

УДК 004.021

Doi: 10.31772/2712-8970-2021-22-1-32-46

Для цитирования: Карцан И. Н., Ефремова С. В. Мультиверсионная модель программного обеспечения систем управления космическим аппаратом с ранжированием принятия решения // Сибирский аэрокосмический журнал. 2021. Т. 22, № 1. С. 32–46. Doi: 10.31772/2712-8970-2021-22-1-32-46.

For citation: Kartsan I. N., Efremova S. V. Multiversion model of software control systems for space vehicles with range of decision-making // Siberian Aerospace Journal. 2021, Vol. 22, No. 1, P. 32–46. Doi: 10.31772/2712-8970-2021-22-1-32-46.

Мультиверсионная модель программного обеспечения систем управления космическим аппаратом с ранжированием принятия решения*

И. Н. Карцан^{1, 2, 3**}, С. В. Ефремова²

¹ФГБУН ФИЦ «Морской гидрофизический институт РАН»

Российская Федерация, 299011, г. Севастополь, ул. Капитанская, 2

²Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева
Российская Федерация, 660037, г. Красноярск, просп. им. газ. «Красноярский рабочий», 31

³Севастопольский государственный университет

Российская Федерация, 299053, г. Севастополь, ул. Университетская, 33

**E-mail: kartsan2003@mail.ru

В статье представлена мультиверсионная модель с ранжированием альтернатив в порядке предпочтения с учетом зависимости атрибутов при проектировании программного обеспечения для системы управления космическими аппаратами различного класса. Применяемое программное обеспечение с набором алгоритмов, базирующихся на общей схеме метода ветвей и границ, позволяет определять точное решение оптимизационной задачи.

Для достижения наибольшей надежности программной составляющей систем управления космическими аппаратами, построенной с использованием методологии мультиверсионного программирования, в единую структуру объединяется большое количество версий программных модулей.

В то время как программные комплексы даже без введения избыточных элементов характеризуются как сложные системы, говорить о широком использовании переборных методов для их формирования не приходится.

Использование предложенного модифицированного метода упорядоченного предпочтения через сходство с идеальным решением позволит решить задачу выбора лучшей вычислительной системы из ряда доступных систем. Данный подход становится все более возможным по причине колоссального прогресса в технологиях проектирования и производства вычислительной техники. Даже так называемые персональные компьютеры предоставляют вычислительные возможности, которые некоторое время назад казались невозможными для компьютеров – представителей значительно более мощного класса вычислительной техники – суперкомпьютеров.

Ключевые слова: мультиверсионная модель, программное обеспечение, системы управления космическими аппаратами, алгоритм, многоатрибутивные методы принятия решений.

* Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-37-90076.

Работа выполнена в рамках государственного задания по теме № 0555-2021-0005.

The research was carried out with the financial support of the RFBR in the framework of scientific project No. 19-37-90076.

This study was supported by the Russian Federation State Task № 0555-2021-0005.

Multiversion model of software control systems for space vehicles with range of decision-making

I. N. Kartsan^{1,2**}, S. V. Efremova²

¹Marine Hydrophysical Institute, Russian Academy of Sciences
2, Kapitanskaya St., Sevastopol, 299011, Russian Federation

²Reshetnev Siberian State University of Science and Technology
31, Krasnoyarskii Rabochi Prospekt, Krasnoyarsk, 660037, Russian Federation

³Sevastopol State University
33, Universitetskaya St., Sevastopol, 299053, Russian Federation

**E-mail: kartsan2003@mail.ru

The paper presents a multiversion model with ranking of alternatives in order of preference, taking into account the dependence of the attributes in the design of software for spacecraft control systems of various classes. The applied software with a set of algorithms, based on the general scheme of the method of branches and borders allow to determine the exact solution of the optimization problem.

To achieve the highest reliability of the software component of spacecraft control systems built with the use of multiversion programming methodology, a large number of versions of software modules are combined into a single structure.

While software complexes even without introduction of redundant elements are characterized as complex systems, there is no need to speak about wide use of enumerative methods for their formation.

Using the proposed modified method of ordered preference through similarity to an ideal solution, will allow to solve the problem of choosing the best computing system from a number of available systems. This approach is becoming increasingly possible because of the tremendous progress in computing design and manufacturing technology. Even the so-called personal computers provide computational capabilities that some time ago seemed impossible even for computers representing a much more powerful class of computing equipment – supercomputers.

Keywords: multiversion model, software, spacecraft control systems, algorithm, multi-attribute decision-making methods.

Введение. На этапе проектирования облика систем управления космических аппаратов происходит принятие решений по выбору состава мультиверсионного программного обеспечения системы с использованием метода нечеткого программирования. Это позволяет проектировщику задать степень «предпочтения атрибутов» и возможный «процент достижимости» целей при выборе того или иного варианта формирования состава мультиверсионного программного обеспечения [1–3]. Результатом, как правило, является множество недоминируемых решений задачи формирования мультиверсионного программного обеспечения системы управления.

Задача формирования комплекса мультиверсионного программного обеспечения систем управления космическими аппаратами представляет собой задачу условной оптимизации монотонной псевдобулевой функции.

Наиболее универсальным из всех методов дискретной оптимизации является метод полного перебора, когда для нахождения наилучшего значения целевой функции обследуются все допустимые решения [4].

В этом направлении разрабатывались алгоритмы двух основных типов: неявный полный перебор, где увеличение скорости нахождения оптимального решения реализовано через сужение

области поиска и сокращение общего числа вычислений в алгоритме, и схема метода ветвей и границ с разбиением области поиска решения на меньшие подобласти.

Среди точных алгоритмов первого типа следует выделить процедуры, использующие для усиления вычислительных возможностей точных алгоритмов идеи сужения области поиска за счет особенностей функций-элементов решаемой оптимизационной задачи и наискорейшего обхода области поиска по соседним точкам, с использованием отношений на пространстве булевых переменных [5].

К алгоритмам схемы ветвей и границ [6] отнесены алгоритмы поиска решений на подкубах [7–8]. Процедуры этого вида различаются лишь способом организации разбиения области поиска решения на подкубы, т. е. способом представления исходной оптимизационной задачи в виде некоторого числа задач меньшей размерности [9].

Большое количество модулей мультиверсионного программного обеспечения систем управления космическими аппаратами, их избыточные версии, а также ограничения реальных потребностей такие, например, как стоимость разработки, внедрения и модификации системы, ставят перед проектировщиком задачу принятия решений по выбору оптимального состава мультиверсионного программного обеспечения системы с учетом ряда атрибутов [10–13].

При существовании достаточно большого числа альтернатив при их оценке в значительной мере сказываются качества самого лица, принимающего решение, его субъективные предпочтения. Поэтому, производя выбор лучшей альтернативы из ряда предложенных, нужно учитывать субъективность в оценках лиц, принимающих решений [14].

Модель общего ранжирования при выборе оптимального решения. В новых информационных технологиях принятием решений считают набор решений в условиях определенности, позволяющих выбрать однозначные, непротиворечивые, корректные решения на основе формализованных моделей объектов управления и окружающей их среды. Принятием решений следует считать и формирование неких вероятностных решений в условиях неопределенности [11].

К задачам поддержки принятия решений в новых информационных технологиях относятся все задачи, включая класс задач в условиях неопределенности, окончательное решение которых осуществляется вне используемой технологии. В этих случаях информацию преобразуют к виду, упрощающему и облегчающему принятие решений иными методами.

Можно констатировать, что в настоящее время для поддержки принятия решений используют различные методы и подходы, которые в совокупности дополняют друг друга. Принятие решений включает выбор последовательности действий и ее реализацию. Поддержка принятия решений основана на получении многовариантных решений с использованием разных методов.

Методология поддержки принятия решений включает разнообразные схемы и технологии, может реализоваться частично или полностью с использованием программно-информационных систем.

Простейшая модель принятия решения включает четыре основных, циклически повторяющихся этапа:

- сбор, анализ и преобразование данных;
- получение вариантов решений (альтернатив);
- разработка критериев оценки решений;
- выбор одного из вариантов на основе выбранных критериев.

Возможны разные условия и ситуации, в которых требуется принимать решения. Существуют разные уровни принятия решений. Верхним является концептуальный уровень. Он позволяет составить обобщенную схему. Технология принятия решений в общем случае включает следующие этапы:

- возникновение проблемы или задачи, требующей решения;
- формирование проблемы на вербальном уровне;
- поиск информации, необходимой для принятия решения;
- формализация постановки задач;
- анализ и обработка информации;
- формирование набора альтернатив;
- получение прогнозных оценок;
- оценка результатов принятия решения.

На первом этапе возникает проблема или задача, которая требует решения. Как правило, она формируется на вербальном (неформализованном) уровне общения.

После этого осуществляют поиск и сбор информации в аспекте данной проблемы или задачи. При этом собирают не только данные, необходимые для решения, но и информацию о методах решения подобных задач. После этого осуществляют формализацию или формирование проблемы на формальном уровне [15].

На этапе анализа и обработки информации осуществляют комплексную обработку, включающую компьютерные методы и экспертный анализ; прямой расчет и эвристические методы; оптимизационные методы.

Возможное изменение условий предусматривается получением наборов решений. Взаимоисключающие решения называют альтернативами. Поэтому для полноты и учета изменяющихся условий в результате анализа осуществляют формирование альтернатив, что представлено как важный этап в схеме принятия решений. Представление обобщенных данных осуществляют в виде, удобном для принятия решений.

Следующим этапом и одним из наиболее важных является этап получения прогнозных оценок. На этом этапе с использованием имеющихся вариантов решений, информации об условиях и методах получения прогноза получают прогнозные оценки, а также обосновывают выбор метода прогнозирования и метода верификации прогноза.

После получения комплекса данных: альтернатив, информации о динамике условий, прогнозных оценок, оценки надежности прогноза и др., производят комплексную обработку с использованием экспертного подхода. Эта схема может работать при разных технологиях и уровнях управления.

Следует подчеркнуть связь перечисленных этапов. Они образуют иерархическую последовательность. Если на одном из этапов отсутствует возможность его реализации, то переход к последующим не осуществляется. Поддержка при принятии решений означает, что группа этих методов направлена не только на получение решений, но и на подготовку рекомендаций для лица, принимающего решения. Таким образом, поддержка принятия решений включает три группы задач.

1. Получение не одного, а совокупности решений.
2. Подготовка критериев для оценки полученных решений.
3. Выбор решения из имеющейся совокупности.

При формировании решений и их оценок существует проблема корректного извлечения необходимой информации.

Применение современных технологий и методов принятия решений требует применения формализованных данных. Поэтому эффективность методов получения решения и поддержки принятия решений зависит от формализации задачи. Так, при декомпозиции сложной задачи до операционного уровня и полной формализации условий решения задачи на этом уровне эффективно для получения решений использование методов исследования операций. При невозмож-

ности декомпозиции и полной формализации используют методы статистической оценки, теорию нечетких множеств и др.

Выбор лучшего варианта формирования мультиверсионного программного обеспечения системы из всего множества возможных реализаций возможно произвести с помощью компенсационной модели многоатрибутивного принятия решений, которая позволяет выполнить общее ранжирования альтернатив на основе порядка их предпочтения по отдельным атрибутам и взаимосвязи между ними и определить наилучший вариант формирования мультиверсионного программного обеспечения.

Разрабатываемая модель принятия решений по составу мультиверсионного программного обеспечения системы предполагает ранжирование возможных вариантов их формирования в порядке предпочтения. Вариант, получивший первый ранг, является наилучшим. Использование в качестве входных данных только порядка предпочтения позволяет избежать масштабирования атрибутов качественного типа [16; 17].

Данная модель основывается на ранжировании альтернатив по отдельным атрибутам. На основе такого «частного» ранжирования определяется общий порядок предпочтительности альтернатив с учетом всех атрибутов и связи между ними.

Существует линейный метод назначения, который позволял выполнить общее ранжирование альтернатив [18]. Согласно данному методу, общий ранг рассчитывался как сумма рангов по отдельным атрибутам. При этом информация о взаимосвязи между атрибутами игнорировалась:

$$r_{i\text{общ}} = \sum_{j=1}^n r_{ij}, \quad i = \overline{1, m}, \quad (1)$$

где m – количество альтернатив; n – количество атрибутов; r_{ij} – ранг i -й альтернативы по j -му атрибуту; r – количество рангов ($r = m$).

Однако для большинства задач принятия решений и, в частности, для выбора варианта формирования мультиверсионного программного обеспечения системы, важно учитывать данную зависимость [19]. В связи с этим, на основе данного метода была разработана компенсационная многоатрибутивная модель общего ранжирования альтернатив в порядке предпочтения с учетом зависимости атрибутов.

Идея компенсации в данном случае заключается в учете зависимости между атрибутами: изменение значения одного из них приводит к изменению значений каких-либо других атрибутов.

Определим матрицу π как квадратную неотрицательную матрицу $m \times m$, чьи элементы π_{ik} представляют количество (или частоту) ранжирования альтернативы A_i по r -му рангу. Матрица π формируется на основе матрицы ранжирования альтернатив по отдельным атрибутам D :

$$\pi_{ik} = \sum_{l=1}^n I(D_{jl}^i) \cdot w_l; \quad i = \overline{1, m}; \quad j = \overline{1, r}, \quad (2)$$

$$I(D_{jl}^i) = \begin{cases} 1, & \text{если } D_{jl}^i = i, \\ 0, & \text{если } D_{jl}^i \neq i, \end{cases} \quad (3)$$

где $I(x)$ – индикаторная функция; w_l – весовой коэффициент l -го атрибута.

При различных весовых коэффициентах элементы матрицы π представляют сумму весов атрибутов соответствующего ранга. Предполагается, что весовые коэффициенты нормированы.

Очевидно, что π_{ik} определяет вклад альтернативы A_i в общее ранжирование. Чем больше значение π_{ik} , тем более справедливо назначение альтернативе A_i r -го ранга.

Определим матрицу перестановки Q , как квадратную матрицу $m \times m$, чьи элементы $Q_{ir} = 1$, если альтернативе A_i назначается общий ранг r , и $Q_{ir} = 0$, в противном случае. Целевая функция может быть записана в следующем виде:

$$\max_{Q^r} \sum_{i=1}^m \sum_{j=1}^r \pi_{ik} Q_{ir} \quad (4)$$

при условиях

$$\sum_{j=1}^r Q_{ir} = 1; i = \overline{1, m}, \quad (5)$$

$$\sum_{i=1}^m Q_{ir} = 1; j = \overline{1, r}. \quad (6)$$

Условия означают, что альтернативе A_i может быть назначен только один ранг и ранг r может быть назначен только одной альтернативе.

Обозначим оптимальную матрицу перестановки, представляющую решение указанной выше задачи линейного программирования, как Q^* . Тогда, оптимальное упорядочение может быть достигнуто перемножением матрицы A , содержащей номера альтернатив, на Q^* .

Рассмотрим пример модельный применения предлагаемой компенсационной многоатрибутивной модели общего ранжирования при выборе одной из трех альтернатив. Пусть все атрибуты имеют возрастающее предпочтение, т. е. чем больше значение атрибута, тем более предпочтительна альтернатива.

Положим, что ранги этих альтернатив по трем отдельным атрибутам соответствуют приведенным в табл. 1. Так, первая альтернатива имеет по первому атрибуту первый ранг, по второму – первый ранг и по третьему – второй.

Таблица 1

Ранжирование альтернатив по отдельным атрибутам

Атрибут		1	2	3
Ранг	1	A1	A1	A2
	2	A2	A3	A1
	3	A3	A2	A3

Ранжирование по отдельным атрибутам может быть представлено в виде матрицы D , элементами которой являются индексы ранжируемых альтернатив:

$$D = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 3 & 2 & 3 \end{bmatrix}. \quad (7)$$

На основе данной матрицы можно получить матрицу π , элементы которой представляют собой количество назначения альтернативе каждого из рангов. Так, первой альтернативе первый ранг был назначен дважды, второй ранг – один раз и третий – ни разу, что соответствует значения в первой строке матрицы:

$$\pi = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}. \quad (8)$$

Для весовых коэффициентов $w_1 = 0,2$, $w_2 = 0,4$, $w_3 = 0,4$ элементы матрицы π изменятся следующим образом:

$$\pi = \begin{bmatrix} 0,2+0,4 & 0,4 & 0 \\ 0,4 & 0,2 & 0,4 \\ 0 & 0,4 & 0,2+0,4 \end{bmatrix} = \begin{bmatrix} 0,6 & 0,4 & 0 \\ 0,4 & 0,2 & 0,4 \\ 0 & 0,4 & 0,6 \end{bmatrix}. \quad (9)$$

Оптимальная матрица перестановок Q^* , определяющая общий ранг каждой из альтернатив, имеет вид

$$Q^* = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

Видно, что первая альтернатива (ей соответствует первый столбец) имеет общий ранг, равный единице (первая строка), вторая альтернатива получила второй ранг и третья – третий. На основе матрицы Q^* получаем следующий порядок предпочтений:

$$A1 \succ A2 \succ A3. \quad (11)$$

Компенсационная модель многоатрибутивного принятия решений позволяет выполнить общее ранжирование альтернатив в порядке предпочтения, основываясь на ранжировании по отдельным атрибутам и взаимосвязи между атрибутами. Достоинством данной модели являются уникальные преимущества при практическом применении. Все, что требуется от эксперта при сборе данных – задание порядка предпочтения альтернатив по атрибутам. Такой подход позволяет избежать трудностей, возникающих при создании шкалы оценки атрибутов.

Алгоритмы оптимизации. *Алгоритмы неявного полного перебора.* В модели с последовательной организацией программных модулей комплекс мультиверсионного программного обеспечения определяется состоящим из набора программных модулей последовательного исполнения, для чего вводится множество I , $card(I) = I$. Множество модулей делится на классы, т. е. вводится множество классов программных модулей (J , $card(J) = J$). Относя некий программный модуль к определенному классу, ему назначается решение соответствующей промежуточной «типовой» задачи управления или обработки информации. Объединение «типовых» программных модулей в комплексы способствует формированию процесса решения общей целевой задачи системы управления.

С целью реализации модулей одного класса назначается программный модуль, который, в свою очередь, для обеспечения гарантоспособности исполнения реализуется с применением методологии мультиверсионного программирования. Для каждого программного модуля в соответствии с исходными спецификациями разрабатывается S_j , ($j = \overline{1, J}$) функционально эквивалентных версий. Таким образом, вводится вектор $\mathbf{S} = \{S_j\}$, ($j = \overline{1, J}$), элементами которого являются числа, равные количеству версий программных модулей (S_j – число версий модуля, решающего задачу класса j – реализующих модуль класса j).

В результате определяется множество всех программных модулей \mathbf{K} , тогда $K = card(K)$ характеризует общее число программных модулей, включаемых в проектируемый программный комплекс.

Кроме того, определяется множество \mathbf{B} , $card(B) = I$ принадлежности задач классам, мощность которого равна числу задач в системе, а каждый элемент этого множества равен номеру класса, которому принадлежит задача. Таким образом, элемент B_i множества \mathbf{B} представляет номер типового модуля, с помощью которого решается i -я задача управления в общем управляющем комплексе.

Введем булевы переменные:

$$X_s^i = \begin{cases} 1, & \text{если для реализации } i\text{-й } (i = \overline{1, I}) \text{ задачи} \\ & \text{используется } s\text{-я } (s = \overline{1, S_{B_i}}) \text{ версия модуля } B_i \\ 0, & \text{в противном случае.} \end{cases} \quad (12)$$

Введенные переменные разворачиваются в вектор участия, формально описывающий возможные варианты формирования состава мультиверсионного комплекса программ. Задача оптимального формирования мультиверсионного программного обеспечения состоит в нахождении набора мультиверсий, который определяет наибольшую надежность программного комплекса при соблюдении определенного уровня финансовых затрат на создание системы [7; 8; 20]. Сформулированная таким образом задача с использованием введенных ранее обозначение была формализована следующим образом:

$$\max R_{NVS1}(X) = \prod_{i=1}^I R_i(X), \quad (13)$$

где $R_i(X) = 1 - \prod_{s=1}^{S_{B_i}} (1 - R_{B_iS})^{X_s^i}$ – оценка надежности i -го программного модуля в составе мультиверсионного комплекса программ; R_{B_iS} – оценка надежности s -й версии i -го программного модуля.

Ограничение на стоимость проектируемого комплекса программ имеет вид

$$C_{NVS1}(X) \leq B, \quad (14)$$

где

$$C_{NVS1}(X) = \sum_{i=1}^I \sum_{s=1}^{S_{B_i}} X_s^i \cdot C_{B_iS}, \quad (15)$$

C_{B_iS} – уровень финансовых затрат на реализацию s -й версии i -го программного модуля проектируемого комплекса программ.

Таким образом, в формальной постановке данного вида задача оптимального формирования мультиверсионного программного обеспечения формулируется как нахождение значений компонент вектора X , таких, что функция надежности программного комплекса (13) принимает свое наибольшее значение при непревышении функцией стоимости (14) определенного значения B .

Для однозначного описания задач формирования комплексов мультиверсионного программного обеспечения, равно как и для конвертирования двух индексов булевых переменных X_s^i , определенных в (12), в один номер компоненты вектора участия необходимо определить процедуру формирования вектора участия и, соответственно, алгоритм определения индексов компонент этого вектора [21–23].

Во второй оптимизационной модели, модели с последовательно-параллельной организацией программных модулей, мультиверсионный программный комплекс также рассматривается состоящим из набора задач управления последовательного исполнения (множество \mathbf{I} , $card(\mathbf{I}) = I$). Подобно первой модели модули программного обеспечения по потребным функциям делятся на J типов, т. е. определяется множество типовых задач управления – множество классов задач – \mathbf{J} , $card(\mathbf{J}) = J$.

Но, в отличие от модели с последовательной организацией модулей, каждая задача $I = \overline{1, I}$ программного комплекса реализуется не одним модулем, а некоторой совокупностью

программных модулей, которая задается в векторе \mathbf{J}_i , где $\mathbf{J}_i = \{j_1^i, \dots, j_{J_i}^i\}$, следовательно, $card(\mathbf{J}_i) = J_i$ – число программных модулей, участвующих в решении i -й задачи и $j_k^i, k = \overline{1, J_i}$ – номер класса, к которому принадлежит k -й программный модуль.

Каждая типовая задача из множества \mathbf{J} реализуется с помощью отказоустойчивого программного модуля, разработанного с использованием подхода мультиверсионного программирования, т. е. каждому элементу множества \mathbf{J} поставлен в соответствие набор версий определяемого им модуля (\mathbf{V}_k – множество версий модуля $k, k = \overline{1, J}, S_k = card(\mathbf{V}_k)$ число версий модуля k). Таким образом, для каждого программного модуля в соответствии с исходными спецификациями разрабатывается $S_j, (j = \overline{1, J})$ функционально эквивалентных версий, т. е. вводится вектор $\mathbf{S} = \{S_j\}, (j = \overline{1, J})$, элементами которого являются числа равные количеству версий программных модулей (S_j – число версий модуля, реализующих модуль класса j).

Как и в первой модели, для формального описания структуры проектируемого комплекса мультиверсионного программного обеспечения вводится вектор участия X , компонентами которого являются булевы переменные X_{kS}^i , такие что

$$X_{kS}^i = \begin{cases} 1, & \text{если для реализации } i\text{-й } (i = \overline{1, I}) \text{ задачи} \\ & \text{используется } s\text{-я } (s = \overline{1, S_{j_k^i}}) \text{ версия модуля } k = (k = \overline{1, J_i}), \\ 0, & \text{в противном случае.} \end{cases} \quad (16)$$

Аналогично модели с последовательной организацией программных модулей, задача оптимального формирования мультиверсионного программного обеспечения в данной модели заключается в определении набора мультиверсий, характеризующего наибольшую надежность программного комплекса при неперевышении установленного бюджета на создание системы, т. е. формально задача оптимизации ставится следующим образом:

$$\max R_{NVS2}(X) = \prod_{i=1}^I R_i(X), \quad (17)$$

где

$$R_i(X) = \prod_{k=1}^{J_i} R_k^i(X), \quad (18)$$

а

$$R_k^i(X) = 1 - \prod_{s=1}^{S_{j_k^i}} (1 - R_{j_k^i, s}^i)^{X_{kS}^i}, \quad (19)$$

при том, что $R_i(X)$ – оценка надежности i -го набора программных модулей в составе мультиверсионного комплекса программ; $R_k^i(X)$ – оценка надежности k -го программного модуля в составе i -го набора, $R_{j_k^i, s}^i$ – оценка надежности s -й версии k -го программного модуля в составе i -го набора.

Ограничение на стоимость проектируемого комплекса программ определяется выражением

$$C_{NVS2}(X) \leq B, \quad (20)$$

где

$$C_{NVS2}(X) = \sum_{i=1}^I \sum_{k=1}^{J_i} \sum_{s=1}^{S_{j_k^i}} X_{kS}^i \cdot C_{j_k^i, s}^i, \quad (21)$$

$C_{j_k^i, s}$ – уровень затрат на реализацию s -й версии k -го программного модуля в составе i -го набора проектируемого комплекса мультиверсионного программного обеспечения.

Очевидно, что если среди множества всех граничных точек выделить две наиболее близкие к точке X_o , все координаты которой равны нулю, и наиболее удаленную от X_o граничную точку и определить уровни I_{\min} и I_{\max} точки X_o , соответствующие этим двум точкам, то станет возможным сужение области поиска решений, так как, бесспорно, решение полученной оптимизационной задачи будет принадлежать множеству, определяемому следующим выражением:

$$S = \bigcup_{i=I_{\min}}^{I_{\max}} O_i(X^o), \quad (22)$$

где X^o – точка пространства булевых переменных, такая что $X_i^o = 0, i = \overline{1, n}$.

Таким образом, чтобы найти решение задачи достаточно определить уровни I_{\min} и I_{\max} точки X_o и сравнить значения целевой функции в элементах множества S , определяемого выражением (22).

Данное свойство позволяет значительно уменьшить вычислительные затраты на поиск точного решения, ускоряя тем самым выполнение процедуры поиска. Мощность множества S определяется следующим выражением

$$\text{card}S = \sum_{k=I_{\min}}^{I_{\max}} C_n^k, \quad (23)$$

где C_n^k – число сочетаний из n элементов по k .

Также неоспоримым является тот факт, что по причине аддитивного характера функций (15) и (19) ближайшая к точке X_o граничная точка будет характеризоваться соответствием ее единичных координат наибольшим слагаемым указанных выражений. Т. е. для того, чтобы достичь из X_o ближайшей к ней граничной точки, необходимо следовать пути наибольшего возрастания функций (15) и (21).

Алгоритм 1. Алгоритм определения уровня I_{\min} точки X_o .

1. Принимаем $I = 0, X \in B_2^n : X_i = 0, i = \overline{1, n}$.

2. Формируем множество стоимостей версий программных модулей проектируемой системы

$C, \text{card}(C) = n$, где n определяется как $n = \sum_{i=1}^I S_{B_i}$ или $n_{NVS2} = \sum_{i=1}^I \left(\sum_{k=1}^{M_i} S_{j_k^i} \right)$, в зависимости от решаемой задачи.

3. Из условия $C_k = \max C_i, i = \overline{1, n}, X_i = 0$ определяем k .

4. Назначаем $X_k = 1$.

5. Если X_k принадлежит множеству допустимых решений, то устанавливаем $I = I + 1$ и переходим к 3, в противном случае, остановка алгоритма.

Алгоритм определения наиболее удаленной от X_o граничной точки (алгоритм определения уровня I_{\max} точки X_o) строится аналогично алгоритму 1, за исключением того, что в пункте 3 следует определять k из условия минимального прироста функции, т. е. определяется k , такое, что $C_k = \min C_i, i = \overline{1, n}, X_i = 0$.

Схема поиска на множестве решений, определяемом выражением (22), реализована в алгоритме усеченного полного перебора.

Алгоритм 2. Алгоритм усеченного полного перебора.

1. Определяем параметры I_{\min} и I_{\max} , соответствующие решаемой задаче.
2. Принимаем $I = I_{\min}$.
3. Определяем вектор X_I^* , такой, что $R(X_I^*) = \max R(X)$, $X \in O_I(X^0)$.
4. Повторяем пункты 2, 3 для всех $I = \overline{I_{\min}, I_{\max}}$.
5. За решение задачи принимается вектор X^* , определяемый из условия $R(X^*) = \max R(X_I^*)$, $I = \overline{I_{\min}, I_{\max}}$.

Следует также выделить следующее свойство псевдобулевых функций (15) и (21): если две соседние друг к другу точки X и Y отличаются значением некоторой i -й компоненты, причем $X_i = 0$, то $C(Y) = C(X) + C_i$, где C_i – стоимость включения в структуру мультиверсии, соответствующей i -и компоненте вектора участия. Таким образом, вид ограничения в построенных оптимизационных задачах позволяет избавиться от полного вычисления выражений (15) и (21) в каждой точке, если обход области поиска решения организован как перемещение по соседним точкам. Полностью значение функции-ограничения для каждой из моделей вычисляется в некоторой начальной точке поиска. Все же последующие значения этой функции получаются прибавлением либо вычитанием соответствующей величины при переходе по соседним точкам.

С целью утилизации вышеописанного свойства построенной оптимизационной задачи была реализована процедура обхода графа булевых переменных по соседним точкам, причем каждая точка области решений просматривается алгоритмом обхода только один раз, что позволяет избежать повышения вычислительной сложности поисковых процедур. В основе этого алгоритма обхода области поиска лежит возможность представления любого подкуба в виде объединения его нижней базовой точки и объединения непересекающихся подкубов меньшей размерности. Такой способ обхода области поиска реализован в алгоритме неявного перебора с обходом области поиска по соседним точкам.

Алгоритм 3. Алгоритм усеченного неявного перебора с обходом области поиска по соседним точкам.

1. Определяем параметры I_{\min} и I_{\max} , соответствующие решаемой задаче.
2. Принимаем $X \in B_2^n : X_i = 1 \forall i \leq I_{\min}$, $k = 0$.
3. С помощью алгоритма неявного перебора с обходом области поиска по соседним точкам определяем X_k^* – точку, дающее наибольшее значение целевой функции на данном этапе.
4. Генерируем следующую точку $X \in O_{X^0}(I_{\min})$, $k = k + 1$.
5. Шаги 3, 4 повторяем $C_n^{I_{\min}}$ раз, где C_n^k – число сочетаний из n по k .
6. За решение принимаем точку $X^* = \max X_k^*$, $k = \bar{1}$, $C_n^{I_{\min}}$.

Алгоритмы схемы метода ветвей и границ. Во второй группе точных поисковых процедур начальная задача разбивается на несколько меньших по размерности подзадач через разбиение области поиска. Разбиение области поиска решения, в свою очередь, реализовано как разбиение пространства булевых переменных на множество непересекающихся подкубов, полностью покрывающих все пространство B_2^n . Общая схема алгоритмов, построенных с использованием данного метода, выглядит следующим образом.

Алгоритм 4. Схема метода ветвей и границ (разбиение булевого гиперкуба на подкубы).

1. Область поиска, представленная (в общем случае) подкубом в бинарном пространстве, разбивается на R непересекающихся подкубов.

2. Определяется принадлежность граничных точек каждому из подкубов.

3. Если r -му ($r = \overline{1, R}$) подкубу принадлежит хотя бы одна граничная точка, то с использованием методов неявного полного перебора находится и запоминается точка X_r подкуба, дающая лучшее значение целевой функции, удовлетворяющее ограничениям.

4. Лучшее из «локальных» решений принимается за решение задачи $f(X^*) = \min, r = \overline{1, R}$.

В общем случае предлагаются два метода разбиения нулевого гиперкуба на подкубы: разбиение B_2^n на $2^{n-n_{sub}}$ подкубов одинаковой размерности $n_{sub} < n$ и метод рекурсивного деления подкубов на два равных по мощности подкуба, размерность которых, очевидно, на единицу меньше размерности делимого подкуба. В основе разбиения пространства булевых переменных на подкубы лежит соответствующее свойство вектора-маски.

Заключение. Оптимизационные задачи, к которым сводится задача формирования комплекса мультиверсионного программного обеспечения, представляют собой задачи так называемого «рюкзачного» типа. Однако особенности целевых функций в оптимизационных моделях не позволяют для их эффективной реализации использовать алгоритмы решения задачи о рюкзаке, разработанные ранее.

Утилизация свойств пространства булевых переменных, в том числе и свойств и отношений подкубов, позволяет разработать эффективные регулярные процедуры реализации построенных оптимизационных моделей, определяющие точное решение задач оптимизации.

Выбор лучшего варианта формирования мультиверсионного программного обеспечения системы основывается на реализации многовариантных решений при применении различных методов многоатрибутивного принятия решений. Компенсационная модель многоатрибутивного принятия решений учитывает информацию о взаимосвязи между атрибутами. Идея компенсации в данном случае заключается в учете зависимости между атрибутами: изменение значения одного из них приводит к изменению значений каких-либо других атрибутов. Модифицированный многоатрибутивный метод упорядоченного предпочтения через сходство с идеальным решением позволяет найти наилучший вариант формирования мультиверсионного программного обеспечения системами в множестве альтернатив любой мощности.

Экспериментальные данные показывают преимущество по времени поиска решения алгоритма неявного перебора с обходом области поиска по соседним точкам перед алгоритмами схемы метода ветвей и границ.

Библиографические ссылки

1. Волков В. А. Многоатрибутивный выбор компонент отказоустойчивого программного обеспечения // Вестник университетского комплекса. 2006. Вып. 8 (22). С. 208–211.

2. Ковалев И. В., Савин С. В. Оптимальное формирование избыточной структуры для отказоустойчивых информационных систем // Исследовано в России. 2004. Т. 7. С. 1123–1129.

3. The hardware and software implementation of the adaptive platform for an onboard spacecraft control system / I. N. Kartsan, A. O. Zhukov, O. A. Platonov, S. V. Efremova // Journal of Physics : Conference Series. 2019. Vol. 1399, No. 3. P. 033071.

4. Choice of optimal multiversion software for a small satellite ground-based control and command complex / I. N. Kartsan, S. V. Efremova, V. V. Khrapunova, M. I. Tolstopiatov // IOP Conference Series: Materials Science and Engineering. 2018. Vol. 450, No. 2. P. 022015.

5. Formation of optimal composition of the modules of single-function multiversion software for automated control system of the satellite communication system / V. I. Kudymov, V. V. Brezitskaya, P. V. Zelenkov, I. N. Kartsan, Yu. N. Malanina // IOP Conference Series: Materials Science and Engineering. 2018. Vol. 450, No. 5. P. 052009.
6. Kovalev I., Davydenko O. Interactive system for spacecraft technological control cycle construction // Program and Abstracts of Int. Symposium SOR'96. TU-Braunschweig (4–6 Sept. 1996). 1996. P. 195.
7. Stupina A. Realization of conventional pattern of random search methods in the space of Boolean variables // Optimization Days. Montreal, 1997. P. 98–112.
8. Юнусов Р. В. Моделирование программных архитектур автоматизированных систем управления // Управляющие и вычислительные системы. Новые технологии : материалы Всерос. электрон. науч.-техн. конф. Вологда : ВоГТУ, 2001. С. 60–61.
9. Stupina A., Antamoshkin A. The random search algorithms in the space of Boolean variables // Symp. OR'97. Jena, 1997. P. 112–118.
10. How to use neural network and web technologies in modeling complex technical systems / M. G. Semenenko, I. V. Kniazeva, L. S. Beckel et al. // IOP Conference Series : Materials Science and Engineering. 2019. Vol. 537, No. 3. P. 032095.
11. Семенько Т. И. Многоатрибутивный подход к формированию программного обеспечения отказоустойчивых систем управления // Успехи современного естествознания, 2004. Вып. 6. С. 32–33.
12. Ковалев И. В., Царев Р. Ю. Комбинированный метод формирования мультиверсионного программного обеспечения управления космическими аппаратами // Авиакосмическое приборостроение. 2006. № 9. С. 8–14.
13. Царев Р. Ю. Многоатрибутивное принятие решений в мультиверсионном проектировании : монография. Красноярск : ИПЦ КГТУ, 2005. 156 с.
14. Efremova S. V., Kartsan I. N., Zhukov A. O. An ordered ranking multi-attributive model for decision-making systems with attributes of control systems software // IOP Conference Series: Materials Science and Engineering. 2021. Vol. 1047. P. 012068.
15. Kartsan I. N. Models for Estimating the Reliability of the Software of an Onboard Control System // Research journal of pharmaceutical biological and chemical sciences. 2018. Vol. 9, No. 5. P. 2357–2367.
16. Царев Р. Ю. Анализ качественных и количественных атрибутов на этапе проектирования отказоустойчивых программных систем // Системы управления и информационные технологии. 2006. № 3 (25). С. 95–101.
17. Царев Р. Ю. Компенсационная модель многоатрибутивного принятия решений при формировании информационных систем управления // Проблемы теории и практики управления. 2007. № 9. С. 63–68.
18. Царев Р. Ю. Преобразование атрибутов при многоатрибутивном принятии решения / Решетневские чтения : тез. докл. V Всерос. научн.-практ. конф. студентов, аспирантов молодых специалистов (12–15 ноября 2001, г. Красноярск). Красноярск, 2001. С. 119–120.
19. Ching-Lai Hwang, Kwangsun Yoon. Multiple Attribute Decision Making. Methods and Application. Berlin : Springer-Verlag, 1981. 255 p.
20. Stupina A., Volf P. Random point processes // САКС : материалы междунар. науч.-практ. конф. Красноярск, 2001. С. 273–276.
21. The multi-objective optimization of complex objects neural network models / V. S. Tynchenko, V. V. Tynchenko, V. V. Bukhtoyarov et al. / Indian Journal of Science and Technology. 2016. Vol. 9, No. 29. P. 99467.

22. Карасева М. В., Карцан И. Н., Зеленков П. В. Метапоисковая мультилингвистическая система // СибГАУ. 2007. № 3 (16). С. 69–70.

23. Карцан И. Н. Мультиверсионное программное обеспечение бортового комплекса управления с генетическим алгоритмом // Решетневские чтения : материалы XXI междунар. науч.-практ. конф. (08–11 ноября 2017, г. Красноярск). Красноярск, 2017. Т. 1. С. 372–373.

References

1. Volkov V. A. [Multiattribute selection of fault-tolerant software components]. *Vestnik universitetskogo kompleksa*. 2006, No. 8 (22), P. 208–211. (In Russ.)

2. Kovalev I. V., Savin S. V. [Optimal formation of redundant structure for fault-tolerant information systems]. *Issledovano v Rossii*. 2004, No. 7, P. 1123–1129. (In Russ.)

3. Kartsan I. N., Zhukov A. O., Platonov O. A., Efremova S. V. The hardware and software implementation of the adaptive platform for an onboard spacecraft control system. *Journal of Physics: Conference Series*. 2019, Vol. 1399, No. 3, P. 033071.

4. Kartsan I. N., Efremova S. V., Khrapunova V. V., Tolstopiatov M. I. Choice of optimal multiversion software for a small satellite ground-based control and command complex. *IOP Conference Series: Materials Science and Engineering*. 2018, Vol. 450, No. 2, P. 022015.

5. Kudymov V. I., Brezitskaya V. V., Зеленков П. В., Kartsan I. N., Malanina Yu. N. Formation of optimal composition of the modules of single-function multiversion software for automated control system of the satellite communication system. *IOP Conference Series: Materials Science and Engineering*. 2018, Vol. 450, No. 5, P. 052009.

6. Kovalev I., Davydenko O. Interactive system for spacecraft technological control cycle construction. *Program and Abstracts of Int. Symposium SOR'96*. TU-Braunschweig (4–6 Sept. 1996). 1996, P. 195.

7. Stupina A. Realization of conventional pattern of random search methods in the space of Boolean variables. *Optimization Days*. 1997, P. 98–112.

8. Yunusov R. V. [Modeling of software architectures of automated control systems]. *Materialy Vserossiiskoi elektronnoi nauch.-tekhn. konf. "Upravlyayushchie i vychislitel'nye sistemy. Novye tekhnologii"* [Materials of the All-Russian Electronic Scientific and Technical Conf. "Management and computing systems. New Technologies"]. 2001, P. 60–61. (In Russ.)

9. Antamoshkin A., Stupina A. The random search algorithms in the space of Boolean variables. *Symp. OR'97*. Jena, 1997, P. 112–118.

10. Semenenko M. G., Kniazeva I. V., Beckel L. S., Rutskiy V. N., Tsarev R. Y., Yamskikh T. N., Kartsan I. N. How to use neural network and web technologies in modeling complex technical systems. *IOP Conference Series: Materials Science and Engineering*. 2019, Vol. 537, No. 3, P. 032095.

11. Semenko T. I. [Multiattributive approach to the formation of fault-tolerant software control systems]. *Successes of modern natural science*. 2004, No. 6, P. 32–33. (In Russ.)

12. Kovalev I. V., Tsarev R. Yu. [Combined method of multi-version spacecraft control software formation]. *Aviation and Space Instrument Engineering*. 2006, No. 9, P. 8–14. (In Russ.)

13. Tsarev R. Yu. *Mnogoatributivnoe prinyatie reshenii v mul'tiversionnom proektirovanii: monografiya* [Multi-attributive decision making in multiversion design : monograph]. Krasnoyarsk, IPC KSTU, 2005, 156 p.

14. Efremova S. V., Kartsan I. N., Zhukov A. O. An ordered ranking multi-attributive model for decision-making systems with attributes of control systems software. *IOP Conference Series: Materials Science and Engineering*. 2021, Vol. 1047, P. 012068.

15. Kartsan I. N. Models for Estimating the Reliability of the Software of an Onboard Control System. *Research journal of pharmaceutical biological and chemical sciences*. 2018, Vol. 9, No. 5, P. 2357–2367.

16. Tsarev R. Yu. [Analysis of qualitative and quantitative attributes in the design phase of fault-tolerant software systems]. *Control systems and information technologies*. 2006, No. 3(25), P. 95–101. (In Russ.)
17. Tsarev R. Yu. [Compensation model of multi-attribute decision making in the formation of management information systems]. *Problems of the theory and practice of management*. 2007, No. 9, P. 63–68. (In Russ.)
18. Tsarev R. Yu. [Attribute transformation in multi-attribute decision making]. *Materialy V Mezhdunar. nauch. konf. "Reshetnevskie chteniya"* [Materials V Intern. Scientific. Conf "Reshetnev reading"]. Krasnoyarsk, 2001, P. 119–120. (In Russ.)
19. Ching-Lai Hwang, Kwangsun Yoon. *Multiple Attribute Decision Making. Methods and Application*, Springer-Verlag, Berlin, 1981, 255 p.
20. Stupina A., Volf P. Random point processes. *Siberian Aviation and Space Salon*. 2001, P. 273–276.
21. Tynchenko V. S., Tynchenko V. V., Bukhtoyarov V. V., Tynchenko S. V., Petrovskiy E. A. The multi-objective optimization of complex objects neural network models. *Indian Journal of Science and Technology*. 2016, Vol. 9, No. 29, P. 99467.
22. Karaseva M. V., Kartsan I. N., Zelenkov P. V. [Meta-search multi-linguistic system]. *Vestnik SibGAU*. 2007, No. 3 (16), P. 69–70. (In Russ.)
23. Kartsan I. N. [The multiversion software of the onboard control complex with genetic algorithm]. *Materialy XXI Mezhdunar. nauch. konf. "Reshetnevskie chteniya"* [Materials XXI Intern. Scientific. Conf "Reshetnev reading"]. Krasnoyarsk, 2017, P. 372–373. (In Russ.)

© Карцан И. Н., Ефремова С. В., 2021

Карцан Игорь Николаевич – доктор технических наук, старший научный сотрудник, Морской гидрофизический институт РАН; профессор, Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева; профессор, Севастопольский государственный университет. E-mail: kartsan2003@mail.ru.

Ефремова Светлана Владимировна – аспирант, Сибирский государственный университет науки и технологий имени академика М. Ф. Решетнева. E-mail: efremova_svet@sibsau.ru.

Kartsan Igor' Nikolaevich – Dr. Sc., Senior Researcher, Marine Hydrophysical Institute, Russian Academy of Sciences, Professor, Reshetnev Siberian State University of Science and Technology; Professor, Sevastopol State University. E-mail: kartsan2003@mail.ru.

Efremova Svetlana Vladimirovna – Postgraduate student, Reshetnev Siberian State University of Science and Technology. E-mail: efremova_svet@sibsau.ru.
