

УДК 004.051

Васильев В. В., Никифоров К. А.

### **Автоматизированная система для оценки эффективности алгоритмов на основе эмпирического анализа**

**1. Введение.** Разработка алгоритма является ключевым шагом в решении задачи. Как только получен алгоритм, можно перевести его в компьютерную программу на каком-либо языке программирования. Однако на практике перед использованием алгоритма необходимо знать, сколько временных и пространственных ресурсов требуется для его выполнения. Данная информация позволяет определить применимость алгоритма в тех или иных условиях.

Один из новейших используемых подходов к оценке качества компьютерных алгоритмов по критерию трудоемкости основан на рассмотрении значений доверительной трудоемкости вместо трудоемкости в среднем случае [1]. Данный метод включает в себя два этапа. Первый этап — предварительное исследование, целью которого является проверка гипотезы о виде закона распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины [2]. Второй этап — основное исследование, в ходе которого значения доверительной трудоемкости  $f_\gamma(n)$  определяются в зависимости от входной длины алгоритма [1].

Однако использование указанного метода является трудозатратным процессом, требующим к тому же определенной квалификации от персонала. Построение системы для автоматизированного проведения анализа по описанной методике позволяет значительно сократить время оценки качества алгоритмов. В данной статье описывается опыт создания прототипа такой системы, которая является необходимым базисом для быстрого внедрения программных реализаций компьютерных алгоритмов.

---

*Васильев Василий Васильевич* — студент, Санкт-Петербургский государственный университет; e-mail: vasar007@yandex.ru, тел.: +7(996)407-29-11

*Никифоров Константин Аркадьевич* — доцент, Санкт-Петербургский государственный университет; e-mail: k.nikiforov@spbu.ru, тел.: +7(812)328-31-84

**2. Постановка задачи.** Необходимо создать прототип системы для проведения следующих операций над программными реализациями алгоритмов:

- проведение предварительного этапа исследований трудоемкости по методике [2];
- проведение основного этапа исследований трудоемкости по методике [1];
- обработка результатов анализа;
- оформление результатов анализа алгоритма в виде отчета, включающего в себя итоговые значения требуемых вычислительных ресурсов для входных данных определенной размерности.

Отметим, что предполагается наличие реализации компьютерного алгоритма, которая будет использоваться для проведения анализа. Также модуль с реализацией должен содержать компонент для генерации входных данных заданной системой размерности.

**3. Существующие решения.** Необходимо обратить внимание на существующие информационные системы, выполняющие схожие задачи.

В работе [4] автор использует подход для анализа программных реализаций алгоритмов, созданных с помощью системы Nuprl. Данный подход применим только при работе с указанной системой и не подходит для реализаций алгоритмов, разработанных с помощью других систем или языков программирования.

В другой работе [5] авторы демонстрируют похожий на предыдущий подход для анализа программ, написанных на специальном функциональном языке программирования. Стоит отметить, что в данном случае проведение анализа алгоритма также затруднительно, поскольку узкоспециализированные системы и языки программирования практически не используются при решении большинства задач на практике. К тому же такой подход удваивает объем необходимых работ для анализа и внедрения программной реализации алгоритма в существующую систему для решения поставленной разработчиком задачи.

Заметим, что в обоих работах авторы рассматривают трудоемкость в среднем случае, что, согласно исследованиям, является не оптимальным подходом для оценки качества алгоритмов [1].

**4. Архитектура системы.** Одной из главных задач является проектирование приложения с учетом поддержки наибольшего количества программных реализаций алгоритмов для проведения анализа.

Для выполнения этого требования программный модуль должен иметь определенный интерфейс, т.е. принимать аргументы из командной строки. Они требуются для проведения экспериментальных исследований трудоемкости в различных конфигурациях алгоритма.

В процессе разработки использованы следующие технологии: Windows Presentation Foundation (WPF) для реализации пользовательского интерфейса, EPPlus для формирования отчета в виде документа пакета Microsoft Excel [3]. Такой стек технологий позволяет вести разработку и поддержку специалисту, владеющему только одним языком программирования. Схема архитектуры прототипа представлена на рис. 1.



Рис. 1. Схема архитектуры системы

**5. Описание реализации.** На основе приведенной архитектуры разработано приложение на языке C# [6], позволяющее проводить анализ алгоритмов и получать результаты в виде отчета для изучения их применимости на заданных входных данных.

В первую очередь, нужно получить программную реализацию алгоритма. Язык программирования или технологии, используемые для получения реализации, не имеют значения. Единственное требование — возможность принимать входные параметры для алгоритма для проведения экспериментальных исследований. В случае отсутствия возможности принимать параметры система не сможет провести анализ для предоставленной программной реализации.

Система имеет predetermined набор программных реализаций алгоритмов, которые можно использовать для анализа. Для добавления новой реализации достаточно поместить разработанный модуль в каталог с установленной системой. При запуске система автоматически обнаружит новый модуль и добавит его в качестве одного из вариантов для проведения анализа.

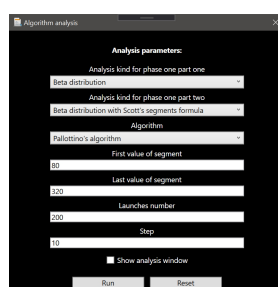


Рис. 2. Интерфейс настройки параметров анализа

Параметры для экспериментальных исследований формирует информационная система на основе пользовательского ввода и передает их программе в качестве аргументов командной строки при запуске. Пример набора параметров показан на рис. 2. Суть экспериментальных исследований заключается в многократном запуске алгоритма и вычислении количества затраченных им операций для определенного входа. Указанные значения сохраняются в временный файл с результатами запуска. Заметим, что подсчет основных операций осуществляется в модуле с алгоритмом, система не ведет их подсчет ввиду отсутствия прямого доступа к алгоритму.

Сохраненные данные загружаются в систему для проведения основного этапа исследования. По окончании анализа формируется отчет с подробными результатами (рис. 3).

Код	Число	Экспериментальные частоты	Теоретические частоты	Значения для $\chi^2$	Дополнительные параметры	Значения
0.000240606	0	0	0.0018983	0.0018983	Размер входа	80
0.000214004	3	0.003	0.007085023	0.002254061	min f(n)	80
0.000307532	13	0.013	0.020979473	0.003034966	max f(n)	212800
0.000300297	47	0.047	0.046176528	1.461070E-05	max f(n)	20224000
0.000317428	80	0.08	0.079871185	2.71344E-07	Кол-во экспериментов	1000
0.000407384	130	0.13	0.113387104	0.011973001	Коэффициент детерминации	0.95
0.000412381	146	0.146	0.134035011	0.000790138	Критерий значимости	0.05
0.000474723	134	0.134	0.140847837	0.000343484	Ер	0.001
0.000508161	105	0.105	0.139505632	0.004051283	Максимальные значения выборки	0.000340056
0.000541399	110	0.11	0.108000661	8.98321E-05	Максимальные значения выборки	0.00080002
0.000575917	77	0.077	0.080235066	0.000317637	Длина векторов	3.84862E-05
0.000608475	59	0.059	0.055336322	0.000216022	Число параметров для гистограммы	19
0.000641314	27	0.027	0.034527707	0.00009982	$\chi^2$ наблюдений	37.8084349
0.000676352	17	0.017	0.021451285	0.000010119	Кол-во случайных выборок	16
0.00070879	13	0.013	0.012118622	6.47003E-05	Критические значения $\chi^2$	26.2962376
0.000716238	10	0.01	0.006480701	0.000007017	Проверка гипотезы функции ТЕСТ	1
0.000775666	4	0.004	0.00330966	0.000143994		
0.000809104	1	0.001	0.001612977	0.000032949		
0.000842542	3	0.003	0.000754123	0.00068852		
0.000875981	1	0.001	0.000090165	0.000284607		

Рис. 3. Пример результатов анализа

**6. Заключение.** Реализована система для оценки качества алгоритмов с использованием доверительной трудоемкости. В дальнейшем планируется добавить новые реализации алгоритмов для анализа, а также расширить настраиваемый набор параметров для проведения более детального анализа. Исходный код и документация системы находятся в публичном репозитории в GitHub [6].

## Литература

1. Петрушин В. Н., Ульянов М. В., Кривенцов А. С. Доверительная трудоемкость — новая оценка качества алгоритмов // Информационные технологии и вычислительные системы. 2009. № 2. С. 23–37.
2. Петрушин В. Н., Ульянов М. В. Планирование экспериментального исследования трудоемкости алгоритмов на основе бета-распределения // Информационные технологии и вычислительные системы. 2008. № 2. С. 81–91.
3. Приложение для работы с электронными таблицами Microsoft Excel [Электронный ресурс]: URL:<https://products.office.com/ru-ru/excel> (дата обращения: 08.03.2020).
4. Benzinger R. Automated complexity analysis of Nuprl extracted programs // Journal of Functional Programming. 2001. Vol. 11. № 1. P. 3–31.
5. Hickey T., Cohen J. Automating program analysis // Journal of the ACM. 1988. Vol. 35. № 1. P. 185–220.
6. Репозиторий проекта в системе контроля версий GitHub [Электронный ресурс]: URL:[https://github.com/Vasar007/algorithm\\_analysis](https://github.com/Vasar007/algorithm_analysis) (дата обращения: 08.03.2020).