**Alice Kjar**

# An experimental analysis into the efficiency of Jarvis' March and Graham's Scan algorithms

## Introduction

A convex hull is an important geometric concept which has a wide range of applications in a variety of fields. A convex hull (of a particular set of points *S*) is defined as the subset of *S* which forms a convex polygon, containing all other points in *S* (Fitzpatrick, 2024). This structure has many real-world applications such as pattern recognition, image processing and collision detection (Fitzpatrick, 2024). Hence, it is important that we have an efficient algorithm for generating a convex hull to reduce processing time and computational power required when processing large amounts of data.

Two popular algorithms, which we will be analysing in this report, are Jarvis' March and Graham's Scan. These two algorithms have varying implementations and complexities and therefore may be compared in their suitability.

Jarvis' March starts from the leftmost point in the set, then iterates through each of the other points to find the one with the greatest counterclockwise angle and adds this to the hull, repeating until the hull is complete. This algorithm has an overall time complexity of $O(nh)$ where $n$ is the size of the initial set of points and $h$ is the size of the final convex hull (Fitzpatrick, 2024). This will result in a worst-case complexity of $O(n^2)$ when the convex hull contains all points in *S*.

Graham's Scan approaches this problem by starting with the lowest point in *S*, then sorting the remaining points with respect to their polar angle. This sorted list is then iterated through, either adding the points to the hull or removing the last point to ensure that the points form a counterclockwise angle. The time complexity of this algorithm varies, depending on the sorting algorithm used, but assuming that the sorting algorithm is efficient, we get a complexity of $O(n \log n)$ where $n$ is the size of *S* (Fitzpatrick, 2024).

The aim of this investigation is to compare the empirical performance of these two algorithms for a variety of inputs and see how this varies from the theoretical 'worst-case' values, giving us insight into their efficiencies in a real-world context.

## Methodology

- Algorithm Implementation
    - Programs for both Jarvis' March and Graham's Scan were implemented in C
    - Implementation of both algorithms utilised two unsorted arrays to store the x and y coordinates of the points in *S*.
    - A linked list was utilised in Jarvis' March to store the convex hull
    - The sorting algorithm used in Graham's Scan was mergesort
    - A stack and a linked list were utilised in Graham's Scan to store convex hull
    - Designed under the assumption that all points in *S* were unique
- Dataset Generation
    - Datasets of size 10, 100, 300 and 1000 were generated
    - Unique randomiser seed was used for each trial to ensure independence

- For each size there were three distributions
  - Circle: created by randomly selecting a series of angles and finding their positions on a unit circle
  - Points within a simple hull: created by generating a rectangle of random size and randomly selecting points within it.
  - Random points: created by selecting a random point and repeatedly selecting a random vector from previous point to add to *S*.

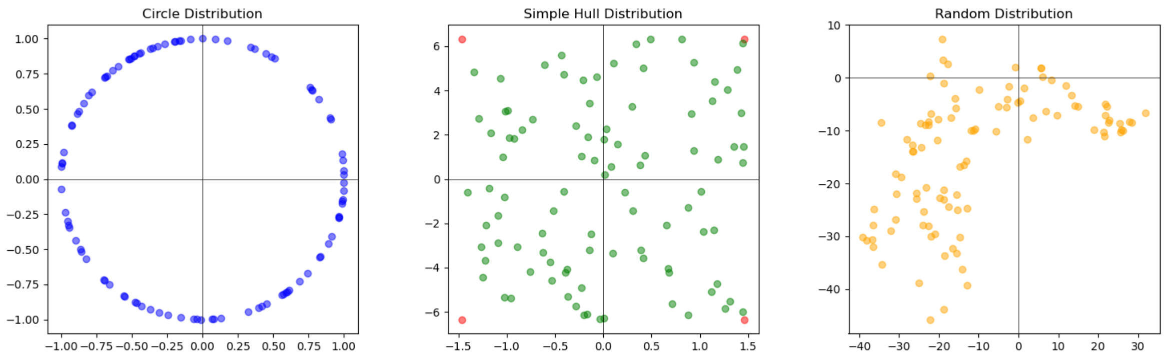


***Figure 1:*** *Plots of different distributions for sample size n = 100*

- Basic Operation Count
  - Count of occurrences of basic operation was recorded for each dataset.
    - Jarvis' March: basic operation was calling of function 'orientation()'
    - Graham's Scan: basic operation was comparison of angles in sorting process

## Results

| **Jarvis March** | **n = 10** | **n = 100** | **n = 300** | **n = 1000** |
|---|---|---|---|---|
| **Circle** | 89 | 9,899 | 89,699 | 998,999 |
| **Simple Hull** | 35 | 395 | 1,195 | 3,995 |
| **Random** | 53 | 792 | 2,990 | 15,984 |
| **Theoretical 'worst-case'** ($O(n^2)$) | 100 | 10,000 | 90,000 | 1,000,000 |

| **Grahams Scan** | **n = 10** | **n = 100** | **n = 300** | **n = 1000** |
|---|---|---|---|---|
| **Circle** | 21 | 546 | 2,099 | 8,692 |
| **Simple Hull** | 24 | 542 | 2,091 | 8,708 |
| **Random** | 19 | 441 | 1,740 | 7,179 |
| **Theoretical 'worst-case'** ($O(n \log n)$) | ~34 | ~665 | ~2,469 | ~9,966 |

***Figure 2:*** *Tabular results of operation count for each dataset*

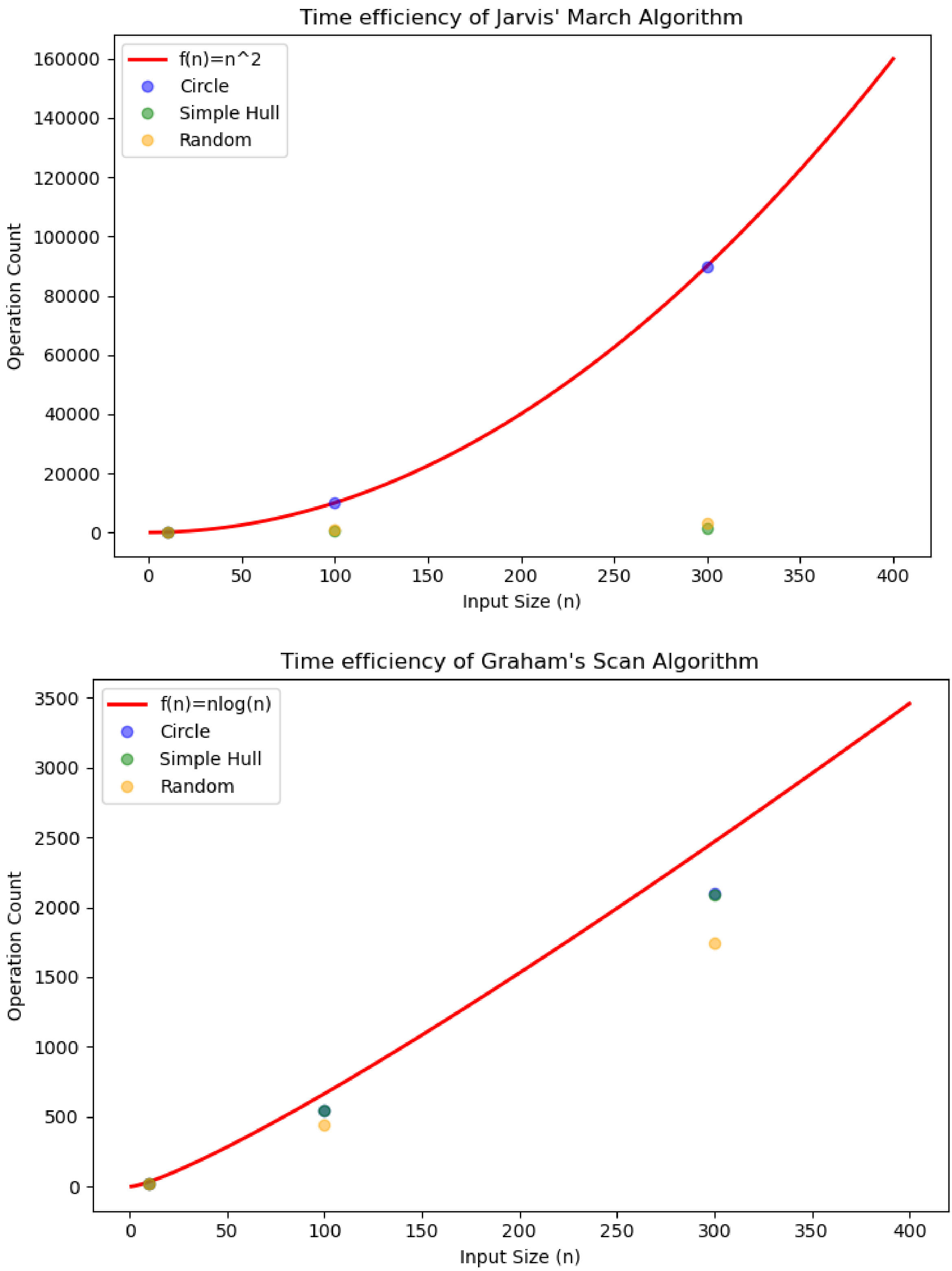


***Figure 3:*** *Graphical comparison between experimental results and theoretical bounds for Jarvis' March and Graham's Scan algorithms. (Note: datasets of size n=1000 were omitted to aid in readability of graph)*

## Discussion

The empirical analysis of the two algorithms largely aligned with the predicted theoretical efficiencies for most of the inputs.

For Jarvis' March, the operation count grew at an increasingly steep rate. This was most notable for the circular distribution of points which grew approximately quadratically with respect to the input size. This was expected as a set of points on a circle is itself a convex hull and so the theoretical complexity is $O(n^2)$, making this the worst case. Conversely, the experimental performance of this algorithm for the simple hull distribution was close to linear, due to the fixed number of points contained within the convex hull (in this case, 4). The distribution of random points produced results somewhere between these two relationships, however they were much more varied due to the inconsistent number of points in the randomly sampled convex hull.

For most test cases, Graham's Scan was much more efficient than Jarvis' March. Due to the mergesort algorithm used, the theoretical complexity of the algorithm was $O(n \log n)$ (Levitin, 2012) and the results closely mimicked the predicted values. This is likely because the complexity of mergesort is independent of the size of the resulting convex hull, and hence any variation would have been largely random.

However, Jarvis' March performed experimentally much better than Graham's Scan for simple hull distributions of larger size. This aligns with what was predicted by the theoretical complexities, as Jarvis' March has a complexity of $O(nh)$ and Graham's Scan has a complexity of $O(n \log n)$ (Fitzpatrick, 2024). From this we can see that generally, Jarvis' March will be more efficient when $h < \log n$ where $h$ is the size of the convex hull and $n$ is the input size.

For both Jarvis' March and Graham's Scan, a linked list was implemented to store the final convex hull. While this did allow new elements to quickly be inserted, as elements were always inserted in sorted order, an array would have been equally as efficient. Graham's Scan also used an array to store the polar angles and a stack to aid in the building of the convex hull. An array was chosen for this first task as it allowed for easy access to elements through indexing, where a linked list would have needed to cycle through each element. A stack was used to form the convex hull due to its last-in-first-out structure which allowed for the newest additions to the stack to be compared first.

Beyond the scope of this analysis, it should also be noted that additional operations performed within the algorithms may impact the experimental performance. For instance, Graham's scan requires additional calculations for the polar angle of each point and must carry out angle comparisons when it undergoes stack operations. While these are not frequent enough to change the 'big-O' time complexity they may impact the experimental time efficiency. Further analysis of the impact of these operations should be carried out including an analysis of specific operations and the wall time of the program. The latter of these however should be done over a large sample size to account for the variations in the machine's run time (Kulik, 2024).

Space efficiency is also not analysed in depth in this experiment and could have an impact on the real-world viability of the two algorithms. Graham's Scan required space to be allocated for an array of all polar angles, an array of indexes which was sorted, and a stack to store the convex hull in addition to the linked list that Jarvis' March also required. This however is less likely to be of concern due to the rapid expansion of data storage technology in recent years (Kulik, 2024).

## Conclusion

In the average and worst–cases, Graham's Scan was generally more efficient than Jarvis' March. However, in the best-case scenario (when the resulting convex hull is small), Jarvis' March is significantly more efficient. Additionally, Graham's Scan poses other problems such as more additional operations and a worse space efficiency. In conclusion, the selection of a particular algorithm should be done with the context of the problem in mind as this will influence the more suitable choice.

## References

Fitzpatrick, G. (2024). *Assignment 1 – Convex Hull* [Assignment notes]. https://edstem.org/au/courses/15984/lessons/50785/slides/344316

Kulik, K. (2024). *Design of Algorithms COMP20007 Lecture 3 Semester 1 - 2024* [Lecture slides]. https://edstem.org/au/courses/15984/lessons/49888/slides/337554

Levitin, A. (2012). *Introduction to the Design and Analysis of Algorithms* (3rd ed.). Pearson.