

ИНФОРМАТИКА,
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И УПРАВЛЕНИЕ

INFORMATICS,
COMPPUTER ENGINEERING
AND CONTROL

УДК 004.054+658.512.6

DOI: 10.17212/1814-1196-2020-4-65-80

Оценка меры различия алгоритмов в многовариантной системе составления производственных планов*

Д.В. ГРУЗЕНКИН^а, А.С. КУЗНЕЦОВ^б, И.В. СЕЛЕЗНЕВ^с

660074, РФ, г. Красноярск, ул. Киренского, 26, Институт космических и информационных технологий, Сибирский федеральный университет

^а dgruzenkin@sfu-kras.ru ^б askuznetsov@sfu-kras.ru ^с seleznevigor2001@gmail.com

В процессе проектирования производственных планов одним из важных шагов является составление расписаний выполнения технологических операций. Расписание может составляться как вручную, так и с помощью программного обеспечения. Если расписание составляется программным средством, то для исключения возможных ошибок используют несколько алгоритмов генерации расписания. Набор таких алгоритмов называют «пачкой». В «пачку» целесообразно включать только различные алгоритмы. Это необходимо для исключения однотипных ошибок. Поэтому поиск клонов алгоритмов в «пачке» является актуальной производственной задачей. Для ее решения в ходе выполнения настоящей работы была разработана метрика диверсифицированности алгоритмов. Такая метрика численно (в процентах) определяет, насколько алгоритмы различны. Данная метрика основывается на характеристиках трасс выполнения алгоритмов. Трассы алгоритмов строятся в N -мерном пространстве по полученным точкам. Координатами точек трасс являются значения, с которыми работает алгоритм на каждом шаге своего выполнения или в каждой из контрольных точек выполнения алгоритма. Для подтверждения корректности работы данной метрики был поставлен эксперимент. В его рамках были вычислены характеристики трасс трех алгоритмов сортировки. На основании полученных характеристик были определены показатели для сравнения алгоритмов в метрическом пространстве. Эксперимент подтвердил эффективность использования метрики диверсифицированности для нахождения клонов в «пачке алгоритмов». Область применения данной метрики не ограничивается лишь поиском клонов. Она может быть применена как самостоятельный показатель качества программного обеспечения.

Ключевые слова: алгоритм, мера различия, метрика, диверсификация, метрика диверсифицированности, производственный план, составление расписания, диверсификация на уровне алгоритмов

* Статья получена 20 августа 2020 г.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-07-00226.

ВВЕДЕНИЕ

В процессе проектирования производственных планов одним из важных шагов является составление расписаний выполнения технологических операций, что особенно важно для мелкосерийного производства с изменяющимся портфелем заказов [1]. Расписание может составляться вручную или генерироваться автоматически при наличии соответствующих программных средств в соответствии с теми или иными алгоритмами. К настоящему моменту известно большое количество алгоритмов планирования технологических операций [2–6].

Введем понятие варианта расписания, под которым будем понимать алгоритмическое описание технологического расписания для случая одной машины. Отсюда можно сформулировать понятие многовариантного расписания, когда одно и то же расписание с одинаковыми ограничениями и (при необходимости) критериями оптимальности получено разными алгоритмами.

В производственных условиях проблемой является эффективное выявление логических ошибок в программной реализации того или иного алгоритма, применяемого для составления расписания. Это может выполняться разными способами, одним из которых является одновременное использование нескольких алгоритмов планирования («пачка алгоритмов»). При этом если для одного и того же входа алгоритмы выдают разные результаты, то делается вывод о наличии ошибки (ошибок) в одном или нескольких алгоритмах в «пачке». Благодаря этому в значительной мере повышается надежность программного средства планирования производственных процессов.

В случае, когда разрабатывается новый алгоритм планирования, он включается в состав «пачки» и проверяется на наличие логических ошибок описанным выше образом путем сравнения с базовыми алгоритмами. Однако в таком случае вероятна ситуация, когда «новый» алгоритм является «клоном» (копией) другого алгоритма, например, одного из базовых. Очевидно, такой алгоритм не должен представлять практического интереса. Отсюда возникает проблема выявления «клонов» в «пачке».

1. ОПРЕДЕЛЕНИЕ МЕТРИКИ ДИВЕРСИФИЦИРОВАННОСТИ АЛГОРИТМОВ

Таким образом, для решения вышеописанной задачи необходимо ввести меру (метрику) различия алгоритмов в «пачке», которая однозначно в числовом виде определяла бы, насколько сравниваемые алгоритмы различны между собой. В качестве такой метрики может быть использована метрика из статьи [7].

Для определения метрики из [7] необходимо, чтобы каждый алгоритм во время своего выполнения на каждом своем шаге или в каждой контрольной точке фиксировал текущий набор данных, с которым работает. В качестве примера авторами используются алгоритмы сортировки одномерного массива, поэтому на каждом шаге фиксируется текущее состояние массива (порядок элементов в массиве). Каждое число в массиве является значением одной из координат в N -мерном пространстве, мерность пространства определяется количеством элементов в массиве. Из полученного набора точек в N -мерном пространстве строятся трассы выполнения всех алгоритмов.

В общем случае фиксируемый на каждом шаге (или в каждой контрольной точке) набор данных может содержать и различное количество значений, поэтому для определения метрики, описываемой в работе [7], будем за мерность метрического пространства брать наибольшее количество элементов, зафиксированных на каком-либо шаге любого из алгоритмов в «пачке», а недостающие координаты в других наборах данных будем заполнять нулями. Это справедливо, так как если мерность пространства одной точки больше, чем у другой, это значит, что все точки пространства с меньшей мерностью имеют одну и ту же координату (нулевую) в пространстве с большей мерностью. Как, например, при переводе двух точек на плоскости в трехмерное пространство может быть добавлено нулевое значение третьей координаты в декартовом пространстве, поскольку они в действительности лежат в одной плоскости и удалены по оси Z на одно и то же (нулевое) расстояние от начала координат по вертикали.

Метрика различия мультиверсий в мультиверсионном программном обеспечении на уровне алгоритмов позволяет определять следующие показатели для описания полученных трасс:

1) отношение мерности пространства точек к количеству точек (узлов) в трассе, т. е. условная скорость прохождения трассы

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

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

2) отношение количества общих отрезков у двух трасс к количеству ребер (отрезков) одной из трасс

$$S(V) = \frac{|SV|}{|V|},$$

где $|V|$ – количество ребер трассы 1, $|SV|$ – количество ребер трассы 1, совпадающих с ребрами трассы 2;

3) длина трасс (сумма расстояний между точками в евклидовом пространстве)

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

где l – длина всей трассы, $|steps|$ – количество точек в трассе, p_i – текущая точка трассы, p_{i+1} – 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 [7];

5) отношение количества общих точек у двух трасс (по аналогии с пунктом 2) к количеству точек в одной из трасс:

$$S(E) = \frac{|SE|}{|E|},$$

где $|E|$ – количество точек трассы 1, $|SV|$ – количество точек трассы 1, совпадающих с точками трассы 2;

6) количество точек (узлов) в трассе;

7) количество отрезков (ребер) в трассе;

8) длина совпадающих отрезков двух трасс;

9) отношение длины совпадающих отрезков двух трасс к длине одной из них:

$$S(V) = \frac{l_{CV}}{l_i},$$

где l_{CV} – суммарная длина совпадающих ребер трасс 1 и 2, l_i – длина i -й трассы, $i = \overline{1, 2}$;

Показатели 1, 3, 4, 6 и 7 определяют индивидуальные характеристики трасс, которые могут относиться лишь к каждой конкретной трассе в отдельности. Их можно применять для сравнения трасс алгоритмов, однако в случае наличия различий в них данные показатели не смогут определить, насколько велики данные различия.

Показатели 2, 5, 8 и 9 определяют меру схожести двух трасс, причем показатель 8 (длина совпадающих отрезков у двух трасс) является абсолютным. Если использовать его и только его для сравнения трасс, нельзя в точности утверждать, насколько в действительности схожи трассы, как в случае с показателями 1, 3, 4, 6, 7. Показатели 2, 5 и 9 при этом отражают именно степень схожести двух сравниваемых трасс друг относительно друга. В случае нахождения такого отношения значения данных показателей будут лежать в промежутке от 0 до 1, значит, их легко перевести в процентный показатель, численно отражающий меру схожести двух трасс. Для нахождения меры различия полученный показатель схожести необходимо вычесть из 100 % или из единицы (в зависимости от выбранных единиц измерения):

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

где D_i – мера диверсифицированности алгоритмов по i -му показателю; S_i – мера схожести алгоритмов, определяемая на основании i -го показателя трасс алгоритмов.

Несмотря на то что эти показатели подходят для решения прикладной задачи нахождения клонов алгоритмов в «пачке алгоритмов», поставленной в настоящей работе, мы полагаем, что для определения полноценной метрики различия алгоритмов их все-таки недостаточно. Поскольку при вводе данной метрики должны соблюдаться все аксиомы метрического пространства, в том числе и аксиома симметрии, расстояние от точки a до точки b должно быть таким же, как расстояние от b до a [8]. При использовании показателей 2, 5 и 9 эта аксиома нарушается, так как длина, количество отрезков (ребер) и ко-

личество точек (узлов) у двух сравниваемых трасс может быть различным, поэтому знаменатель у всех этих показателей будет различаться в зависимости от порядка, в котором мы сравниваем трассы.

Таким образом, для определения полноценной метрики диверсифицированности алгоритмов необходимо определить точку начала координат, чтобы находить удаленность каждого показателя трассы в метрическом пространстве относительно начала координат, а не относительно аналогичных показателей какой-то другой трассы. Иными словами, необходимо определить такую трассу гипотетического алгоритма, которая имела бы что-то общее с каждой из сгенерированных трасс алгоритмов, чтобы каждую из трасс алгоритмов можно было сравнивать с ней.

Такой трассой, с которой будет сравниваться каждая из трасс алгоритмов из «пачки», может стать минимальное дерево Штейнера [9]. Для его построения по имеющимся точкам на сегодняшний день разработано достаточно большое количество численных и аналитических методов, а также их программных реализаций. Такое дерево будет содержать все точки всех трасс исследуемых алгоритмов. Благодаря этому показатели трасс 2, 5 и 9 могут быть применены не для сравнения трасс двух конкретных алгоритмов, а для сравнения каждой трассы алгоритма из «пачки» с минимальным деревом Штейнера.

Таким образом, при использовании минимального дерева Штейнера для сравнения с ним трасс алгоритмов формулы показателей 2, 5 и 9 соответственно приобретают следующий вид.

Отношение количества общих отрезков (ребер) у сравниваемой трассы с минимальным деревом Штейнера к количеству ребер в минимальном дереве Штейнера:

$$S(V) = \frac{|SV|}{|V_{ST}|},$$

где $|V_{ST}|$ – количество ребер в минимальном дереве Штейнера; $|SV|$ – количество ребер трассы, совпадающих с ребрами минимального дерева Штейнера;

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

$$S(E) = \frac{|SE|}{|E_{ST}|},$$

где $|E_{ST}|$ – количество точек в минимальном дереве Штейнера; $|SE|$ – количество точек сравниваемой трассы, совпадающих с точками в минимальном дереве Штейнера;

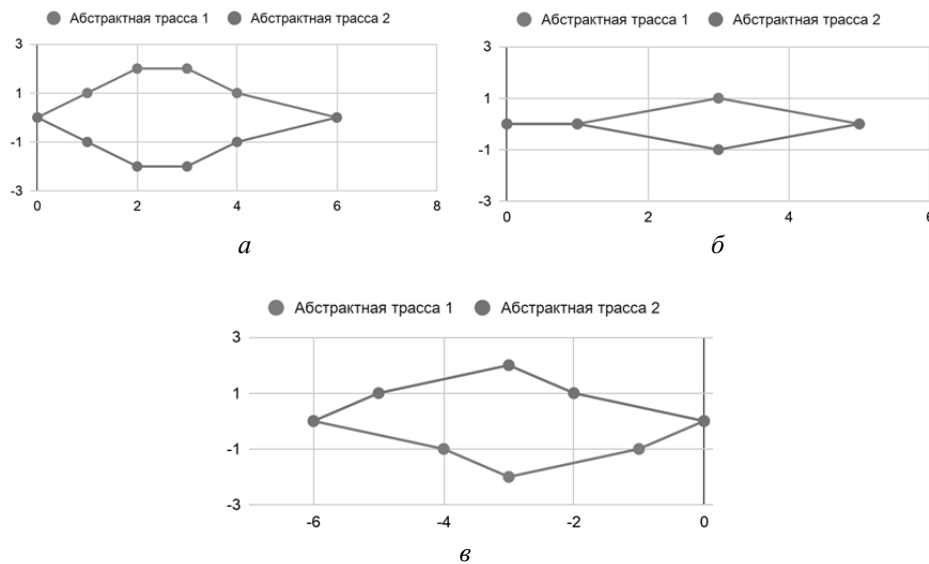
Отношение длины совпадающих отрезков у сравниваемой трассы и минимального дерева Штейнера к длине минимального дерева Штейнера:

$$S(V) = \frac{l_{CV}}{l_{ST}},$$

l_{CV} – суммарная длина совпадающих ребер трассы, сравниваемой с минимальным деревом Штейнера; l_{ST} – длина минимального дерева Штейнера.

Теперь, применяя измененные показатели, возможно определить удаленность трассы от минимального дерева Штейнера как от начала координат сразу по трем координатам. Таким образом может быть определено трехмерное метрическое пространство, в котором соблюдаются все аксиомы метрических пространств, в том числе и аксиома симметрии [8]. Соответственно, в таком пространстве уже может быть определена метрика схожести и диверсифицированности алгоритмов. Кроме того, в данное метрическое пространство могут быть добавлены и остальные вышеперечисленные показатели в качестве дополнительных координат по которым также может производиться сравнение. Получается девятимерное метрическое пространство, в котором мера различия алгоритмов может быть выражена численно.

При его применении на практике в представленном виде могут возникать неточности в случаях, когда сравниваемые алгоритмы имеют трассы одинаковой длины и с одинаковым количеством точек. Так, например, на рисунке приведены абстрактные гипотетические трассы прохождения алгоритмов в двумерном пространстве, которые по всем указанным выше параметрам будут считаться клонами, однако фактически ими не являются.



Трассы гипотетических алгоритмов в двумерном пространстве, которые схожи по всем показателям, но фактически не являются клонами

Traces of hypothetical algorithms in a two-dimensional space which are similar in all characteristics, but are not clones

Для исключения подобных неточностей целесообразно ввести несколько дополнительных показателей, определяющих характеристики трасс:

Отношение количества ребер сравниваемой трассы, совпадающих с ребрами всех других сравниваемых трасс, к среднему количеству ребер во всех сравниваемых трассах:

$$S(V_{Avg}) = \frac{|SV_j|}{|V_{AVE}|},$$

где $|V_{AVE}|$ – среднее количество ребер во всех сравниваемых трассах; $|SV_j|$ – количество ребер сравниваемой трассы, совпадающих с ребрами всех остальных сравниваемых трасс; $j = \overline{1, m}$, m – количество сравниваемых трасс;

11) отношение количества точек (узлов) сравниваемой трассы, совпадающих с точками всех других сравниваемых трасс, к среднему количеству узлов во всех сравниваемых трассах:

$$S(E_{Avg}) = \frac{|SE_j| - 2}{|E_{AVE}|},$$

где $|E_{AVE}|$ – среднее количество точек во всех сравниваемых трассах; $|SE_j|$ – количество точек сравниваемой трассы, совпадающих с точками всех других сравниваемых трасс; $j = \overline{1, m}$, m – количество сравниваемых трасс; 2 – количество точек, которые должны совпадать во всех трассах (точка начала и конца);

12) отношение длины совпадающих отрезков сравниваемой трассы и всех остальных сравниваемых трасс к средней длине всех сравниваемых трасс:

$$S(L_{Avg}) = \frac{l_j^S}{l_{AVE}},$$

где l_j^S – суммарная длина совпадающих ребер сравниваемой трассы с ребрами всех остальных сравниваемых трасс; l_{AVE} – средняя длина всех сравниваемых трасс.

Преимущество этих трех показателей перед описанными выше заключается в следующем. Мы полагаем, что здесь не просто какая-то из характеристик сравниваемой трассы соотносится с такой же характеристикой «точки начала координат». Вместо этого с началом координат соотносится именно та часть сравниваемой трассы, которая является общей для всех трасс алгоритмов из «пачки». Очевидно, что с помощью данных показателей можно сравнивать на различимость любое количество алгоритмов в «пачке», однако наиболее точные результаты будут давать при сравнении только двух алгоритмов.

Таким образом, итоговая мера схожести двух алгоритмов определяется как евклидово расстояние между точками в многомерном пространстве. Такими точками являются меры диверсифицированности по каждому из показателей трасс алгоритмов из «пачки», вычисленные по формуле (1):

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

где D_k^1 – мера диверсифицированности трассы 1 по k -му показателю; D_k^2 – мера диверсифицированности трассы 2 по k -му показателю.

2. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТА

Для апробации предложенной метрики диверсифицированности и проверки ее эффективности был проведен эксперимент, в рамках которого предполагалось использование одной сортировочной машины и стратегии планирования SJF («кратчайшее задание – первым») [10] для сортировки вторсырья [11]. Имеется последовательность операций с оценочным временем их выполнения. Следуя стратегии SJF, задания не имеют приоритетов, а машина будет отодвигать выполнение длительной операции на более поздний срок. В ходе проведения эксперимента были использованы три алгоритма упорядочения последовательности по длительности каждой из операций: алгоритмы сортировки вставкой, групповой сортировки и гномьей сортировки [12]. После выполнения данных алгоритмов были получены три трассы в шестимерном пространстве, состоящие из значений продолжительности заданий, которые определяют координаты точки. Полученные трассы представлены в табл. 1.

Таблица 1

Table 1

Трассы алгоритмов сортировок
Sorting Algorithms Traces

Шаг	Сортировка вставкой	Групповая сортировка	Гномья сортировка
№	Трасса A	Трасса B	Трасса C
0	{5, 0, 2, 4, 1, 3}	{5, 0, 2, 4, 1, 3}	{5, 0, 2, 4, 1, 3}
1	{0, 5, 2, 4, 1, 3}	{0, 5, 2, 4, 1, 3}	{0, 5, 2, 4, 1, 3}
2	{0, 2, 5, 4, 1, 3}	{0, 2, 5, 4, 1, 3}	{0, 2, 5, 4, 1, 3}
3	{0, 2, 4, 5, 1, 3}	{0, 2, 4, 5, 1, 3}	{0, 2, 4, 5, 1, 3}
4	{0, 1, 2, 4, 5, 3}	{0, 2, 4, 1, 5, 3}	{0, 2, 4, 1, 5, 3}
5	{0, 1, 2, 3, 4, 5}	{0, 2, 1, 4, 5, 3}	{0, 2, 1, 4, 5, 3}
6		{0, 1, 2, 4, 5, 3}	{0, 1, 2, 4, 5, 3}
7		{0, 1, 2, 4, 3, 5}	{0, 1, 2, 4, 3, 5}
8		{0, 1, 2, 3, 4, 5}	{0, 1, 2, 3, 4, 5}

Значения показателей 1, 3, 4, 6, 7 и 8 трасс алгоритмов могли бы быть при необходимости включены в *метрическое* пространство (не в пространство, в котором проходят трассы), как уже было сказано выше, для определения меры схожести, а на ее основе и меры различия алгоритмов. Фактически же они представляют собой те компоненты, из которых строятся показатели, определяющие меру схожести алгоритмов относительно «начала координат». В рамках предложенного подхода для определения меры различия будут применяться только показатели трасс 2, 5, 9, 10, 11 и 12.

Для краткости трассы алгоритмов сортировки вставкой, групповой сортировки и гномьей сортировки далее обозначаются как трассы A , B и C соответственно.

Для составления сравнительных характеристик между наборами точек, составляющих эти три трассы, требуется найти количество точек, отрезков и длины этих трасс. Последние рассчитываются как суммы евклидовых расстояний между парами следующих друг за другом точек. Таким образом, для трасс A , B и C получаем соответствующие длины:

$$l_A = \sqrt{50} + \sqrt{18} + \sqrt{2} + \sqrt{22} + \sqrt{6} \approx 19,8678;$$

$$l_B = \sqrt{50} + \sqrt{18} + \sqrt{2} + \sqrt{32} + \sqrt{18} + \sqrt{2} + \sqrt{8} + \sqrt{2} \approx 28,2843;$$

$$l_C = \sqrt{50} + \sqrt{18} + \sqrt{2} + \sqrt{32} + \sqrt{18} + \sqrt{2} + \sqrt{8} + \sqrt{2} \approx 28,2843.$$

Также для последующего сравнения трасс было построено минимальное дерево Штейнера с помощью инструмента ESMT-Smith [13].

Перечисленные данные трасс и графа Штейнера приведены в табл. 2.

Таблица 2

Table 2

Значения некоторых индивидуальных параметров сравниваемых трасс и дерева Штейнера

Values of some individual parameters of the compared traces and the Steiner tree

Характеристика	Трасса сортировки вставкой	Трасса групповой сортировки	Трасса гномьей сортировки	Граф Штейнера
Количество точек (вершин)	6	9	9	20
Количество отрезков (ребер)	5	8	8	19
Суммарная длина	19,8678	28,2843	28,2843	21,9351

Затем были рассчитаны неиндивидуальные характеристики, основанные на непосредственном сравнении трасс между собой и деревом Штейнера: средняя длина трасс, среднее количество точек и отрезков, количество общих точек и отрезков, длина общих отрезков.

Средняя длина трасс была получена как среднее арифметическое длин трасс:

$$l_{cp} = \frac{19,8678 + 28,2843 + 28,2843}{3} \approx 25,4788.$$

Аналогичным образом были рассчитаны средние количества точек и отрезков: 8 и 7 соответственно.

Для расчета 12-го показателя была найдена суммарная длина отрезков, общих для всех трех трасс:

$$l_j^S = \sqrt{50} + \sqrt{18} + \sqrt{2} \approx 12,7279.$$

Некоторые сравнительные показатели трасс, фигурирующие в дальнейших расчетах, представлены в табл. 3.

Таблица 3

Table 3

Некоторые сравнительные показатели трасс

Some comparative measures of traces

Характеристика	Сортировка вставкой	Групповая сортировка	Гномья сортировка
Средняя длина трасс	25,4788		
Среднее количество точек	8		
Среднее количество отрезков	7		
Количество точек, общих для всех трасс	4		
Количество отрезков, общих для всех трасс	3		
Суммарная длина отрезков, общих для всех трасс	12,7279		
Количество точек, общих с минимальным деревом Штейнера	6	9	9
Количество отрезков, общих с минимальным деревом Штейнера	0	0	0

На основании полученных данных для определения меры различия были установлены вышеперечисленные показатели трасс.

Стоит отметить, что для трасс трех данных алгоритмов не существует отрезков, общих с минимальным деревом Штейнера. Это объясняется относительно небольшим количеством точек в трассах, взятых для эксперимента. Отсюда же следует, что показатели 2 и 9, представляющие собой отношения, в которых в качестве числителей выступают длина и количество общих с минимальным деревом Штейнера отрезков, будут равны нулю для трасс небольшого размера.

Что касается общих точек у сравниваемой трассы и минимального дерева Штейнера, то их число всегда будет равно числу точек сравниваемой трассы по определению самой задачи построения минимального дерева Штейне-

ра. Показатель отношения их числа к числу точек в минимальном дереве Штейнера для трасс A , B и C будет соответственно равен:

$$S_A(E) = \frac{6}{20} = 0,3;$$

$$S_B(E) = \frac{9}{20} = 0,45;$$

$$S_C(E) = \frac{9}{20} = 0,45.$$

Характеристики трасс 10, 11 и 12 используют общие для всех трех трасс значения, поэтому также являются общими для них. Их значения соответственно равны:

$$S(V_{Avg}) = \frac{3}{7} \approx 0,4286;$$

$$S(E_{Avg}) = \frac{4-2}{8} = 0,25;$$

$$S(L_{Avg}) = \frac{12.7279}{25.4788} \approx 0,4995.$$

В результате измерений были получены значения вышеперечисленных характеристик для каждой из трасс. Результаты приведены в табл. 4.

Таблица 4

Table 4

Неиндивидуальные характеристики сравниваемых трасс

Non-individual characteristics of the compared traces

Характеристика	Сортировка вставкой	Групповая сортировка	Гномья сортировка
2. Отношение количества общих отрезков у сравниваемой трассы с минимальным деревом Штейнера к количеству ребер в минимальном дереве Штейнера	0	0	0
5. Отношение количества общих точек у сравниваемой трассы и минимальным деревом Штейнера к количеству узлов в минимальном дереве Штейнера	0,3	0,45	0,45
9. Отношение длины совпадающих отрезков у сравниваемой трассы и минимального дерева Штейнера к длине минимального дерева Штейнера	0	0	0

Окончание табл. 4

End of Tab. 4

Характеристика	Сортировка вставкой	Групповая сортировка	Гномья сортировка
10. Отношение количества ребер сравниваемой трассы, совпадающих с ребрами всех других сравниваемых трасс, к среднему количеству ребер во всех сравниваемых трассах	0,4286		
11. Отношение количества точек сравниваемой трассы, совпадающих с точками всех других сравниваемых трасс, к среднему количеству узлов во всех сравниваемых трассах	0,25		
12. Отношение длины совпадающих отрезков сравниваемой трассы и всех остальных сравниваемых трасс к средней длине всех сравниваемых трасс	0,4995		

Полученные результаты показывают, что алгоритмы гномьей и групповой сортировок равны по всем характеристикам. Очевидно, что мера диверсифицированности этих алгоритмов равна нулю, так как согласно формуле (2) это то же самое, что евклидово расстояние между двумя совпадающими точками. Мера диверсифицированности для трассы сортировки вставкой относительно каждой из двух других трасс была найдена как

$$D_A = \sqrt{(0,3 - 0,45)^2} = 0,15.$$

В получившемся подкоренном выражении осталось лишь одно слагаемое, так как диверсифицированность общих характеристик всегда будет равна нулю, а все остальные рассматриваемые характеристики, кроме пятой, оказались равны нулю из-за небольшого размера трасс.

Итоговые значения позволяют заключить, что алгоритмы гномьей и групповой сортировок являются клонами.

Проведенный эксперимент подтвердил, что описанная в статье метрика может использоваться для определения диверсифицированности алгоритмов, так как сделанный вывод о диверсифицированности трех алгоритмов, рассмотренных в эксперименте, соответствует их действительным различиям и результатам проведенного ранее исследования [7].

ЗАКЛЮЧЕНИЕ

Предложенная в данной работе метрика позволяет численно определять меру различия программных алгоритмов по трассам их выполнения, не вдаваясь в детали реализации. Описанная метрика позволяет находить клоны в «пачках алгоритмов», предназначенных для построения компонентов производственных

планов. Данное утверждение было подтверждено экспериментально. В ходе проведения эксперимента сравнивались трассы выполнения трех алгоритмов сортировки, которые использовались для определения очередности взятия в работу заказов с наименьшим временем выполнения по методу SJF. Результаты эксперимента согласуются с фактическим наличием различий в исследуемых алгоритмах, а также с результатами предыдущего исследования [7].

В перспективе описанная в настоящей работе метрика диверсифицированности алгоритмов может применяться не только для поиска клонов в «пачках алгоритмов» формирования компонентов производственных планов, но и для определения меры различия версий на уровне алгоритмов в мультиверсионных программных системах [14], а также она может использоваться как отдельный показатель качества программного обеспечения в дополнение к уже используемым [15, 16].

СПИСОК ЛИТЕРАТУРЫ

1. Kuznetsov A.S., Noskova E.E. Assessment of planning methods at job-order manufacturing facilities // IOP Conference Series: Materials Science and Engineering. – 2020. – Vol. 862. – P. 042024. – DOI: 10.1088/1757-899X/862/4/042024.
2. Цуканов М.А., Божкова О.А. Разработка и реализация алгоритма построения расписания сталеплавильного производства на основе адаптации фрактала Кантора // Вестник Воронежского государственного технического университета. – 2017. – № 6. – С. 32–37.
3. Стохастическая модель оптимизации затрат при планировании технологических процессов лесозаготовок / И.В. Бачериков, Ф.В. Свойкин, А.Р. Бирман, В.А. Соколова // Системы. Методы. Технологии. – 2017. – № 4 (36). – С. 182–186.
4. Обоснование эффективности планирования технологических процессов водопользования и оперативное управление водораспределением на базе использования метода Монте-Карло / В.И. Ольгаренко, И.Ф. Юрченко, И.В. Ольгаренко, Г.Г. Костюнин, М.С. Эфендиев, В.И. Ольгаренко // Научный журнал Российского НИИ проблем мелиорации. – 2018. – № 1 (29). – С. 49–65.
5. Бутаков А.С. Автоматизация планирования технологических процессов сборки радиоэлектронной аппаратуры: выпускная квалификационная работа бакалавра / Сибирский федеральный университет. – Красноярск, 2018. – URL: http://elibrary.sfu-kras.ru/bitstream/handle/2311/72191/vkr_butakov.pdf?sequence=1 (дата обращения: 14.12.2020).
6. Егоршин А.А., Атеняев Е.С., Носкова Е.Е. Автоматизация планирования технологических процессов сборки радиоэлектронной аппаратуры // Современные проблемы радиоэлектроники / науч. ред. А.И. Громыко. – Красноярск: СФУ, 2018. – С. 388–390.
7. 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.
8. Васильев Н. Метрические пространства // Квант. – 1990. – № 1. – С. 16–23.
9. Багов М.А., Кудяев В.Ч. Математическое моделирование и оптимизация трубопроводной сети Штейнера // Известия Кабардино-Балкарского научного центра РАН. – 2017. – № 1 (75). – С. 5–11.
10. Астриков Д.Ю., Кузьмин Д.А., Панасюк А.И. Моделирование системы планирования распределенного высокопроизводительного вычислительного комплекса // Доклады Академии наук высшей школы Российской Федерации. – 2014. – № 2–3 (23–24). – С. 34–41.
11. Григорьева М.В. Эколого-экономическая оценка методов ручной и оптической сортировки твердых бытовых отходов // Вестник Пермского национального исследовательского политехнического университета. Урбанистика. – 2012. – № 3. – С. 62–73.
12. Дупленко А.Г. Сравнительный анализ алгоритмов сортировки данных в массивах // Молодой ученый. – 2013. – № 8. – С. 50–53.
13. Fonseca R. ESMT-Smith: Smiths algorithm for the Euclidean Steiner Minimal Tree problem. – 2015. – 23 March. – URL: <https://github.com/RasmusFonseca/ESMT-Smith> (accessed: 20.11.2020).

14. Chen L., Avizienis A. N-version programming: A fault-tolerance approach to reliability of software operation // FTCS-8: the Eighth Annual International Conference on Fault-Tolerant Computing: digest of papers. – Toulouse, France, 1978. – Vol. 1. – P. 3–9.

15. Бураков Д.П., Кожомбердиева Г.И. Использование формулы Байеса при оценивании качества программного обеспечения согласно стандарту ISO/IEC 9126 // Программные продукты и системы. – 2019. – Т. 32, № 1. – С. 34–41.

16. Грузенкин Д.В., Михалев А.С. Определение метрики диверсифицированности мультиверсионного программного обеспечения на уровне языков программирования // Программная инженерия. – 2019. – Т. 10, № 9–10. – С. 384–390.

Грузенкин Денис Владимирович, старший преподаватель кафедры информатики Института космических и информационных технологий Сибирского федерального университета. Основное направление научных исследований – надежность программного обеспечения, мультиверсионное программирование, качество программного обеспечения. Имеет более 30 печатных работ (РИНЦ, ВАК, Scopus, WoS), является соавтором одной монографии. E-mail: dgruzenkin@sfu-kras.ru

Кузнецов Александр Сергеевич, кандидат технических наук, доцент, заведующий кафедрой информатики Института космических и информационных технологий Сибирского федерального университета. Основное направление научных исследований – оценка качества, стандартизация и сопровождение программных систем; проблемы производственного планирования. Имеет более 35 печатных работ (РИНЦ, ВАК, Scopus, WoS), монографий и учебных пособий. E-mail: askuznetsov@sfu-kras.ru

Селезнев Игорь Валерьевич, студент направления «Программная инженерия» в Институте космических и информационных технологий Сибирского федерального университета. Основное направление научных исследований – качество программного обеспечения. Научных работ и учебных пособий пока не имеет. E-mail: seleznevigor2001@gmail.com

Gruzenkin Denis V., senior teacher at the Institute of Space and Information Technology of Siberian Federal University, Department of Informatics. His research interests are currently focused on software reliability, N-version programming, and software quality. He has more than 30 publications and one monograph. E-mail: dgruzenkin@sfu-kras.ru

Kuznetsov Aleksandr S., associate professor, PhD (Eng.), Head of the Department of Informatics at the Institute of Space and Information Technology of Siberian Federal University. His research interests are currently focused on quality assessment, standardization and maintenance of software systems and production planning problems. He has more than 35 publications, monographs and teaching manuals. E-mail: askuznetsov@sfu-kras.ru

Seleznev Igor V., software engineering student at the Institute of Space and Information Technology of Siberian Federal University. His research interest is currently focused on software quality. He has not published any papers yet. E-mail: seleznevigor2001@gmail.com

DOI: 10.17212/1814-1196-2020-4-65-80

An assessment of the algorithm difference measure in a multivariate system for production plans scheduling*

D.V. GRUZENKIN^a, A.S. KUZNETSOV^b, I.V. SELEZNEV^c

Siberian Federal University, School of Space and Information Technology, Department of Informatics, 26 Kirensky Street, 660074 Krasnoyarsk, Russian Federation

^a dgruzenkin@sfu-kras.ru ^b askuznetsov@sfu-kras.ru ^c seleznevigor2001@gmail.com

Abstract

In the process of designing a production plan, one of the important steps is scheduling the execution of technological operations. The schedule can be created either manually or by using software. If the schedule is compiled by software, then several schedule generation algorithms

* Received 20 August 2020.

Acknowledgments: The reported study was funded by RFBR, project number 20-07-00226.

are used to eliminate possible errors. A set of such algorithms is called a "batch". It is advisable that only different algorithms should be included in the batch. This is necessary to eliminate errors of the same type. Therefore, the search for clones of algorithms in the batch is an urgent production task. To solve it a diversity metric of algorithms was developed in the course of this work. Such a metric numerically (as a percentage) determines how much the algorithms differ. This metric is based on the properties of the algorithm execution. Algorithm traces are constructed in the N-dimensional space using the obtained points. The coordinates of the trace points are the values with which the algorithm works at each step of its execution or each of the control points of the algorithm execution. An experiment was performed to confirm the correctness of this metric. Within this experiment, the trace properties of three sorting algorithms were calculated. Based on the properties obtained, indicators were determined for comparing algorithms in the metric space. The experiment confirmed the effectiveness of using the diversity metric to find clones in the algorithms batch. The scope of this metric is not limited to clone searches. It can be used as an independent indicator of software quality.

Keywords: algorithm, measure of difference, metric, diversification, diversity, diversity metric, production plan, scheduling, algorithms diversity

REFERENCES

1. Kuznetsov A.S., Noskova E.E. *Assessment of planning methods at job-order manufacturing facilities // IOP Conference Series: Materials Science and Engineering*, 2020, vol. 862, p. 042024. DOI: 10.1088/1757-899X/862/4/042024.
2. Tsukanov M.A., Bozhkova O.A. Razrabotka i realizatsiya algoritma postroeniya raspisaniya staleplavil'nogo proizvodstva na osnove adaptatsii fraktala Kantora [Development and implementation of the algorithm design compiling schedule of steelmaking based on adaptation of the fractal Cantor]. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta = The Bulletin of Voronezh State Technical University*, 2017, no. 6, pp. 32–37.
3. Bacherikov I.V., Svoikin F.V., Birman A.R., Sokolova V.A. Stokhasticheskaya model' optimizatsii zatrat pri planirovanii tekhnologicheskikh protsessov lesozagotovok [Stochastic model of cost optimization in the planning of logging processes]. *Sistemy. Metody. Tekhnologii = Systems. Methods. Technologies*, 2017, no. 4 (36), pp. 182–186.
4. Olgarenko V.I., Yurchenko I.F., Olgarenko I.V., Kostyunin G.G., Efendiev M.S., Olgarenko V.I. Obosnovanie effektivnosti planirovaniya tekhnologicheskikh protsessov vodopol'zovaniya i operativnoe upravlenie vodoraspredeleniem na baze ispol'zovaniya metoda Monte-Karlo [Planning effectiveness substantiation of technological processes of water use and operating control of water distribution using the Monte Carlo method]. *Nauchnyi zhurnal Rossiiskogo NII problem melioratsii = Scientific Journal of Russian Scientific Research Institute of Land Improvement Problems*, 2018, no. 1 (29), pp. 49–65.
5. Butakov A.S. *Avtomatizatsiya planirovaniya tekhnologicheskikh protsessov sborki radioelektronnoi apparatury* [Automation of planning of technological processes of assembly for radio-electronic equipment. Bachelor thesis]. Siberian Federal University. Krasnoyarsk, 2018. Available at: http://elib.sfu-kras.ru/bitstream/handle/2311/72191/vkr_butakov.pdf?sequence. (accessed 14.12.2020).
6. Egorshin A.A., Atenyaev E.S., Noskova E.E. Avtomatizatsiya planirovaniya tekhnologicheskikh protsessov sborki radioelektronnoi apparatury [Automation of planning of technological processes of assembly for radio-electronic equipment]. *Sovremennye problemy radioelektroniki* [Modern problems of radio electronics]. Krasnoyarsk, Siberian Federal University Publ., 2018, pp. 388–390.
7. Gruzenkin D.V., Yakimov I.A., Kuznetsov A.S., Tsarev R.Yu. Opredelenie metriki diversifitsirovannosti mul'tiversiionnogo programmnoho obespecheniya na urovne algoritmov [N-version software diversity metric definition on the algorithm level]. *Fundamental'nye issledovaniya = Fundamental research*, 2017, no. 6, pp. 36–40.
8. Vasil'ev N. Metricheskie prostranstva [Metric spaces]. *Kvant = Quantum*, 1990, no. 1, pp. 16–23.

9. Bagov M.A., Kudaev V.Ch. Matematicheskoe modelirovanie i optimizatsiya truboprovodnoi seti Shteinera [Mathematical simulation and optimization of Steiner pipeline network]. *Izvestiya Kabardino-Balkarskogo nauchnogo tsentra RAN = News of the Kabardin-Balkar scientific center of RAS*, 2017, no. 1 (75), pp. 5–11.
10. Astrikov D.Yu., Kuzmin D.A., Panasyuk A.I. Modelirovanie sistemy planirovaniya raspredelenogo vysokoproizvoditel'nogo vychislitel'nogo kompleksa [Simulation of a scheduling system of the distributed high-performance computing system]. *Doklady Akademii Nauk Vysshei Shkoly Rossiiskoi Federatsii = Proceedings of the Russian Higher School Academy of Sciences*, 2014, no. 2–3 (23–24), pp. 34–41.
11. Grigoryeva M.V. Ekologo-ekonomicheskaya otsenka metodov ruchnoi i opticheskoi sortirovki tverdykh bytovykh otkhodov [Environmental and economic evaluation of manual and optical methods of sorting of municipal solid waste]. *Vestnik Permskogo natsional'nogo issledovatel'skogo politekhnicheskogo universiteta. Urbanistika = Bulletin of Perm National Research Polytechnic University. Urban Development*, 2012, no. 3, pp. 62–73.
12. Duplenko A.G. Sravnitel'nyi analiz algoritmov sortirovki dannykh v massivakh [Comparative analysis of algorithms to sorting data in arrays]. *Molodoi uchenyi = Young Scientist*, 2013, no. 8, pp. 50–53.
13. Fonseca R. *ESMT-Smith: Smiths algorithm for the Euclidean Steiner Minimal Tree problem*. 2015, 23 March. Available at: <https://github.com/RasmusFonseca/ESMT-Smith> (accessed 20.11.2020).
14. Chen L., Avizienis A. N-version programming: A fault-tolerance approach to reliability of software operation. *FTCS-8: the Eighth Annual International Conference on Fault-Tolerant Computing: digest of papers*, Toulouse, France, 1978, vol. 1, pp. 3–9.
15. Burakov D.P., Kozhombardieva G.I. Ispol'zovanie formuly Baiesa pri otsenivanii kachestva programmogo obespecheniya soglasno standartu ISO/IEC 9126 [Using the Bayes' theorem within software quality evaluation according to ISO/IEC 9126 standard]. *Programmnye produkty i sistemy = Software and Systems*, 2019, vol. 32, no. 1, pp. 34–41.
16. Gruzenkin D.V., Mikhalev A.S. Opredelenie metriki diversifitsirovannosti mul'tiversion-nogo programmogo obespecheniya na urovne yazykov programmirovaniya [N-Version software diversity metric definition at the programming languages level]. *Programmnyaya inzheneriya = Software Engineering*, 2019, vol. 10, no. 9–10, pp. 384–390.

Для цитирования:

Грузенкин Д.В., Кузнецов А.С., Селезнев И.В. Оценка меры различия алгоритмов в многовариантной системе составления производственных планов // Научный вестник НГТУ. – 2020. – № 4 (80). – С. 65–80. – DOI: 10.17212/1814-1196-2020-4-65-80.

For citation:

Gruzenkin D.V., Kuznetsov A.S., Seleznev I.V. Otsenka mery razlichiya algoritmov v mnogovariantnoi sisteme sostavleniya proizvodstvennykh planov [An assessment of the algorithm difference measure in a multivariate system for production plans scheduling]. *Nauchnyi vestnik Novosibirskogo gosudarstvennogo tekhnicheskogo universiteta = Science bulletin of the Novosibirsk state technical university*, 2020, no. 4 (80), pp. 65–80. DOI: 10.17212/1814-1196-2020-4-65-80.