.m – нахождение размера графа;

GR\_medianDistance.m – нахождение минимального среднего расстояния между вершинами графа;

GR\_View.m – визуальное отображение графа;

GR\_WorstMat.m – создание графа с худшей конфигурацией;

GR\_mat2vector.m – преобразование матрицы смежности в хромосому;

GR\_vector2mat.m – восстановление матрицы смежности из хромосомы;

GA\_CostF\_MDistMDiamMEdge\_.m – целевая функция;

GR\_Filter.m – фильтрация графов по параметрам в GR\_OPTIONS (отбрасывание заведомо неоптимальных топологий);

GR\_rand.m – генерация случайного числа в заданном диапазоне;

GA\_FilterTwins.m – удаление одинаковых экземпляров из популяции;

GA\_FindBestIndivid.m – поиск лучших экземпляров в заданной популяции;

GA\_GenerateStartPop.m – генерация начальной популяции;

GA\_Crossover.m – генетический оператор кроссовера;  
GA\_Inversion.m – генетический оператор инверсии;  
GA\_Mutation.m – генетический оператор мутации;  
GA\_SaveResult.m – сохранение результатов поиска;  
GA\_startEvolution.m – скрипт запуска генетического алгоритма;  
Shuffle.m, Shuffle.c, Shuffle.mexw32, Shuffle.mexw64 – файлы необходимые для работы GA\_GenerateStartPop.m;  
init\_gr.m – файл настройки параметров генетического поиска.

Остальные файлы являются вспомогательными.

### **Содержимое файла настроек init\_gr.m**

init\_gr.m содержит следующие переменные настроек:

grSize – задает размерность размер графа (N);

grPowLim – максимальный степень вершин графа;

grEdgesLim – ограничение по максимальному количеству ребер;

grDiamLim – максимальный диаметр графа;

BEST\_START – процент лучших индивидов с начальной популяции, будут отобраны для следующего цикла;

BEST\_GENERATED – процент лучших индивидов с созданной популяции, будет выбрано в новой;

WORST\_GENERATED – процент индивидов которые будут выбраны случайно из созданной новой популяции;

start\_pop\_c – размер начальной популяции;

generated\_pop\_c – размер популяции которая будет генерироваться;

EPOCH\_C – количество циклов эволюции;

CROSSOVER\_GENERATE – количество индивидов которое будет генерироваться кроссовером;

MUTATION\_CHANCE – вероятность мутации;

INVERSION\_CHANCE – вероятность инверсии.

## **Параметры целевой функции**

Выражение для нахождения целевой функции имеет вид:

costFRes = ...

options.diam \* (graphStruct.Diameter / MAX\_D) ...

options.edgeC \* (graphStruct.EdgeCount / MAX\_E) ...

options.medDs \* (graphStruct.MedDist / MAX\_DS) ...

options.power \* (graphStruct.Power / MAX\_P);

Целевая функция учитывает 4 характеристики графа (диаметр, степень, минимальное расстояние, количество ребер), для улучшения поиска оптимальной

конфигурации графа по определенному параметру необходимо увеличить его коэффициент важности, что можно сделать изменив соответствующее значение в следующей структуре (для исключения воздействия необходимо указать 0).

```
COST_F_OPT = struct (...
```

```
'power', 0.1, ...
```

```
'diam', 0.1, ...
```

```
'medDs', 0.7, ...
```

```
'edgeC', 0.1 ...
```

```
);
```

Рекомендуется выбирать коэффициенты таким образом, чтобы их сумма равнялась 1.

## **Параметры структуры GR\_OPTIONS**

В структуре GR\_OPTIONS содержатся верхнее и нижнее ограничения по диаметру, количества ребер и степени вершин, необходимых для отбрасывания заведомо неоптимальных топологий.

```
GR_OPTIONS = struct (...
```

```
'dl', [], ...
```

```
'de', [], ...
```

```
'dg', [], ...
```

```
'pl', [], ...
```

'pe', [], ...

'pg', [], ...

'el', [], ...

'ee', [], ...

'eg', [] ...

);

d – ограничение диаметра;

e – ограничение количества ребер;

p – ограничение максимального степеня;

l – параметр должен быть меньше указанного;

p – параметр должен быть больше чем указанный;

e – параметр должен быть равным указанному.

## **Запуск синтеза топологии с заданными параметрами**

Для запуска работы генетического алгоритма необходимо:

- 1) Изменить CurrentFolder на папку с проектом;
- 2) Открыть файл init\_gr.m и выполнить необходимые настройки;



3) Открыть файл GA\_startEvolution.m, указать настройки и нажать клавишу F5.

4) Для продолжения синтеза в файле GA\_startEvolution.m необходимо переменной LOAD\_FILE указать путь к файлу с предварительным результатам синтеза.

```
LOAD_FILE = 'd: \ .. \ 100_evolution.mat';
```

## **Отображение результатов работы GeNoC**

Во время работы алгоритма в командной строке будет отображаться информация о текущем состоянии эволюции, например:

Generation of start population

```
=====
=====
```

Start of genetic algorithm

```
-----
```

1 evolution:

Goal function: 8.705474e-01

Evolution time is 6.327844 seconds

```
-----
```

2 evolution:

Goal function: 8.685965e-01

Evolution time is 6.065170 seconds

...

100 evolution:

Goal function: 6.863480e-01

Evolution time is 6.141921 seconds

=====  
===

Genetic evolution finished

Number of evolutions: 100

Time 615.849595 seconds

Goal function: 6.863480e-01

Vertex Count: 64

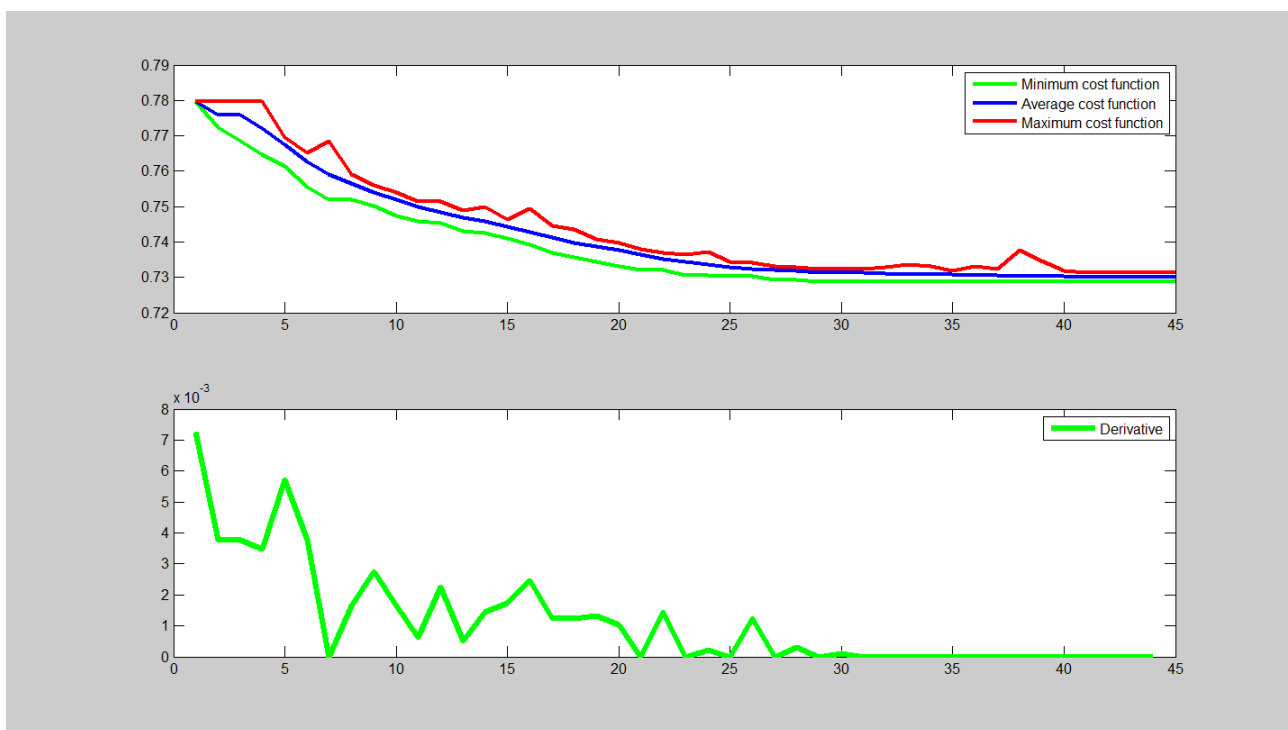
Diameter: 5

Edge Count: 119

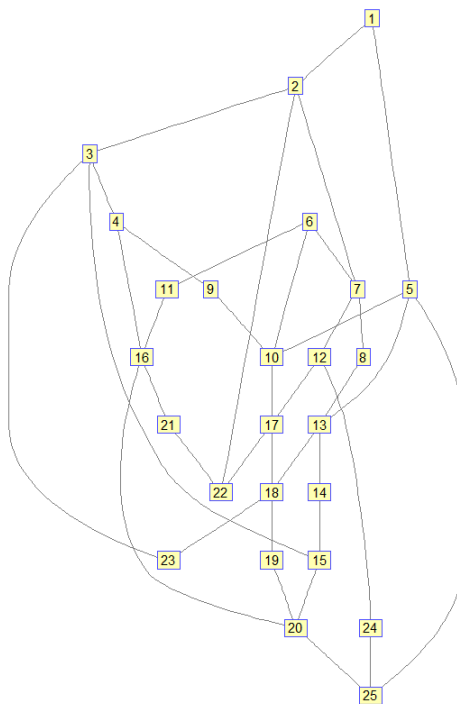
Max power: 4

Median distance: 3.301742e +00

Ход синтеза отображается графически в виде значения целевой функции и изменения целевой функции между двумя итерациями.



Полученная топология отображается в виде графа:



Сохранение результатов синтеза

Результаты синтеза сохраняются в папке, название которой состоит из количества вершин графа, даты и времени синтеза, например,

25\_11-Jan-2014 1\_00 AM

или

25\_11-Jan-2014 1\_46 AM

В папке содержатся файлы:

graph.bmp – изображение графа;

evolution.bmp – изображение графиков изменения значений целевой функции и изменения целевой функции при синтезе;

Matrix.xls – матрица смежности полученного графа;

param.txt – параметры полученного графа;

grStruct.mat – расширенные параметры полученного графа, представленные в формате Matlab;

\_evolution.mat – файлы промежуточных результатов. Их можно указать в качестве исходных, для следующей итерации синтеза.

## **@ Copyright**

Разработчики проекта:

Автор проекта - Романов Александр Юрьевич, к.т.н.,  
доц. МИЭМ НИУ ВШЭ (г. Москва).

Все материалы предоставляются на основе GNU General Public License (GPL).

Разработчики не предоставляют ни каких гарантий относительно правильности работы модели или пригодности для конкретной задачи.