МОДИФИЦИРОВАННЫЙ АЛГОРИТМ ЧЕНА ПОСТРОЕНИЯ ВЫПУКЛОЙ ОБОЛОЧКИ НА ПЛОСКОСТИ

Н.С. Мирза, А.В. Скворцов, Р.В. Чаднов

Томский государственный университет

Задача построения выпуклой оболочки является фундаментальной задачей вычислительной геометрии. Важность этой задачи происходит не только из-за огромного количества приложений (в распознавании образов, обработке изображений, базах данных, в задаче раскроя и компоновки материалов, математической статистике), но также и из-за полезности выпуклой оболочки как инструмента решения множества задач вычислительной геометрии.

Эта задача позволяет разрешить целый ряд других, иногда с первого взгляда не связанных с ней вопросов: построение диаграмм Вороного, построение триангуляций и т.д. Построение выпуклой оболочки конечного множества точек на плоскости довольно широко исследовано и имеет множество приложений. Очень широко алгоритмы построения выпуклой оболочки используются в геоинформатике и геоинформационных системах.

Задача построения выпуклой оболочки имеет давнюю историю. Она является одной из первых задач вычислительной геометрии, с которой начала зарождаться эта наука. В настоящее время известно достаточно большое число алгоритмов построения выпуклой оболочки. Основная масса этих алгоритмов может быть разделена на два типа:

1. Алгоритм Грэхема, его модификации, алгоритм «разделяй и властвуй», быстрый алгоритм построения выпуклой оболочки. Эти алгоритмы имеют трудоемкость $O(n \log n)$.
2. Обход Джарвиса. Если за h обозначить количество вершин выпуклой оболочки, то этот алгоритм имеет трудоемкость $O(nh)$. Такие алгоритмы, трудоемкость которых зависит и от n (размера входных данных), и от h (размера выходных данных), называются *чувствительными к результату*.

Стоит заметить две вещи. Во-первых, алгоритмы первого типа оптимальны в среднем. Однако, если предположить, что h, – количество вершин выпуклой оболочки мало, то трудоемкость алгоритма Джарвиса меньше, чем у алгоритмов первого типа. К примеру, если h известно заранее или постоянно, то алгоритм Джарвиса имеет линейную трудоемкость, тогда как алгоритмы из первой группы всё равно имеют логарифмическую трудоемкость. С другой стороны, если h гораздо больше, чем $\log n$, то алгоритм Джарвиса проигрывает. К примеру, если $h=n$, то этот алгоритм показывает квадратичную трудоемкость.

Эти рассуждения наводят на следующую проблему: существует ли алгоритм, по крайней мере такой же быстрый, как оба вышеприведенных класса алгоритмов на любых входных данных.

Однако существуют даже более быстрые алгоритмы. Это было показано в 1986 году в [1] Киркпатриком и Зейделем. Они описали весьма алгоритм с трудоемкостью $O(n \log h)$. Однако этот алгоритм имел чрезвычайно сложную логику. Позже, в 1994 году, в [2, 3, 4] Тимоти Чен предложил *чрезвычайно* *простой* алгоритм с трудоемкостью O(n log h). Алгоритм Чена является оригинальным обобщение алгоритмов Джарвиса и Грэхема.

Проанализировав алгоритм Чена, который имеет оптимальную трудоемкость O(n log h), авторы пришли к выводу, что этот алгоритм можно модифицировать, увеличив его быстродействие.

Несмотря на приемлемую теоретическую трудоемкость, алгоритм Чена делает слишком много излишних действий, которые, несомненно, уменьшают его практическое быстродействие. В данной работе описывается модифицированный алгоритм Чена, имеющий трудоемкость $O(n^{1/3} \log h)$ для точек, равномерно распределенных внутри единичного круга.

Авторами был проведен количественный анализ предложенного алгоритма путём моделирования работы различных алгоритмов построения выпуклых оболочек на различных распределениях точек. При этом анализировалась скорость работы алгоритмов.

Результаты проведенных экспериментов показали, что различные алгоритмы ведут себя на различных классах входных данных. Предложенный алгоритм показал лучшую скорость работы на всех распределениях. Однако, несмотря на использование множества улучшений, модифицированный алгоритм Чена остался чувствительным к результату, что показали итоги его тестирования.

Таким образом, можно сказать, что модифицированный алгоритм Чена является наиболее быстрым практически применимым алгоритмом, чувствительным к результату.

Литература

1. Kirkpatrick D.G., Seidel R. The ultimate planar convex hull algorithm? // SIAM Journal on Computing. 1986. Vol. 15. P. 287–299.
2. Chan T.M. Output-Sensitive Construction of Convex Hulls: Ph.D. thesis / Department of Computer Science, University of British Columbia. 1995. 104 p.
3. Chan T.M. Optimal output-sensitive convex hull algorithms in two and three dimensions // Discrete & Computational geometry. 1995.
4. Chan T.M. Output-sensitive results on convex hulls, extreme points and related problems // Proceedings of the 11th Annual ACM symposium on Computational Geometry. 1995. P. 10–19.
5. Dobkin D.P., Kirkpatrick D.G. Fast detection of polyhedral intersection // Theoretical computer science. 1983. Vol. 27. P. 241–253.
6. O’Rourke J. Computational geometry in C. Cambridge University Press. 1994. 376 p.