

## **Алгоритм уплотнения расчетной сетки тестирования программных модулей расчета и контроля полетного задания, построенной на основе приоритетов входных данных**

**Ляпин А.А.**

*Государственный ракетный центр имени академика В.П. Макеева,*

*Тургоякское шоссе, 1, Миасс, Челябинская обл., 456300, Россия*

*e-mail: [lyapin-sasha@mail.ru](mailto:lyapin-sasha@mail.ru)*

***Статья поступила 01. 07.2019***

### **Аннотация**

В статье предложен алгоритм улучшения метода построения сетки тестирования программных модулей из состава программного обеспечения расчета и контроля полетного задания, основанного на построении приоритетов входных данных. Разработанный алгоритм позволяет уплотнять сетку тестирования в областях вариаций входных данных приводящих к наибольшим отклонениям результата, что позволяет эффективнее использовать отведенные под тестирование временные ресурсы. Уплотнение сетки реализуется за счет построения для каждого параметра внутри области его изменения дискретного набора приоритетов (вместо одного значения приоритета). Таким образом, появляется возможность строить переменный шаг вариаций каждого параметра внутри области его изменения и сгущать сетку тестирования в областях, потенциально приводящих к ошибкам.

**Ключевые слова:** полетное задание, алгоритм, блок-схема, сетка тестирования, приоритеты данных.

### Постановка задачи

Программное обеспечение расчета и контроля полетного задания (далее ПО РКПЗ) состоит из множества взаимосвязанных элементов (программных модулей) [1]. Программный модуль (далее ПМ) – функционально завершённая программная реализация частной задачи (алгоритма) РКПЗ. Отработка и тестирование программных модулей может осуществляться при помощи использования специального программного средства (программный комплекс VTTesting [2-7]). Заложенная в VTTesting методология тестирования (автоматизированное исследовательское тестирование [8-15], автоматизация расчетов в аэрокосмической технике [16]) подразумевает формирование расчетной сетки на основе построения приоритетов входных данных тестируемого ПМ (VTTesting может применяться для оценки эффективности современных задач баллистики и движения КА [17-19]).

Методика построения приоритетов входных данных и алгоритм генерации расчетной сетки представлены в работе [3]. Указанный алгоритм позволяет увеличить количество шагов тестирования для входных параметров с высоким приоритетом (чем выше возмущения результирующих данных при изменении входного параметра, тем выше его приоритет). Однако, возмущения результирующих данных в зависимости от области изменения входного параметра распределены неравномерно. Знание указанной зависимости позволит строить расчетную сетку тестирования с неравномерным (дискретным) шагом изменения и,

таким образом, позволит увеличить количество узлов в областях с наибольшим отклонением результирующих данных (рис. 1).

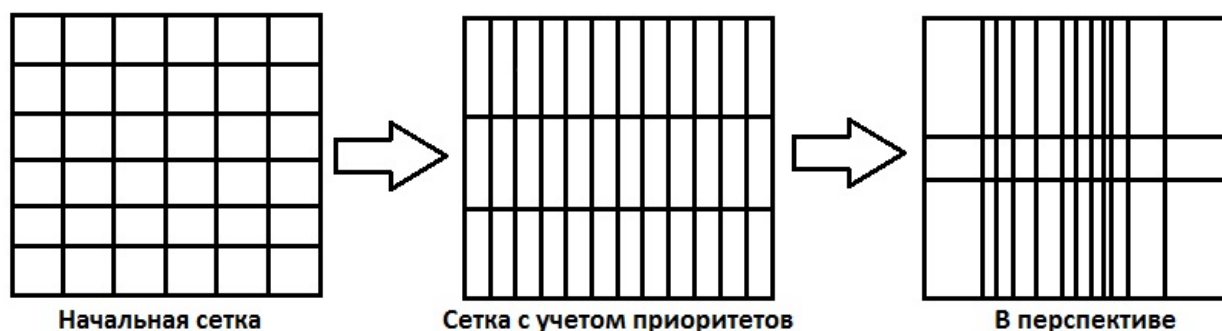


Рис. 1. Построение сетки тестирования

В данной работе представлен алгоритм построения указанной зависимости и дискретной расчетной сетки, основанный на предположении, что в выделенных областях (с наибольшими вариациями результата) – наибольшее количество сбоев (некорректных решений).

### Алгоритм построения дискретной расчетной сетки

Пусть ПМ имеет  $n$  входных параметров  $P\{p_1, p_2 \dots p_n\}$ , а также известно максимально возможное значение приоритета  $C_{max}$ . Для построения расчетной сетки по алгоритму [3] необходимо наличие начальной сетки тестирования и результатов расчета по ней.

Обозначим начальную расчетную сетку как  $\{p_i^1, p_i^2 \dots p_i^{K_i}, i = 1..n\}$ , где  $K_i$  – количество вариаций  $i$ -того параметра в расчетной сетке.

1) Для каждого входного параметра строим дискретный набор приоритетов.

В цикле по  $i = 1$  до  $n$ :

- разбиваем заданные вариации текущего параметра на отрезки:

$$\{p_i^1, p_i^2 \dots p_i^{K_i}\} \rightarrow (p_i^1, p_i^2), (p_i^2, p_i^3), \dots, (p_i^{K_i-1}, p_i^{K_i});$$

- используя алгоритм построения приоритетов, приведенный в [3], строим приоритет параметра на каждом отрезке  $\{C_i^1, C_i^2 \dots C_i^{K_i-1}\}$  с учетом  $C_{max}$ ;

- 2) Определим приоритет параметра как среднее арифметическое на

полученном отрезке:  $C_i = \frac{\sum_{j=1}^{K_i-1} C_i^j}{K_i-1}$ .

- 3) По алгоритму построения расчетной сетки [3] на основе  $C_i$  получим новое количество вариаций параметра (новая размерность сетки)  $K_i^{нов}$ .

- 4) Определим количество вариаций  $N_i^j$  каждого  $i$ -того параметра внутри каждого его  $j$ -того отрезка.

Определяем коэффициент значимости  $L_i$  каждого отрезка (весовой коэффициент):

$$L_i = \frac{K_i^{нов} + K_i - 1}{\sum_{j=1}^{K_i-1} C_i^j};$$

$$N_i^j = L_i \cdot C_i^j, \text{ где } j = 1..(K_i - 1).$$

Конец цикла по  $i$ .

- 5) Определим общее количество тестов как  $K_{реал} = \prod_{i=1}^n K_i^{нов}$

- 6) Формируем новую расчетную сетку, разбивая каждый отрезок каждого параметра на найденное количество вариаций  $N_i^j$ .

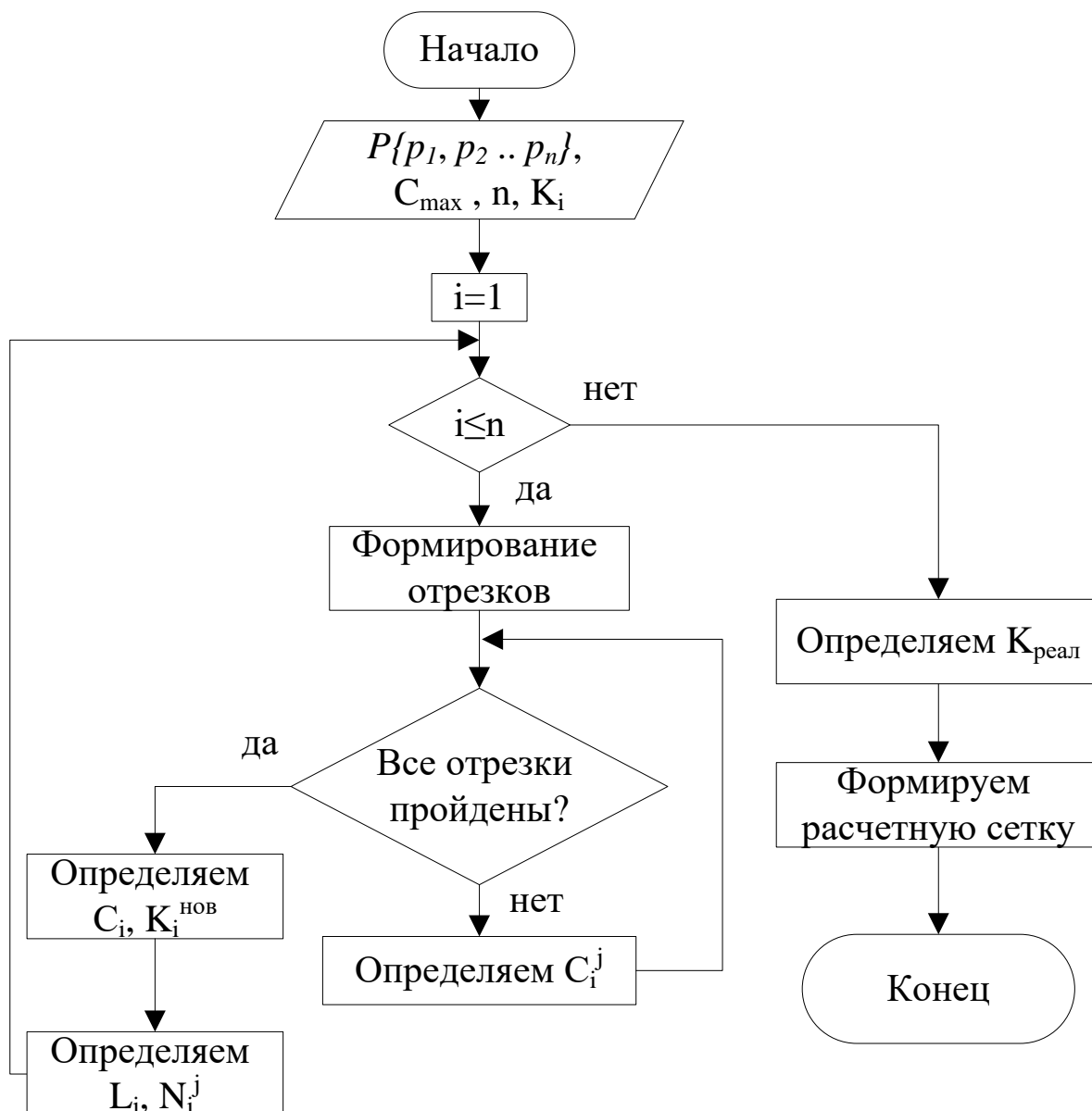


Рис. 2. Блок-схема алгоритма

## Результат

На основе приведенного алгоритма был модифицирован программный комплекс ВТTesting [20-25]. Ниже приведен пример построения сетки тестирования для программного модуля решения обратной баллистической задачи (при заданном кол-ве тестов 500) для сетки с учетом приоритетов (таблица 2) и полученной по результатам алгоритма (таблица 3). В качестве варьируемых параметров взяты широта  $\varphi$  и долгота  $\lambda$  начальной точки. Статистика решений приведена в таблице 4.

Таблица 1 – Значения варьируемых параметров

№	Равномерная сетка		Дискретная сетка	
	$\varphi$ (град.)	$\lambda$ (град)	$\varphi$ (град.)	$\lambda$ (град)
1	0	0	0	0
2	4.21	7.5	17.78	13.33
3	8.42	15	20.74	26.67
4	12.63	22.5	23.70	40
5	16.84	30	26.67	52
6	21.05	37.5	29.63	64
7	25.26	45	32.59	76
8	29.47	52.5	35.55	88
9	33.68	60	38.52	100
10	37.89	67.5	41.48	108
11	42.10	75	44.44	116
12	46.31	82.5	46.67	124
13	50.52	90	48.89	132
14	54.73	97.5	51.11	140
15	58.94	105	53.33	144.44
16	63.15	112.5	55.56	148.89
17	67.36	120	57.78	153.33
18	71.57	127.5	60	157.78
19	75.78	135	62.22	162.22
20	80	142.5	68.15	166.67
21		150	74.07	171.11
22		157.5	80	175.56
23		165		180
24		172.5		
25		180		



Таблица 3 – Дискретная сетка

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	2	1	2	2	1	1	2	2	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
21	1	1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
22	1	1	1	2	1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
23	1	1	1	2	1	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1

Таблица 4 – Статистика решений

	Равномерная сетка	Дискретная сетка
Проведено тестов	500	506
Некорректные завершения	14 (2.8 %)	44 (8.7%)

Анализ результатов показывает, что в областях вариаций входных данных, потенциально приводящих к некорректному решению, происходит уплотнение расчетной сетки. Построения расчетной сетки для других программных модулей (приведенных в [3]) подтверждает полученный результат.



## Заключение

Разработанный алгоритм позволяет уплотнять сетку тестирования в областях вариаций входных данных, приводящих к наибольшим отклонениям результата, что позволяет находить большее количество некорректных решений за ту же единицу времени и, таким образом, эффективнее использовать отведенные под тестирование временные ресурсы.

## Библиографический список

1. Ляпин А.А., Голунов М.С. Технология разработки программного обеспечения расчета и контроля полетного задания для межконтинентальных баллистических ракет // V Ежегодный форум «Информационные технологии на службе оборонно-промышленного комплекса – 2016» (Челябинск, 17-20 мая 2016): тезисы докладов. – Челябинск: Издательский дом Connect, 2016. С. 62.
2. Ляпин А.А. Методология тестирования ПО РКПЗ // Материалы XX Юбилейной Международной конференции по вычислительной механике и современным прикладным программным системам, ВМСППС'2017 (Алушта, 24-31 мая 2017) – М.: Изд-во МАИ, 2017. С. 154 - 156.
3. Ляпин А.А. Тестирование программных модулей расчета и контроля полетного задания на основе построения приоритетов исходных // Труды МАИ. 2017. № 97. URL: <http://trudymai.ru/published.php?ID=87359>

4. Тюгашев А.А., Ильин И.А., Ермаков И.Е. Пути повышения надежности и качества программного обеспечения в космической отрасли // Управление большими системами. 2012. № 39. С. 288 – 299.
5. Ляпин А.А. Алгоритм построения стохастических вариаций вектора с заданными ограничениями // Труды МАИ. 2018. № 102. URL: <http://trudymai.ru/published.php?ID=99221>
6. Ляпин А.А. Программа VТTesting: программа для ЭВМ № 2017664117. Правообладатель: Акционерное Общество «Государственный ракетный центр имени академика В.П. Макеева», 18.12.2017.
7. Ляпин А.А., Тюлькин Б.М. Интегрирование пассивного участка траектории на основе сочетания метода ks-регуляризации с методом Энке // Научные ведомости Белгородского государственного университета. Математика. Физика. 2018. Т. 50. № 1. С. 88 – 96.
8. James Bach. What is Exploratory Testing? [URL: http://www.satisfice.com/articles/what\\_is\\_et.shtml](http://www.satisfice.com/articles/what_is_et.shtml)
9. Савин Р. Тестирование Dot Ком, или Пособие по жестокому обращению с багами в интернет-стартапах. - М.: Дело, 2007. - 312 с.
10. Куликов С.С. Тестирование программного обеспечения. Базовый курс. - Минск: Четыре четверти, 2015. – 294 с.
11. Дастин Э., Рэшка Д., Пол Д. Автоматизированное тестирование программного обеспечения: внедрение, управление и эксплуатация / Пер. с англ. – М.: Изд-во Лори, 2003, - 567 с.

12. Бейзер Б. Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. – СПб.: Питер, 2004. – 318 с.
13. Котляров В.П., Коликова Т.В. Основы тестирования программного обеспечения – М.: Интернет-Университет Информационных Технологий. БИНОМ. Лаборатория знаний, 2006. – 285 с.
14. Блэк Р. Ключевые процессы тестирования. – М.: Изд-во Лори, 2006.– 544 с.
15. Николаев М.А., Юферов А.Г. Алгоритмы организации вариантных проектных расчетов // Научно-технический вестник Поволжья. 2013. № 6. С. 388 – 393.
16. Полоник Е.Н., Суренский Е.А., Федотов А.А. Автоматизация расчетов усталостной долговечности механических соединений элементов авиаконструкций // Труды МАИ. 2017. № 92. URL: <http://trudymai.ru/published.php?ID=76755>
17. Глущенко А.А., Хохлов В.П. Метод обнаружения маневра космического аппарата на основе текущих траекторных измерений // Труды МАИ. 2019. № 109. URL: <http://trudymai.ru/published.php?ID=111402>, DOI: [10.34759/trd-2019-109-17](https://doi.org/10.34759/trd-2019-109-17)
18. Сапожников В.Б., Корольков А.В., Константинов С.Б., Александров Л.Г. Экспериментальная оценка невырабатываемых остатков топлива разгонных блоков на заключительных этапах полета // Труды МАИ. 2020. № 110. URL: <http://trudymai.ru/published.php?ID=112838>. DOI: [10.34759/trd-2020-110-7](https://doi.org/10.34759/trd-2020-110-7)
19. Виноградов А.В., Борукаева А.О., Бердииков П.Г. Математическая модель движения баллистического летательного аппарата и алгоритмов расчета номинальных и возмущенных параметров движения баллистического летательного

аппарата // Труды МАИ. 2019. № 109. URL:

<http://trudymai.ru/published.php?ID=111430>. DOI: [10.34759/trd-2019-109-25](https://doi.org/10.34759/trd-2019-109-25)

20. Соммервилл И. Инженерия программного обеспечения. - М.: Издательский дом «Вильямс», 2002. - 624 с.
21. Бланшет Ж., Саммерфилд М. QT 4: программирование GUI на C++. Спб.: Изд-во Кудиц-Пресс, 2007. - 628 с.
22. Земсков Ю.В. Qt 4 на примерах. - СПб.: БХВ-Петербург, 2008. - 608 с.
23. Шлее М. Qt 4.5. Профессиональное программирование на C++. - СПб.: БХВ-Петербург, 2010. - 896 с.
24. Косова А.С. Особенности отработки программно-алгоритмического обеспечения системы управления движением и формирования консервативной информации одного типа летательных аппаратов // Труды ФГУП "НПЦАП" Системы и приборы управления. 2017. № 2. С. 27 - 36.
25. Макконнелл С. Совершенный код. Мастер класс. – М.: Издательско-торговый дом «Русская редакция». - СПб.: Питер, 2005. – 896 с.