

УДК 004.3:681.518

Д.Е. ИВАНОВ

Институт прикладной математики и механики НАН Украины, г. Донецк

СИНТЕЗ ЭВОЛЮЦИОННЫХ МЕТОДОВ ДИАГНОСТИРОВАНИЯ СБИС

Розроблено методику синтезу еволюційних методів діагностування сучасних великих цифрових пристроїв. Методика включає розробку шаблонів одно- та дворівневих еволюційних методів, компонентів таких методів (кодування рішень та еволюційні операції), формальний апарат побудови функцій оцінювання потенційних рішень та низку застосованих евристик.

Ключові слова: діагностування, цифровий пристрій, еволюційний алгоритм, компоненти алгоритму, синтез.

D.E. IVANOV

Institute of Applied Mathematics and Mechanics, National Academy of Sciences of Ukraine, Donetsk

SYNTHESIS OF EVOLUTIONARY METHODS FOR DIAGNOSING OF VLSI

Abstract – The aim of the paper is to development of methodology for the synthesis of evolutionary methods for diagnosing of large digital devices.

A technique for the synthesis of new evolutionary methods of diagnosing of digital devices is proposed. It includes templates of one- and two-level evolutionary methods of diagnosing of digital devices and implementation-dependent components. These components include: encoding of potential solutions, evolutionary operations, evaluation functions and heuristics. To formalize the construction of evaluation functions the behavior functions of digital device at the structural level of representation are introduced. Then a new method is constructed as the union of the selected template and implementation-dependent components.

Proposed methodology allows building new effective evolutionary methods for diagnosing of large digital devices, which are based on a genetic algorithm as well as on the method of simulated annealing.

Keywords: diagnostics, digital device, evolutionary algorithm, components of algorithm, synthesis of methods.

Введение и анализ публикаций

Эволюционные методы [1] находят широкое применение при решении различных прикладных задач. В технической диагностике цифровых устройств (ЦУ), в частности, развивается направление, связанное с построением входных диагностических последовательностей различных классов [2-3] с помощью эволюционных алгоритмов (ЭА).

Преимущество ЭА перед методами, основанными на построении деревьев обходов или преобразовании булевых функций представления ЦУ, заключается в следующем. В отличие от методов синтеза решений эволюционные алгоритмы используют подход, который заключается в итеративном улучшении качества некоторых потенциальных решений. При решении задач диагностирования качество входных последовательностей определяется на основании моделирования поведения ЦУ. Поскольку считается, что задача моделирования для больших ЦУ решена, то ЭА диагностирования ЦУ позволяют проводить обработку очень больших ЦУ, промышленной реализацией которых являются СБИС.

Разработка методов построения идентифицирующих последовательностей (ИдП) с применением эволюционных вычислений идёт в двух направлениях. Во-первых, рассматриваются задачи построения различных классов последовательностей, например, тестовых [3-4], инициализирующих [5-6], верификации эквивалентности поведения [7], диагностических (локализирующих неисправность) [8-9]. Во-вторых, при разработке методов построения ИдП ЦУ авторы используют различные эволюционные алгоритмы. В частности, используются генетические алгоритмы (ГА) [2-5, 8], алгоритмы симуляции отжига (СО) [10-11], муравьиные алгоритмы и алгоритмы роя частиц [6, 12] и т.д. Работа отечественных авторов в этом направлении обобщена в монографии [2].

Методы метаэвристического поиска [13], такие как муравьиные алгоритмы и алгоритмы роя частиц, в данной работе рассматриваться не будут, поскольку, во-первых, их применение в задачах технической диагностики ещё очень ограничено, а во-вторых, строго говоря, они не являются эволюционными алгоритмами (ЭА) по устоявшейся классификации. Далее в данной статье мы будем в качестве популяционных ЭА рассматривать генетический алгоритм, а в качестве ЭА с эволюцией одного решения – метод симуляции отжига, называя их обобщённо ЭА-методами.

Формулировка целей статьи

Часто разрабатываемые ЭА-методы очень близки идеологически и отличаются только в некоторых деталях: метод моделирования, используемый для оценки особей, вид оценочной функции, применяемые эвристики и т.п. Также отметим, что построение существующих эволюционных методов основано, в основном, на эвристическом подходе, они несут точечный характер применения и слабо взаимосвязаны между собой, что затрудняет их эффективное применение к решению новых задач. Таким образом, возникает задача объединения данных методов на единой методологической основе, что позволит строить новые методы диагностирования современных больших ЦУ. В данной статье предлагается методика синтеза

новых эволюционных методов диагностирования цифровых устройств, которая учитывает большой опыт авторов в разработке таких методов.

Шаблоны одно- и двухуровневых ЭА

Схемы применения ЭА-методов. Анализ применения ЭА-методов в диагностике ЦУ позволяет разделить их на два больших класса. В одном случае решение задачи может быть найдено за один вызов ЭА. В другом случае требуется итеративный вызов некоторого ЭА. Для таких классов введём две модели применения ЭА.

В одноуровневой модели (схеме применения) ЭА-методов (рис.1а) единственный цикл эволюции непосредственно формирует метод решения задачи. К методам данного класса относятся относительно простые задачи: инициализация ЦУ, проверка эквивалентности поведения и т.п.

В том случае, если сложность задачи не позволяет методу найти решение за один вызов ЭА поиска, то будем говорить о двухуровневой модели (двухуровневой схеме, рис.1б), предполагающей итеративную двухфазовую реализацию. В первой фазе происходит поиск промежуточной (локальной) цели. Если такая цель найдена, то для неё вызывается ЭА поиска решения, формирующий вторую фазу итерации. Будем называть фазу 1 верхним уровнем, а фазу 2 – нижним уровнем ЭА. При этом структура фазы 2 метода соответствует одноуровневому ЭА построения ИдП. В задачах построения входных ИдП решение (последовательность) часто строится по аддитивному принципу, т.е. данные задачи естественным образом проецируются на двухуровневую схему ЭА. Примером может служить задача построения проверяющих тестов.

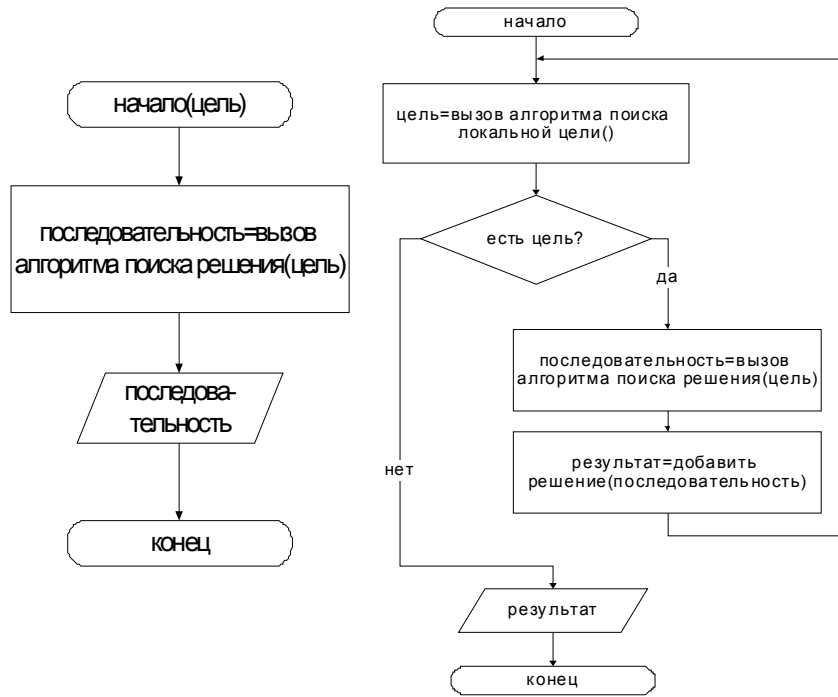
В работе рассмотрены одно- и двухуровневые ЭА-методы как для популяционных алгоритмов (ГА), так и для методов с эволюцией одного решения (СО).

Шаблон

одноуровневых ГА-методов. Первым и основным элементом в рассматриваемой методологии являются шаблоны ЭА-методов. Вначале рассмотрим шаблоны для одноуровневой схемы.

Шаблон

одноуровневого ГА-метода должен показывать итеративное построение новых популяций особей-последовательностей при условии, что выбраны следующие элементы: представление особей, эволюционные операции, оценочная функция, параметры эволюции. Будем такие элементы, в отличие от шаблона, называть зависящими от реализации. Укрупненная схема такого шаблона в виде псевдокода представлена ниже.



а) одноуровневая б) двухуровневая
Рис.1. Модели применения ЭА

Схема А1

```

ГА_построения_ИдП( A0 , Параметры) {
    ПредварительнаяОбработка( A0 );
    Popнач = ПостроениеНачальнойПопуляции( Nособ , L );
    ОценитьПопуляцию( Popнач , A0 , Nособ );
    Popтек = Popнач ;
    НомерПопуляции=0;
    while( НеДостигнутКритерийОстановки() ){ // цикл по поколениям
        ВычислитьФитнессФункцию( Popтек , Nособ );
        while( СтроитсяНоваяПопуляция() ){ //построение новой популяции
            Родители=ОперацияСелекции( Popтек );
            Потомки=ОперацияСкрещивания( Родители );
            Потомки=ОперацияМутации( Потомки );
            ДобавлениеВНовуюПопуляцию( Потомки );
        }
    }
}
    
```

```

} // конец while - построение новой популяции
Popтек = ПостроитьНовуюПопуляцию();
ОценитьПопуляцию( Popтек, A0, Nособ );
АдаптацияПараметров();
НомерПопуляции++;
} // конец while - достигнут критерий остановки
СортироватьПопуляциюПоОценке( Popтек );
Решение = Popтек[0]; // лучшая особь в посл. популяции
} // конец одноуровневого ГА построения ИдП

```

В данном псевдокоде: A_0 - обрабатываемое ЦУ; $Pop_{нач}$, $Pop_{тек}$ - начальная и текущая популяции; $N_{особ}$ - число особей в популяции; L - начальная длина строящихся последовательностей.

Совместное конструктивное задание зависящих от реализации компонент со схемой А1 означает построение соответствующего одноуровневого ГА-метода. На основе данного шаблона разработаны методы логической инициализации, достижения состояний и верификации эквивалентности ЦП [2].

Шаблон одноуровневого СО-метода разрабатывается аналогичным образом. Однако от представленного выше он показывает итеративное улучшение одного решения – конфигурации.

Шаблон двухуровневых ЭА-методов. Аналогично рассмотренному выше разрабатывается шаблон верхнего уровня для двухуровневых ЭА-методов построения ИдП, который представлен ниже.

Схема А2

```

Двухуровневый_ЭА_построения_ИдП( A0, Параметры ) {
    ПредварительнаяОбработка( A0 );
    while( НеДостигнутКритерийОстановки() ) {
        Цель = ВыборТекущейЦели( A0, Параметры );
        if( Цель == НемЦели )
            return; // нет цели - завершение работы алгоритма
        else {
            // цель найдена - вызов ГА поиска локального решения
            S = ЭА_Построение_ИдП( A0, Цель, НачальнаяПопуляция );
            if( S == NULL ) // последовательность не построена
                ОтметитьКакНедостижимую( Цель );
            else {
                ДобавитьВТест( S );
                ДополнительнаяПроверка( A0, S );
            } // конец else - последовательность построена
        } // конец else - выбрали цель
    } // конец while - не достигли критерия остановки
} // конец двухуровневого ГА-метода

```

Шаблон показывает итеративный выбор некоторой промежуточной цели метода и её достижение с помощью некоторого ЭА. Поэтому данный шаблон применим как для ГА так и для СО, поскольку он абстрагируется от такой поисковой процедуры и вызывает её в качестве вспомогательной.

Объект *Цель* может иметь различную природу: непроверенная неисправность в методе построения проверяющих тестов, арифметическая операция в методе построения функциональных тестов АЛУ и т.д.

Построение двухуровневого ЭА-метода включает шаблоны верхнего и нижнего уровней (как на основе ГА, так и СО), а также зависящие от реализации компоненты. По данной схеме разрабатываются методы построения проверяющих и диагностических тестов, функциональных тестов арифметико-логических устройств, энергоэффективных тестов и т.п.

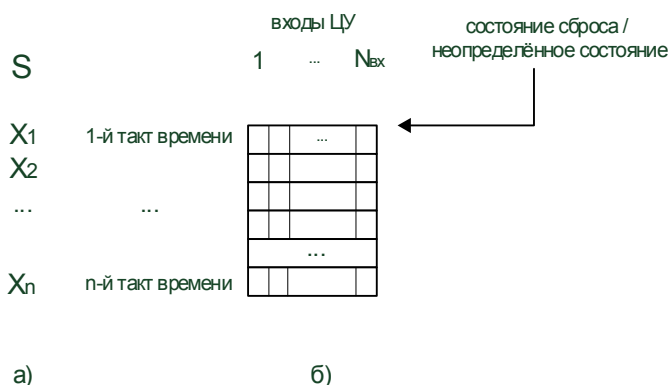


Рис.2. Представление особи в ГА построения ИдП

Компоненты эволюционных алгоритмов диагностирования ЦУ

Кодирование решений и эволюционные операции. Опишем теперь компоненты ЭА построения ИдП. В качестве потенциального решения рассматривается упорядоченное множество входных наборов.

Для абстрактного уровня задания ЦУ потенциальному решению соответствует входная последовательность $S = (X_1, \dots, X_n)$ (рис.2а), а на структурном – матрица, каждый элемент которой представляет символ из алфавита кодирования (рис.2б). В ГА потенциальное решение называется особью, а их набор образует популяцию. В СО потенциальное решение называется конфигурацией.

Операции скрещивания разделим на два класса: 1) имеющие аналоги для структурного и абстрактного уровней задания ЦУ (горизонтальное скрещивание); 2) операции, реализуемые только на структурном уровне и не имеющие явного аналога для абстрактного уровня (различные реализации вертикального скрещивания). Аналогично, операции мутации также делятся на два класса: добавление/удаление/изменение вектора (абстрактный и структурный уровни) и изменение столбца (структурный уровень). Часто элементом исследования при разработке ЭА-метода является поиск таких эволюционных операций, которые наилучшим образом работают в конкретном методе.

Более подробно данные компоненты методов рассмотрены в [2].

Таблица 1

Динамические параметры оценочных функций.

параметр	интерпретация
$U_{N_{вых}}, U_{N_{вх}}, U_{N_{тр}}, U_{N_{к.т.}}, U_{N_{эл}}$	число компонент, для которых установлено заданное значение в алфавите моделирования
$R_{N_{вых}}, R_{N_{вх}}, R_{N_{тр}}, R_{N_{к.т.}}, R_{N_{эл}}$	число соответствующих компонент, для которых получены различные значения
$N_{собр}$	число событий
L	длина последовательности
$ F_{тест} $	число обнаруженных неисправностей
$ F_{акт} $	число активизированных неисправностей
$N_{нер}^i$	число множеств неразличимых неисправностей мощности i

Построение оценок входных диагностических последовательностей. Анализ известных методов построения ИдП показывает, что все параметры, используемые при построении оценок, могут быть разделены на две группы: статические и динамические. Статические параметры являются характеристиками обрабатываемого ЦУ A_0 и не изменяются в процессе работы ГА: $N_{вых} / N_{вх} / N_{тр} / N_{к.т.} / N_{эл}$ - число внешних выходов / входов / элементов состояний / контрольных точек / логических элементов (комбинационных блоков); $|F|$ - число рассматриваемых неисправностей; H / I - управляемость / наблюдаемость элемента и т.п. Динамические параметры (табл.1) являются свойством оцениваемой особи-последовательности S и вычисляются на основании моделирования поведения исследуемого ЦУ.

Для формализации информации о поведении ЦУ введем понятия булевых функций поведения компонент. Основными такими функциями являются: функция установки компоненты в заданное значение, функция активности компоненты и функция различающей активности. Данные функции вычисляются по некоторому контрольному множеству компонент устройства. Обычно рассматривают множества внешних выходов, элементов состояний, выходов логических элементов и контрольных точек. Если оставить для рассмотрения только первых два множества, то по уровню абстракции целевое ЦУ будет выглядеть как конечный автомат, для которого не известна внутренняя реализация. Принимая во внимание поведение ЦУ по другим из названных множеств, к рассмотрению принимается существенно большая информация о его поведении. Именно учёт в функциях оценки такой дополнительной информации о поведении ЦУ на структурном уровне даёт преимущество данных методов над автоматными.

Пусть функция $g(X_j, A_0)$ показывает значение на выходе компоненты g в заданном алфавите после моделирования A_0 на наборе X_j . Функция установки в заданное значение $U^{(v)}(g, X_j, A_0)$ равна 1, если после моделирования A_0 на входном наборе X_j выход элемента g равен v :

$$U^{(v)}(g, X_j, A_0) = \begin{cases} 1 & \Leftrightarrow g(X_j, A_0) = v; \\ 0, & \text{иначе.} \end{cases} \quad (1)$$

При использовании в качестве основного трёхзначного алфавита E_3 целевыми являются значения 0 и 1, снимающие неопределённость начального состояния, т.е. используются функции $U^{(0)}$ и $U^{(1)}$.

Функция установки значений по множеству компонент имеет следующий вид:

$$U_G = \sum_{g \in Z} (U^{(1)}(g, X_j, A_0) \vee U^{(0)}(g, X_j, A_0)), \quad (2)$$

где G - исследуемое множество в ЦУ.

Функция активности $N_{собр}$ ЦУ A_0 по заданному множеству при моделировании на входной последовательности S показывает число изменений значений сигнала:

$$N_{\text{cob}}(A_0, S) = \sum_{j=0}^{\text{длина}(S)} \sum_{g \in G} ((U^{(1)}(g, X_{j-1}, A_0) = U^{(0)}(g, X_j, A_0)) \vee (U^{(0)}(g, X_{j-1}, A_0) = U^{(1)}(g, X_j, A_0))), \quad (3)$$

где G - исследуемое множество компонент в ЦУ.

Функция различия $r()$ компоненты g по паре устройств A_1 и A_2 при использовании двухзначного алфавита моделирования E_2 определяется как:

$$r(g, X_j, A_1, A_2) = g(X_j, A_1) \oplus g(X_j, A_2). \quad (4)$$

При использовании многозначных алфавитов (4) должна быть переопределена с их учётом. Аналогично (3) определяется функция различия поведения двух устройств A_1 и A_2 при подаче тестовой последовательности S . Также в некоторых задачах требуется определять различие в поведении не двух устройств, а целого множества F . В этом случае (4) следует переопределить для множества устройств:

$$R(g, X_i, A_0, F) = \begin{cases} 0, & \text{если выходы элементов } g \text{ множества ЦУ} \\ & \text{из } F, \text{ одинаковы после подачи набора } X_i; \\ 1, & \text{иначе.} \end{cases} \quad (5)$$

Видно, что функция $r()$ различия двух устройств (4) является частным случаем функции различия $R()$ для множества устройств.

Введённые функции поведения (1)-(5) позволяют выразить все динамические параметры (табл.1), которые используются при построении оценок качества последовательностей. Оценочные функции обычно представляются как взвешенная сумма параметров установки / активности / различающей активности по контрольным множествам при моделировании устройства (устройств) на заданной оцениваемой последовательности. Часто оценки дополняются эвристиками, включающими длину последовательности, число проверенных / непроверенных неисправностей и т.д. Включение всех таких параметров в оценочную функцию при решении конкретной задачи может оказаться неэффективным и необходим их отбор.

Подробнее построение оценочных функций на основании функций поведения ЦУ структурного уровня рассмотрено в [14].

Методика синтеза эволюционных методов диагностирования ЦУ

После определения компонент опишем методику синтеза новых ЭА-методов и их алгоритмических реализаций. Укрупнённо алгоритм синтеза приведён на рис.3. Поясним каждый из этапов.

На первом шаге исследователь выбирает общую структуру будущего ЭА-метода: одно- либо двухуровневую. Результаты данного этапа существенно зависят от знаний и навыков разработчика, который в данной ситуации выступает в роли эксперта. Это связано с тем, что методы решения одной и той же задачи могут строиться на основании различных подходов. Например, метод достижения заданного состояния ЦУ может синтезироваться как по одноуровневой, так и по двухуровневой схеме [15].

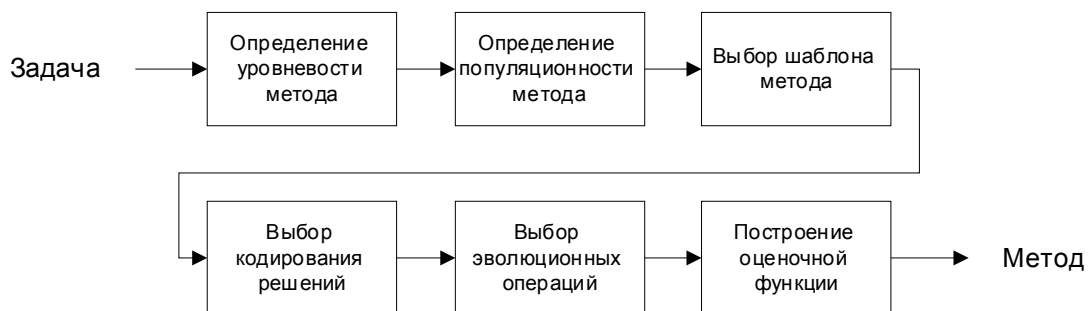


Рис.3. Алгоритм покомпонентного синтеза ЭА-метода

Второй шаг предполагает выбор между популяционными и непопуляционными методами. Споры в научной среде об эффективности того или иного эвристического подхода к решению конкретной задачи не утихают до сих пор. В том числе такой вопрос встаёт и при выборе генетических алгоритмов или методов симуляции отжига. Авторы проводили практическое сравнение эффективности ГА- и СО-методов при решении задач построения ИдП различных классов. Результаты этих исследований приведены в [11].

На шаге 3 на основании предыдущих решений выбирается шаблон соответствующего метода. Шаги 4 и 5 путём выбора кодирования решений и вида оценочной функции доопределяют шаблон до полноценного метода.

Реализация выполняется на основании программно реализованных шаблонов схем методов, структур данных, реализующих кодирование особей, процедур, реализующих эволюционные операции и эвристики.

Синтезированный по такой схеме метод и его программная реализация позволяет находить решение исходной задачи. Однако на этом этапе ни о какой эффективности нового метода речь не ведётся. Построенный ЭА-метод традиционно требует «настройки», которая заключается в экспериментальном

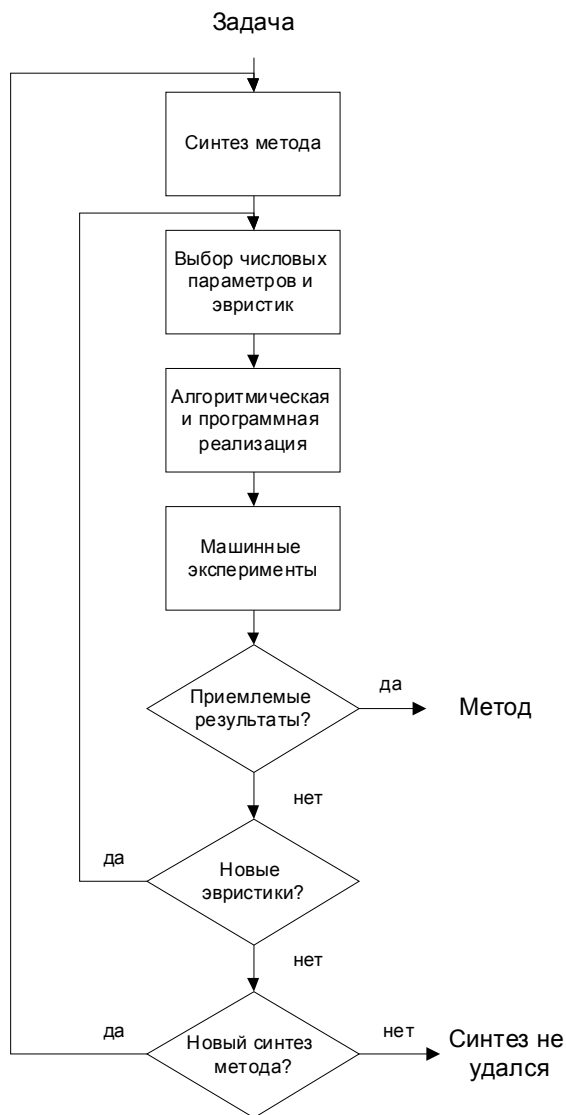


Рис.4. Итеративный алгоритм синтеза и настройки ЭА-метода

обосновании его параметров: условие окончания работы, число особей в популяции, схема отбора, вероятности применения эволюционных операций и т.д. После такой настройки синтезированный ЭА-метод будет являться эффективным в некотором практическом смысле.

Однако предлагаемая методика синтеза позволяет получить несколько методов, решающих поставленную задачу. В общем случае выбор наиболее эффективного метода включает двухуровневую итеративную схему их синтеза и настройки (рис.4). Нижний уровень представляет собой описанную выше настройку одного ЭА-метода. На верхнем уровне в качестве вариативной части выступают структура метода и его популяционность. При этом четкого критерия остановки здесь нет ввиду особенностей самих методов. Выход из алгоритма происходит, когда разработчик решает, что синтезированный ЭА-метод имеет приемлемую эффективность.

Выводы

В работе представлена методика синтеза эволюционных методов диагностирования ЦУ. Она основывается на разработке шаблонов и компонент таких методов, включая кодирование решений, эволюционные операции и оценочные функции. Для формального построения оценочных функций введены функции поведения цифрового устройства на структурном уровне описания. Предложен итеративный алгоритм синтеза новых методов и их оптимизация. Предложенная методика позволяет синтезировать новые одно- и двухуровневые эволюционные методы генерации ИдП. Разработаны шаблоны и компоненты как для популяционных, так и для непопуляционных эволюционных методов построения идентифицирующих последовательностей.

Литература

1. Скобцов Ю.А. Основы эволюционных вычислений. – Донецк: ДонНТУ, 2008. – 326с.
2. Иванов Д.Е. Генетические алгоритмы построения входных идентифицирующих последовательностей цифровых устройств / Иванов Д.Е. – Донецк, 2012. – 240с.
3. Corno F. Experiences in the use of evolutionary techniques for testing digital circuits / F. Corno, M. Sonza Reorda, M. Rebaudengo // Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA. – 1998. – P.128-139.
4. Rudnick E.M. Sequential Circuit Test Generation in a Genetic Algorithm Framework / E.M. Rudnick, J.H. Patel, G.S. Greenstein, T.M. Niemann // Proc. Design Automation Conf.- 1994. – P.698–704.
5. Corno F. A Genetic Algorithm for the Computation of Initialization Sequences for Synchronous Sequential Circuits / F. Corno, P. Prinetto, M. Rebaudengo etc. // Proceeding ATS '01 Proceedings of the 10th Anniversary Compendium of Papers from Asian Test Symposium 1992-2001. – 2001. – P.213.
6. Xiaojing H. Ant Colony Optimizations for Initialization of synchronous sequential circuits / H. Xiaojing, S. Zhengxiang // IEEE Circuits and Systems International Conf., – 2009. – P. 5–18.
7. Corno F. Evolutionary Simulation-Based Validation / F. Corno, M. Sonza Reorda, G. Squillero // International Journal on Artificial Intelligence Tools (IJAIT). – 2004. – Vol.14, 1-2, Dec. – P.897 916.
8. Corno F. GARDA: a Diagnostic ATPG for Large Synchronous Sequential Circuits / F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proc. of IEEE European Design and Test Conference, Paris, March 1999. – P.267 - 271.
9. Yu X. Diagnostic Test Generation for Sequential Circuits / X. Yu, J. Wu, E.M. Rudnick // Proc. of International Test Conference. – Atlantic City, NJ. – 2000. – P.225 – 234.
10. Corno F. SAARA: a simulated annealing algorithm for test pattern generation for digital circuits / F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda // Proceedings of the 1997 ACM symposium on Applied

computing, San Jose, California. – 1997. – P.228-232.

11. Иванов Д.Е. Применение алгоритмов симуляции отжига в задачах идентификации цифровых схем // Вестник Национального технического университета «Харьковский политехнический институт». Сборник научных трудов. Тематический выпуск: Информатика и моделирование. – Харьков: НТУ "ХПИ", 2011. – № 17. – С.60-69.

12. Xin F. A Sequential Circuits Test Set Generation Method Based on Ant Colony Particle Swarm algorithm / Fu Xin, Fu Shuai // Proc. of National Conference on Information Technology and Computer Science (CITCS 2012). – Atlantis Press, 2012. – P.205-209.

13. Скобцов Ю.А. Метаэвристики / Ю.А. Скобцов, Е.Е. Федоров. – Донецк: Ноулидж, 2013. – 426 с.

14. Иванов Д.Е. Применение информации структурного уровня в алгоритмах построения идентифицирующих последовательностей / Д.Е. Иванов // Известия ЮФУ. Технические науки. – 2013. – №1. – С.149-160.

15. Иванов Д.Е. Алгоритмы достижения состояний в цифровых устройствах и их применение в задачах диагностики / Д.Е. Иванов // Вісник Хмельницького національного університету. Технічні науки. – Хмельницький, 2012. – №3(189). – С.104-110.

References

1. Yu.A. Skobcov, *Osnovy e'volucionnykh vychislenij*, Doneck: DonNTU, 2008.
2. D.E. Ivanov, *Geneticheskie algoritmy postroeniya vxođnykh identifičiruyushhix posledovatel'nostej cifrovyyh ustrojstv*, Doneck: IAMM NASU, 2012.
3. F. Corno, M. Sonza Reorda and M. Rebaudengo, "Experiences in the use of evolutionary techniques for testing digital circuits", Proc. of Conf. Applications and science of neural networks, fuzzy systems, and evolutionary computation, San Diego CA, 1998, pp. 128-139.
4. E.M. Rudnick, J.H. Patel, G.S. Greenstein and T.M. Niermann, "Sequential Circuit Test Generation in a Genetic Algorithm Framework", Proc. Design Automation Conf, 1994, pp. 698-704.
5. F. Corno, P. Prinetto and M. Rebaudengo, "Genetic Algorithm for the Computation of Initialization Sequences for Synchronous Sequential Circuits", Proceeding ATS '01 Proceedings of the 10th Anniversary Compendium of Papers from Asian Test Symposium 1992-2001, 2001, pp. 213.
6. H. Xiaojing and S. Zhengxiang, "Ant Colony Optimizations for Initialization of synchronous sequential circuits", // IEEE Circuits and Systems International Conf., 2009, pp. 5-18.
7. F. Corno, M. Sonza Reorda and G. Squillero, "Evolutionary Simulation-Based Validation", International Journal on Artificial Intelligence Tools (IJAIT), 2004, Vol.14, 1-2, pp. 897-916.
8. F. Corno, P. Prinetto, M. Rebaudengo and M. Sonza Reorda, "GARDA: a Diagnostic ATPG for Large Synchronous Sequential Circuits", Proc. of IEEE European Design and Test Conference, Paris, March 1999, pp. 267 - 271.
9. X. Yu, J. Wu and E.M. Rudnick, "Diagnostic Test Generation for Sequential Circuits", Proc. of International Test Conference, Atlantic City, NJ, 2000, pp. 225 - 234.
10. F. Corno, P. Prinetto, M. Rebaudengo and M. Sonza Reorda, "SAARA: a simulated annealing algorithm for test pattern generation for digital circuits", Proceedings of the 1997 ACM symposium on Applied computing, San Jose, California, 1997, pp. 228-232.
11. D.E. Ivanov, "Primenenie algoritmov simulyacii otzhiga v zadachax identifikacii cifrovyyh sxem", Vestnik Nacional'nogo texnicheskogo universiteta «Xar'kovskij politexnicheskij institut». Sbornik nauchnyx trudov. Tematicheskij vypusk: Informatika i modelirovanie, Xar'kov: NTU "XPI", 2011, Vol.17, pp. 60-69.
12. Fu Xin and Fu Shuai, "A Sequential Circuits Test Set Generation Method Based on Ant Colony Particle Swarm algorithm", Proc. of National Conference on Information Technology and Computer Science (CITCS 2012), Atlantis Press, 2012, pp. 205-209.
13. Yu.A. Skobcov and E.E. Fedorov, *Metaevristiki*, Doneck: Noulidzh, 2013.
14. D.E. Ivanov, "Primenenie informacii strukturnogo urovnya v algoritmax postroeniya identifičiruyushhix posledovatel'nostej" Izvestiya YuFU. Texnicheskie nauki, 2013, Vol.1, pp. 149-160.
15. D.E. Ivanov, "Algoritmy dostizheniya sostoyanij v cifrovyyh ustrojstvax i ix primenenie v zadachax diagnostiki", Visnyk Khmelnytskoho natsionalnoho universytetu. Tekhnichni nauky. – Khmelnytskyi, 2012, Vol.№3(189), pp. 104-110.

Рецензия/Peer review : 25.6.2014 р.

Отпечатана/Printed : 12.7.2014 р.

Рецензент: д.т.н., проф., зав. кафедрою автоматизованих систем управління ДонНТУ Ю.О. Скобцов