

СЖАТИЕ СПИСКА НЕИСПРАВНОСТЕЙ С ПОМОЩЬЮ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

161-167

Иванов Д.Е., к.т.н., доц., кафедра АСУ, Донецкий государственный
технический университет, Скобцов В.Ю., к.т.н., н.с. отдела ТУС,
Институт прикладной математики и механики НАН Украины
E-mail: ivanov@iamm.ac.donetsk.ua, skobtsov@iamm.ac.donetsk.ua

Предложен алгоритм сжатия списка неисправностей для синхронных последовательностных схем, основанный на генетической стратегии.

An algorithm for the reduction of fault list that is used in test generation procedure for sequential digital circuits is suggested. For solving this problem is adopted genetic approach.

Введение. В последнее время широкое распространение получили методы генерации тестов, основанные на моделировании [1]. Их широкое распространение стало возможным благодаря созданию алгоритмов моделирования неисправных цифровых схем с низкими временными затратами типа PROOFS и подобных ему [2-3]. В генераторах тестов, основанных на моделировании, происходит псевдослучайная генерация тестовых последовательностей, а затем проверка их качества путём моделирования с неисправностями. Даже для цифровых устройств средней размерности число моделируемых неисправностей может быть достаточно велико, что приводит к чрезмерным затратам времени и памяти при работе таких программ генерации тестов. Одним из способов преодоления указанного недостатка является применение методов параллельного моделирования [4]. Другим возможным решением является уменьшение числа неисправностей, которые необходимо моделировать в процессе построения теста. Это возможно благодаря выбору таких неисправностей, тестовые последовательности для которых проверяют большое количество неисправностей, не входящих в группу моделируемых неисправностей.

Предлагается метод формирования списка моделируемых константных одиночных неисправностей, в котором учитываются топо-

логические свойства схемы и показатели тестируемости. Процедура формирования основана на применении генетического алгоритма.

Генетические алгоритмы. Предложенный Голдбергом в [5] генетический подход хорошо зарекомендовал себя при решении широкого спектра NP полных комбинаторных задач. В широком смысле под термином «генетический алгоритм» понимается алгоритм поиска, который основан на эволюционном моделировании популяции особей, которая является выборкой точек в пространстве поиска, с использованием операций селекции и рекомбинации для построения новой выборки. Опишем кратко классическую реализацию генетического алгоритма.

Пусть задана произвольная задача и её пространство поиска, в котором будет производиться поиск. Решение задачи с помощью генетического алгоритма предполагает, что любая точка в пространстве решений может быть закодирована как строка битов (x_1, x_2, \dots, x_n) , называемая особью или хромосомой. Битовые переменные $x_i, i = \overline{1, n}$ называются генами. Некоторый набор особей называется популяцией. Каждой особи ставится в соответствие значение некоторой оценочной функции $f(x_1, x_2, \dots, x_n)$, которое показывает насколько близко данная особь приблизилась к решению поставленной задачи. Таким образом, решение задачи предполагает поиск экстремума некоторой многомерной функции f . Часто вычисление оценочной функции сводится к моделированию решения задачи в заданной точке пространства. На основе оценочных функций вычисляется фитнес-функция, показывающая качество данной особи относительно других особей популяции. Исходя из ее значения, принимается решение о выживании данной особи, т.е. о её переходе в следующее поколение. В классическом генетическом алгоритме фитнес-функция особи полагается равной f_s / \bar{f} , где f_s - оценочная функция особи s , \bar{f} - средняя оценочная функция популяции.

Поиск решения задачи с помощью генетического алгоритма состоит из итеративного процесса порождения новой популяции на основании текущей. Выполнение одной итерации генетического алгоритма, как правило, разбивают на два шага (рис.1). Первый шаг состоит в построении промежуточной популяции из текущей. Выбор

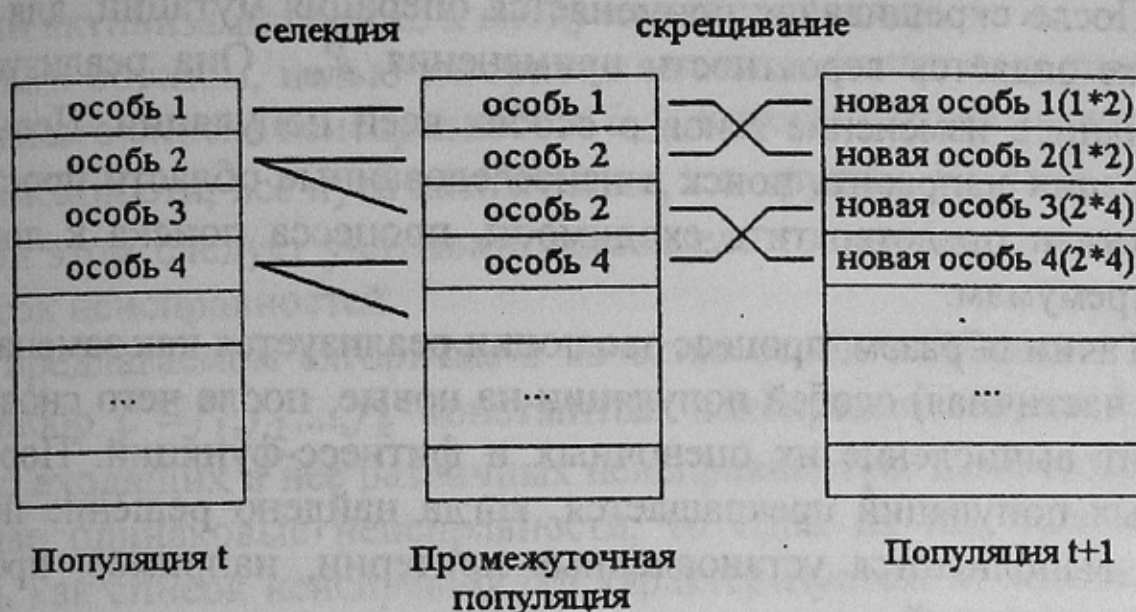


Рисунок 1 - Построение нового поколения особей

особей для промежуточной популяции называется селекцией. На втором шаге из промежуточной популяции строится популяция следующего поколения. Для этого над особями промежуточной популяции выполняются некоторые трансформирующие операции, под воздействием которых особи могут получить новые свойства и, тем самым, улучшить свою оценочную функцию. Такими операциями являются операции скрещивания и мутации.

Скрещивание – это операция обмена двух особей-родителей (иногда более) своими фрагментами, в результате которого получают особи-потомки с новыми свойствами. Реализуют как одноместную, так и многоместную операцию скрещивания (рис.2). Цель операции скрещивания заключается в том, чтобы направить процесс поиска в те регионы пространства, где предположительно находится решение задачи

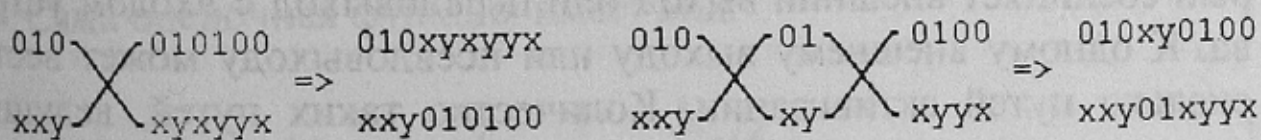


Рисунок 2 - Операции одноместного и двуместного скрещивания

После скрещивания применяется операция мутации, для которой также задаётся вероятность применения P_m . Она реализуется как случайное изменение генов в особях всей популяции. Цель данной операции направить поиск в неисследованные области пространства поиска и предотвратить сходимость процесса поиска к локальным экстремумам.

Таким образом, процесс эволюции реализуется как замена (полная или частичная) особей популяции на новые, после чего снова происходит вычисление их оценочных и фитнес-функций. Порождение новых популяций прекращается, когда найдено решение проблемы или выполняются установленные критерии, например, превышено число итераций.

Следовательно, чтобы задать генетический алгоритм, необходимо определить понятия особи, популяции, операции скрещивания и мутации, задать оценочную функцию. Очевидно также, что эффективность генетического алгоритма зависит от целого ряда параметров: размера популяции, метода выбора особей из предыдущей популяции, скрещивания и мутации, а также вида оценочной функции и т.д.

Формирование списка неисправностей. На предварительном этапе полный список $F = \{f_1, f_2, \dots, f_N\}$ константных неисправностей разбивается на группы F_1, F_2, \dots, F_n следующим образом. После преобразования синхронной последовательностной схемы в псевдокомбинационный эквивалент строится список всех внешних выходов и псевдовыходов. При этом внешние выходы считаются конечными точками распространения неисправностей, а псевдовыходы – точками активизации неисправностей. В одну группу неисправностей F_i входят все неисправности, находящиеся на пути активизации, который соединяет внешний выход или псевдовыход с входом устройства. К одному внешнему выходу или псевдовыходу может вести несколько путей активизации. Количество таких путей, ведущих ко всем внешним выходам и псевдо-выходам от входов схемы равно числу n групп предварительной сортировки $F_i, i = \overline{1, n}$. Предполагается, что если некоторый входной набор проверит (активизирует) неисправность $f_{ij} \in F_i$, то с большой вероятностью он проверит (акти-

визирует) другие неисправности из данной группы F_i , находящиеся на пути активизации ближе к входу или псевдовходу.

Таким образом, целью алгоритма является построение такого минимального списка неисправностей L_{\min} , который будет представлять, по возможности, все пути активизации, т.е. группы F_i , $i = \overline{1, n}$. Однако, при этом следует учитывать показатель тестируемости входящих в список неисправностей.

В предлагаемом алгоритме в качестве особи выбирается некоторый набор $L = \bar{f}_1, \bar{f}_2, \dots, \bar{f}_k$ константных неисправностей, где k - количество входящих в неё различных неисправностей. Если в особь входят две одинаковые неисправности, то одна из них исключается. Особь как список неисправностей характеризуется: а) числом входящих неисправностей; б) числом представленных групп предварительного разбиения, в которые входят неисправности; в) оценочным параметром входящих неисправностей. В качестве последнего используется показатель тестируемости неисправности, равный произведению наблюдаемости на управляемость линии, на которой расположена неисправность. Здесь параметр наблюдаемости – это мера наблюдаемости различия значений в исправном и неисправном устройствах на его выходах, то есть это мера возможности распространения соответствующей константной неисправности на выходы схемы. Параметр управляемости – это мера возможности установления заданного значения на линии устройства с помощью некоторой входной последовательности. Таким образом, оценочная функция определяется как взвешенная смесь трёх параметров:

- k - число неисправностей, входящих в особь;
- n' - число групп, представленных неисправностями, входящими в особь;
- $t(\bar{f}_i)$ - мера тестируемости неисправности \bar{f}_i из списка-особи L .

Сама оценочная функция имеет вид:

$$H(L) = (c_1 - k) + c_2 * n' + \frac{1}{n \cdot t_{cp}} \cdot \sum_{i=1}^k t(\bar{f}_i), \quad (1)$$

где k – размер особи, c_1, c_2 – нормализующие константы: c_1 – мощность полного множества неисправностей в схеме, c_2 – отношение полного числа неисправностей к числу групп предварительного разбиения, t_{cp} – средняя тестируемость по всем неисправностям (определяется принадлежностью к соответствующей линии схемы).

Заметим также, что вычисление построенной таким образом оценочной функции не требует моделирования анализируемой схемы и, следовательно, использование временных ресурсов очень экономично.

Операция скрещивания реализуется как обмен произвольными неисправностями в особях с вероятностью 0.5. В результате особи обмениваются приблизительно половиной неисправностей. При этом размеры наборов-особей не изменяются.

Возможно три операции мутации:

- удаление неисправности из особи; размер особи k уменьшается на 1;
- добавление произвольной неисправности в особь; размер особи k может не увеличиться, если в качестве новой неисправности выбрана, уже присутствующая в особи;
- замена неисправности на другую; размер особи может уменьшиться, если для замены выбрана неисправность, уже присутствующая в особи..

Выбор между операциями мутации происходит равновероятно, т.е. с вероятностью $p = \frac{1}{3}$. Вероятность операции мутации $P_{мут} = \frac{1}{k}$.

Ниже приведён псевдокод генетического алгоритма сжатия списка неисправностей. Начальная популяция формируется включением в особи неисправностей из всех групп предварительного разбиения.

Сжатие_списка_неисправностей()

```
{
    предварительное_разбиение_на_группы_неисправностей();
    формирование_начальной_популяции();
    for( i=0 ; i<MAX_ПОКОЛЕНИЙ ; i++)
    {
        for( каждой особи L в популяции P )
```



```

вычислить_оценку  $H(L)$ ;
newP= $\emptyset$ ; // новая популяция
for( j=0 ; j<ЧИСЛО_НОВЫХ_ОСОБЕЙ ; j++ )
{
   $(L_1, L_2)$ =выбрать_две_последовательности_в_P();
  применить_операцию_скрещивания( $L_1, L_2$ ); // порождаются
  // 2 новые особи
  применить_операцию_мутации( $L_1, L_2$ );
  newP=newP $\cup$ ( $L_1, L_2$ );
}
P=(лучшие k особей из newP и P )
} // конец цикла по поколениям
Lmin=P;
} // конец алгоритма

```

Выводы. Предложен новый метод формирования (сжатия) списка неисправностей, существенно сокращающий число неисправностей, которые необходимо моделировать в процессе построения теста. Процесс поиска оптимального списка основан на применении эффективной генетической стратегии поиска. Данный метод важен для процедур построения проверяющих тестов, основанных на моделировании. Подобные алгоритмы тестирования получили большое распространение в настоящее время, в силу резкого увеличения степени интеграции разрабатываемых устройств.

Список источников.

1. E. M. Rudnick, J. G. Holm, D. G. Saab, J. H. Patel, "Application of Simple Genetic Algorithm to Sequential Circuit Test Generation", Proc. European Design & Test Conf., 1994, pp. 40-45.
2. Cheng W.T., Patel J.H. PROOFS: A Super Fast Fault Simulator for Sequential Circuits. // Proc. The European Conference on Design Automation. – 1990. – p.475-479.
3. Иванов Д.Е., Скобцов Ю.А. Параллельное моделирование неисправностей для последовательностных схем // Искусственный интеллект. – 1999. – №1. – С.44-50
4. Иванов Д.Е., Скобцов Ю.А. Генерация тестов цифровых устройств с использованием генетических алгоритмов // Труды института прикладной математики и механики НАН Украины. – Т.4. – Донецк, ИПММ. – 1999. – С.82-88.
5. Goldberg D.E. Genetic algorithms in search, optimization and machine learning. – Addison-Wesley Publishing Company Inc., 1989. – 142p.