

УДК 519.718.7:681.3.069

**А. Н. Мартынюк**, канд. техн. наук,  
**А. С. Сугак**

### ЭВОЛЮЦИОННАЯ СЕТЕВАЯ МОДЕЛЬ ТЕСТИРОВАНИЯ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

**Аннотация.** Предложена сетевая алгоритмическая модель, основанная на иерархической композиции модели тестирования и модели эволюций и использующая тестовые контрольные и транспортные фрагменты экспериментов для компонентных автоматов сети в качестве базиса для эволюционной системы на основе генетического программирования. Модель позволяет определять условия тестирования и построить метод синтеза функциональных тестов распределенных информационных систем. Реализация модели для сетевых сервисов управления подтвердила целесообразность исследования.

**Ключевые слова:** распределенные информационные системы, функциональные тесты, модель тестирования, эволюционная система, генетическое программирование

**N. Martynyuk**, PhD,  
**A. S. Sugak**

### EVOLUTIONARY NETWORK MODEL OF TESTING FOR DISTRIBUTED INFORMATION SYSTEMS

**Abstract.** Proposed network algorithmic model based on hierarchical composition model of testing and model of evolution and used the test check and transport experiments for component automata of network, as a basis for evolutionary system based on genetic programming. The model allows to determine the conditions of the test and build a method for the synthesis of functional tests of distributed information systems. Implementation of the model for network services management has confirmed the feasibility of the study.

**Keywords:** distributed information systems, functional tests, model testing, evolutionary system, genetic programming

**О. М. Мартинюк**, канд. техн. наук,  
**Г. С. Сугак**

### ЕВОЛЮЦІЙНА МЕРЕЖЕВА МОДЕЛЬ ТЕСТУВАННЯ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

**Анотація.** Запропоновано мережеву алгоритмічну модель, засновану на ієрархічній композиції моделі тестування та моделі еволюцій і що використовує тестові контрольні і транспортні фрагменти експериментів для компонентних автоматів мережі як базису для еволюційної системи на основі генетичного програмування. Модель дозволяє визначати умови тестування і побудувати метод синтезу функціональних тестів розподілених інформаційних систем. Реалізація моделі для мережевих сервісів управління підтвердила доцільність дослідження.

**Ключові слова:** розподілені інформаційні системи, функціональні тести, модель тестування, еволюційна система, генетичне програмування

#### Введение

Системы диагностирования, осуществляющие рабочий и тестовый контроль, составляют весомую часть в числе разнообразных средств обеспечения работоспособности распределенных информационных систем (РИС). Быстро растущая сложность РИС, общность, неполнота, нечеткость, многоуровневость спецификаций компонентов РИС [1], зачастую ограниченных функциональным описанием, наряду с имеющейся невозможностью прямого наблюдения и управления большинством из них предполагает использование и развитие аналитических моделей и методов тестирования РИС, основанных, в частности, на композициях

объектов и процессов, структурных, функциональных, информационных представлениях, подходах искусственного интеллекта [2].

Сложность тестирования и, в частности, синтеза тестов, особенно в случае функционально-поведенческих спецификаций, порождающих перебор [3], обуславливает целесообразность развития декомпозиционных моделей и оптимизации методов синтеза функциональных тестов (ФТ) [4], снижающих экспоненциальную сложность задач тестового анализа.

В этой связи представляется актуальным исследование моделей и методов синтеза ФТ, использующих идентификацию свойств [5] в сетевых автоматных экспериментах [6, 7] на основе псевдослучайного целевого поиска

© Сугак А.С., Мартынюк А.Н., 2014

эволюционных систем (ЭС) [8] и методов генетического программирования (ГП) [9].

### Постановка задачи

Целью работы является сокращение времени и, как следствие, вычислительных затрат на синтез ФТ для поведенческого диагностирования компонентов РИС в классе ошибок их функциональных отображений за счет конструктивного описания ФТ как композиции реализуемых и распознаваемых автоматных контрольных и транспортных экспериментов в сети автоматов (СА), представляющей РИС, на основе идентификации специфицируемых поведенческих свойств и использования моделей и методов ЭС/ГП.

Поставленная цель может быть достигнута на основе решения задачи построения сетевой алгоритмической модели (АМ):

во-первых, основанной на иерархической композиции модели тестирования СА и модели сетевой тестовой эволюции в СА;

во-вторых, обладающей особенностями использования идентификации специфицируемых свойств в контрольных и транспортных экспериментах, построенных на базе тестовых фрагментов и ЭС/ГП в СА.

Компонентные тестовые фрагменты и ЭС, реализуемые и распознаваемые в СА, могут применяться в качестве объектного базиса объединенной АМ.

Таким образом, АМ в части модели тестирования позволяет определить условия функционального тестирования РИС, а в части модели ЭС – для синтеза ФТ РИС.

Под тестовым фрагментом понимается часть тестового вход-выходного поведения компонентного автомата (КА) в СА, лежащая в диапазоне от некоторого неделимого тестового примитива, идентифицирующего некоторое свойство, до некоторого полного контрольного или транспортного эксперимента для этого компонента.

Построение АМ использует два представления, то есть  $AM = (CT, ЭСА)$ .

Первое – представление тестирования СТ для СА, формирующее общие объекты ЭС/ГП – реализуемые (управляемые) и распознаваемые (наблюдаемые) в СА тестовые контрольные и транспортные тестовые фрагменты для КА в СА.

Второе – представление сетевой эволюции ЭСА, особыми которой являются указанные общие объекты ЭС/ГП – реализуемые и распознаваемые контрольные и транспортные тестовые фрагменты для КА в СА.

### Определение формальной сетевой модели тестирования

В представлении тестирования могут быть определены различные типы целевых объектов, в частности, в поведенческой проверке на основе контрольных или транспортных экспериментов в качестве объектов определены тестовые фрагменты для КА в СА.

Формально модель для представления тестирования СТ – сетевая модель тестирования – содержит четыре компонента [6], приобретающих собственную интерпретацию размерности и полноты тестирования всех КА – объектов СА:

$$CT = (CA, \{\delta^\wedge, \lambda^\wedge\}, \{Id^\wedge, \{R^\wedge Tr^\wedge\}\}, Fr^\wedge),$$

где

–  $CA = (X, Y, A^\wedge, \alpha^\wedge)$  – сеть взаимодействующих автоматов, в которой:

а)  $X, Y$  – общие входной и выходной алфавиты СА;

б)  $A^\wedge = \cup_{i \in I} A_i$  множество компонентных автоматов (КА) вида  $A_i = (S_i, X_i, Y_i, \delta_i, \lambda_i, S_{oi})$  с соответствующим множеством абстрактных вход-выходных поведений  $Be^\wedge = \cup_{i \in I} Be_i$  в объединенных алфавитах вида  $U_i = X_i \times Y_i$ , то есть, языков, структурированных в соответствующие сети на основе начальных состояний и детерминированности поведения КА (очевидно,  $\min(Be_i) = A_i$ );

в)  $\alpha^\wedge = \cup_{i \in I} \alpha_i$  – множество алфавитных отображений (структура связей в СА) вида  $\alpha_i: X \times Y_i \rightarrow X_i$ , где  $i' \in I \setminus \{i\}$ , причем,  $Y = X_k$  для некоторого  $k \in I$ ;

–  $\delta^\wedge = \cup_{i \in I} \delta_i$  и  $\lambda^\wedge = \cup_{i \in I} \lambda_i$  – проверяемые сетевые свойства для всех  $A_i \in A^\wedge$  в составе их контролируемых функций соответственно переходов и выходов;

–  $Id^\wedge = \cup_{i \in I} Id_i$  (где для любого  $i \in I$   $Id_i \leq Be_i$ ) контрольные идентификаторы, проверяющие свойства для всех  $A_i \in A^\wedge$ , в составе их идентифицирующих характеристических окрестностей внутренних состояний  $s_i \in S_i$ ; где  $i \in I$ .

–  $R^\wedge = \cup_{i \in I} R_i$  и  $Tr^\wedge = \cup_{i \in I} Tr_i$  (где для любого  $i \in I$   $R_i \leq (Be_i)^{-1}$ ,  $Tr_i \leq Be_i$ ) – соответствен-

но реализуемые и распознаваемые транспортные примитивы-идентификаторы для всех  $A_i \in A^\wedge$ , то есть множества соответственно их реализуемых и распознаваемых слов во вход-выходных алфавитах;

–  $Fr^\wedge = \cup_{i \in I} Fr_i$  (где для любого  $i \in I$   $Fr_i \leq Be_i$ ) – формируемые в СА целевые сетевые тестовые фрагменты на основе сетевых контрольных ( $Id^\wedge$ ) и транспортных ( $R^\wedge, Tr^\wedge$ ) идентификаторов, то есть реализуемые и распознаваемые тестовые фрагменты, в частности, тестовые примитивы  $Pr^\wedge = \cup_{i \in I} Pr_i$  для всех  $A_i \in A^\wedge$  ( $Pr^\wedge \subseteq Fr^\wedge$ ), которые позволяют строить в СА контрольные и транспортные эксперименты  $Ex^\wedge = \cup_{i \in I} Ex_i$  (где для любого  $i \in I$   $Ex_i \leq Be_i$ ).

Для конструктивного описания контрольных идентификаторов состояний каждого  $A_i$  из СА использованы автоматы Рабина-Скотта [5] простого или общего [6] вида  $H(A_i) = (\Psi_{Hi}, R_i, U^+_i, \Delta_i, \psi_{0i}, F_i)$ .

В этом случае анализ поведения, представленного множеством вход-выходных последовательностей  $W_i$ , выполняется на основании априорной информации о входном  $X_i$  и выходном  $Y_i$  алфавитах, множествах приведенных начальных  $Id_{Hi}$  и конечных  $Id_{Ki}$  идентификаторов, отношениях совместимости  $\sigma_i$  и различимости  $\eta_i$ , определенных на объединенном множестве  $Id_i = Id_{Hi} \cup Id_{Ki}$  [5].

Прямая однозначность функций

$$\delta^\wedge = \cup_{i \in I} \delta_i, \lambda^\wedge = \cup_{i \in I} \lambda_{i-1}$$

и обратная неоднозначность функций

$$\delta^{\wedge^{-1}} = \cup_{i \in I} \delta_i^{-1}, \lambda^{\wedge^{-1}} = \cup_{i \in I} \lambda_i^{-1}$$

предполагает при построении  $\{R^\wedge, Tr^\wedge\}$  на основе идентифицируемого входного поведения для всех  $A_i \in A^\wedge$  использование множества  $G^\wedge = \cup_{i \in I} G(A_i)$  проверочных графов [5] одного из возможных видов [7]

$$G(A_i) = (B(S_i), Y_i \times X_i \times S_i^2, \Delta_i, \{S_i\}).$$

Сетевая модель тестирования  $CT$  принимает в качестве входных (выше представленных) данных тестовые примитивы для всех  $A_i \in A^\wedge$ . Алгоритмическая модель  $AM_i$  отдельного компонентного автомата  $A_i \in A^\wedge$  в

составе его модели  $CT_i$  тестирования и тестовой эволюции рассмотрена в [10].

В сетевой  $AM$  в отличие и в дополнение к множеству моделей  $AM_i$  для всех  $i \in I$  акцентируется определение и использование условий реализуемости и распознаваемости тестовых фрагментов всех  $A_i \in A^\wedge$ , в частности, примитивов в композиции СА.

Сетевая композиция моделей тестирования  $CT_i$  и эволюций для всех  $A_i \in A^\wedge$  в соответствии со структурой и отображениями сетевой модели тестирования  $CT$  дает возможность декомпозиции задачи синтеза тестов для СА в целом и, как следствие, полиномиального понижения NP-трудности этой задачи.

### Определение формальной сетевой модели эволюций

В сетевой модели основной объект эволюций – популяция особей ЭС – автономных тестовых эволюций для всех  $A_i \in A^\wedge$ .

В представлении эволюций в качестве базового конструктивного описания (метода синтеза) ФТ применена сетевая эволюционная система на основе генетического программирования (ЭС/ГП) для СА, определяющая механизмы псевдослучайного целенаправленного поиска оптимального развития и взаимодействия особей ЭС.

Особи ЭС – это эволюционирующие по внутренним (в  $A_i \in A^\wedge$ ) [10] и внешним (в СА) правилам контрольные и транспортные фрагменты и их группы (агрегации), для которых выполняются условия реализуемости и распознаваемости в СА.

В ЭС/ГП определена сигнатура сетевых генетических операций и функций.

Формально сетевая модель представления эволюций – сетевая модель эволюций ЭСА – включает три компонента с корректируемой интерпретацией значений фитнес-функций для всех  $A_i \in A^\wedge$  и в целом СА:

$$\text{ЭСА} = (Fr^\wedge, Vr^\wedge, \{\mu_n, \kappa_n, \varphi_n, \sigma_n\}),$$

где –  $Fr^\wedge = \cup_{i \in I} Fr_i$  (где для любого  $i \in I$   $Fr_i \leq Be_i$ ) – тестовые контрольные и транспортные фрагменты, в частности, примитивы  $Pr^\wedge = \cup_{i \in I} Pr_i$  для всех  $A_i \in A^\wedge$ , эволюционирующие по их внутренним правилам;

–  $Vr^\wedge = \cup_{i \in I} Vr_i$  (где для любого  $i \in I$   $Vr_i \leq Be_i$ ) – вирусы, то есть фрагменты поведения, соответствующие соответствующим  $A_i \in A^\wedge$  и играющие в ЭС/ГП роль внешних мутационных модификаторов фрагментов поведения  $A_i \in A^\wedge$ , модификация использует информацию, во-первых, структурированную под внешние цели и функции вирусов; во-вторых, модифицирующую или расширяющую основные свойства фрагментов поведения  $A_i \in A^\wedge$ ;

–  $\{\mu_n, \kappa_n, \varphi_n, \sigma_n\}$  – сигнатура сетевых действий в ЭС в составе двух операций и двух функций:

а)  $\mu_n: Fr_i \times Vr_i \rightarrow Fr_i$  – операция вирусной мутации;

б)  $\kappa_n: Fr_i \times Fr_i \rightarrow Fr_i$  – операция кроссинговера;

в)  $\varphi_n: Fr_i \times \Pi_i \rightarrow Fr_i'$  – фитнес функция;

г)  $\sigma_n: Fr_i' \times \Sigma_i \rightarrow Fr_i$  – функция выбора,

где  $Fr_i' = Fr_i \times N$  ( $N$  – множество натуральных чисел, характеризующих упорядочение выбора). В функциях  $\varphi_n$  и  $\sigma_n$  присутствуют управляющие параметры ЭС: приоритеты –  $\Pi_i$  и директивный выбор –  $\Sigma_i$ .

Множество  $AM_i$  для всех  $A_i \in A^\wedge$  [10] вместе с сетевой  $AM$  для  $CA$  позволяет строить объединенную двухуровневую модель тестирования, в которой иерархия моделей тестирования может определять сквозные условия синтеза ФТ, а иерархия ЭС/ГП – задавать сквозной метод синтеза ФТ.

### Реализация сетевой алгоритмической модели эволюций

В ходе реализации элементов информационной технологии тестирования РИС на базе предложенных сетевых моделей тестирования и эволюции разработан параллельный комплексный метод построения ФТ на основе ЭС/ГП, представляемых в виде сетевой композиции тестовых контрольных и транспортных фрагментов, которые допустимо реализовать и распознать в заданной автоматной сети.

В составе комплексного метода – совокупность задач:

– синтеза автономных контрольных фрагментов;

– синтеза автономных транспортных фрагментов;

– прямого распознавания сетевых тестовых фрагментов;

– обратной реализации сетевых тестовых фрагментов;

– агрегации сетевых тестовых фрагментов в целевые группы эволюции.

Детерминированные сетевые реализации частных методов для решения первых четырех задач, предложенные, в частности, в [3], позволили за счет декомпозиции снизить NP – сложность синтеза функциональных тестов. Последующее ее уменьшение может дать целенаправленный случайный поиск ЭС/ГП, который реализуется в представленной выше модели эволюции и принят в качестве базового метода для решения всех пяти задач.

В общем виде, применимом ко всем пяти задачам, представление базового метода в формальной модели незамкнутой эволюции ЭСА (без ограничений ресурсов), содержит следующие шаги:

*Шаг 1.* Формируется текущая популяция ЭСА, как начальная популяция ЭС – стартовая совокупность множеств начальных тестовых контрольных и транспортных примитивов всех  $A_i \in A^\wedge$  из  $CA$ .

*Шаг 2.* Для текущей популяции ЭСА – совокупности множеств вновь полученных контрольных, транспортных и объединенных (общих) фрагментов для  $A_i \in A^\wedge$  из  $CA$ , частей  $CA$  и  $CA$  в целом, а также вновь полученных агрегаций фрагментов запускаются процессы расчета фитнес-функций (возможность параллелизма), однокомпонентных для фрагментов и многокомпонентных для агрегаций.

Значения фитнес-функций в общем представлении прямо пропорциональны полноте тестирования и обратно пропорциональны длине и кратности (в агрегациях) фрагментов.

*Шаг 3.* На текущей общей совокупности всех тестовых контрольных, транспортных и общих фрагментов всех  $A_i \in A^\wedge$  из  $CA$ , частей  $CA$  и  $CA$  в целом, а также агрегаций фрагментов выполняется их частное, по

отдельным критериям, и полное, по интегральному критерию, упорядочение.

*Шаг 4.* Если получена целевая часть упорядоченной текущей популяции ЭС – часть упорядоченной совокупности множеств тестовых контрольных, транспортных и общих фрагментов для всех  $A_i \in A^{\wedge}$  из СА, частей СА и СА в целом, а также агрегаций фрагментов, удовлетворяющая критериям цели по полноте, длине и кратности, принимается решение о переходе к шагу 10.

*Шаг 5.* Для упорядоченной текущей популяции ЭСА – упорядоченной совокупности множеств тестовых контрольных, транспортных и общих фрагментов для всех  $A_i \in A^{\wedge}$  из СА, частей СА и СА в целом, а также агрегаций фрагментов запускаются процессы функций выбора (возможность параллелизма) для операций кроссинговера, мутации и агрегации.

*Шаг 6.* Для текущей популяции ЭСА – совокупности множеств тестовых контрольных, транспортных и общих фрагментов всех  $A_i \in A^{\wedge}$  из СА, частей СА и СА в целом, а также агрегаций фрагментов – все фрагменты, в том числе из состава агрегации, запускают два процесса-поиска (возможность параллелизма с другими фрагментами), формирующих новые экземпляры фрагментов (без ликвидации старых):

а) автономный процесс внутри компонентного развития фрагмента на основе автономной идентификации для внутри компонентного многоточечного кроссинговера;

б) сетевой процесс внешнего развития многокомпонентного фрагмента на основе сетевой идентификации для многокомпонентного многоточечного кроссинговера.

*Шаг 7.* Для текущей популяции ЭСА – совокупности множеств тестовых контрольных, транспортных и общих фрагментов всех  $A_i \in A^{\wedge}$  из СА, частей СА и СА в целом, а также агрегаций фрагментов запускаются процессы внутри компонентных мутаций (возможность параллелизма), расширяющие эволюцию новыми экземплярами фрагментов и агрегаций.

*Шаг 8.* Для текущей популяции ЭСА – совокупности множеств тестовых, транспортных и общих фрагментов всех  $A_i \in A^{\wedge}$  из СА, частей СА и СА в целом, а также агрегаций фрагментов запускаются сетевые процессы внешнего развития комбинаций многокомпонентных фрагментов (возможность параллелизма) на основе множественной сетевой совместимой агрегации, расширяющие эволюцию новыми тестовыми фрагментами и агрегациями.

*Шаг 9.* Переход к шагу 2.

*Шаг 10.* Завершение эволюции.

В случае параллельного представления пока экземпляр фрагмента или агрегации модифицируется, он не связывается ни в сети, ни в агрегации.

Операции и функции ЭС/ГП, используя свойства совместимости и идентификации, критерии полноты, длины и кратности тестирования, определяют предшествование и экстремумы фрагментов, агрегаций и популяции в целом в пространстве значений фитнес-функций.

### **Выводы**

Предложенная в работе сетевая алгоритмическая модель, основанная на взаимодействии модели тестирования и модели эволюции, выполняя конструктивное описание ФТ, позволяет полиномиально уменьшить:

– размерность (NP-трудность) функционального тестирования за счет сетевой декомпозиции модели тестирования РИС в выбранном классе ошибок;

– время синтеза ФТ, образующих покрытие тестовых примитивов, за счет применения псевдослучайного целевого поиска тестовых фрагментов в сетевой эволюции вместо детерминированного переборного поиска покрытия такого покрытия.

Возможность ограничения ресурсов ЭС – числа особей и агрегаций, частной для особей, групповой для агрегаций и общей для популяции длины и кратности, параметры фитнес-функций и функций выбора, а также возможность останова эволюции при достижении определенных показателей полноты тестирования, длины и кратности, времени синтеза обеспечивают завершение построения функциональных тестов.

К достоинствам предлагаемого подхода, кроме уменьшения размерности и сокращения времени синтеза тестов, можно отнести относительную простоту и параллелизм вычислительной организации поиска в ЭС. Недостатками остаются высокая сложность построения тестовых контрольных и транспортных примитивов и фрагментов, усиленная условиями реализуемости и распознаваемости в СА, а также недетерминированность получения ФТ необходимых полноты и ресурсоемкости в псевдослучайном целевом поиске ЭС.

Экспериментальная программно-алгоритмическая реализация модели и метода, выполненная с применением компонентного программирования для сервисов мониторинга и контроля работы протокольных реализаций в средствах РИС, подтвердила целесообразность развития исследований и реализаций в данном направлении.

#### Список использованной литературы

1. Bakker A.A., Steen M., and Tanenbaum A.S., (2006), Wide-Area Distribution Network for Free Software, *ACM Trans. on Internet Technology*, Vol. 6, Aug. 2006, pp. 259 – 281.

Url: <http://www.cs.vu.nl/~ast/publications/transit-2006.pdf>.

2. Ranjan Praveen, and Srivastava Km Baby, (2010), Automated Software Testing Using Metaheuristic Technique Based on An Ant Colony Optimization Electronic System Design (ISED), *International Symposium*, 20-22 Dec. 2010, Bhubaneswar, pp. 235 – 240.

Url: <http://arxiv.org/ftp/arxiv/papers/1106/1106.2686.pdf>.

3. Thomas Müller, Rex Black, Sigrid Eldh, Dorothy Graham, Klaus Olsen, Maaret Pyhäjärvi, Geoff Thompson, and Erik van Veendendal, (2005), Certified Tester Foundation Level Syllabus. Version 2005, *International Software Testing Qualifications Board*. 2005. – 73 p.

Url: <http://www.bcs.org/upload/pdf/istqbsyll.pdf>.

4. Mats Grindal, (2007), Handling Combinatorial Explosion in Software Testing, *Printed by LiU-Tryck, Linköping*, 89 p.

Url: <http://www.divaportal.org/smash/get/diva2:17568/FULLTEXT01.pdf>.

5. Кудрявцев В. Б. Анализ поведения автоматов / В. Б. Кудрявцев, И. С. Грунский, В. А. Козловский. // *Дискретная математика* – М. : РАН, Отделение математических наук. Т. 21. – № 1. – 2009. – С. 3 – 35.

Url: [http://intsys.msu.ru/magazine/archive/v10\(14\)/grunskiy-345-448.pdf](http://intsys.msu.ru/magazine/archive/v10(14)/grunskiy-345-448.pdf).

6. Мартынюк А. Н. Сетевые автоматные модели синтеза тестов / А. Н. Мартынюк // *Холодильна техніка і технологія*. – Одеса : Изд-во Одеська Академія Харчових Технологій. – 2007. – № 2(106). – С. 94 – 97.

7. Мартынюк А. Н. Модели тестирования для композиций сетей Петри, / А. Н. Мартынюк, Васим Аль Шариф // *Холодильна техніка і технологія*. – Одесса : Изд-во Одеська Академія Харчових Технологій – 2013. – № 2(142). – С. 84 – 87.

8. Снитюк В. Е. Аспекты эволюционного моделирования в задачах оптимизации / В. Е. Снитюк // *Искусственный интеллект*. – М. : Изд-во Наука. – 2005. – № 4. – С. 284 – 291.

Url: [http://artint.com.ua/pdf/Russian/evolmodel\\_rus.pdf](http://artint.com.ua/pdf/Russian/evolmodel_rus.pdf).

9. Riccardo Poli, William B. Langdon, Nicholas F. McPhee, and John R. Koza, (2008), A Field Guide to Genetic Programming, *Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 UK: England March 2008* – 250 p.

Url: [http://dces.essex.ac.uk/staff/rpoli/gp-field-guide/A\\_Field\\_Guide\\_to\\_Genetic\\_Programming.pdf](http://dces.essex.ac.uk/staff/rpoli/gp-field-guide/A_Field_Guide_to_Genetic_Programming.pdf).

10. Мартинюк О. М. Модель тестування еволюційної системи, / О. М. Мартинюк, Г. С. Сугак, О. В. Дрозд // *Науковий вісник Чернівецького національного ун-ту. Серія : Комп'ютерні системи та компоненти*. – 2014. – Т. 5. – Вип. 1. – Чернівці : Видавн. Чернівецький нац. ун-т. – С. 49 – 53.

Получено 30.10.2014

1. Bakker A.A., Steen M., and Tanenbaum A.S., (2006), Wide-Area Distribution Network for Free Software, *ACM Trans. on Internet Technology*, Vol. 6, Aug. 2006, pp. 259 – 281.

Url:<http://www.cs.vu.nl/~ast/publications/trans-it-2006.pdf>.

2. Ranjan Praveen, and Srivastava Km Baby, (2010), Automated Software Testing Using Metaheuristic Technique Based on An Ant Colony Optimization Electronic System Design (ISED), *International Symposium*, 20-22 Dec. 2010, Bhubaneswar, pp. 235 – 240. Url:<http://arxiv.org/ftp/arxiv/papers/1106/1106.2686.pdf>.

3. Thomas Müller, Rex Black, Sigrid Eldh, Dorothy Graham, Klaus Olsen, Maaret Pyhäjärvi, Geoff Thompson, and Erik Van Veendendal, (2005), Certified Tester Foundation Level Syllabus. Version 2005, *International Software Testing Qualifications Board*. 2005. – 73 p.

Url:<http://www.bcs.org/upload/pdf/istqbsyll.pdf>.

4. Mats Grindal, (2007), Handling Combinatorial Explosion in Software Testing, *Printed by LiU-Tryck*, Linköping. 2007. – 89 p. Url: <http://www.divaportal.org/smash/get/diva2:17568/FULLTEXT01.pdf>.

5. Kudryavtsev V.B., Grunskii I.S., and Kozlovskii V.A. Analiz Povedeniya Avtomatov [Automata Behavior Analysis], (2009), *Diskretnaya Matematika*, Moscow, Russian Federation, RAN, Otdelenie Matematicheskikh Nauk. Tom 21, No. 1. pp. 3 – 35 (In Russian). Url: [http://intsys.msu.ru/magazine/archive/v10\(1-4\)/grunskiy-345-448.pdf](http://intsys.msu.ru/magazine/archive/v10(1-4)/grunskiy-345-448.pdf).

6. Martynyuk A.N. Setevye avtomatnye modeli sinteza testov [Network Automata Models for Test Synthesis], (2007), *Kholodil'na Tekhnika I Tekhnologiya*, Odesa, Ukraine, *Isd. Odesk. Akad. Harch. Tehnolog.* No. 2(106), pp. 94 – 97 (In Russian).

7. Martynyuk A.N., Vasim Al Sharif Modeli testirovaniya dlya kompozitsii setei Petri [Models of Testing for Petri Net Compositions], (2013), *Kholodil'na Tekhnika I Tekhnologiya, Isd. Odesk. Akad. Harch. Tehnolog.*, Odesa, Ukraine, No. 2(142), pp. 84 – 87 (In Russian).

8. Snityuk V.E. Aspekty evolyutsionnogo modelirovaniya v zadachakh optimizatsii [Aspects

of Evolutionary Modeling in Optimization Problems], (2005). *Iskusstvennyi Intellekt, Nauka Publ.*, Moscow, Russian Federation, *Isd. Nauka*, No. 4, pp. 284 – 291 (In Russian). . Url:[http://artint.com.ua/pdf/Russian/evolmodel\\_rus.pdf](http://artint.com.ua/pdf/Russian/evolmodel_rus.pdf).

9. Riccardo Poli, William B. Langdon, Nicholas F. McPhee, and John R. Koza, (2008), A Field Guide to Genetic Programming, *Creative Commons Attribution-Noncommercial-No Derivative Works 2.0 UK*: England March 2008 – 250 p.

Url: [http://dces.essex.ac.uk/staff/rpoli/gp-field-guide/A\\_Field\\_Guide\\_to\\_Genetic\\_Programming.pdf](http://dces.essex.ac.uk/staff/rpoli/gp-field-guide/A_Field_Guide_to_Genetic_Programming.pdf).

10. Martynyuk A.N., Sugak A.S., and Drozd A.V. Model' testuvannya evoljucijnoi systemy [Model of Testing for Evolution System], (2014), *Naukovyj Visnyk Chernivc'kogo Nacional'nogo Universytetu, Vipusk "Komp'yuterni systemy ta komponenty"*, Chernivtzi, Ukraine, Tom 5, Vip. 1, *Vidavn. Chen. Natz. Univ.* pp. 48 – 53 (In Ukrainian).



Сугак  
Анна Сергеевна;  
аспирант каф. компьютерных интеллектуальных систем и сетей Одесского нац. политехнич. ун-та,  
тел.: +38(093) 705-15-93.  
E-mail:  
[sugak.anna@mail.ru](mailto:sugak.anna@mail.ru)



Мартынюк  
Александр Николаевич,  
канд. техн. наук, доц.  
каф. компьютерных интеллектуальных систем и сетей Одесского нац. политехн. ун-та,  
тел.: (067) 489-81-69  
E-mail:  
[anmartynyuk@ukr.net](mailto:anmartynyuk@ukr.net)