

На правах рукописи

Панфилова Татьяна Александровна

**СТОХАСТИЧЕСКИЕ АДАПТИВНЫЕ АЛГОРИТМЫ
ПОВЫШЕНИЯ НАДЕЖНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Специальность: 05.13.01 – Системный анализ, управление и обработка информации (космические и информационные технологии)

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Красноярск – 2017

Работа выполнена в Сибирском государственном университете науки и технологий им. академика М.Ф. Решетнева.

Научный руководитель – доктор технических наук, профессор
Ковалев Игорь Владимирович

Официальные оппоненты – **Терсков Виталий Анатольевич**
доктор технических наук, профессор
Красноярский институт железнодорожного
транспорта, филиал ФГБОУ ВО ИРГУПС
профессор кафедры управления персонала

Кравец Олег Яковлевич
доктор технических наук, профессор,
ФГБОУ ВО «Воронежский государственный
технический университет»
профессор кафедры автоматизированных
и вычислительных систем

Ведущая организация – Федеральное государственное автономное
образовательное учреждение высшего
образования «Сибирский федеральный
университет», г. Красноярск.

Защита состоится "08" декабря 2017 г. в 15 часов на заседании совета по защите докторских и кандидатских диссертаций Д 212.249.05 Сибирского государственного университета науки и технологий им. академика М.Ф.Решетнева. по адресу: 660037, г.Красноярск, просп.им.газеты Красноярский рабочий, 31.

С диссертацией можно ознакомиться в библиотеке Сибирского государственного университета науки и технологий им. академика М.Ф. Решетнева.

Автореферат разослан “ ____ ” _____ 2017 г.

Ученый секретарь
диссертационного совета,
канд. тех. наук, доцент

И.А. Панфилов

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность. Конкуренция на рынке производителей программного обеспечения (ПО) сейчас высока как никогда. Сокращение сроков разработки ПО, необходимость адаптации программ под различные устройства, необходимость учета программной совместимости и повышенные требования к информационной безопасности – вот неполный перечень причин, почему вопрос о надежности программного обеспечения стоит так остро.

Важнейшим фактором, определяющим качество современного программного продукта, является надежность его функционирования. Этой проблемой занимаются исследователи и производители ПО по всему миру. Программные сбои, приводящие к остановке производств или обслуживанию клиентов, ошибки проектировщиков, приводящие к утечкам клиентских данных – эти проблемы несут колоссальные финансовые и репутационные риски, как крупным, так и малым компаниям.

В то же время производители программного обеспечения вынуждены сокращать сроки разработки ПО, использовать унифицированные решения для обеспечения совместимости программных систем, привлекать к разработке удаленных специалистов. То есть, несмотря на высокий запрос на надежное ПО, разработчики вынуждены экономить.

Самый эффективный способ обеспечить надежность функционирования программного обеспечения – это проведение работ на этапе проектирования ПО. Выбор надежной архитектуры программ позволит избежать большинства трудностей и сократить затраты на устранение проблем надежности в будущем. Однако у современных исследователей нет единого подхода даже к определению термина «надежность программного обеспечения». Одни вкладывают в это понятие невосприимчивость к внешним воздействиям, другие – простоту и прозрачность программных решений, исключаящую возможность возникновения ошибок.

Причин возникновения ошибок и сбоев в программных системах также великое множество: проблемы, связанные с проектированием, с аппаратным обеспечением, с поведением пользователей в системе, с внешними воздействиями на систему. Отсюда и разнообразие подходов к обеспечению надежности. Это требует развития модельно-алгоритмического обеспечения, методов и средств выбора надёжного варианта ПО

Степень разработанности темы исследования. Задача оценки надежности программного обеспечения (ПО) возникла вместе с появлением первых программ для ЭВМ. Решению данной проблемы посвящены исследования, таких ученых как Авижиенис А., Майерс Г., Дилон Б., Берман О., Липаев В.В., Луи М.Р., Боэм Б.В., Черкесов Г.Н., Ковалев И.В., Орлов С.А. и других.

Г. Майерс в своей ставшей классической книге «Надежность программного обеспечения»(1976) выделяет несколько подходов к оцениванию надежности: динамические методы, статистические методы, архитектурные методы и другие. В данной работе рассмотрены архитектурные подходы к оценке надежности.

Архитектурные методы выполняют декомпозицию программной системы на компоненты, выполняют оценку надежности каждого компонента в отдельности, определяют взаимное влияние компонентов и формируют общую оценку надежности всей системы. Для оценки отдельных компонентов системы могут быть задействованы как динамические, так и статистические методы. А для моделирования всей системы целиком используется представление системы в виде Марковских цепей, ГЕРТ-сетей и сетей Петри.

Одним из распространенных подходов к обеспечению надежности ПО является мультиверсионный подход. Концепция мультиверсионного программирования, как подход к реализации программной отказоустойчивости, была введена А. Авижиенисом в 1977. Он определил мультиверсионное программирование как независимую генерацию $N \geq 2$ функционально эквивалентных программ. В настоящее время большое количество работ по данному направлению принадлежит Ковалеву И.В. и его научной школе.

Второй проблемой, рассматриваемой в данной работе, стала задача проектирования сложных систем эволюционными алгоритмами. Изначально эволюционные алгоритмы (D.E. Goldberg, 1989), в том числе и генетический алгоритм (ГА), разрабатываемый в данном исследовании, рассматривались как методы решения задач оптимизации. Однако, многие исследователи с успехом стали использовать эволюционный подход для проектирования сложных систем различной природы. Большое количество работ по данному направлению проведено в научных школах Курейчика В.М. и Семенкина Е.С.

Объектом диссертационного исследования является отказоустойчивая архитектура сложных программных систем, **предметом исследования** являются стохастические алгоритмы моделирования и выбора надежного варианта программного обеспечения.

Целью работы является повышение обоснованности принятия решений при моделировании процесса функционирования программных систем, а также при выборе надежного варианта программного обеспечения за счет использования стохастических алгоритмов моделирования и оптимизации.

В процессе достижения поставленной цели решались следующие **задачи**:

1. Провести анализ подходов оценки надежности программного обеспечения;
2. Модифицировать модель оценки надежности функционирования программной системы так, чтобы она позволяла учитывать как экспертные оценки надежности отдельных компонентов, так и оценки, полученные в результате статистического анализа экспериментальных данных;
3. Модифицировать модель оценки надежности функционирования программной системы на основе оценок надежности функционирования ее компонентов;
4. Разработать стохастические алгоритмы многокритериальной условной оптимизации, позволяющие находить надежные варианты архитектуры ПО с учетом трудозатрат на реализацию ПО и с возможностью использования различных подходов к обеспечению отказоустойчивости;
5. Применить разработанные модели и алгоритмы при решении тестовых и реальных практических задач проектирования архитектуры ПО.

Методы исследования. При выполнении работы использовались методы прикладного системного анализа, элементы теории вероятностей, теория надёжности программного обеспечения, методология мультиверсионного проектирования отказоустойчивого программного обеспечения, эволюционные алгоритмы, методология разработки программного обеспечения, методы стохастического моделирования.

Новые научные результаты и положения, выносимые на защиту:

1. Разработан новый стохастический алгоритм моделирования процесса функционирования программной системы для оценки надёжности архитектуры программного обеспечения, позволяющий учитывать как экспертные оценки

надежности отдельных компонентов, так и статистические оценки на основе экспериментальных данных. Алгоритм позволяет учитывать оценки надежности отдельных связанных компонент программной системы для получения оценок надежности системы в целом.

2. Предложена новая схема оценивания решений в многокритериальном генетическом алгоритме, отличающаяся от известных учетом одновременно всего множества критериев и позволяющая избегать преждевременной сходимости алгоритма. Применение модифицированного подхода к оценке решений в генетическом алгоритме многокритериальной оптимизации повышает эффективность его работы, а оператор процентного скрещивания расширяет возможности его применения.

3. Разработан специализированный генетический алгоритм многокритериальной оптимизации, позволяющий осуществлять поиск надежного варианта программной архитектуры путем реализации нескольких вариантов мультиверсионного подхода для обеспечения избыточности программных систем. Алгоритм позволяет осуществлять поиск эффективных по надежности программных систем, обеспечивая приемлемые затраты на разработку.

Значение для теории результатов диссертационного исследования состоит в разработке специализированных генетических алгоритмов решения задач многокритериальной оптимизации с переменной длиной хромосомы, позволяющих получать в результате своей работы репрезентативную аппроксимацию множества и фронта Парето.

Предложенный в ходе диссертационного исследования подход к оценке надежности программного обеспечения позволяет использовать экспертную и статистическую информацию о функционировании программной системы.

Результаты, полученные при выполнении диссертационной работы, создают теоретическую основу для разработки моделей, методов и алгоритмов, направленных на повышение эффективности процессов разработки и модернизации программных систем.

Практическая ценность Разработанное в результате работы над диссертацией алгоритмическое обеспечение позволяет использовать результаты тестирования программных систем для оценки надежности не только тестируемых компонентов, но и программной системы в целом. Разработанный алгоритм выбора надежного варианта программного обеспечения позволяет учесть программные сбои, как возникающие на этапе разработки программного обеспечения, так и появляющиеся в ходе его эксплуатации.

Практическая реализация результатов: Алгоритмы моделирования функционирования и выбора надежной структуры программных систем реализованы в ходе выполнения работ по проекту «Разработка протокола безопасного обмена данными в распределенной информационно-вычислительной системе на основе технологии защиты с использованием движущейся цели» (2014-2016 гг., ФЦП, соглашение № 14.574.21.0126).

Предложенный в диссертационном исследовании подход к проектированию сложных систем с помощью генетического алгоритма был реализован в виде программной системы и зарегистрирован в РОСПАТЕНТе, что делает его доступным широкому кругу специалистов по системному анализу, архитектурному проектированию и планированию разработки сложных информационных систем.

Решения, полученные в ходе выполнения диссертационного исследования, были реализованы в виде алгоритмического обеспечения, переданного в рамках реализации соглашения о предоставлении субсидии от 27.11.2014 г. (№14.574.21.0126) организации - индустриальному партнеру ООО «Гуарднет», являющемуся резидентом Инновационного центра Сколково, и представлены Министерству науки и образования РФ.

Апробация работы. Результаты проведенных исследований докладывались на научных конференциях различного уровня, в том числе:

– Международная конференция "Вычислительные и информационные технологии в науке, технике и образовании", Алматы, Казахстан 2008;

– 6-я Международная научно-практическая конференция «Ресурсы недр России: экономика и геополитика, геотехнологии и геозология, литосфера и геотехника», Пенза, Сентябрь 2007;

– Научно-практическая конференции «Наука – взгляд в будущее», филиал РГГУ, Красноярск, 8-9 октября 2011;

– XVI Международная научная конференция «Решетневские чтения», Красноярск, 2012;

– Международная научная конференция «Молодежь. Общество. Современная наука, техника и инновации», Красноярск, 2013.

Ход выполнения диссертационной работы и диссертация в целом обсуждались на научных семинарах кафедры «Бизнес информатика» Сибирского федерального университета (2015-2017 гг.) и кафедры «Системный анализ и исследование операций» Сибирского государственного аэрокосмического университета (2010-2017 гг.).

Публикации. Основные теоретические и практические результаты диссертации опубликованы в 9 печатных работах (также имеется 1 свидетельство о государственной регистрации программ для ЭВМ), среди которых 3 работы в ведущих рецензируемых изданиях, рекомендуемых в действующем Перечне ВАК.

Структура и объем диссертации. Диссертация состоит из введения, трех глав, заключения, списка использованной литературы и приложения. Она изложена на 142 листах машинописного текста, содержит список литературы из 123 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во **введении** обоснована актуальность, научная новизна и практическая значимость работы, изложены цели, задачи и методы исследования.

Глава 1 носит обзорный характер. Дается обзор различных подходов к определению понятия надежности программных систем и к методам оценки программной надежности. Приводятся описания причин возникновения программных сбоев и методов обеспечения отказоустойчивости.

Одним из эффективных подходов к повышению надежности и снижению затрат на эти работы является обеспечение надежности на этапе проектирования программного обеспечения. В работе применяется мультиверсионный подход к обеспечению программной избыточности. Простое дублирование компонентов, как при аппаратном резервировании, недопустимо, так как в отличие от аппаратуры, программные ошибки имеют внутреннюю природу и при дублировании не исчезнут. При введении программной избыточности возникновение сбоя в функционально

эквивалентных модулях (версиях) на одних и тех же входных данных происходит в различных точках его исполнения, что снижает вероятность общей ошибки системы.

Создание функционально-эквивалентных, но, тем не менее, разных модулей может быть достигнуто при разработке версий одного модуля разными специалистами или применением разных языков программирования. Программную избыточность используют для разработки компонентов, к которым происходит наиболее частое обращение, или результаты работы которого участвуют в критически важных циклах управления. В диссертации используется два распространенных подхода в мультиверсионном программном обеспечении: первый подход – мультиверсионное программирование (*N-version programming, NVP*), второй подход – блок восстановления (*Recovery Block*).

В исследовании используются модели надежности программного обеспечения использующие для получения оценок различные источники данных. Одна из моделей – коэффициент надежности архитектуры ПО:

$$R = \sum_{j=1}^M \sum_{i=1}^{N_j} PU_{ij} \times R_{ij} \quad (1.1)$$

где M - число уровней архитектуры ПО;

N_j - число компонент на уровне j , $j=1, \dots, M$;

PU_{ij} – вероятность того, что компонент i на уровне j , $i \in \{1, \dots, N_j\}$, $j \in \{1, \dots, M\}$, будет использоваться;

R_{ij} – надежность компонента i уровня j .

Значения надежности отдельных компонент ПО R_{ij} могут быть получены в результате статистической обработки результатов тестирования или могут быть присвоены экспертом. В обоих случаях для расчёта надежности программной системы в целом придется использовать вероятности задействования программных компонент, оценить которые экспертам непросто.

Другая модель оценки надежности ПО позволяет дать оценку на стадии проектирования. На надежность влияют ошибки, допущенные на фазе кодирования ПО. Количество ошибок, в свою очередь, зависит от квалификации и опыта разработчиков, а также от способа тестирования ПО. Одним из самых часто используемых параметров оценки корректности ПО является плотность ошибок.

Начальную плотность ошибок можно оценить как:

$$D = C \times F_{pr} \times F_{pt} \times F_m \times F_s, \quad (1.2)$$

где F_{pt} – коэффициент фазы тестирования;

F_{pr} – коэффициент командного программирования;

F_m – коэффициент опытности и «зрелости» процесса разработки ПО;

F_s – коэффициент структурирования;

C – константа, определяющая количество ошибок/*KLOC* (ошибок на тысячу строк исходного кода).

Оценить данные параметры на фазе разработки архитектуры невозможно, численные показатели в модель берутся непосредственно на фазах кодирования и тестирования. Коэффициенты F_{pr} , F_m , F_s и C зависят только от мастерства и опыта команды разработчиков. Все числовые показатели данной модели определяются и задаются экспертом, что можно рассматривать как серьезный недостаток, ограничивающий применение данной модели.

В диссертации предлагается метод получения количественных оценок надежности для отдельных программных компонентов и для программных комплексов. В качестве метода моделирования предложены ГЕРТ-сети.

На рисунке 1 представлен пример фрагмента работы программы для ЭВМ. Состояние «1» - соответствует работе модуля «ввода данных». Состояние «2» - это следующий по очередности модуль информационной системы, переход к нему возможен с вероятностью p_1 . Состояние «3» - соответствует системному сбою, который приводит к остановке работы всей программы. Состояние «4» - описывает работы модуля по корректировке вводимых данных, после которого возможен переход к состоянию «2», «3» или к состоянию «5», соответствующему вызову модуля по «ручной корректировке вводимых данных». При этом $p_1 + p_2 + p_3 = 1$ и $p_4 + p_5 + p_6 = 1$. Состояния «1» и «4» - носят вероятностный, а состояния «2», «3», «5» - детерминированный характер.

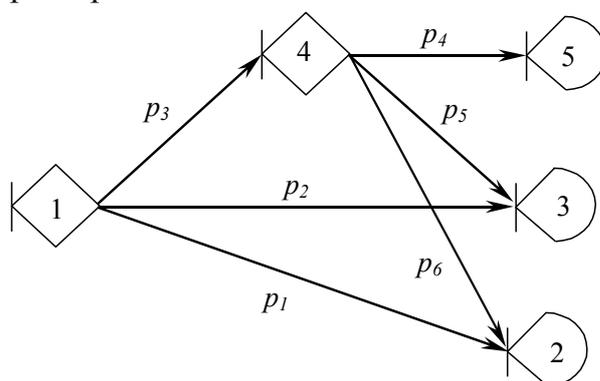


Рисунок 1 – Пример GERT-сети

Моделирование процесса функционирования программного обеспечения ГЕРТ-сетью позволяют ответить на следующие вопросы:

1. Какова вероятность того, что работа программы приведет к системному сбою?
2. Какова вероятность того, что программа продолжит штатную работу?
3. Чему равны математическое ожидание и дисперсия времени, необходимого для продолжения штатной работы?

Одним из недостатков моделей ГЕРТ-сетей является их высокая вычислительная сложность. В диссертации предлагаются прямой и обратный алгоритмы, позволяющие учитывать как экспертные оценки надежности отдельных компонентов, так и статистические оценки на основе экспериментальных данных. Алгоритм позволяет учитывать оценки надежности отдельных связанных компонент программной системы для получения оценок надежности системы в целом.

В главе 2 дается оптимизационная постановка задачи выбора надежного варианта программного обеспечения. В качестве критериев задачи оптимизации предложено использовать надежность программного обеспечения и стоимость его разработки. Стоимость разработки предлагается считать как трудоемкость (в человеко-часах) программной реализации:

$$T_S = \sum_{j=1}^M \sum_{i=1}^{N_j} \left(B_{ij} T_{ij} + (NVP_{ij} + RB_{ij}) \cdot \left(NVX_{ij} + \sum_{k \in Z_{ij}} T_{ij}^k \right) \right) \quad (2.1)$$

при ограничении $Z_{ij} \geq 1$

где M – количество архитектурных уровней в архитектуре ПО;

N_j – количество компонентов на уровне $j, j \in \{1, \dots, M\}$;

K_{ij} – глубина программной избыточности компонента i , на уровне j ;

Z_{ij} – множество версий компонента i , на уровне j ;

T_{ij} – трудоемкость разработки компонента i на уровне j ;

T_{ij}^k – трудоемкость разработки версии k компонента i на уровне j , $k \in Z_{ij}$, в часах;

NVX_{ij} – трудоемкость разработки среды исполнения версий (приемочного теста для RB (*recovery block*, блок восстановления) или алгоритма голосования для NVP (*N-version programming*, N -версионное программирование));

B_{ij} – бинарная переменная, принимающая значение 1 (тогда $NVP_{ij}=0$, $RB_{ij}=0$), если в программном компоненте не используется программная избыточность, иначе равна 0;

NVP_{ij} – бинарная переменная, принимающая значение 1 (тогда $B_{ij}=0$, $RB_{ij}=0$), если в программном компоненте введена программная избыточность методом NVP , иначе равна 0.

RB_{ij} – бинарная переменная, принимающая значение 1 (тогда $B_{ij}=0$, $NVP_{ij}=0$), если в программном компоненте введена программная избыточность методом RB , иначе равна 0.

T_s – общая трудоемкость реализации программной системы.

Надежность в данном исследовании оценивается с учетом ошибок допущенных на этапе разработки (1.2) и коэффициента надежности архитектуры ПО (1.1):

$$H_{no} = R + D \rightarrow \max, \quad (2.2)$$

$$T_s \rightarrow \min, \quad (2.3)$$

где

$$R = \sum_{j=1}^M \sum_{i=1}^{N_j} PU_{ij} \times R_{ij},$$

$$D = T_s \cdot F_{pr}.$$

Коэффициент D связывает показатель надежности со стоимостью (трудоемкостью) разработки.

В данной постановке оптимизируемыми параметрами будут выступать:

- N_j – количество компонентов на уровне j , $j \in \{1, \dots, M\}$;
- Z_{ij} – множество версий компонента i , на уровне j ;
- Бинарные переменные B_{ij} , NVP_{ij} , RB_{ij} , характеризующие применение подходов программной избыточности при разработке.

Для решения задачи (2.2-2.3) в диссертационном исследовании был выбран генетический алгоритм. В тексте диссертации описываются особенности применения ГА, рассматриваются подходы для решения задач многокритериальной оптимизации. В данной работе были предложены оригинальные модификации многокритериального ГА, отличающиеся от известных учетом одновременно всего множества критериев и позволяющие избегать преждевременной сходимости алгоритма.

Для оценки решений ГА в многокритериальном пространстве предлагается использовать метрики GD и IGD

Generational Distance (GD):

$$GD(A) := \frac{1}{N} \left(\sum_{i=1}^N d_i^p \right)^{1/p},$$

$A=\{a_1, \dots, a_N\}$ – аппроксимация фронта Парето (результат работы алгоритма), $F(P_Q)=\{y_1, \dots, y_M\}$ – Истинный фронт Парето, где d_i – минимальное расстояние от a_i до $F(P_Q)$ в евклидовой метрике.

Другими словами, в метрике GD измеряется расстояние от каждой точки полученного набора решений до каждой точки истинного фронта Парето, а затем оно усредняется по количеству точек, что даёт примерную погрешность между найденным и истинным решениями.

Inverted Generational Distance (IGD):

$$IGD(A) := \frac{1}{M} \left(\sum_{i=1}^M \tilde{d}_i^p \right)^{1/p}$$

где d_i – минимальное расстояние от y_i до A в евклидовой метрике.

В метрике IGD берется обратное расстояние, то есть вычисляется расстояние от каждой точки полученного решения до каждой точки истинного решения и усредняется уже по количеству точек найденного решения, что дает более обширную информацию о полученных результатах, так как такой подход к измерению расстояния позволяет узнать равномерность покрытия найденных решений.

Данные способы оценки решений ГА в многокритериальном пространстве были реализованы при модификации алгоритмов *SPEA* и *SPEA2*. На множестве тестовых задач были проведены численные исследования генетических алгоритмов, показана эффективность предложенного способа оценки решений. Для реальных задач в качестве «известных» множества и фронта Парето выбираются недостижимые («идеальные») точки, что соответствует подходу идеального проектирования.

Также в данной работе была предложена модификация ГА, позволяющая работать с хромосомами переменной длины. Это позволило существенно сократить пространство поиска, мощность которого, для задачи в главе 3, оценивалось как 2^{175} . Длина хромосомы в ГА зависит от количества архитектурных уровней M и количества N_i компонент на i -м уровне. При этом сами M и N_i выступают как переменные задачи.

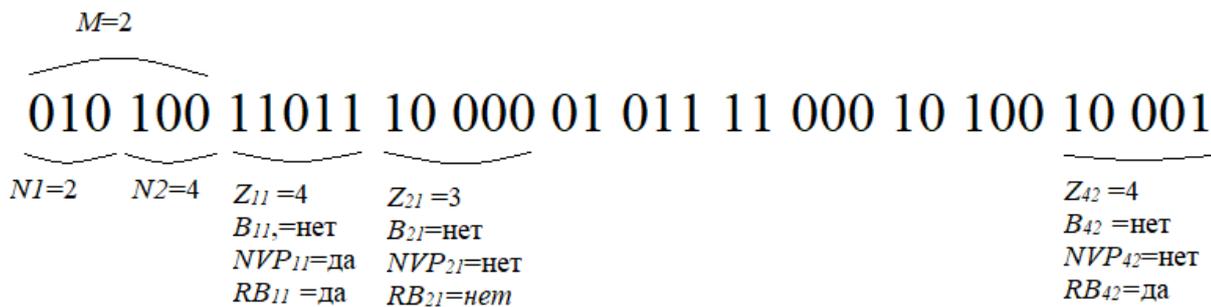


Рисунок 2 – Пример хромосомы в генетическом алгоритме

Для ГА с переменной длиной хромосом использовался специальный оператор процентного скрещивания.

Глава 3 посвящена решению практической задачи разработки надежного программного обеспечения. Решалась задача разработки экспериментального образца протокола безопасного обмена данными (ЭО ПБОД), созданного в ходе исполнения соглашения о предоставлении субсидии от 27.11.2014 г. №14.574.21.0126 «Разработка протокола безопасного обмена данными в распределенной информационно-вычислительной системе на основе технологии защиты с использованием

движущейся цели». На рисунке 3 представлена логическая структура разрабатываемой программы.

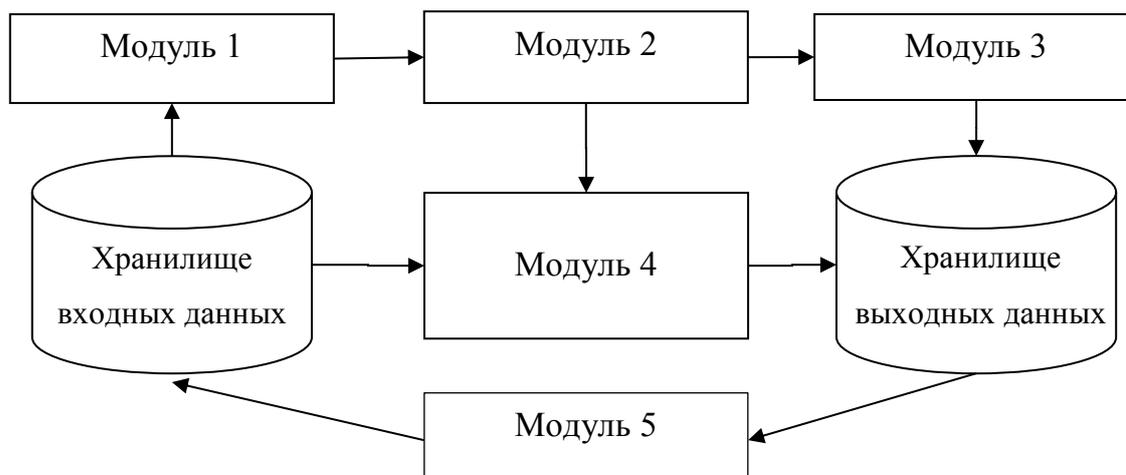


Рисунок 3 – Логическая структура программы

Каждый программный модуль является самостоятельной программной системой со своим собственным функционалом:

1. Модуль аутентификации пользователя по логину и паролю
2. Модуль авторизации пользователя с использованием принадлежности пользователя к группам и его ролям в них
3. Модуль делегирования прав пользователя, полученных на основе аутентификации и авторизации, связанным с пользователем программным компонентам
4. Модуль определения динамического адресного пространства в распределенной информационно-вычислительной системе
5. Модуль защиты от исследования программных компонентов распределенной информационно-вычислительной системы

В ходе проектирования данной программной системы каждый программный модуль был декомпозирован на функциональные блоки (от 5 до 8 компонент в каждом модуле). В рамках диссертационного исследования и в ходе выполнения работ по соглашению №14.574.21.0126 были проведены исследовательские испытания ЭО ПБОД. В соответствии с Техническим заданием на разработку, в ходе проведенных экспериментов, были получены статистические оценки надежности функционирования для четырнадцати функциональных блоков ЭО ПБОД.

Было проведено моделирование процесса функционирования программных модулей ЭО ПБОД в виде ГЕРТ-сетей. На рисунке 4 в качестве иллюстрации представлена модель функционирования программного модуля №5.

В таблице 1 приведены значения статистических оценок надежности для всех 14 экспериментально оцененных функциональных компонентов. В программный модуль №5 из этого списка попал параметр из 11-го эксперимента. Он соответствует узлу №5 на рисунке 4.

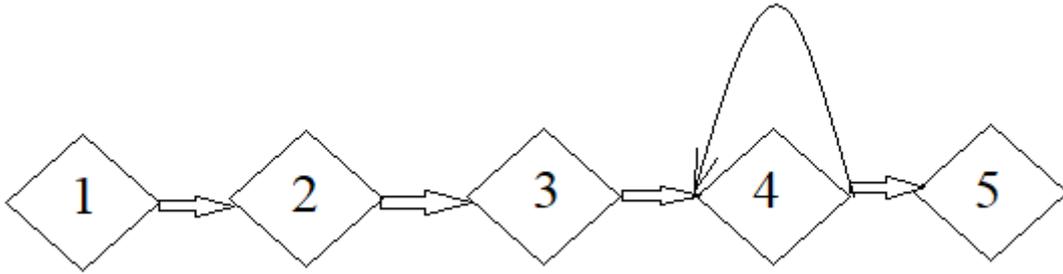


Рисунок 4 – Модель функционирования программного модуля №5 (1. Блок доступа к адресной информации, 2. Блок доступа к базе входных данных, 3. Входной интерфейс блока, 4. Блок проверки формата, 5. Блок модификации программного кода)

Таблица 1 - Моменты и производящие функции моментов

№	Тип распределения	$M_E(s)$	Математическое ожидание	Второй момент
1	Экспоненциальное	$\left(1 - \frac{s}{a}\right)^{-1}$	3	0,16
2	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	100	0
3	Константа	n	5	0
4	Экспоненциальное	$\left(1 - \frac{s}{a}\right)^{-1}$	95	10
5	Экспоненциальное	$\left(1 - \frac{s}{a}\right)^{-1}$	0	0
6	Экспоненциальное	$\left(1 - \frac{s}{a}\right)^{-1}$	0	0,1
7	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	5	0,05
8	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	1000	10
9	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	100	6
10	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	1000	20
11	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	45	0,5
12	Нормальное	$e^{sm+(1/2)s^2\sigma^2}$	20	10
13	Константа	n	20	0
14	Экспоненциальное	$\left(1 - \frac{s}{a}\right)^{-1}$	5	0,15

Было составлено топологическое уравнение Мейсона и получена оценка надежности всего программного модуля №5.

$$\mu_E = \frac{\partial}{\partial s} \left[\frac{1 - 0.95}{0.95 \cdot 0.85 \cdot 0.99 \cdot e^{45s + 0.5 \cdot 0.5s^2}} \right]_{s=0}.$$

Подобные расчеты были проведены для всех программных модулей ЭО ПБОД. Оценки для недостающих узлов были назначены экспертом. Результаты были сопоставлены с оценками надежности, полученными по формуле (1.1). Расхождения оказались несущественными. Таким образом моделирование функционирования

программного обеспечения ГЕРТ-сетями дает результаты не хуже, чем оценки, предложенные экспертами.

Далее был получен коэффициент надежности $R=0.62007891$ (формула 1.1) архитектуры всего ЭО ПБОД. Трудозатраты для представленной программной архитектуры составили $T_s=92$ человеко-часа (формула 2.1). Значения оценки надежности ЭО ПБОД оказались недостаточными, поэтому был применен разработанный в данном исследовании модифицированный генетический алгоритм для задач многокритериальной оптимизации.

В результате многократных запусков генетического алгоритма были получены альтернативные существующей архитектуры программной системы. В таблице 2 представлены значения характеристик надежности (коэффициент надежности архитектуры ПО R) и трудозатрат T_s . Решения, полученные алгоритмом, представленные в таблице, являются элементами множества Парето (недоминируемых решений) данной задачи, аппроксимация которого была получена в ходе многократного ее решения алгоритмом. Наилучшие результаты показал алгоритм MGAGD с настройками: 200 индивидов в популяции, низкая мутация.

Таблица 2 - Характеристики оптимальных вариантов модернизации

№	Коэффициент готовности системы	Трудозатраты на модернизацию, чел-час.
1	0.92007891	489
2	0.87100579	367
3	0.95411304	523
4	0.96486553	597
5	0.90338740	454

Анализ полученных решений позволил убедиться, что генетический алгоритм действительно находит решения, не только удовлетворяющие критерию надежности, но и уменьшающие трудозатраты. Так для полученного варианта программной системы, алгоритм не ввел обеспечение избыточности для компонентов, которые носят детерминированный характер. То есть для тех программных модулей, для которых коэффициент надежности архитектуры, заданный экспертом или полученный в результате анализа статистических данных $R_{ij}=1$.

ЗАКЛЮЧЕНИЕ

В ходе выполнения диссертационного исследования получены следующие результаты.

1. На основе проведенного анализа существующих подходов к оценке надежности программного обеспечения была предложена модификация модели оценки надежности функционирования программной системы так, чтобы она позволяла учитывать как экспертные оценки надежности отдельных компонентов, так и оценки, полученные в результате статистического анализа экспериментальных данных;

2. Был предложен подход к моделированию процесса функционирования программного обеспечения ГЕРТ-сетью. Применение модифицированного алгоритма расчета ГЕРТ-сетей позволяет получать оценки надежности функционирования программной системы на основе оценок надежности функционирования ее компонентов;

3. Был предложен новый способ оценивания решений в многокритериальном генетическом алгоритме, отличающийся от известных учетом одновременно всего множества критериев и позволяющий избегать преждевременной сходимости алгоритма.

4. Был разработан стохастический алгоритм многокритериальной условной оптимизации, позволяющий находить надежные варианты архитектуры ПО с учетом трудозатрат на реализацию ПО и возможностью использования различных подходов к обеспечению отказоустойчивости. Применение в генетическом алгоритме хромосом переменной длины и оператора процентного скрещивания позволило существенно сократить пространство поиска;

5. Разработанный алгоритм был применен для проектирования программной системы с высокими требованиями к надежности. В ходе проведенных с системой экспериментов были получены статистические оценки надежности отдельных ее компонентов, составлена модель функционирования программной системы. В результате применения разработанного генетического алгоритма были предложены архитектуры программной системы, позволяющие существенно улучшить надежность ее функционирования.

Таким образом, в диссертации разработаны стохастические алгоритмы моделирования и оптимизации, позволяющие повысить обоснованность принятия решений при выборе надежных вариантов программного обеспечения, что имеет существенное значение для теории и практики разработки систем обработки информации.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

В изданиях из перечня ВАК:

1. Панфилова Т.А. Прямой и обратный алгоритм расчета стохастических сетей /Царев Р.Ю., Штарик А.В., Штарик Е.Н., Хасанова Е.Р., Панфилова Т.А. // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2013. № 1 (47). С. 91-96.

2. Панфилова Т.А. Анализ вероятностно-временных характеристик отказоустойчивого программного обеспечения распределенных вычислительных систем /Царев Р.Ю., Штарик А.В., Штарик Е.Н., Кочергина М.А., Панфилова Т.А. // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2012. № 4 (44). С. 64-70.

3. Панфилова Т.А. Формализация задачи выбора надежного варианта программного обеспечения / Панфилов И.А., Панфилова Т.А. // Вестник Сибирского государственного аэрокосмического университета им. академика М.Ф. Решетнева. 2008. № 2 (19). С. 26-28.

В других изданиях:

4. Панфилова Т.А. Модификация генетического алгоритма для решения задач многокритериальной оптимизации / Слободина И.С., Панфилов И.А., Панфилова Т.А.// Решетневские чтения. 2012. Т. 2. № 16. С. 514-515.

5. Панфилова Т.А. Применение многоагентных стохастических алгоритмов при проектировании систем обработки и анализа слабоструктурированной информации / Панфилов И.А., Панфилова Т.А. // Материалы научно-практической конференции II «Наука – взгляд в будущее», филиал РГГУ, Красноярск, 8-9 октября 2011, С. 45-51.

6. Панфилова Т.А. Разработка надежного варианта программного обеспечения / Панфилов И.А., Панфилова Т.А. // Вестник университетского комплекса: Сб. научн. трудов; Красноярск: ВСФ РГУИТП, НИИ СУВПТ. – 2006. – вып. 7(21).

7. Панфилова Т.А. Разработка универсальной системы поддержки принятия решений, обработки информации и управления / Липинский Л.В., Панфилов И.А., Панфилова Т.А. // Международная конференция "Вычислительные и информационные технологии в науке, технике и образовании", Алматы, Казахстан 2008.

8. Панфилова Т.А. О задаче выбора надежного варианта программного обеспечения / Панфилов И.А., Панфилова Т.А. // Вестник университетского комплекса: Сб. научн. трудов; Красноярск: ВСФ РГУИТП, НИИ СУВПТ. – 2006. – вып. 6(20).

9. Панфилова Т.А. Разработка и оптимизация специализированной вычислительной техники для решения задач геофизики на примере многопроцессорной вычислительной системы ПС-2000 / Панфилов И.А., Панфилова Т.А. // Сборник статей 6-й Международной научно-практической конференции «Ресурсы недр России: экономика и геополитика, геотехнологии и геоэкология, литосфера и геотехника», Пенза, Сентябрь 2007.

Свидетельства о государственной регистрации программ для ЭВМ:

1. Панфилова Т.А. Система поддержки принятия решений на основе методов нейросетевого моделирования / Л.В. Липинский, С.Ю. Кузин, А.С. Егоров, Т.А. Панфилова, В.С. Тынченко. – М.: РОСПАТЕНТ. – 2009. – Свидетельство о государственной регистрации программы для ЭВМ № 2009611013 от 16.02.2009.