

УДК 004.052.42

DOI 10.47813/nto.3.2022.6.458-469 EDN [COVBDW](#)



## Повышение надёжности систем управления БПЛА экологического мониторинга путём применения общей метрики диверсифицированности для модификации алгоритма голосования согласованным большинством

Д.В. Грузенкин\*

Сибирский федеральный университет, ул. Академика Киренского, 26, к.1,  
Красноярск, 660074, Россия

\*E-mail: [gruzenkin.denis@good-look.su](mailto:gruzenkin.denis@good-look.su)

**Аннотация.** В работе рассматривается проблема повышения надёжности систем управления беспилотными летательными аппаратами, которые применяются для экологического мониторинга и сельскохозяйственных нужд. Повышение надёжности программных систем такого рода может быть достигнуто за счёт применения мультиверсионного программного обеспечения, в котором нередко используется алгоритм голосования согласованным большинством. Однако данный алгоритм может производить выбор предположительно верного ответа всего модуля случайным образом. Это происходит, когда несколько групп версий одинаковой численности выдают различные ответы. Для решения данной проблемы в работе предлагается модифицировать алгоритм голосования согласованным большинством путём применения общей метрики диверсифицированности. В работе описана общая метрика диверсифицированности, а также пара частных метрик, входящих в её состав (метрика диверсифицированности мультиверсий на уровне языков программирования и метрика диверсифицированности на уровне алгоритмов). Также приведена методика модификации алгоритма голосования согласованным большинством. В завершении работы подведены итоги, и сделан вывод о целесообразности применения общей метрики диверсифицированности для модификации алгоритма голосования согласованным большинством с целью повышения надёжности беспилотных летательных аппаратов экологического мониторинга и сельскохозяйственного назначения.

**Ключевые слова:** мультиверсионное программирование, алгоритм голосования, общая метрика диверсифицированности, методика модификации алгоритмов голосования.

## Improving UAVs control systems reliability for environmental monitoring by applying a common diversity metric to modify the agreed by the majority vote algorithm

D.V. Gruzenkin\*

Siberian Federal University, 26 Kirensky st., Krasnoyarsk, 660074, Russia

\*E-mail: [gruzenkin.denis@good-look.su](mailto:gruzenkin.denis@good-look.su)

**Abstract.** The article considers the problem of control systems for unmanned aerial vehicles reliability increasing, which are used for environmental monitoring and agricultural needs. Software systems reliability improving can be achieved through the use of N-version software, which often uses an algorithm for voting by an agreed majority. However, this algorithm can select the presumably correct answer of a entire module randomly. This happens when several groups of versions of the same amount give different answers. To solve this problem, the article proposes to modify the voting algorithm by an agreed majority by applying a common diversity metric. The paper describes the common diversity metric, as well as a couple of particular metrics included in it (the programming languages diversity metric and the algorithm diversity metric). The method of modifying the voting algorithm by an agreed majority is also given. In the article conclusion the results were summed up, and a conclusion was made about the expediency of using a common diversity metric to modify the vote algorithm by an agreed majority in order to increase the reliability of unmanned aerial vehicles for environmental monitoring and agricultural purposes.

**Keywords:** N-version programming, vote algorithm, common diversity metric, voting algorithms modification method.

## 1. Введение

На сегодняшний день активно используются беспилотные летательные аппараты (БПЛА), как для мониторинга лесов и пожарной обстановки в них [10], так и для сельскохозяйственной деятельности [11]. Кроме того, БПЛА могут использоваться и для тушения лесных пожаров [12, 3]. Вообще, их спектр применения очень широк, а требования к надёжности систем управления БПЛА достаточно высоки. Так для повышения надёжности систем управления беспилотными летательными аппаратами может быть использована методология мультиверсионного программирования [2, 7, 8].

Сама по себе методология мультиверсионного программирования является одним из хорошо себя зарекомендовавших подходов для повышения надёжности программного обеспечения, поскольку сбои в функционально эквивалентных версиях одного модуля, содержащих различия в своей реализации, являются независимыми [1]. Однако выбор итогового ответа всего модуля производит какой-либо алгоритм голосования [9].

Одним из часто применяемых алгоритмов голосования в мультиверсионных программных системах является алгоритм голосования согласованным большинством, который может применяться также и в системах беспилотных летательных аппаратов. Однако, как указано в [9], данный алгоритм голосования может осуществлять выбор предположительно верного варианта случайным образом, когда несколько групп версий одинаковой численностью выдали различные результаты. Такое поведение по мнению автора является недопустимым при работе программного обеспечения на реальном БПЛА. Решению указанной проблемы и посвящена данная работа.

## 2. Постановка задачи

Для решения описанной выше проблемы необходимо ввести дополнительные критерии, на основании которых алгоритм голосования согласованным большинством смог бы делать выбор в условиях неопределённости, то есть когда несколько групп версий одинаковой численности выдали разные результаты. Следовательно, необходимо задать эти самые критерии и модифицировать алгоритм голосования согласованным большинством таким образом, чтобы введённые критерии и их значения учитывались при выборе предположительно верного ответа всего модуля – таким образом сформулирована цель работы.

### 3. Методы и материалы исследования

В качестве критериев было принято использовать значения метрик диверсифицированности на уровне языков программирования [5] и на уровне алгоритмов [4]. Стоит отметить, что совместное применение указанных метрик позволит наиболее точного нахождения меры различия между версиями. Поэтому целесообразно ввести общую метрику диверсифицированности, которая включала бы в себя данные метрики и предоставляла бы возможность расширения путём включения в неё новых метрик путём увеличения мерности метрического пространства с введением каждой новой метрики.

#### 3.1. Метрика диверсифицированности на уровне алгоритмов

Для того, чтобы можно было сравнивать алгоритмы выполнения двух программных функционально эквивалентных решений, предлагается на этапе разработки мультIVERсий заложить в них фиксацию текущего состояния (значений переменных) в заранее определённых контрольных точках, количество и места вставки которых определяются индивидуально для каждой версии. По этим контрольным точкам строится трасса выполнения каждого алгоритма, которую уже можно сравнивать с другими трассами [4].

С целью выполнения анализа трасс, (для введения меры различия между алгоритмами, т.е. метрики) для предметной области программных алгоритмов было определено следующее [4]:

1. Значения набора переменных (состояние версии) в каждой контрольной точке – точка в многомерном пространстве.
2. Каждые две точки в многомерном пространстве образуют вектор.
3. Совокупность изменений состояния версии за время работы алгоритма образует трассу.
4. Если в течение нескольких шагов алгоритма состояние остаётся неизменным, точка остаётся на месте [4].

Стоит отметить, что, если разные мультIVERсии возвращают разное количество значений переменных в контрольных точках, то для помещения их в одно метрическое пространство производится допущение, что недостающие координаты у версий с

меньшим числом переменных, определяющих текущее состояние, равны нулю, как при переводе двумерного графика в трёхмерный может быть добавлена нулевая координата  $Z$  (высота), поскольку все точки плоского изображения находятся на одной высоте, которую можно принять за нулевую.

Таким образом, определение метрики различия между алгоритмами может быть осуществлено по нескольким признакам [4]:

1. отношение длины набора переменных состояния (мерности пространства точки) к количеству шагов, т.е. условная скорость прохождения трассы, которая определяется по формуле:

$$v = \frac{|S|}{|steps|},$$

где  $|S|$  – мерность пространства точки,  $|steps|$  – количество точек в трассе;

2. количество и продолжительность общих отрезков у двух трасс, определим степень схожести алгоритмов ( $D$ ) как

$$D = \frac{|SV|}{|V|},$$

где  $|V|$  – количество рёбер трассы 1,  $|SV|$  – количество рёбер трассы 1, совпадающих с рёбрами трассы 2;

3. длина (продолжительность) трасс:

$$l = \sum_{i=1}^{|steps|-1} \sqrt{\sum_{k=1}^n (p_i^k - p_{i+1}^k)^2},$$

где  $l$  – длина всей трассы,  $|steps|$  – количество точек в трассе,  $p_i^k$  – текущая точка трассы,  $p_{i+1}^k$  – следующая после текущей точка трассы,  $n$  – мерность точек,  $k$  – индекс измерения соответствующей точки;

4. отношение длины прямого пути от начальной до конечной точки и длины всей трассы:

$$I = \frac{l_{forward}}{l},$$

где  $l_{forward}$  – длина прямого пути от первой до последней точки трассы,  $l$  – длина трассы.

Для нахождения меры различия полученный показатель схожести необходимо вычесть из 100% или из единицы, в зависимости от выбранных единиц измерения (формула 1):

$$D_i = 1 - S_i, \quad (1)$$

где  $D_i$  - мера диверсифицированности алгоритмов по  $i$ -му показателю,  $S_i$  - мера схожести алгоритмов, определяемая на основании  $i$ -го показателя трасс алгоритмов [4].

Однако, описанной модели недостаточно для введения полноценной метрики, поскольку метрическое пространство может считаться таковым только тогда, когда выполняются 3 аксиомы, в числе которых присутствует и аксиома симметрии, гласящая, что расстояние от точки  $a$  до точки  $b$  должно быть равно расстоянию от точки  $b$  до точки  $a$ . А в текущей модели, например, количество общих отрезков у двух трасс может быть одинаковым при различном количестве отрезков в каждой трассе, поэтому первая трасса со второй может совпадать, например, наполовину, в вторая с первой – на треть [Chen].

Для устранения этого недостатка в метрическое пространство вводится начало координат. Им предложено считать минимальное остовное дерево Штейнера, поскольку оно может включать в себя все вершины всех трасс мультиверсий и при этом связывает их. Таким образом схожесть (отличие) каждой трассы становится возможным определять относительно одного эквивалента – минимального остовного дерева Штейнера [6].

Таким образом, при использовании минимального дерева Штейнера для сравнения с ним трасс алгоритмов могут быть введены следующие показатели (критерии) для сравнения [6]:

1. отношение количества общих отрезков (рёбер) у сравниваемой трассы с минимальным деревом Штейнера к количеству ребер в минимальном дереве Штейнера:

$$S(V) = \frac{|SV|}{|V_{ST}|}, \text{ где } |V_{ST}| \text{ – количество рёбер в минимальном дереве Штейнера, } |SV| \text{ –}$$

количество рёбер трассы, совпадающих с рёбрами минимального дерева Штейнера;

2. отношение количества общих точек (узлов) у сравниваемой трассы и минимальным деревом Штейнера к количеству узлов в минимальном дереве Штейнера:

$$S(E) = \frac{|SE|}{|E_{ST}|}, \text{ где } |E_{ST}| - \text{ количество точек в минимальном дереве Штейнера, } |SE| -$$

количество точек сравниваемой трассы, совпадающих с точками в минимальном дереве Штейнера;

3. отношение длины совпадающих отрезков у сравниваемой трассы и минимального дерева Штейнера к длине минимального дерева Штейнера:

$$S(Vl) = \frac{l_{CV}}{l_{ST}}, \text{ } l_{CV} - \text{ суммарная длина совпадающих рёбер трассы, сравниваемой с}$$

минимальным деревом Штейнера,  $l_{ST}$  - длина минимального дерева Штейнера [6].

В результате итоговая мера диверсифицированности двух алгоритмов определяется как Евклидово расстояние между точками в многомерном пространстве. Такими точками являются вычисленные по (формуле 1) меры диверсифицированности по каждому из показателей трасс алгоритмов (формула 2):

$$D_A = \sqrt{\sum_{k=1}^n (D_k^1 - D_k^2)^2}, \quad (2)$$

где  $D_k^1$  - мера диверсифицированности трассы 1 по  $k$ -му показателю,  $D_k^2$  - мера диверсифицированности трассы 2 по  $k$ -му показателю [6].

### 3.2. Метрика диверсифицированности на уровне языков программирования

Метрика диверсифицированности на уровне языков программирования вводится на основании возможности сравнения языков программирования, на которых реализованы мультиверсии, по определённым признакам. Например, по способам работы с памятью или возможностью программирования в нескольких парадигмах [5].

Таким образом, каждому признаку каждого языка программирования может быть поставлено в соответствие некоторое непустое множество значений. Что по своей сути является формализацией задачи сравнения языков программирования между собой по некоторому набору признаков. То есть путём нахождения мощности множества, которое образовалось на пересечении двух множеств значений определенного признака двух

конкретных языков программирования, может быть получено количество сходных значений данного признака. При делении этого значения на общее число значений данного признака для конкретного языка может быть найдено процентное соотношение схожести одного языка с другим [5].

То есть степень схожести двух языков программирования по  $k$ -му признаку определяется как (формула 3):

$$S_{ij}^k = \frac{|set_i(k) \cap set_j(k)|}{|set_i(k)|}, i = \underline{1,2}, j = \underline{1,2}, i \neq j, \quad (3)$$

где  $i$  и  $j$  – индексы языков программирования,  $k$  – номер признака из всего множества признаков,  $set_i(k)$  – множество значений  $k$ -го признака  $i$ -го языка программирования [5].

А для расчета степени схожести двух языков программирования с учетом всех признаков используется (формула 4):

$$S_{ij} = \frac{\sum_{k=1}^m S_{ij}^k}{m}, \quad (4)$$

где  $i$  и  $j$  – индексы языков программирования,  $k$  – номер признака из всего множества признаков,  $m$  – число признаков,  $S_{ij}^k$  – степень схожести двух языков программирования по  $k$ -му признаку [5].

Например, один язык позволяет писать код в функциональном и объектно-ориентированном стиле, а другой – в объектно-ориентированном, процедурном и функциональном. Тогда можно утверждать, что язык 1 схож со вторым языком на 100% по заданному критерию, в второй язык совпадает с первым лишь на 2/3, поскольку у него с первым совпали только две из трёх возможных парадигмы программирования. Поэтому вероятно, что и написанные на этих языках версии будут отличаться друг от друга по заданному критерию в том же соотношении. Однако снова встаёт вопрос определения единой точки начала координат в данном метрическом пространстве [5].

В качестве такой точки отсчёта было предложено использовать «идеальный» несуществующий язык программирования, который включает в себя все значения всех критериев сравниваемых языков программирования. И уже с эти «идеальным» языком



сравниваются все остальные языки программирования, на которых были реализованы мультиверсии [5].

Описанное логическое заключение приводит (формулу 4) к следующему виду (формула 5):

$$S_{ij} = \frac{\sum_{k=1}^m (S_{ip}^k - S_{jp}^k)}{m}, \quad (5)$$

где  $i$  и  $j$  – индексы языков программирования,  $k$  – номер признака из всего множества признаков,  $m$  – число признаков,  $S_{ip}^k$  и  $S_{jp}^k$  – степени схожести  $i$ -го и  $j$ -го языков программирования, соответственно, по  $k$ -му признаку с «идеальным» языком программирования, который является началом координат в метрическом пространстве [5].

В итоге метрика диверсифицированности двух мультиверсий одного модуля мультиверсионного программного обеспечения на уровне языков программирования ( $D_L$ ) может быть рассчитана по (формуле 6) [5]:

$$D_L = 1 - S_{ij}. \quad (6)$$

### 3.3. Общая метрика диверсифицированности

Общая метрика диверсифицированности подразумевает использование конечных значений метрик диверсифицированности на уровне алгоритмов и языков программирования, а также любых других метрик диверсифицированности, которые могут быть введены в будущем, как значений координат в  $N$ -мерном метрическом пространстве. Мерность пространства равна количеству используемых частных метрик.

При использовании данной модели каждая мультиверсия представляется точкой со своими координатами в  $N$ -мерном пространстве, благодаря чему между ними может быть найдено Евклидово расстояние, которое показывает степень удалённости версий друг от друга в общей системе координат.

Таким образом, общая метрика диверсифицированности двух мультиверсий определяется как Евклидово расстояние между точками в многомерном пространстве. Такими точками, вычисленными по (формуле 2) и (формуле 6), являются меры диверсифицированности на каждом из уровней диверсификации (формула 7):



$$D = \sqrt{\sum_{k=1}^n (D_n^1 - D_n^2)^2}, \quad (7)$$

где  $D_n^1$  - мера диверсифицированности мультиверсии 1 по  $n$ -ой метрике,  $D_n^2$  - мера диверсифицированности мультиверсии 2 по той же,  $n$ -ой, метрике,  $n = \overline{1, N}$ ,  $N$  – количество метрик диверсифицированности, входящих в общую метрику.

#### 4. Полученные результаты

Для достижение поставленной в данной работе цели необходимо в процессе голосования использовать описанную выше общую метрику диверсифицированности для определения группы наиболее различных между собой версий с целью определения верного ответа с более высокой вероятностью, поскольку мультиверсионная методология основана на утверждении, что чем более разнообразны версии в модуле, тем ниже вероятность возникновения в них зависимых или идентичных сбоев.

Результатом данной работы стал ввод метрики диверсифицированности в алгоритм голосования согласованным большинством, для чего необходимо выполнить следующие действия:

1. Определить набор частных метрик диверсифицированности, которые отражают наиболее значимые различия мультиверсий для решения поставленной задачи;
2. Рассчитать до запуска модуля (или системы в целом) те метрики диверсифицированности, которые не меняются после исполнения мультиверсий (как метрика диверсифицированности на уровне языков программирования);
3. Обеспечить расчёт метрик, пересчитываемых после каждого выполнения мультиверсий (как метрика диверсифицированности на уровне алгоритмов);
4. Обеспечить расчёт общей метрики диверсифицированности для каждой мультиверсии;
5. Результаты выполнения мультиверсий сравниваются между собой по заданному алгоритму для нахождения единственного верного ответа. Если единственный результат алгоритма голосования найден, то он принимается за результат работы модуля. Если в ходе выполнения алгоритма голосования выяснилось, что несколько групп мультиверсий, куда входит одинаковое количество версий, выдали разные результаты, происходит их сравнение с использованием метрики диверсифицированности;

6. Если необходимо сравнить между собой несколько групп мультиверсий по уровню их диверсифицированности, то необходимо определить, какая величина будет взята за уровень диверсифицированности группы версий, исходя из специфики конкретной задачи. Такой величиной могут быть, например, максимальное расстояние между двумя мультиверсиями входящими в одну группу, а также среднее или медианное расстояние между всеми версиями в группе в пространстве общей метрики диверсифицированности;
7. В качестве решения принимается тот результат, который был получен группой мультиверсий, где уровень диверсифицированности версий по определённому на шаге 6 признаку является максимальным.

Стоит отметить, что для модификации с применением метрики диверсифицированности подходит не только алгоритм голосования согласованным большинством, но и другие алгоритмы, позволяющие сравнивать в том числе одинаковые по количеству группы версий, например,  $t/(n-1)$ -вариантное программирование.

## 5. Выводы

В заключении следует сказать, что может быть определена одна или несколько частных метрик диверсифицированности, оказывающих наибольшее влияние на общую метрику диверсифицированности, по средствам регрессионного анализа.

При последующих выполнениях мультиверсионного модуля расчёт общей метрики диверсифицированности, вычисляемой на шаге 4, может быть заменён расчётом одной или нескольких частных метрик для более оптимального использования вычислительных ресурсов.

С целью контроля эффективности алгоритма принятия решения (голосования) могут вычисляются невязки – определяется отклонение значения общей метрики диверсифицированности, вычисленной на основе частных метрик, от значения данной метрики, полученного на основании накопленной статистики; что позволяет определить возможные скрытые ошибки в мультиверсиях, нестандартные входные данные или недавние изменения в составе мультиверсий модуля.

Таким образом, применение модифицированного с помощью общей метрики диверсифицированности алгоритма голосования в мультиверсионных системах

управления беспилотными летательными аппаратами может помочь не только в вопросе повышения надёжности программных систем такого рода, но и в выявлении зависимых ошибок в группах мультиверсий.

### Список литературы

1. Chen, L. N-version programming: A fault-tolerance approach to reliability of software operation / L. Chen, A. Avizienis // Proc. 8th IEEE Int. Symp. on Fault-Tolerant Computing (FTCS-8). – 1978. – Т. 1. – С. 3-9.
2. Kovalev, I. Model implementation of the simulation environment of voting algorithms, as a dynamic system for increasing the reliability of the control complex of autonomous unmanned objects / I. Kovalev, V. Losev, M. Saramud, M. Petrosyan // MATEC Web of Conferences. – EDP Sciences, 2017. – Т. 132. – С. 04011.
3. Альтмаер, Е.Э. Использование беспилотных летательных аппаратов при тушении пожаров / Е.Э. Альтмаер, И.Д. Мягких, М.В. Назаров // Повышение качества образования, современные инновации в науке и производстве. – 2021. – С. 390-397.
4. Грузенкин, Д.В. Определение метрики диверсифицированности мультиверсионного программного обеспечения на уровне алгоритмов / Д.В. Грузенкин, И.А. Якимов, А.С. Кузнецов, Р.Ю. Царев // Фундаментальные исследования. – 2017. – №. 6. – С. 36-40.
5. Грузенкин, Д. В. Определение метрики диверсифицированности мультиверсионного программного обеспечения на уровне языков программирования / Д.В. Грузенкин, А.С. Михалев // Программная инженерия. – 2019. – Т. 10. – №. 9-10. – С. 384-390.
6. Грузенкин, Д.В. Оценка меры различия алгоритмов в многовариантной системе составления производственных планов / Д.В. Грузенкин, А.С. Кузнецов, И.В. Селезнев // Системы анализа и обработки данных. – 2020. – №. 4 (80). – С. 65-80.
7. Ковалев, И.В. К вопросу реализации мультиверсионной среды исполнения бортового программного обеспечения автономных беспилотных объектов средствами операционной системы реального времени / И.В. Ковалев, В.В. Лосев, М.В. Сарамуд [и др.]. // Сибирский журнал науки и технологий. – 2017. – Т. 18. – №. 1. – С. 58-61.
8. Ковалев, И. К вопросу формирования блочно-модульной структуры системы управления беспилотных летательных объектов / И. Ковалев, В. Лосев, М. Сарамуд

- [и др.]. // Современные инновации, системы и технологии. – 2021. – Т. 1. – №. 3. – С. 48-64.
9. Котенок, А.В. РЕАЛИЗАЦИЯ АЛГОРИТМОВ МУЛЬТИВЕРСИОННОГО ГОЛОСОВАНИЯ / А.В. Котенок // Современные наукоемкие технологии. – 2007. – № 8. – С. 44-45
10. Кретинин, А.А. Цифровизация управления охраной лесов на базе мониторинга лесных пожаров / А.А. Кретинин, Т.Л. Безрукова // Актуальные направления научных исследований XXI века: теория и практика. – 2022. – Т. 10. – №. 1. – С. 139-152.
11. Тимирбаев, А.М. Беспилотный летательный аппарат в сельском хозяйстве / А.М. Тимирбаев // Наука, техника и образование. – 2022. – №. 4(87). – С. 77-78.
12. Чжан, С. Перспективы применения миниатюрных беспилотных летательных аппаратов в роботизированных системах обнаружения и тушения пожара / С. Чжан, М.А. Васильев, О.А. Зыбина // Научно-аналитический журнал «Вестник Санкт-Петербургского университета Государственной противопожарной службы МЧС России». – 2022. – №. 2. – С. 17-24.