



УДК 517.63

ГЕОМЕТРИЧЕСКИЕ СВОЙСТВА АЛГОРИТМОВ

Е. А. Мирошников*, В. В. Яновский**

*Харьковский национальный университет им. Каразина,
пл. Свободы, 4, 61022, Харьков, Украина, e-mail: Skull@jandex.ua;

**Институт монокристаллов НАНУ,
пр. Ленина, 60, Харьков, 61001, e-mail: yanovsky@isc.kharkov.ua

Аннотация. В работе рассмотрены характеристики алгоритмов такие, как зависимость времени выполнения алгоритма от начальных данных, где под временем выполнения понимается количество шагов машины Тьюринга. Выбор теоретической модели алгоритмов, основанный на машине Тьюринга, обусловлен универсальностью и близостью ее к реальным вычислительным устройствам. Показано, что возможно существование алгоритмом с фрактальной геометрией устройства уровня постоянного значения времени вычислений над входными данными. Обсуждаются глобальные свойства алгоритмов.

Ключевые слова: машина Тьюринга, алгоритм, эффективность алгоритма, фрактальная размерность, мультифрактал.

1. Введение. Теория сложности вычислений – бурно развивающаяся область теоретической информатики и охватывает как чисто теоретические вопросы, так и вопросы, непосредственно связанные с практикой. Среди наиболее важных приложений этой теории можно выделить способы построения эффективных алгоритмов, а также развитие методов анализа их эффективности. Существуют различные подходы к пониманию эффективности вычислений (см. например [1], [2]). С общей точки зрения эффективность алгоритмов оценивается с помощью глобальных средств. Так, например, принято считать эффективные алгоритмы относятся к классу P [3], с полиномиальным ростом времени вычислений при увеличении «объема» входных данных. Аналогично эффективность классов $PSPACE$ определяется полиномиальным ростом требуемых ресурсов [3]. Такие подходы оставляют открытыми различия в выполнении алгоритмов при работе с одинаковыми по объему, но всевозможными входными данными. В определенном смысле, возникающие различия при этом можно рассматривать как полезные характеристики, позволяющие устанавливать более тонкие различия между алгоритмами одного класса. В этой статье предложен именно такой подход к анализу алгоритмов. В качестве универсальной модели алгоритма используется машина Тьюринга. Учитывая тезис Черча, который гласит, что всякий словарный, дискретный детерминированный алгоритм может быть реализован машиной Тьюринга [4], что позволяет перенести результаты на все типы алгоритмов. Основная идея состоит в изучении выполнения алгоритма на всевозможных входных данных. Другими словами, множество входных данных отображается в скалярную характеристику, связанную с выполнением алгоритма на входном значении. В качестве такой характеристики можно использовать такие общепринятые как время вычисления или потребляемая память. Разумеется, можно использовать и другие характеристики, например, даже результат вычисления. Таким образом, основное внимание сосредоточено на изменении этих характеристик при изменении входных данных без изменения их «объема». Для классификации алгоритмов предлагается использовать геометрический подход, заключающийся в

изучении геометрии множеств уровней постоянного значения выбранной характеристики. Основной вопрос, который обсуждается в работе, это какие типы алгоритмов с этой точки зрения могут встречаться и какими свойствами они могут отличаться. Обратим внимание, что в определенном смысле, такой подход позволяет ввести более тонкую классификацию алгоритмов находясь даже в одном классе сложности.

2. Построение и анализ алгоритмов. Для анализа свойств используем общий подход к описанию алгоритма в терминах машины Тьюринга [5]. На языке машины Тьюринга сформулируем ряд простых алгоритмов свойства которых и будем анализировать далее. В качестве таких тестовых алгоритмов выберем алгоритм сложения двух целых неотрицательных чисел в двоичном представлении, алгоритм поразрядного «И» над неотрицательными дробными числами в двоичном представлении и некоторые другие. Также будем использовать машины Тьюринга, программы которых генерируются случайным образом. Это позволит убедиться в типичности анализируемых свойств алгоритмов. Основное свойство используемых алгоритмов состоит в том, что исходные данные для таких алгоритмов взаимнооднозначно сопоставляются точкам единичного квадрата. Другими словами, на вход таких алгоритмов подаются два числа принадлежащие единичному отрезку.

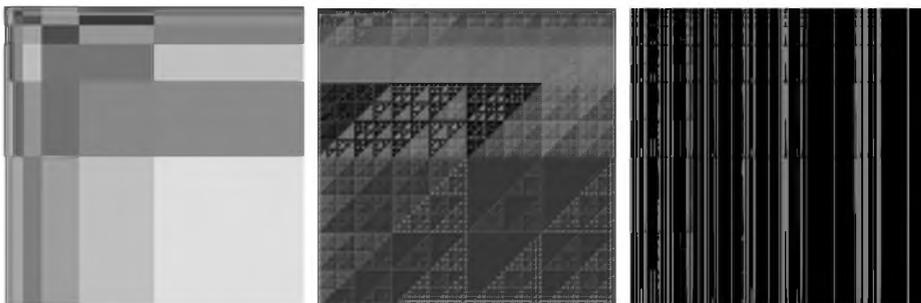


Рис. 1. Примеры трех типов полей. Левый соответствует алгоритму логического «И» в двоичной системе исчисления, центральный сложение двух целых неотрицательных чисел и правый один из алгоритмов, полученных при случайной генерации множества программ машины Тьюринга. Градация серого соответствует времени вычислений соответствующего результата.

Таким образом единичный квадрат разделим на клеточки, соответствующие начальным точкам, определяемыми с некоторой заданной точностью. Все точки принадлежащие одной клеточке имеют одинаковые координаты с точностью до размера клеточки. Чем выше точность, тем больше число клеточек. Естественно, выбор точности удобно ограничить степенями двойки. Так, на вход целочисленных алгоритмов подавались входные данные из интервалов $[2^n, 2^{n+k}]$, где $n, k \in \mathbb{N}$ с единичным шагом. А на вход алгоритмов с дробными входными данными (меньше единицы), точки единичного квадрата с шагом 2^{-n} , где $n = 1, 2, 3, \dots$, т.е. каждая точка принадлежит соответствующей области единичного квадрата. Выбор такого шага обусловлен тем, что при таком разбиении наиболее удобен переход в двоичную систему исчисления.

Осуществляя вычисление с использованием выбранного алгоритма над каждой входной клеточкой единичного квадрата \vec{x} , получим поле значений соответствующей характеристики $I(\vec{x})$. Для начала выберем в качестве такой характеристики время вычисления над входными данными $T(\vec{x})$. Это поле легко визуализировать раскрашивая клеточку в зависимости от времени вычислений над соответствующими исходными данными. При анализе большого числа

алгоритмов обнаруживается существование полей времени вычислений, которые принципиально отличаются геометрическими свойствами. Грубо такие множества можно разделить на 3 качественно различных типа. Типичные примеры этих типов показаны на рис. 1.

Так из приведенных примеров можно заметить, что распределение входных данных, требующих одинаковых временных затрат является простым для а) и сложным для б), в) алгоритмов. Поля а), б) симметричны относительно диагонали, что обусловлено симметрией алгоритма на входных данных. Как, например, симметрия алгоритма сложения чисел. Для более детального исследования удобно зафиксировать время вычисления и выделить в единичном квадрате входные данные, требующие одинаковых временных затрат. Ясно, что геометрическое устройство множеств фиксированного уровня должно быть подобными независимо от выбранного значения времени вычисления. Далее покажем множества заданного уровня для приведенных выше полей на рис. 1. Данные, требующие одинаковых временных затрат, визуализированы черным цветом, а все остальные белым.

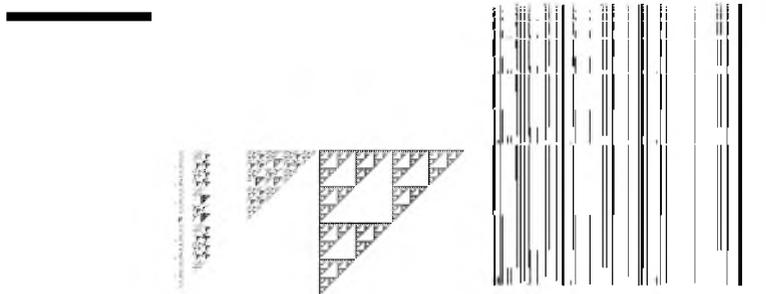


Рис. 2. Множества фиксированного уровня некоторых алгоритмов имеют простую структуру, другие – самоподобную структуру как и более сложную мультифрактальную. Здесь используются те же алгоритмы что и для построения рис. 1.

В результате упрощения получена сложная самоподобная структура для б) и в), а для а) простая структура. Геометрическая самоподобная структура б) похожа на известный фрактал - прямоугольную салфетку Серпинского [6], структура в) также похожа на фрактал. Для проверки, является ли данные множества фракталами, были построены поля выводов с большим разрешением. Используя специальную подпрограмму, методом клеточного покрытия [6] было проведено определение зависимости числа покрытий от выбора масштаба покрытия. Такая зависимость в логарифмических координатах хорошо аппроксимируется линейной зависимостью (см. например рис. 3). Тангенс угла наклона этой прямой и совпадает с фрактальной размерностью построенного множества. В результате вычислений было установлено, что структура б) является фракталом с фрактальной размерностью $D_F \approx 1.55 \pm 0.03$, что в пределах погрешности соответствует размерности фрактала «салфетка Серпинского». Значение этой размерности сохраняется и для множеств исходных данных, вычисление над которыми осуществляется за другое фиксированное время. Другими словами множества уровней фиксированных времен выполнения алгоритма являются фрактальными множествами определенной фрактальной размерности.

Подобный анализ структуры, приведенной на рис. 1 в), указывает на более сложную структуру множества, относящегося к мультифракталам. Дальнейший анализ машин Тьюринга с



генерируемыми случайным образом программами, позволяет установить типичность такого деления алгоритмов на три класса (простые, фрактальные и мультифрактальные). Для алгоритмов с обычной геометрией множеств фиксированного уровня, время выполнения зависит только от разрядности входных данных: чем больше разрядность – тем больше время выполнения. Для алгоритмов с фрактальной и мультифрактальной геометрией время выполнения зависит как от разрядности входных данных, так и от их значения. Другими словами, время выполнения этих алгоритмов чувствительно к изменению входных данных.

Принадлежность алгоритма к первой группе означает, что время затрачиваемое на вычисления над близкими начальными условиями фиксированной разрядности одинаково или очень близко. Данные алгоритмы будем считать простыми.

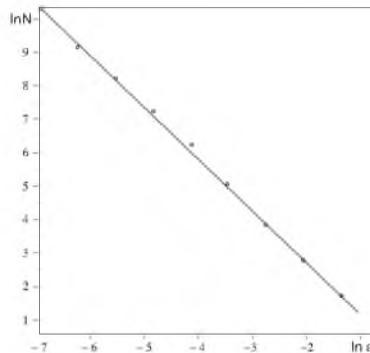


Рис. 3. К определению фрактальной размерности. Вертикальная ось – количество клеточек, содержащих точки соответствующего множества в логарифмическом масштабе, горизонтальная – размер клеток также в логарифмическом масштабе. Угол наклона прямой определяет клеточную или фрактальную размерность.

Принадлежность ко второму и третьему классу означает, ввиду фрактальной и мультифрактальной структуры множества исходных данных на которых алгоритм выполняется за фиксированное время, что время вычислений над сколь угодно близкими входными данными может быть сколь угодно близким или наоборот далеким. Другими словами, в любой окрестности исходных данных находятся данные со всеми значениями времен выполнения алгоритма над ними. Данные алгоритмы будем считать сложными.

3. Выводы

Таким образом, алгоритмы можно классифицировать по типам геометрии исходных данных, выполнение алгоритма над которыми происходит за определенное фиксированное время. Удобной характеристикой таких множеств, а следовательно и алгоритмов, можно рассматривать фрактальную размерность, а в более сложных и весь спектр размерностей Реньи [6] или другие характеристики мультифрактальных множеств. В качестве таковых могут выступать спектр сингулярностей или показатель масс [6]. Следует заметить, что алгоритмы относящиеся к фрактальным и мультифрактальным типам, допускают новый способ уменьшения времени выполнения алгоритмов путем малого «шевеления» входных данных, что означает новый способ оптимизации ресурсов при выполнении таких алгоритмов. Кроме этого можно предложить естественную гипотезу, что значения на которых машина Тьюринга не останавливается или



не приводит к результату вычислений, распределены фрактальным или мультифрактальным образом для таких алгоритмов.

Литература

1. А.А.Разборов, О сложности вычислений // Математическое просвещение. – 1999. – МЦНМО, No.3, с.127-141.
2. Яновский В.В. Квантовая механика алгоритмов / Харьков: ИСМА, 2009.
3. Верешагин Н.К., Шень А. Лекции по математической логике и теории алгоритмов. Вычислимые функции / М.: МЦНМО, 1999. (2-е изд.: М.: Наука, 1982).
4. А.Черч Введение в математическую логику / М.: ИЛ, 1960.
5. Колмогоров А.Н., Драгалин А.Г. Математическая логика:дополнительные главы / М.: изд-во МГУ, 1984.
6. Федер Е. Фракталы / М.: Мир, 1991.

GEOMETRIC PROPERTIES OF ALGORITHMS

Е.А. Miroshnikov*, **V.V. Yanovsky****

*V.N. Karazin Kharkiv National University,
Svobody Sq., 4, Kharkov, 61022, Ukraine, e-mail: xxxx@ccas.ru;

**Institute for Single Crystals of NASU,
Lenina Av., 60 Kharkov, 31001, Ukraine, e-mail: yanovsky@isc.kharkov.ua

Abstract. Algorithms characteristics like the dependence of algorithm execution time on initial data is under consideration when it refers to the number of Turing's machine steps. Algorithms theoretical models based on Turing's machine. It is done due to its flexibility and affinity to real computing devices. It is shown that there exists an algorithm having fractal geometry of its device connected with the fixed value of computation time over input data. Some global properties of algorithms are discussed.

Key words: Turing's machine, algorithm efficiency, fractal dimension, multifractal.