

УДК 004.415.23

Васильев В. В.

Проектирование и разработка информационной системы для оценки качества алгоритмов

Рекомендовано к публикации доцентом Никифоровым К. А.

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.

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

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

- 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.

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 разработчику задачи.

Заметим, что в обеих 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.

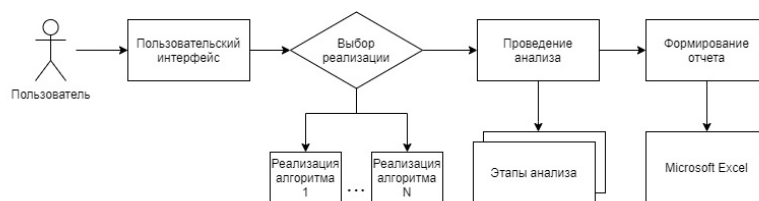


Рис. 1. Схема архитектуры системы

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 для анализа. Язык программирования или технологии, используемые для получения реализации, не имеют значения. Единственное требование для программы — 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.

Algorithm analysis

Analysis parameters:

Analysis kind for phase one part one

Beta distribution

Analysis kind for phase one part two

Beta distribution with Scott's segments formula

Algorithm

Pallottino's algorithm

First value of segment

80

Last value of segment

320

Launches number

200

Step

10

Show analysis window

Run

Reset

Рис. 2. Интерфейс настройки параметров анализа

Параметры для экспериментальных исследований формирует ин-формационная система на основе пользовательского ввода и переда-ет их программе в качестве аргументов командной строки при за-пуске. Пример набора параметров показан на рис. 2. Суть экспе-риментальных исследований заключается в многократном запуске алгоритма и вычислении количества затраченных им операций для определенного входа. Указанные значения сохраняются в временный файл с результатами запуска.

Сохраненные данные загружаются в систему для проведения ос-новного этапа исследования. По окончании анализа формируется от-чет с подробными результатами (рис. 3).

	D	E	F	G	H	I	J
1	Карман	Частота	Эмперические частоты	Теоретические частоты	Значение для χ^2	Дополнительные параметры	Значения
2	0.000240656	0	0	0.002018983	0.0018983	Размер входа	80
3	0.000274094	3	0.003	0.007083503	0.002354061	min f(n)	80
4	0.000307532	13	0.013	0.020979473	0.003034966	average f(n)	252800
5	0.00034097	47	0.047	0.046176928	1.46707E-05	max f(n)	20224000
6	0.000374408	80	0.08	0.079873185	2.01344E-07	Кол-во экспериментов	1000
7	0.000407846	150	0.15	0.113187104	0.011973001	Коэффициент доверия	0.95
8	0.000441285	146	0.146	0.135651011	0.000789538	Уровень значимости	0.05
9	0.000474723	134	0.134	0.140947837	0.000424884	Eps	0.001
10	0.000508161	105	0.105	0.129500432	0.004635283	Минимальное значение выборки	0.000240656
11	0.000541599	110	0.11	0.106900661	8.98582E-05	Максимальное значение выборки	0.00086902
12	0.000575037	77	0.077	0.080325006	0.000137637	Длина интервала	3.34382E-05
13	0.000608475	59	0.059	0.055536322	0.000216022	Число подвыборок для гистограммы	19
14	0.000641914	27	0.027	0.035652707	0.002099962	χ^2 наблюдаемое	37.8084349
15	0.000675352	17	0.017	0.021415285	0.000910319	Кол-во степеней свободы	16
16	0.00070879	13	0.013	0.012114622	6.47065E-05	Критическое значение χ^2	26.2962276
17	0.000742228	10	0.01	0.006480701	0.001897357	Проверка гипотезы функцией TEST	1
18	0.000775666	4	0.004	0.00330966	0.000143994		
19	0.000809104	1	0.001	0.001612977	0.000232948		
20	0.000842542	3	0.003	0.000754123	0.00668852		
21	0.000875981	1	0.001	0.000590165	0.000284607		

Рис. 3. Пример результатов анализа

6. Заключение. Реализована система для оценки качества алгоритмов с использованием доверительной трудоемкости. На данный момент для проведения анализа доступен лишь небольшой набор алгоритмов. 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 (дата обращения: 08.03.2020).