

# СРАВНЕНИЕ ЭФФЕКТИВНОСТИ СТАНДАРТНОГО И МОДИФИЦИРОВАННОГО АЛГОРИТМА МУРАВЬИНОЙ КОЛОНИИ

Е.В.Соловьев

*Институт космических и информационных технологий СФУ*

Мультиверсионное программирование (*N-version programming* или *multiversion programming*) является методологией разработки отказоустойчивого ПО. При ее использовании осуществляется разработка мультиверсионного программного обеспечения (NVS - *n-version software*), которое строится на множестве версий с общей спецификацией, созданными разными командами разработчиков. Эти версии выполняются параллельно. Результат их работы сравниваются с помощью системы согласования, результаты версии, не совпадающие с половиной версий и еще одной или не полученные вовремя, отвергаются. Это наиболее часто используемый подход к созданию отказоустойчивого ПО.

Модель для описания мультиверсионного ПО, которая использовалась для его представления выглядит следующим образом:

$n$  – число модулей NVS - системы программного обеспечения;  
 $m_i$  – число версий,  $i$ -го модуля,  $i = 1, \dots, n$ ;  
 $R_{ij}$  – оценка надежности  $j$ -ой версии  $i$ -го модуля;  
 $X_{ij}$  – булева переменная, равная 1, если  $j$ -я версия выбрана для  $i$ -го модуля, иначе - 0;

$g_{ij}$  – вероятность перехода из  $i$ -го модуля в  $j$ -ый модуль;

$R_i$  – оценка надежности  $i$ -го модуля;

$R$  – оценка надежности NVS - системы ПО;

$C_{ij}$  – стоимость разработки  $j$ -ой версии  $i$ -го модуля;

$B$  - ограничение по стоимости создаваемой NVS - системы ПО.

Модель представляет из себя наиболее общий случай мультиверсионного программного обеспечения.

$$R \rightarrow \max R, C \rightarrow \min C, \quad (1)$$

$$R_{parallel} = \sum_{j \in S_i} g_{ij} R_j, \quad (2)$$

$$R_{coherently} = \prod_{i \in S_k} R_i, \quad (3)$$

при ограничениях

$$\sum_{j=1}^{m_i} X_{ij} \geq 1, i = \{1, \dots, n\}, \quad (4)$$

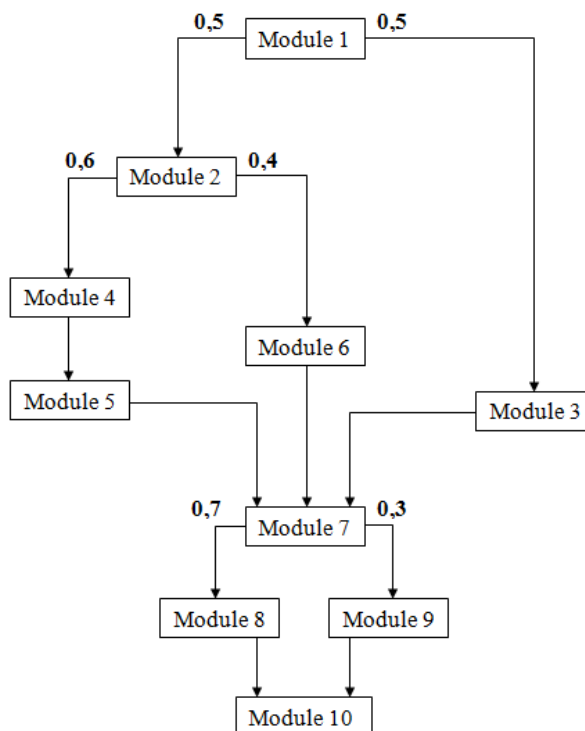
$$\sum_{i=1}^n \sum_{j=1}^{m_i} X_{ij} C_{ij} \leq B, \quad (5)$$

$$X_{ij} = \{0,1\}, i = \{1, \dots, n\}, j = \{1, \dots, m_i\}, \quad (6)$$

где  $R_i$  задаётся выражением

$$R_i = 1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{X_{ij}}. \quad (7)$$

Благодаря наличию данной модели мы имеем возможность сформулировать задачу формирования мультиверсионного программного обеспечения с динамической архитектурой.



**Рисунок 1. Архитектура программы в тестовой задаче**

Алгоритмом с помощью которого будет решаться задача нахождения состава мультиверсионного ПО, стал стандартный и модифицированный алгоритм муравьиной колонии. Для того чтобы получить данные о

сравнении результатов работы стандартного и модифицированного алгоритма, была создана тестовая задача, на которой проводились серии экспериментов для получения показателей эффективности. Задача представляет из себя набор из 100 версий, каждая из которых обладает показателем надежности и стоимости ее использования, эти 100 версий были поделены на группы по 10 версий образующих модули программного обеспечения, модули образуют схему программного обеспечения с вероятностями перехода от одного модуля к другому. Схема программного обеспечения представлена на рисунке 1.

Целью эксперимента стала задача минимизации стоимости, при соблюдении минимально допустимого ограничения на надежность. Параметрами задачи стали :  $P_{min} = 0.95, C \rightarrow min$ .

В результате анализа полученных данных было установлено, что среднее значение стоимости, которое достигалось при использовании стандартного алгоритма  $C = 2392$ , если же использовался модифицированный алгоритм муравьиной колонии, то средним результатом стоимости являлось  $C = 2347$ . Разница в полученных результатах составляет около 2 %. Стоит отметить, что не смотря на наличие дополнительных вычислений модифицированный алгоритм быстрее закончил свою работу и его скорость больше на 13 %. На рисунке 2 представлены графики изменения показателей стоимости от итерации для стандартного и модифицированного алгоритма.

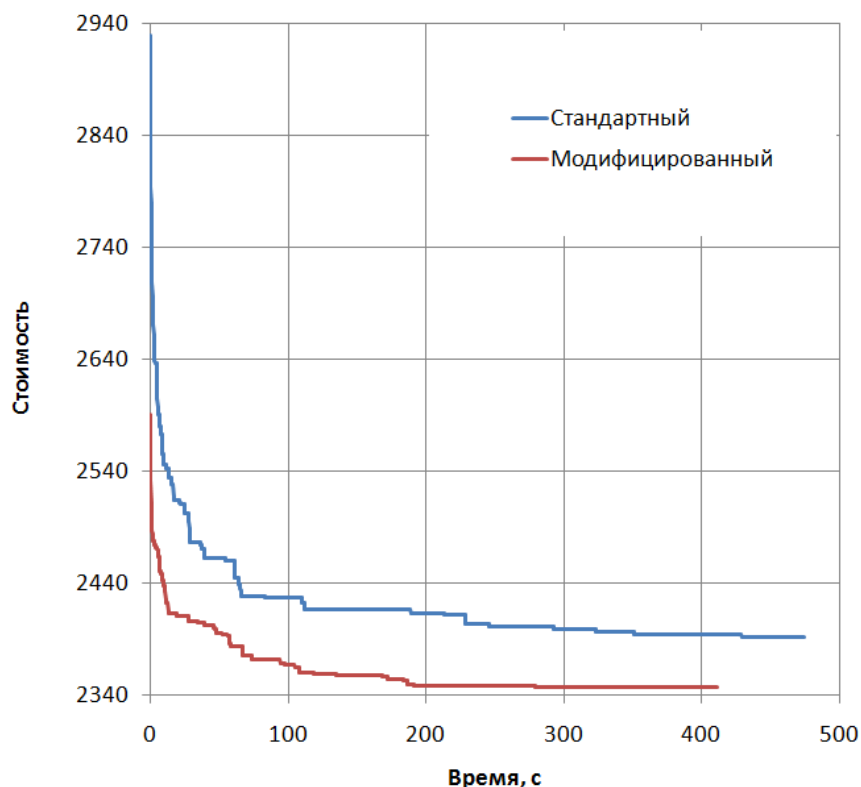


Рисунок 2. Изменение стоимости найденного решения

Как видно из графика, модифицированный алгоритм уже на 57 секунде достигает результата, который удастся достигнуть на стандартном алгоритме только на 428 секунде, что составляет 86% выигрыша по времени его достижения.

На основании проведенных экспериментов показана большая эффективность модифицированного алгоритма в сравнении со стандартными, достигнутая скорость нахождения решений существенно превышает скорость стандартных алгоритмов.

#### Список литературы:

1. Ковалёв, И. В. Мультиверсионный метод обеспечения отказоустойчивости программных архитектур систем управления критичными по надёжности техническими объектами / И.В. Ковалёв, Р.В. Юнусов // Науч. конф. «Научно-инновационное сотрудничество». Москва, 2002. – 172 с.
2. Соловьев Е.В. К вопросу реализации муравьиного алгоритма при выборе состава мультиверсионного программного обеспечения информационно-управляющих систем / И.В. Ковалев, Р.Ю. Царев, А.В. Прокопенко // Приборы и системы. Управление, контроль, диагностика. - 2012. №2. Стр. 01-04.