

# ДИАГНОСТИРОВАНИЕ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СОВРЕМЕННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Л.С. Ломакина, В.П. Губернаторов

Нижегородский государственный технический университет им. Р.Е. Алексева,  
ул. Минина, 24, Нижний Новгород, 603950, Российская Федерация; e-mail: llomakina@list.ru

Рассмотрена задача оптимального диагностирования сложных технических систем. Для решения задачи разработана модификация эволюционно-генетического алгоритма. Выполнено сравнение разработанной модификации с алгоритмами, основанными на концепциях динамического программирования и имитации отжига.

**Ключевые слова:** техническое диагностирование, эволюционно-генетический алгоритм, метод имитации отжига, динамическое программирование

## Введение

Современные сложные технические системы решают задачи, требующие высокого уровня надёжности их функционирования. Одним из методов обеспечения надёжности является диагностирование систем на заданном наборе тестов. Выполнение каждого теста связано с затратой материальных ресурсов, поэтому при необходимости диагностирования, возникает задача построения оптимальной тестовой последовательности, обеспечивающей минимум затрат на определение заданного числа отказов технической системы.

Данной проблеме посвящено большое количество работ отечественных и зарубежных авторов: Г.Ф. Верзаков, П.П. Пархоменко, В.И. Сагунов, А.Ю. Аржененко, Д.В. Сперанский, В.В. Сапожников, М.А. Владимиров, J. Wegener, J. Ribero, A. Arcury, J. Shiozaky, и др. На основании обзора литературных источников можно сделать вывод, что рассматриваемая задача является комбинаторной и может быть решена с помощью известных методов дискретной оптимизации [1], однако при диагностировании сложных систем эти методы не всегда практически реализуемы, так как требуют трудоёмких вычислений. Когда использование точных методов оптимизации становится невозможным, практическую значимость приобретает приближённый подход. В ряде работ предлагается использовать метаэвристические алгоритмы: имитацию отжига, моделирование муравьиной колонии и восхождение к экстремуму, однако рассмотренные методы носят частный характер, поэтому актуальной является разработка нового эффективного метода диагностирования сложных систем, основанного на применении обобщенной модели и современных приближённых методов оптимизации.

Предлагается модификация эволюционно-генетического алгоритма, адаптированная для решения задачи построения оптимальной тестовой последовательности и выполняется сравнение предложенной модификации с другими известными методами.

## Базовая модель сложной системы

Для разработки обобщенного метода оптимального диагностирования сложных технических систем в качестве базовой модели будем использовать графо-матричную модель.

Пусть  $T$  — множество тестов, разработанных для проверки диагностируемой системы. Каждый тест  $t_i \in T$  позволяет определить работоспособность и может быть представлен в виде множества  $\{(X_a, Y_d, Z_d)_g\}$ , где  $X_a$  — определенное воздействие на систему,  $Y_d$  — ожидаемая реакция на это воздействие,  $Z_d$  — контрольная точка, на которой производится сравнение с  $Y_d$  реакции объекта на воздействие  $X_a$ .

Представим сложную техническую систему в виде ориентированного графа  $G(B, U)$  с  $N$  вершинами. Обозначим  $B$  — множество вершин графа,  $U$  — множество рёбер графа. Если можно выделить конструктивные или функциональные блоки объекта, вершины графа соответствуют структурным блокам системы, а рёбра графа — связям между блоками. Если выделить блоки затруднительно — вершины графа соответствуют параметрам системы, а рёбра графа — причинно-следственным связям между параметрами.

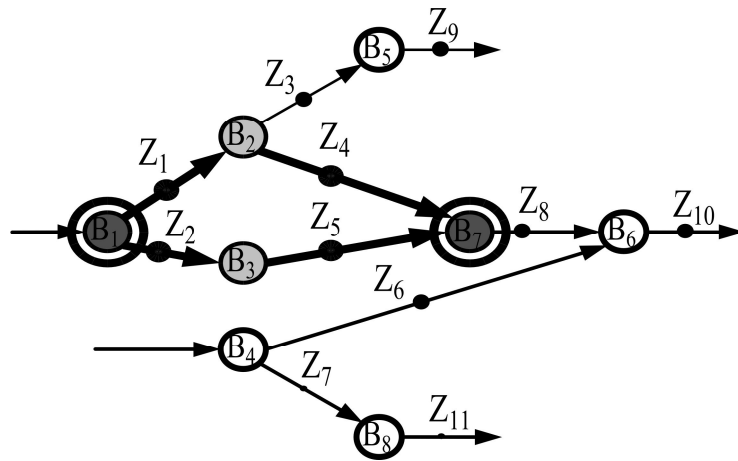
Далее будем рассматривать граф блочной структуры, для графа причинно-следственных связей моделирование осуществляется аналогично.

Пронумеруем блоки объекта, диагностирования и разметим вершины графа  $G(B, U)$  соответствующими номерами  $\{1, \dots, N\}$ . Отметим на графе контрольные точки, соответствующие тестам  $t_i \in T$ . Контрольные точки представляются в виде разметки рёбер графа  $G(B, U)$ . Ребро  $U_\gamma = (B_\alpha, B_\beta)$  размечено контрольной точкой  $Z_\gamma$ , тогда и только тогда, когда при выполнении одного из тестов  $t_i \in T$  осуществляется оценка диагностического параметра  $Y_\gamma$ , являющегося выходным для блока  $B_\alpha$  и входным для блока  $B_\beta$ .

На графе  $G(B, U)$  тест  $t_i$  изображается в виде множества контрольных пар вершин  $(B_\delta, B_\gamma)$ , где  $B_\delta$  — вершина, соответствующая блоку, на вход которого подаётся входной сигнал  $X_{ag}$ ;  $B_\gamma$  — вершина, из которой выходит дуга  $U$ , соответствующая контрольной точке  $Z_{dg}$ . Тест  $t_i$  контролирует состояние блока  $B_j$  тогда и только тогда, когда в графе  $G(B, U)$  существует хотя бы один путь из вершины  $B_\delta$  в вершину  $B_\gamma$ , проходящий через вершину  $B_j$  (рис. 1).

Каждый элементарный тест позволяет установить исправность или неисправность группы из  $k$  контролируемых блоков. Остальные  $(N - k)$  блоков остаются непроверенными.

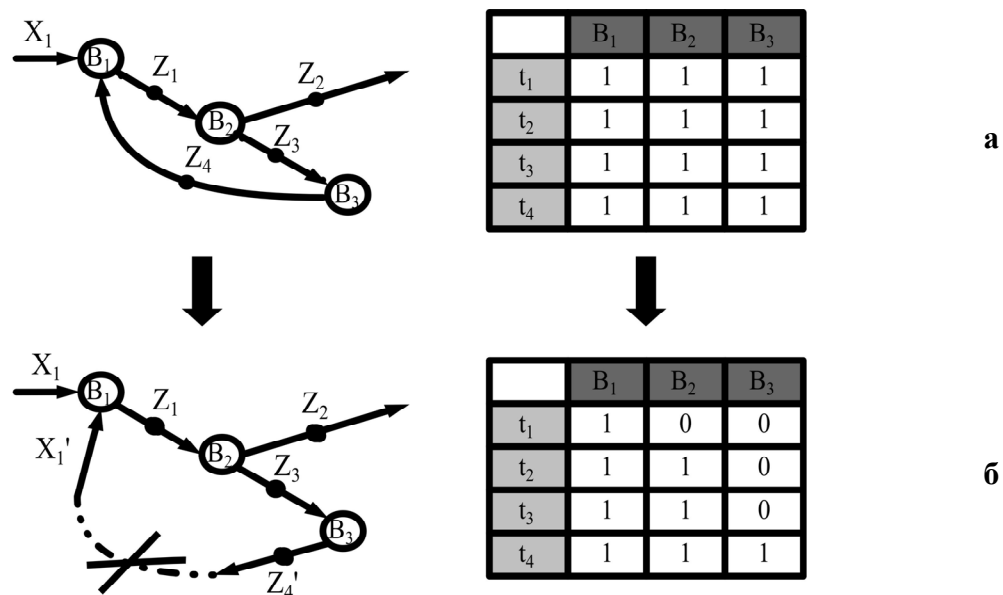
Если число блоков диагностируемой системы  $N$ , тест  $t_i \in T$  может быть представлен в виде  $N$ -мерного вектора  $V_i$ . Ненулевое значение  $j$ -й компоненты вектора  $V_i$  означает, что  $j$ -й блок системы контролируется данным тестом и является исправным, если результат теста положительный. Нулевые значения компонент вектора  $V_i$  соответствует блокам, неконтролируемым элементарным тестом  $t_i$ . При положительном результате теста и неисправном состоянии всей исследуемой системы в целом формулируется вывод о том, что неисправен, по меньшей мере, один из  $(N - k)$  неконтролируемых блоков.



**Рис. 1.** Представление теста  $t_i = \{(X_1, Y_8, Z_8)\} \longrightarrow \{(B_1; B_7)\}$  на графе сложной системы

Положим, что  $|T| = M$ , и на основе модели теста построим модель сложной системы в виде матрицы идентификации. Матрица идентификации – это прямоугольная матрица  $A_{M \times N} = \|a_{ij}\|$  столбцам, которой соответствуют блоки исследуемой системы  $B_j$ , а строкам тесты  $t_i$ .

Если два или более столбца в матрице  $A$  совпадают, то неисправности в соответствующих им блоках являются неразличимыми (рис. 2(а)). Описанная ситуация может возникнуть из-за ориентированных циклов в графе  $G(B, U)$ . Данные циклы свидетельствуют о наличии обратных связей между блоками исследуемой системы. Для возможности обнаружения неисправности данных блоков, необходимо вводить контролируемые разрывы в контуры обратных связей (рис. 2(б)).



**Рис. 2.** Введение контролируемых разрывов в граф исследуемой системы для идентификации состояния блоков, входящих в конуры обратных связей

Выбранная модель представляет информацию о структуре системы, возможных неисправностях, разработанных тестах и стоимости их исполнения. Данная модель позволяет представить тестовую последовательность в виде нагруженного дерева поиска  $S = s(\tau \subseteq T)$ , листьями которого являются номера неисправных блоков системы  $i \in [1, N]$ , а каждой внутренней вершине соответствует тест из некоторого подмножества  $\tau$  множества тестов  $T = \{t_1 \dots t_M\}$ .

При известной стоимости выполнения каждого теста  $C(t_i \in T) = C_i$  и вероятности неисправности каждого блока  $P(B_i, i \in [1, N]) = P_i$  можно вычислить стоимость исполнения тестовой последовательности:

$$C(S) = \sum_{z \in Z \subseteq N} C(B_z) P(B_z), \quad (1)$$

где

$Z$  — множество блоков диагностируемой системы, состояния которых могут быть идентифицированы путём выполнения последовательности  $S$ ;

$C(B_z)$  — стоимость идентификации состояния блока  $z$ , определяемая как суммарная стоимость тестов, принадлежащих пути от корня дерева  $S$  до листа  $B_z$ .

В качестве глубины диагностирования системы, достигаемой при исполнении тестовой последовательности  $S$ , рассматривается величина:

$$\phi(S) = \frac{\sum_{z \in Z \subseteq N} P(B_z)}{\sum_{i \in N} P(B_i)}, \quad (2)$$

где  $Z$  — множество блоков диагностируемой системы, состояния которых могут быть идентифицированы путём выполнения последовательности  $S$ .

*Постановка задачи:* найти тестовую последовательность  $S = s(\tau \subseteq T)$ , позволяющую минимизировать целевую функцию  $C(S)$ , при условии  $\phi(S) = \phi_0$ .

### Алгоритмы построения оптимальных тестовых последовательностей

Для решения поставленной задачи построим модификацию классического эволюционно-генетического алгоритма [2], в качестве альтернативных методов решения для последующего сравнения рассмотрим алгоритмы на основе концепций имитации отжига и динамического программирования.

#### Модификация эволюционно-генетического алгоритма (ЭГА)

Формально эволюционно-генетический алгоритм можно определить следующим образом:

$$\text{ЭГА} = (P^0, K, \lambda, L, Sl, R, f, k), \quad (3)$$

где

$P^0 = \{x_1^0, x_2^0, \dots, x_\lambda^0\}$  — начальная популяция;

$x_\gamma$  — потенциальное решение задачи, представленное в виде хромосомы;

$\lambda$  — размер популяции;

$K$  — биективное отображение множества допустимых решений во множество хромосом, определяющее способ кодирования;

$L$  — длина хромосомы;

$Sl$  — операторы селекции;

$R$  — операторы рекомбинации;

$f = f(x)$  — функция приспособленности (целевая функция для эволюционной оптимизации);

$k$  — критерий останова.

Модификация эволюционно-генетического алгоритма заключается в выборе параметров (3), согласно условиям поставленной задачи.

Рассмотрим в качестве генома популяции множество номеров тестов  $J = \{1, 2, \dots, j, \dots, M\}$ . *Допустимой хромосомой* будем считать любую перестановку из элементов этого множества.

Тест, соответствующий первому гену  $x_0^j$ , становится корнем дерева тестовой последовательности, при этом множество всех блоков исследуемой системы разбивается на классы  $\{B_1^0, B_2^0, \dots, B_j^0, \dots, B_H^0\}$ , и образуется  $H$  ветвей дерева, по которым расположены подозрительные на неисправность блоки диагностируемого объекта. Просматриваются все ветви, тест, соответствующий следующему гену  $x_1^j$ , добавляется к ветви  $j$  только в том случае, если он позволяет выполнить дальнейшее разбиение подмножества  $B_j^0$  на классы  $\{B_1^1, B_2^1, \dots, B_j^1, \dots, B_H^1\}$ . Затем, операция повторяется для тестов, соответствующих оставшимся генам  $x_2^j, \dots, x_k^j, \dots, x_M^j$ .

Процесс прекращается при условии:

$$(\forall(i, j): |B_j^i| = 1) \vee (\phi(S_\gamma) = \phi_0) \vee (k = M). \quad (4)$$

Предложенный метод кодирования позволяет получать хромосомы одинаковой длины  $L = M$  и использовать не декодируемую часть хромосомы, как дополнительное средство выхода из локальных оптимумов.

В качестве операторов  $R$  из различных операторов кроссинговера и мутации были выбраны *жадный и упорядоченный операторы кроссинговера*. Совместное применение выбранных операторов позволяет *комбинировать случайный и направленный поиск* (рис. 3) и обеспечивает достаточное условие выхода из локальных оптимумов:

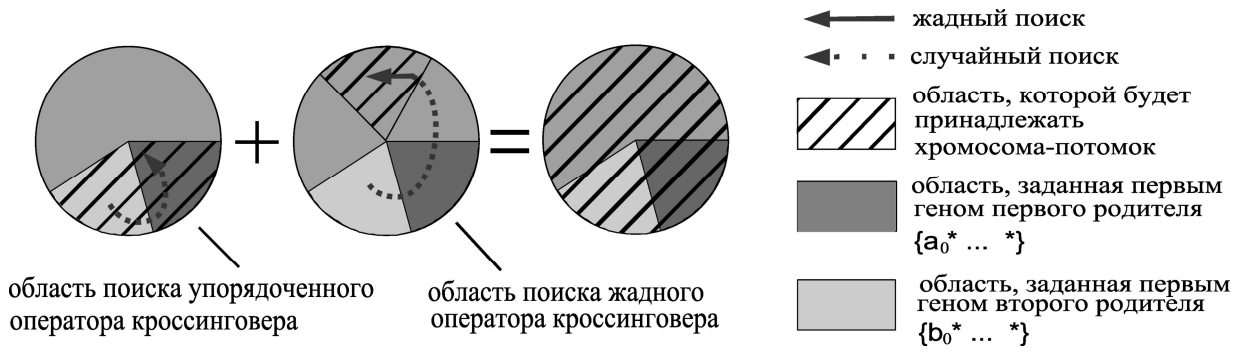
$$\forall \gamma : p(x_\gamma : x_\gamma \xrightarrow{k} (S_\gamma \in U)) \neq 0, \quad (5)$$

где

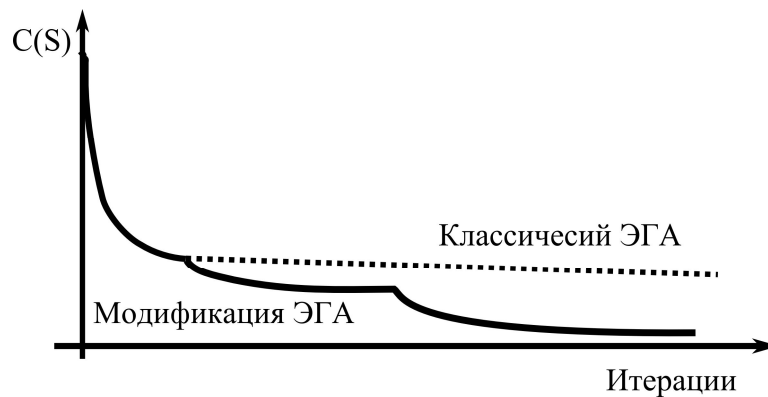
$U$  — множество тестовых последовательностей, удовлетворяющих условиям задачи;

$p(x_\gamma)$  — вероятность появления хромосомы  $x_\gamma$  после применения генетических операторов  $R$ .

Для ускорения выхода из локальных оптимумов предлагается использовать адаптивную поисковую стратегию. При схождении популяции вторые операнды операторов  $R$  генерируются случайно, а не выбираются из текущей популяции. При длительном нахождении в состоянии схождения к популяции применяется оператор геноцида. Предложенная поисковая стратегия основана на принципе *элитизма*, позволяет ускорить выход локальных оптимумов и обеспечивает более быстрое схождение по сравнению с классическими ЭГА, использующими кроссинговер и мутацию (рис. 4).



**Рис. 3.** Область эволюционно-генетического поиска при совместном применении жадного и упорядоченного операторов кроссинговера



**Рис. 4.** График схождения разработанной модификации и классического ЭГА

В качестве оператора  $SI$  предлагается использовать простую элитную селекцию на основе сравнения значений функции приспособленности  $f(x) = C(s)$ , а начальную популяцию  $P^0$  формировать путём применения фрактального оператора кроссинговера к двум случайно сгенерированным решениям. Такой подход позволит эффективно использовать жадный оператор кроссинговера на первой итерации алгоритма, обеспечивать уникальность начальных решений и автоматически определять размер родительской популяции, как функцию от числа строк матрицы идентификации –  $M$ . Останов будем осуществлять при достижении желаемого значения целевой функции  $C(S)$  либо по истечении заданного числа итераций.

#### Имитация отжига (ИО)

Метод имитации отжига является одним из наиболее эффективных методов случайного поиска решения широкого спектра задач. В работе [3] указывается, что на большинстве задач метод имитации отжига не проигрывает эволюционно-генетическим алгоритмам, а на многих оказывается эффективнее. Основным преимуществом метода имитации отжига является возможность выхода из локальных оптимумов. Это достигается за счет принятия решений не только в сторону улучшения, но и в сторону ухудшения целевой функции в зависимости от температуры отжига  $\Delta$ , моделируемого процесса закалки.

Общая схема алгоритма имитации отжига.

1) Выбирается температура отжига в виде дискретного отрезка  $[\Delta_0, \dots, \Delta_n]$ , устанавливается текущая температура  $\Delta' = \Delta_i, i = n$ .

2) Выбирается начальное решение  $x_0$ , оно становится текущим  $x' = x_0$ .

3) С помощью преобразования  $\theta$  получается решение-кандидат  $x'' = \theta(x')$ .

4) Вероятность, с которой решение  $x'' = \theta(x')$  станет текущим, равна  $P(x', x'')$ , где  $P$  — распределение Гиббса:

$$P(x', x'' | x') = \begin{cases} 1 & \text{при } f(x'') \leq f(x') \\ e^\lambda & \text{при } f(x'') > f(x') \end{cases} \quad (6)$$

$$\lambda = -\frac{f(x'') - f(x')}{\Delta'}$$

где  $f(x)$  — значение целевой функции, соответствующее решению  $x$ .

5) Понижаем температуру  $i = i - 1, \Delta' = \Delta_i$ .

6) Процесс прекращается при  $\Delta' = \Delta_0$ , в противном случае переходим к шагу 3.

В начале поиска температура имеет наибольшие значения, и величина  $e^\lambda$  близка к единице, поэтому велика вероятность выбора решения с худшим значением целевой функции  $f(x)$ . При высоких температурах  $\Delta_i, i > n/2$  решение  $x'' = \theta(x')$  может быть выбрано текущим при  $f(x'') > f(x')$ , что может привести к попаданию в локальный оптимум при дальнейшем уменьшении температуры. Чтобы избежать описанной ситуации, предлагается на каждой итерации алгоритма сохранять наилучшее из найденных решений.

#### Динамическое программирование (ДП)

Метод подробно рассмотрен в работе [2]. Для построения оптимальной тестовой последовательности используется уравнение Беллмана вида:

$$C_{opt}(k_{\beta x}, T_{\beta x}) = C_{opt}(S_{\beta x}) = \min_i \left( C(t_i \in T_{\beta x}) + \sum_j P_j \cdot C_{opt}(k_{jh}, T_{jh}) \right), \quad (7)$$

$$P_j = \frac{\sum_{B_y \in k_{jh}} P(B_y)}{\sum_{B_z \in k_{\beta x}} P(B_z)}$$

где

$(k_{\alpha h}, T_{\alpha h})$  — ситуация  $\alpha$ , порядка  $h = |k_{\alpha h}|$ ;

$T_{\alpha h} \subseteq T$  — множество тестов, позволяющих выполнить дальнейшее разбиение подмножества  $k_{\alpha h} \in K$ ;

$K$  — Совокупность подмножеств блоков  $B^i$ , соответствующих всевозможным сочетаниям тестов из  $T$ ;

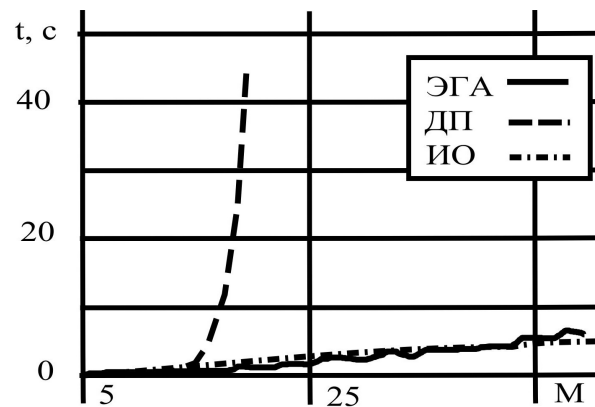
$t_i$  — тест из множества  $T_{\beta x}$ ;

$k_{jh}$  — подмножества, на которые  $t_i$  разбивает множество  $k_{\beta x}$ ;

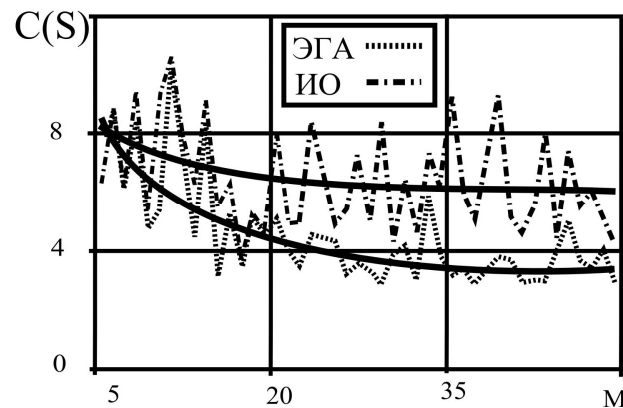
$P_j$  — условная вероятность наличия неисправности в блоках  $B_y \in k_{jh}$ .

## Вычислительный эксперимент

Рассмотрим оценки вычислительной сложности (рис. 5(а)) и сходимости (рис. 5(б)) разработанной модификации ЭГА, ДП и ИО. Модификация ЭГА обладает квадратичной сложностью от числа разработанных тестов  $M$  и линейной относительно числа блоков системы  $N$ . Возможно снижение сложности до  $O(M^2/K)$  при использовании  $K$  параллельных вычислителей. Сложность ДП экспоненциальная относительно  $M$  и линейная относительно  $N$ . ИО имеет линейную сложностью относительно  $M$  и  $N$ . Сравнение целевой функции оптимальной тестовой последовательности, получаемой в результате применения модификации ЭГА и ИО, позволяет сделать вывод, что ЭГА эффективнее ИО.



а

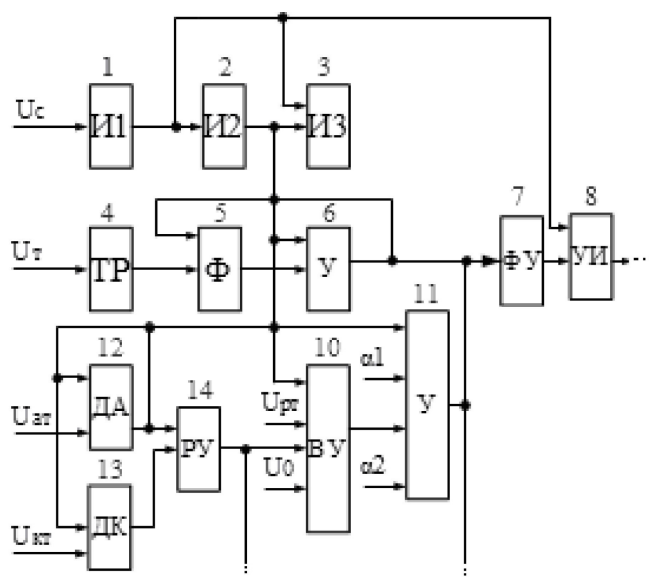


б

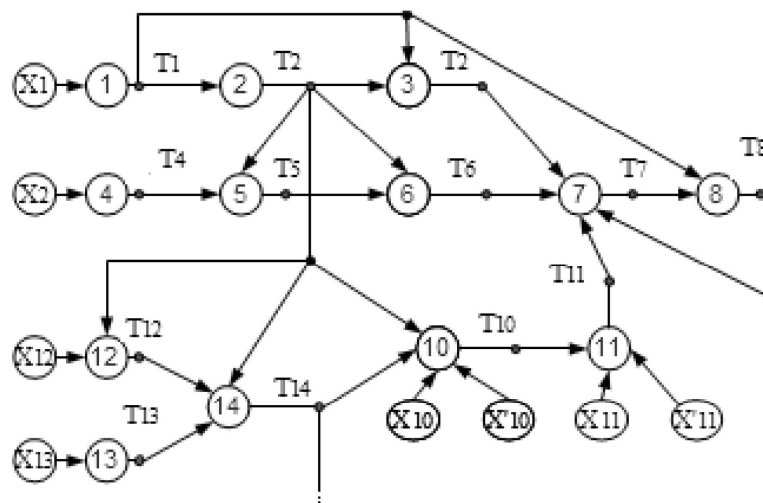
**Рис. 5.** а – оценка времени работы ЭГА, ДП, ИО от числа тестов  $M$ ; б – сравнение значения целевой функции оптимального решения для ИО и ЭГА

В качестве примера рассмотрен автоматизированный электропривод [4], функциональная схема которого изображена на рис. 6(а), граф на рис. 6(б), матрица идентификации на рис. 6(в). Точками контроля являются выходы функциональных блоков. Для каждого блока устанавливаются зависимости между входными и выходными сигналами, а также допустимые диапазоны этих сигналов. Функциональный блок считается дефектным, если при допустимых входных сигналах на выходе данного блока наблюдается недопустимый сигнал. Стоимость тестов соответствует временным затратам на отключение и подключение измерительной аппаратуры. Вероятности неисправностей блоков выбираются, исходя из насыщенности данных блоков элементами.





а



б

| Тесты | Блоки элементов |      |      |      |  | Б48  | с(Т) |
|-------|-----------------|------|------|------|--|------|------|
|       | Б1              | Б2   | Б3   | Б4   |  |      |      |
| Т1    | 1               | 0    | 0    | 0    |  | 0    | 0,04 |
| Т2    | 1               | 1    | 0    | 0    |  | 0    | 0,02 |
| Т3    | 1               | 1    | 1    | 0    |  | 0    | 0,01 |
| Т48   | 1               | 1    | 0    | 0    |  | 1    | 0,03 |
| Р(Б)  | 0,04            | 0,06 | 0,04 | 0,01 |  | 0,18 |      |

в

Рис. 6. Модель автоматизированного электропривода

С помощью предложенных в работе алгоритмов были получены результаты, представленные в таблице. Для ДП результат получить не удалось, так как исполнение алгоритма потребовало более 2 Гб ОЗУ.

Таблица 1.

## Результаты экспериментов

|     | $C(S)$ | $t, c$ |
|-----|--------|--------|
| ДП  | -      | >7200  |
| ЭГА | 28.4   | 10     |
| ИО  | 31.7   | 15     |

Результаты вычислительных экспериментов показали, что предложенный подход позволяет использовать возможности современных вычислительных систем и может эффективно применяться для оптимизации числа контрольных точек при проектировании сложных технических систем, либо для минимизации затрат на тестирование в процессе их эксплуатации.

**Выводы**

1) Разработана модификация ЭГА, для решения задачи построения оптимальных тестовых последовательностей, отличающаяся от существующих: способами кодирования решений и построения целевой функции; использованием специальными генетических операторов. Модификация позволяет рассматривать тестовые последовательности различной длины, учитывать заданную глубину диагностирования и может применяться при неполном или избыточном наборе тестов.

2) Рассмотрены альтернативные методы решения поставленной задачи на основе концепции динамического программирования и метода имитации отжига.

3) Экспериментально получены оценки сходимости и вычислительной сложности, позволяющие сделать вывод об эффективности предложенной модификации ЭГА, по сравнению с другими известными алгоритмами.

4) На основе предложенной модификации разработан новый метод оценки оптимального решения, позволяющий решить задачу диагностирования с наименьшим использованием вычислительных ресурсов, по сравнению с другими известными методами.

**Список литературы**

1. Сигал, И.Х. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы [Текст] : учеб. для студ. вузов, обуч. по напр. и спец. «Прикладная математика и информатика» / И.Х. Сигал, А.П. Иванова. — Изд. 2-е, испр. и доп. — М. : Физматлит, 2007. — 304 с.
2. Ломакина, Л.С. Модификация эволюционно-генетического алгоритма для эффективного диагностирования сложных систем / Л.С. Ломакина, В.П. Губернаторов // Системы управления и информационные технологии. — 2013. — Т. 53, № 3. — С. 59–64.
3. Inber, L. Genetic Algorithms and Very Fast Simulated Reannealing: A comparison / L. Inber, B. Rosen // Mathematical and Computer Modelling. — 1992. — Vol. 16, Iss. 11. — PP. 87–100.
4. Осипов, О.И. Техническая диагностика автоматизированных электроприводов : научное издание / О.И. Осипов, Ю.С. Усынин. — М. : Энергоатомиздат, 1991. — 160 с.

**ДІАГНОСТУВАННЯ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ З ВИКОРИСТАННЯМ СУЧАСНИХ  
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

Л.С. Ломакіна, В.П. Губернаторов

Нижегородський державний технічний університет ім. Р.С. Алексєєв,  
вул. Мініна, 24, Нижній Новгород, 603950, Російська Федерація; e-mail: llomakina@list.ru

Розглянуто задачу оптимального діагностування складних технічних систем. Для вирішення завдання розроблена модифікація еволюційно-генетичного алгоритму. Виконано порівняння розробленої модифікації з алгоритмами, заснованими на концепціях динамічного програмування та імітації відпалу.

**Ключові слова:** технічне діагностування, еволюційно-генетичний алгоритм, метод імітації відпалу, динамічне програмування

**STRUCTURAL TESTING OF SOFTWARE SYSTEMS BASED ON COMPUTER ALGEBRA  
ELEMENTS**

Lyubov S. Lomakina, Vladimir P. Gubernatorov

Nizhny Novgorod State Technical University n.a. R.E. Alekseev,  
24 Minina str., Nizhny Novgorod, 603950, Russian Federation; e-mail: llomakina@list.ru

The problem of complex technical systems optimal diagnosis is concerned. The evolutionary algorithm modification for optimal diagnosis is proposed. The proposed approach is compared with the simulated annealing and dynamic programming methods.

**Keywords:** technical diagnosis, evolutionary algorithm, simulated annealing, dynamic programming