

УДК 681.34

## АЛГОРИТМ ФОРМИРОВАНИЯ ПРОГРАММНОЙ СИСТЕМЫ ПО СХЕМЕ БЛОКА ВОССТАНОВЛЕНИЯ С СОГЛАСОВАНИЕМ НА ОСНОВЕ НЕЧЕТКОЙ ЛОГИКИ

Царев Р.Ю.

*ФГАОУ ВПО «Сибирский федеральный университет», Красноярск, Россия (660074, Красноярск, ул. Академика Киренского, 26, УЛК), e-mail: tsarev.sfu@mail.ru*

---

Программная избыточность как средство обеспечения высокой надежности программных систем приводит к необходимости решения ряда проблем, самой важной из которых является оптимальное использование ресурсов при проектировании и разработке избыточных программных систем. Схема блока восстановления с согласованием является эффективным механизмом отказоустойчивости программных систем, основываясь на двух современных подходах, таких как разработка программного обеспечения с блоком восстановления и мультиверсионное программирование. Данная комбинация позволяет использовать достоинства обоих указанных подходов. Использование в составе программной системы избыточных программных компонент вынуждает решать задачу оптимизации, минимизируя расходы одновременно с повышением уровня надежности программной системы. При решении задачи формирования реальной программной системы зачастую невозможно получить точные значения ее характеристик. Решение задачи оптимизации в условиях неопределенности возможно благодаря использованию аппарата нечеткой логики. В данной статье предложен алгоритм нечеткой оптимизации состава программной системы, создаваемой согласно схеме блока восстановления с согласованием.

---

Ключевые слова: блок восстановления, мультиверсионное программирование, надежность, оптимизация

## ALGORITHM FOR GENERATING SOFTWARE SYSTEM ACCORDING TO CONSENSUS RECOVERY BLOCK SCHEME BASED ON FUZZY LOGIC

Tsarev R.Y.

*Siberian federal university, Krasnoyarsk, Russia (660074, Krasnoyarsk, street A. Kirenskogo, 26, ULK), e-mail: tsarev.sfu@mail.ru*

---

Software redundancy as a means to ensure high reliability of software systems makes it necessary to address a number of issues, the most important of which is the optimal use of resources at the phases of design and development of redundant software systems. Consensus recovery block scheme is an effective mechanism providing the fault tolerance of software systems. The scheme is based on two up to date approaches such as the recovery blocks and multiversion programming. This combination brings us the advantages of both these approaches. The use of redundant software component as part of a software system requires solving the optimization problem minimizing costs while increasing the reliability of the software system. The optimization of a real software system is often difficult due to fuzzy values of its characteristics. Solution of the problem of optimization under uncertainty can be obtained using the fuzzy logic. The article proposes an algorithm of fuzzy optimization of the software system created according the consensus recovery block scheme.

---

Keywords: recovery block, mutiversion programming, reliability, optimization

Одним из эффективных методов обеспечения высокой надежности программных систем является использование блока восстановления с согласованием [4]. В основе данного метода лежит избыточная реализация модулей программной системы в виде набора альтернатив. При исполнении одной альтернативы программного модуля ее результаты оцениваются посредством приемочного теста, в случае ошибки исполняется другая альтернатива. Таким образом, решение задачи модулем программной системы гарантировано даже в случае возникновения ошибки в отдельной альтернативе или даже в ряде альтернатив.

Дальнейшим развитием схемы программной системы с блоком восстановления стало применение мультиверсионного программирования альтернатив. При таком подходе каждая альтернатива представляет собой набор функционально эквивалентных версий, исполнение которых происходит одновременно и параллельно, а результаты оцениваются с помощью алгоритмов голосования [4]. Подобная реализация программных систем получила название блока восстановления с согласованием [5].

Основной проблемой избыточной реализации программных систем является потребность в дополнительных ресурсах. Это приводит к необходимости решения задачи максимизации надежности программной системы с учетом существующих ограничений на стоимость, время реализации, использования компонент собственной разработки и приобретенных программных компонент. Подробно данная задача описана в [2].

Довольно часто при решении практических задач исходная информация является неполной или неточной [3, 8]. Существующие исследования в отношении надежности программных систем показывают, что не все параметры задачи четко заданы. Различные цели и ограничения, задаваемые проектировщиком программных систем, а также показатели затрат, включаемые в функцию стоимости, определяются на основании предыдущего опыта и информации, доступной из различных баз данных [10]. В связи с этим проектировщику сложно задать точные значения возможных финансовых затрат и использования различных ресурсов [6]. Более того, сложность проекта, новизна задачи, возможные изменения требований заказчика, а также ряд факторов, которые неизвестны на начальном этапе проекта, но могут появиться позже, увеличивают степень неточности и неопределенности. Кроме этого, требуемые значения могут быть преднамеренно заданы нечетко, чтобы обеспечить определенный уровень допустимого отклонения по этим параметрам [7].

Все это приводит к формализации задачи как нечеткой, и для ее решения целесообразным становится использование методов нечеткой оптимизации, которые позволяют найти решение задачи в условиях неопределенности [1].

В данной статье рассматривается задача оптимизации состава программной системы по схеме блока восстановления с согласованием, для решения которой предложен алгоритм работы метода нечеткой оптимизации. Предлагаемый алгоритм позволяет сформировать программную систему, отличающуюся высоким уровнем надежности и минимальными затратами на ее реализацию.

#### **Задача оптимизации состава программной системы**

Задача формирования программной системы по схеме блока восстановления с согласованием может быть рассмотрена как двучелевая задача оптимизации:

максимизировать надежность программной системы и одновременно с этим минимизировать ее стоимость. Формально данную задачу можно записать следующим образом:

$$\max \tilde{R} = \sum_{l=1}^L f_l \prod_{i \in s_l} \tilde{R}_i, \quad \min \tilde{C} = \sum_{i=1}^n \sum_{j=1}^{m_i} \left( \tilde{c}_{ij} (t_{ij} + \tau_{ij} N_{ij}) y_{ij} + \sum_{k=1}^{V_{ij}} \tilde{c}_{ijk} x_{ijk} \right), \quad (1)$$

где знак тильды ( $\sim$ ) над коэффициентами целевых функций означает, что они являются нечеткими числами.

Условные обозначения:

$R$  – надежность (вероятность безотказной работы) программной системы;

$R_i$  – надежность  $i$ -го модуля программной системы;

$C$  – стоимость программной системы;

$c_{ij}$  – стоимость  $j$ -й альтернативы  $i$ -го модуля;

$c_{ijk}$  – стоимость  $k$ -й версии  $j$ -й альтернативы  $i$ -го модуля;

$n$  – количество модулей в программной системе;

$m_i$  – количество альтернатив, доступных для  $i$ -го модуля;

$V_{ij}$  – количество версий, доступных для  $j$ -й альтернативы  $i$ -го модуля;

$L$  – количество функций, которые выполняет программная система;

$f_l$  – частота использования функции  $l$ ;

$s_l$  – множество модулей, необходимых для выполнения функции  $l$ ;

$x_{ijk}$  – булева переменная, равная 1, если выбранная  $k$ -я версия  $j$ -й альтернативы  $i$ -го модуля создана сторонними разработчиками, 0 – если версия разработана самостоятельно;

$y_{ij}$  – булева переменная, равная 1, если  $j$ -я альтернатива  $i$ -го модуля разрабатывается собственными силами, 0 – в противном случае;

$t_{ij}$  – расчетное время разработки  $j$ -й альтернативы  $i$ -го модуля;

$\tau_{ij}$  – среднее время тестирования  $j$ -й альтернативы  $i$ -го модуля;

$N_{ij}$  – общее количество тестовых испытаний.

Надежность модуля программной системы, создаваемой согласно схеме блока восстановления с согласованием, вычисляется следующим образом:

$$\tilde{R}_i = 1 + \left[ \sum_{j=1}^{m_i} \frac{1}{(1 - R_{ij})^{z_{ij}}} \left( \prod_{k=1}^{m_i} (1 - R_{ik})^{z_{ik}} \right) (1 - (1 - R_{ij})^{z_{ij}}) + \prod_{j=1}^{m_i} (1 - R_{ij})^{z_{ij}} \right].$$

$$\cdot \left[ \sum_{j=1}^{m_i} z_{ij} \left( \prod_{k=1}^{j-1} P(A_{ik})^{z_{ik}} \right) P(B_{ij}) - 1 \right]; \quad i = 1, 2, \dots, n,$$

$$P(A_{ij}) = (1 - p_1)[(1 - R_{ij})(1 - p_3) + R_{ij}p_2],$$

$$P(B_{ij}) = R_{ij}(1 - p_2),$$

где  $R_{ij}$  – надежность  $j$ -й альтернативы  $i$ -го модуля;  $z_{ij}$  – булева переменная, равная 1, если  $j$ -я

альтернатива присутствует в  $i$ -м модуле, 0 – в противном случае;  $A_{ij}$  – событие, соответствующее отклонению результата  $j$ -й альтернативы  $i$ -го модуля;  $B_{ij}$  – событие, соответствующее принятию корректного результата  $j$ -й альтернативы  $i$ -го модуля;  $p_1$  – вероятность того, что следующая альтернатива не будет вызвана, несмотря на отказ текущей альтернативы;  $p_2$  – вероятность неверной оценки корректного результата;  $p_3$  – вероятность принятия неверного результата за корректный.

### Алгоритм формирования оптимального состава программной системы

Следующий алгоритм описывает последовательность шагов решения задачи формирования оптимального состава программной системы, создаваемой согласно схеме блока восстановления с согласованием, как задачи нечеткого математического программирования.

Шаг 1. Вычисление четких эквивалентов нечетких параметров с помощью функции дефаззификации. Отметим, что для каждого из параметров используется одна и та же функция дефаззификации.

Будем использовать следующую функцию дефаззификации:

$$F_2(A) = \frac{a_l + 2a_m + a_u}{4},$$

где  $a_l, a_m, a_u$  – являются треугольными нечеткими числами.

Шаг 2. Включение целевой функции минимального (максимального) оператора в виде нечеткого ограничения. Тогда задачу формирования оптимального состава избыточного программного обеспечения можно представить как

Найти  $X$

при условиях:

$$R(X) = \sum_{l=1}^L f_l \prod_{i \in S_l} R_i \geq R_0,$$

$$C(X) = \sum_{i=1}^n \sum_{j=1}^m \left( c_{ij} (t_{ij} + \tau_{ij} N_{ij}) y_{ij} + \sum_{k=1}^{V_{ij}} C_{ijk} x_{ijk} \right) \leq C_0,$$

$$X \in S,$$

где  $R_0$  и  $C_0$  – дефаззифицированные уровни желаемой надежности и стоимости.

Шаг 3. Определение соответствующих функций принадлежности для каждого нечеткого неравенств:

$$\mu_R(X) = \begin{cases} 1; & R(X) \geq R_0 \\ \frac{R(X) - R_0^*}{R_0 - R_0^*}; & R_0^* \leq R(X) < R_0, \\ 0; & R(X) < R_0^* \end{cases}$$

где  $R_0$  – желаемый уровень, а  $R_0^*$  – уровень допустимого отклонения к нечеткому ограничению целевой функции надежности.

$$\mu_C(X) = \begin{cases} 1; & C(X) \leq C_0 \\ \frac{C_0^* - C(X)}{C_0^* - C_0}; & C_0 \leq C(X) < C_0^*, \\ 0; & C(X) > C_0^* \end{cases}$$

где  $C_0$  – ограничение и  $C_0^*$  – уровень допустимого отклонения от нечеткого ограничения по бюджету.

Шаг 4. Чтобы найти нечеткое решение, используем принцип расширения, что приводит к задаче четкого математического программирования, выраженной как

$$\max \alpha$$

при условиях:

$$\mu_R(x) \geq \alpha,$$

$$\mu_C(x) \geq \alpha,$$

$$X \in S,$$

где  $\alpha$  отражает степень достижения уровня, заданного лицом, принимающим решение (ЛПР).

Шаг 5. При решении задачи на шагах алгоритма 1–4 целевая функция рассматривается как ограничение. Каждое ограничение рассматривается как цель для ЛПР, а задача может быть представлена как задача нечеткого многокритериального математического программирования. Помимо этого, каждая цель может обладать различным уровнем важности, ей может присваиваться определенный весовой коэффициент, выражающий ее относительную значимость. Полученную задачу можно решить при помощи взвешенного минимаксного подхода. Четкая формулировка взвешенной задачи задается как

$$\max \alpha \tag{2}$$

при условиях:

$$\mu_R(x) \geq \omega_1 \alpha,$$

$$\mu_C(x) \geq \omega_2 \alpha,$$

$$\omega_1 \geq 0; \omega_2 \geq 0; \omega_1 + \omega_2 = 1.$$

Данная задача может быть решена с помощью стандартных подходов математического программирования.

Шаг 6. При подстановке значений для  $\mu_R(x)$  и  $\mu_C(x)$  задача принимает вид:

$$\max \alpha \tag{3}$$

при условиях:

$$R(x) \leq R_0 - (1 - \omega_1 \alpha)(R_0 - R_0^*),$$

$$C(x) \leq C_0 + (1 - \omega_2 \alpha)(C_0^* - C_0),$$

$$\alpha \in [0, 1],$$

$$X \in S,$$

$$\omega_1 \geq 0; \omega_2 \geq 0; \omega_1 + \omega_2 = 1.$$

Шаг 7. В случае, если допустимое решение задачи (2) или (3) не получено, можно использовать метод нечеткого целевого программирования для получения компромиссного решения, например, предложенный Мохамедом в [9].

Так, если задача (3) нерешаема, т.е. цель не может быть достигнута для допустимого значения  $\alpha \in [0, 1]$ , то можно использовать следующий метод нечеткого целевого программирования для получения компромиссного решения. Максимальное значение, которое может принимать функция принадлежности, равно единице. Максимизация  $\alpha \in [0, 1]$  равнозначна задаче максимального приближения его значения к 1. Этого можно добиться путем минимизации отрицательных переменных целевого программирования (т.е.  $\eta$ ), отклоняющихся от единицы. Формулировка задачи (3) в рамках нечеткого программирования с введением отрицательных и положительных переменных отклонения  $\eta_j$  и  $\rho_j$  может быть задана следующим образом:

$$\max u \tag{4}$$

при условиях:

$$\mu_R(X) + \eta_1 - \rho_1 = 1,$$

$$\mu_C(X) + \eta_2 - \rho_2 = 1,$$

$$u \geq \omega_j \cdot \eta_j,$$

$$\eta_j \cdot \rho_j = 0; \eta_j \geq 0; \rho_j \geq 0; j = 1, 2,$$

$$\alpha \in [0, 1],$$

$$X \in S,$$

$$\omega_1 \geq 0; \omega_2 \geq 0; \omega_1 + \omega_2 = 1,$$

$$\alpha = 1 - u.$$

Таким образом, решение задачи (1) может быть получено путем решения задачи (4).

### **Заключение**

Оптимизация состава избыточных программных систем всегда является нетривиальной проблемой, требующей учета ряда параметров системы и специфики самой задачи. В данной статье предложен алгоритм, предназначенный для формирования оптимального состава программной системы схемы блока восстановления с согласованием. Решение данной задачи в условиях неопределенности, связанной с неточностью и неполнотой информации о ряде

характеристик программной системы, стало возможным при использовании математического аппарата нечеткой логики. Предложенный в статье алгоритм нечеткой оптимизации позволяет сформировать программную систему с блоком восстановления с согласованием, обеспечивая высокий уровень надежности системы с минимальными затратами на ее создание. Данный алгоритм может быть использован для формирования программных систем, применяемых в критичных по надежности реализации их целевых функций областях.

### Список литературы

1. Лысенко, И.В. Нечеткая оптимизация: новый подход к постановке и решению задач / И.В. Лысенко // Труды СПИИРАН. – 2004. – Т. 1. – № 2. – С. 90–118.
2. Модель формирования оптимальной программной системы по схеме блока восстановления с согласованием / О.И. Завьялова, С.Н. Гриценко, С.В. Тынченко, Р.Ю. Царев // Современные проблемы науки и образования. – 2015. – № 1; URL: <http://www.science-education.ru/121-18871> (дата обращения: 17.06.2015).
3. Пальчунов Д.Е. Нечеткие алгебраические системы / Д.Е. Пальчунов, Г.Э. Яхьяева // Вестник НГУ. Серия: Математика, механика, информатика. – 2010. – Т. 10. – Вып. 3. – С. 76–93.
4. Царев Р.Ю. Мультиверсионное программное обеспечение. Алгоритмы голосования и оценка надежности: монография / Р.Ю. Царев, А.В. Штарик, Е.Н. Штарик. – Красноярск: Сибирский федеральный университет, 2013. – 120 с.
5. Царев Р.Ю. Учет совместимости COTS-компонентов при формировании избыточных программных систем / Р.Ю. Царев, О.И. Завьялова, А.С. Черниговский // Современные проблемы науки и образования. – 2015. – № 1; URL: <http://www.science-education.ru/125-19746> (дата обращения: 17.06.2015).
6. Царев Р.Ю. Fuzzy-метод формирования структуры мультиверсионного программного обеспечения информационно-управляющих систем / Р.Ю. Царев // Информационно-измерительные и управляющие системы. – 2007. – № 12. – С. 72–75.
7. Bali S., Gupta A., Dinesh Kumar U. Fuzzy multi-objective build-or-buy approach for component selection of fault tolerant modular software system under consensus recovery block scheme (2012) *Advances in Intelligent and Soft Computing*, 130 AISC (VOL. 1), pp. 1025–1036.
8. Bellman R.E., Zadeh L.A. Decision-making in a fuzzy environment (1970) *Management Science*, 17 (4), pp. b-141-64.
9. Mohamed R.H. The relationship between goal programming and fuzzy programming (1997) *Fuzzy Sets and Systems*, 89 (2), pp. 215–222.

10. Reifer D.J., Boehm, B.W., Gangadharan, M. Estimating the cost of security for cots software (2003) Lecture Notes in Computer Science, vol. 2580, pp. 178–186.

**Рецензенты:**

Бронов С.А., д.т.н., профессор, руководитель научно-учебной лаборатории систем автоматизированного проектирования кафедры систем искусственного интеллекта Сибирского федерального университета, г. Красноярск;

Носков М.В., д.ф.-м.н., профессор, заместитель директора по научной работе Института космических и информационных технологий Сибирского федерального университета, г. Красноярск.