

# ИССЛЕДОВАНИЕ И РАЗРАБОТКА ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ПОСТРОЕНИЯ ТЕСТОВ ДЛЯ ЦИФРОВЫХ СХЕМ

Ярославцев В.А.

ДонНТУ

Факультет компьютерных информационных технологий и автоматизации

E-mail: [HraniteIEFA\\_dntu@mail.ru](mailto:HraniteIEFA_dntu@mail.ru)

## **Abstract:**

*Yaroslavtsev V.A. Research and development genetical algorithms of construction of the tests for digital circuits. Clause describes use of genetic algorithms for construction of checking tests of digital schemes. The program which is created builds sets of checking sequences, using three basic operators of a reproduction, crossing and mutations.*

## **Введение**

На сегодняшний день исследования в области эволюционных вычислений является одной из самых перспективных и динамично развивающихся отраслей знаний в мире. То, что впервые описал Гольдберг на основе работ Холланда, теперь нашло своё признание во многих областях науки, организовав самостоятельное направление - эволюционные вычисления. Применение генетических алгоритмов как одного из самых интересных понятий уже привело науку к невероятным результатам. Так в Стэнфордском университете создана компьютерная система способная самостоятельно делать изобретения.

Система основана на использовании генетических алгоритмов. Сначала создаётся "среда", образованная множеством правил. Все вместе они определяют задачу, которую придётся решить. В этот искусственный мир подселяются много сот тысяч "обитателей" - сгенерированных случайным образом программ, каждая из которых соответствует одному из подходов к решению задачи. Затем они начинают "жить": программы скрещиваются, мутируют и отбраковываются поколение за поколением.

Как и при настоящей эволюции, в итоге выживают сильнейшие - программы, определяющие наилучшие решения задачи. Поскольку система использует для работы мощный тысячеузловый кластер, принадлежащего компании Genetic Programming, сроки вычислений вполне разумны.

Хотя сами эти программы и их "мир" сделаны по образу и подобию примитивных одноклеточных существ, результаты, которые даёт система, на удивление разумны. К настоящему моменту система совершенно независимо повторила несколько настоящих изобретений, запатентованных в Америке в период с 1917 по 1974 год.

Цель нашей работы заключается в создании компьютерной системы которая на базе выбора и использования наиболее эффективных и оптимальных методов обработки данных при помощи генетических алгоритмов, сможет продуцировать построение проверяющих тестов для цифровых схем.

Актуальность данной темы диктуется быстрым темпом роста производства цифровых схем с всё более миниатюрными размерами и с всё большим количеством деталей, таких как транзисторы. Но как известно: Количество не всегда значит качество. Создавая всё новые и новые схемы, производители уделяют недостаточно сил, времени и средств, на проверку и до усовершенствование своих изделий, теряя при этом бесчисленные материальные ресурсы. Создание системы, компьютерной системы, которая бы в течении не длительного промежутка времени могла бы с достаточно высокой точностью проверять цифровые схемы при помощи построенных ею проверяющих тестов, могла бы намного сократить затраты производителей на изготовление комплектующих, а следовательно и дала бы возможность снизить стоимость данной продукции, что без сомнения отразилось бы и на конечных пользователях.

Рассмотрим, что же такое генетический алгоритм и каковы его реальные и перспективные возможности построения тестов для цифровых схем.

Генетические алгоритмы (ГА) – алгоритмы поиска, построенные на принципах, сходных с принципами естественного отбора.

Классический “простой” ГА использует двоичные строки для кодирования решения проблемы – особи. Это делает его привлекательным для использования в задачах генерации проверяющих тестов логических схем, где решение задачи представляется в виде двоичных наборов или их последовательностей. На множестве решений определяется целевая (fitness) функция (ЦФ), которая позволяет оценить близость каждой особи к оптимальному решению. Простой ГА использует три основных оператора: репродукция, кроссинговер-скрещивание, мутация. Используя данные операторы, популяция (множество решений данной проблемы)

эволюционирует от поколения к поколению. Таким образом, чтобы задать ГА для решения конкретной проблемы, необходимо определить понятия особи, популяции, операций скрещивания и мутации, задать оценочную функцию.

Предварительно простой ГА случайным образом генерирует начальную популяцию стрингов (хромосом). Затем алгоритм генерирует следующее поколение (популяцию), с помощью трех основных генетических операторов:

- 1) Оператор репродукции (ОР);
- 2) Оператор скрещивания (кроссинговера, ОК);
- 3) Оператор мутации (ОМ).

Рассмотрим кратко каждый из этих операторов рассмотрев процессы которым данные операторы соответствуют.

#### Репродукция

Репродукция – это процесс, в котором хромосомы копируются в промежуточную популяцию согласно их значениям целевой (фитнес-) функции. При этом хромосомы с лучшими значениями целевой функции имеют большую вероятность попадания одного или более потомков в следующее поколение.

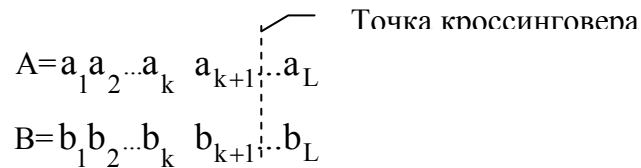
Очевидно, оператор репродукции является искусственной версией натуральной селекции – выживания сильнейших по Дарвину. Этот оператор представляется в алгоритмической форме различными способами. Самый простой (и популярный) метод реализации ОР – построение колеса рулетки, в которой каждая хромосома имеет сектор, пропорциональный ее значению ЦФ.

Для селекции хромосом используем случайный поиск на основе колеса рулетки. При этом колесо рулетки вращается и после останова ее указатель определяет хромосому для селекции в промежуточную популяцию. Очевидно, что хромосома, которой соответствует больший сектор рулетки, имеет большую вероятность попасть в следующее поколение. В результате выполнения оператора репродукции формируется промежуточная популяция, хромосомы которой будут использованы для построения поколения с помощью операторов скрещивания.

#### **Оператор кроссинговера (скрещивания)**

Одноточечный или простой оператор кроссинговера (ОК) с заданной вероятностью  $P_{ok}$  выполняется в 3 этапа:

### 1-й этап. Две хромосомы



выбираются случайно (или одним из методов, рассмотренных в разделе «современные модификации и обобщения ГА») из промежуточной популяции, сформированной при помощи оператора репродукции (ОР). Иногда 1-й этап может выполняться и без ОР.

2-й этап. Случайно выбирается точка скрещивания - число  $k$  из диапазона  $[1, 2 \dots L-1]$ , где  $L$  – длина хромосомы (число бит в двоичном коде)

3-й этап. Две новых хромосомы  $A'$ ,  $B'$  формируются из  $A$  и  $B$  путем обмена подстроками после точки скрещивания:

$$A' = a_1 a_2 \dots a_k b_{k+1} \dots b_L$$

$$B' = b_1 b_2 \dots b_k a_{k+1} \dots a_L$$

Следует отметить, что ОК выполняется с заданной вероятностью  $P_{ок}$  (отобранные два родителя не обязательно производят потомков !!!). Обычно величина  $P_{ок} = 0.5$ .

Т.о. операторы репродукции и скрещивания очень просты – они выполняют копирование особей и частичный обмен частей хромосом.

### Мутация

Далее согласно схеме классического ГА с заданной вероятностью  $P_M$  выполняется оператор мутации. Иногда этот оператор играет вторичную роль. Обычно вероятность мутации  $P_M = 0,001$ .

Оператор мутации (ОМ) выполняется в 2 этапа:

1-й этап. В хромосоме  $A = a_1 a_2 \dots a_L$  случайно выбирается  $k$ -позиция (бит) мутации ( $1 \leq k \leq L$ ).

2-й этап. Производится инверсия значения гена в  $k$ -й позиции.

$$\underline{a'_k} = \bar{a}_k$$

В общем случае ГА работает до тех пор, пока не будет выполнено заданное количество генераций или на некоторой генерации будет получено заданного качества, или

возникает преждевременная сходимость, когда найден локальный оптимум и ГА не может найти из него выход. В отличие от других методов оптимизации ГА, как правило, анализирует различные области пространства решений одновременно и более приспособлены к нахождению новых областей с лучшими значениями целевой функции за счёт объединения квазиоптимальных решений из разных популяций

Генетические алгоритмы отличаются от других оптимизирующих и поисковых процедур следующими особенностями:

- 1) Работают не с параметрами, а с закодированным множеством параметров.
- 2) Осуществляет поиск из популяции точек, а не из единственной точки.
- 3) Использует целевую функцию непосредственно, а не ее непосредственное приращение.
- 4) Использует не детерминированные, а вероятностные правила.

#### ГА в генерации тестов

Суть задачи построения тестов заключается в поиске двоичной входной последовательности, которая для каждой неисправности из заданного множества дает различные выходные значения сигналов в исправной и неисправной схемах. На первом этапе ГА применялись для генерации тестов комбинационных ЦС. Поскольку значения на выходных линиях комбинационной ЦС зависят только от значений на внешних входах и не зависят от внутреннего состояния схемы, в качестве особи выступает одиночный двоичный тестовый набор. Группа тестовых наборов образует популяцию. К выбранным таким образом особям применимы стандартные операторы кроссинговера и мутации ГА.

Поскольку целью генерации тестов является построение последовательности, на которой максимально отличаются значения сигналов в исправной и неисправной схемах, то качество тестовой последовательности (fitness-функция) оценивается как мера отличия значений сигналов в исправной и неисправной схемах. В простейшем случае для этого используются программы логического моделирования исправных цифровых схем, позволяющие оценить значения сигналов на двух соседних (во

времени) тестовых наборах. На основе полученных данных моделирования разработаны fitness-функции следующего вида:

$$h(v, f) = c_1 f_1(v, f) + c_2 * f_2(v, f),$$

где  $f_1$  - и  $f_2$  - число изменений сигналов на выходах логических вентилях и триггеров соответственно ( $c_2 \gg c_1$ ). [2,3]. Целевая функция для последовательности наборов определяется как взвешенная сумма фитнес-функций отдельных наборов

$$H(s, f) = \sum_{i=1}^{i=\text{длина}} L^i * h(v_i, f),$$

где  $s$  - анализируемая последовательность;  $v_i$  - вектор из рассматриваемой последовательности,  $i$  – позиция вектора в последовательности,  $f$  - заданная неисправность,  $0 < L < 1$ .

Если не удастся построить тест с использованием целевой функции указанного типа, то применяются программы моделирования неисправных цифровых схем, требующие большие машинные ресурсы. Целевые функции при этом основаны на подсчете различий сигналов в исправной и неисправной схемах и имеют примерно такой же вид как и в предыдущем случае.

## **Заключение**

Эволюционные вычисления и генетические алгоритмы как часть их – одна из самых перспективных и быстро развивающихся наук, которая уже в начале своего пути начинает приносить определённые плоды в той области знаний в которой применяется.

Создание компьютерных систем которые будут использовать генетические алгоритмы в своей основе является одной из важнейших задач перед научным сообществом и человечеством в целом.

Приминительно к теме нашей магистерской работы генетические алгоритмы и их методы и операторы будут являться основным ядром построения компьютерной системы способной создавать проверяющие тесты для цифровых схем.

## **Литература**

1. Д. -Э. Бэстенс, В. .М. Ван Ден Берг, Д. Вуд. .Нейронные сети и финансовые рынки..., Москва, научное издательство .ТВП., 1997.

2. Галушкин А. И. .Нейрокомпьютеры и их применение. Книга 1. Теория нейронных сетей.. Москва, Издательское предприятие редакции журнала .Радиотехника., 2000.
3. Тейво Кохонен, Гвидо Дебок .Анализ финансовых данных с помощью самоорганизующихся карт., Москва, издательский дом .Альпина., 2001.  
Ф. Уоссерман. .Нейрокомпьютерная техника., Москва, издательство .Мир., 1992.
4. Шумский С. А. .Нейрокомпьютинг и его применение в экономике и бизнесе., Москва, издательство МИФИ, 1998.
5. А. И. Змитрович Интеллектуальные информационные системы. - Минск.: НТООО "Тетра Системс", 1997. - 368с.
6. В. В. Корнеев, А. Ф. Гарев, С. В. Васютин, В. В. Райх Базы данных. Интеллектуальная обработка информации. - М.: "Нолидж", 2000. - 352с.
7. С.А. Закусило, Д.Е. Иванов, В.Ю. Скобцов.. Эволюционный подход к генерации проверяющих тестов цифровых схем (статья) Труды конференций «Интеллектуальные САПР». – Москва: Физматлит. – 2003 – С.76-78.