

УДК 004.051

Васильев В. В., Никифоров К. А.

Автоматизированная система для оценки эффективности алгоритмов на основе эмпирического анализа

1. Введение. Оценка эффективности алгоритмов является важным этапом в создании качественных программных средств, причем один из критериев качества — временная эффективность, особенно актуальная для систем, работающих в режиме реального времени. Очевидно, что временная эффективность компьютерной программы связана с функцией трудоемкости алгоритма, т. е. с точным количеством операций, задаваемых алгоритмом в основе программной реализации. Однако, асимптотические оценки вычислительной сложности, получаемые в теоретическом исследовании алгоритмов, не всегда справедливы для конечного диапазона длин входов, что объясняется большими значениями коэффициентов у компонент функции трудоемкости. В работе предлагается практический подход на основе эмпирического анализа времени выполнения программной реализации, для чего создана автоматизированная система с критерием оценки по величине доверительной трудоемкости в выбранном диапазоне входных данных.

Вычисление доверительной трудоемкости связано с построением доверительных интервалов оцениваемой величины трудоемкости с заданной доверительной вероятностью в классическом подходе математической статистики [1]. Данный метод требует использования репрезентативных выборок достаточно большого объема и многократного запуска программных реализаций исследуемых алгоритмов в соответствующем многоэтапном процессе, который будет описан ниже. Поэтому, несомненно, актуальной является разработка системы для автоматизированного проведения анализа, значительно сокращающего время оценки качества 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. В соответствии с вышеуказанным критерием оценки по доверительной трудоемкости необходимо создать автоматизированную систему для проведения следующих этапов исследования программной реализации алгоритмов:

- предварительный этап, целью которого является проверка гипотезы о виде закона распределения значений трудоемкости алгоритма как дискретной ограниченной случайной величины [2];
- основной этап, в ходе которого значения доверительной трудоемкости вычисляются в зависимости от длины входных данных алгоритма [1];
- этап обработки результатов анализа;
- оформление результатов анализа алгоритма в виде отчета, включающего итоговые значения требуемых вычислительных ресурсов для входных данных определенной размерности.

Необходимо отметить, что предполагается наличие реализации компьютерного алгоритма, которая используется для проведения анализа. Также модуль с реализацией должен содержать компонент для генерации входных данных.

3. Существующие решения. В работах [3, 4] системы автоматизированной оценки эффективности алгоритмов на основе анализа программных реализаций ограничивались использованием специальных языков программирования (Nuprl и функциональный язык), что не подходит для реализаций алгоритмов, разработанных с помощью других систем или языков программирования. Более поздние системы на основе машинного обучения [5] работают как с параметризованными алгоритмами, так и без параметров, причем обладают большой гибкостью при построении регрессионных моделей на основе эмпирического анализа программных реализаций на различных языках программирования. В последнее время в области вычислительного интеллекта развиваются альтернативы статистическим подходам в выборе наилучшего или более подходящего алгоритма для решения конкретной задачи [6].

Однако, в упомянутых системах даются лишь точечные оценки трудоемкости (мода, медиана, математическое ожидание, коэффициент вариации), что не позволяет получить какие-либо сведения о поведении алгоритма на конкретном входе. Преимущество предлагаемой автоматизированной системы заключается в использовании

доверительной трудоемкости как гарантирующей оценки, повышающей точность результатов эмпирического анализа.

4. Архитектура системы. Одной из главных задач является проектирование системы 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 [7]. Такой стек технологий позволяет вести разработку и поддержку специалисту, владеющему только одним языком программирования. Схема архитектуры прототипа представлена на рис. 1.

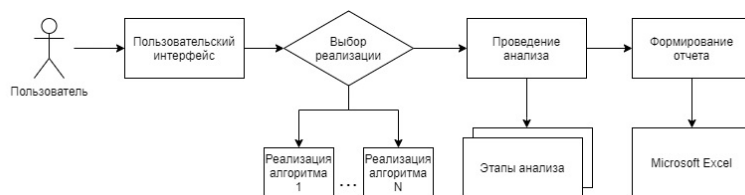


Рис. 1. Схема архитектуры системы

5. Описание реализации. На основе приведенной архитектуры разработано приложение на языке C# [8], 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 проведения экспериментальных исследований. В случае отсутствия возможности принимать параметры система не сможет провести анализ для предоставленной программной реализации.

Система имеет предопределенный 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 parameters

Algorithm
Pallottino's algorithm

The first value of function segment to analyze
80

The last value of function segment to analyze
320

The last extrapolation value of function segment to analyze
2560

The initial launches number
200

The size of step for function segment
10

Analysis parameters

Analysis kind for phase one part one
Normal distribution

Analysis kind for phase one part two
Beta distribution with Scott's segments formula

Analysis kind for phase two
Beta distribution

Goodness of fit
Coefficient of determination

Max degree of parallelism
1 12

Advanced parameters

☐ Show analysis window

☐ Open analysis results

Рис. 2. Интерфейс настройки параметров анализа

Параметры для экспериментальных исследований формирует система на основе пользовательского ввода и передает их программе в качестве аргументов командной строки при запуске. Пример набора параметров показан на рис. 2. Результаты многократного запуска исследуемой программной реализации сохраняются во временный файл, из которого потом загружаются в систему для проведения основного этапа исследования. По окончании анализа формируется отчет с подробным описанием полученных оценок эффективности (рис. 3).

| D | | E | | F | | G | | H | | I | | J | |
|--------|-------------|---------------------|-------|-----------------------|--|-----------------------|--|--------------------------------------|--|-------------|--|---|--|
| Размер | Частота | Затраченные частоты | | Теоретические частоты | | Значения для χ^2 | | Дополнительные параметры | | Значения | | | |
| 2 | 0.000240656 | 0 | | 0.0038983 | | 0.0018983 | | Размер ввода | | 80 | | | |
| 3 | 0.000274894 | 3 | 0.003 | 0.007083503 | | 0.002354061 | | min f(n) | | 80 | | | |
| 4 | 0.000307332 | 13 | 0.013 | 0.020979473 | | 0.003034966 | | интервал f(n) | | 253800 | | | |
| 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 | 0.113871304 | | 0.011977001 | | 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 | | 0.000342484 | | Eps | | 0.001 | | | |
| 10 | 0.000508161 | 105 | 0.105 | 0.129520432 | | 0.000463283 | | Максимальное значение выборки | | 0.002148056 | | | |
| 11 | 0.000541599 | 110 | 0.11 | 0.106900661 | | 8.9582E-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 полуинтервалов для гистограммы | | 19 | | | |
| 14 | 0.000641914 | 27 | 0.027 | 0.035627707 | | 0.000599662 | | χ^2 -наблюдение | | 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 | | 6.47905E-05 | | Критическое значение χ^2 | | 26.2962276 | | | |
| 17 | 0.000742228 | 10 | 0.01 | 0.006490701 | | 0.001897357 | | Проверка гипотезы функцией TEST | | 1 | | | |
| 18 | 0.000775666 | 4 | 0.004 | 0.00330966 | | 0.000143994 | | | | | | | |
| 19 | 0.000809104 | 1 | 0.001 | 0.001812977 | | 0.000232949 | | | | | | | |
| 20 | 0.000842542 | 3 | 0.003 | 0.000754123 | | 0.00646852 | | | | | | | |
| 21 | 0.000875981 | 1 | 0.001 | 0.000590165 | | 0.000284607 | | | | | | | |

Рис. 3. Пример результатов анализа

6. Заключение. Разработана автоматизированная система для оценки качества алгоритмов по доверительной трудоемкости, вычис-

ленной в процессе эмпирического анализа программной реализации. Преимущество системы состоит в получении гарантирующей оценки на основе статистических заключений с заданным уровнем значимости. Исходный код и документация системы находятся в публичном репозитории [8].

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. Benzinger R. Automated complexity analysis of Nuprl extracted programs // Journal of Functional Programming. 2001. Vol. 11. No 1. P. 3–31.
4. Hickey T., Cohen J. Automating program analysis // Journal of the ACM. 1988. Vol. 35. No 1. P. 185–220.
5. Hutter F., Xu L. Hoos H., Leyton–Brown K. Algorithm runtime prediction: Methods & evaluation // Artificial Intelligence. 2014. Vol. 206. No 1. P. 79–111.
6. Oprea M. A general framework and guidelines for benchmarking computational intelligence algorithms applied to forecasting problems derived from an application domain-oriented survey // Applied Soft Computing. 2020. Vol. 89. No 4. P. 106–103.
7. Приложение для работы с электронными таблицами Microsoft Excel [Электронный ресурс]: URL:<https://products.office.com/ru-ru/excel> (дата обращения: 08.03.2020).
8. Репозиторий проекта в системе контроля версий GitHub [Электронный ресурс]: URL:https://github.com/Vasar007/algorithm_analysis (дата обращения: 08.03.2020).