# A Computationally Efficient Gravitational Search Algorithm

Alex Rothwell
Monash University
Caulfield East, Victoria, Australia
alex.rothwell@monash.edu

Aldeida Aleti
Monash University
Caulfield East, Victoria, Australia
aldeida.aleti@monash.edu

## ABSTRACT

Gravitational search algorithm (GSA) is a population-based optimisation technique that was originally developed to deal with high-dimensional search spaces. In recent years, GSA has been successfully applied to a wide range of problems. At every iteration, the algorithm calculates the gravitational force of each solution with respect to all other solutions, which has combinatorial complexity. In this paper, we propose an efficient way for calculating the force component of each solution, reducing the complexity of GSA from $O(N^2)$ to $O(N \log(N))$, where $N$ is the population size. The experimental evaluation shows that the new algorithm is computationally efficient and cost effective.

## CCS CONCEPTS

•**Computing methodologies → Continuous space search;**

## KEYWORDS

Gravitational search algorithm; Barnes-Hut Algorithm

**ACM Reference format:**
Alex Rothwell and Aldeida Aleti. 2017. A Computationally Efficient Gravitational Search Algorithm. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3067720

## 1 INTRODUCTION

The search space of high dimensional optimisation problems increases exponentially with the number of dimensions, which makes complete search impractical. Instead, approximate methods, such as the gravitational search algorithm (GSA) [5], have proven successful in finding solutions to such complex problems.

In the last decade, many enhancements of GSA have been proposed. The algorithm has been hybridised with genetic algorithms [6], enhanced with fuzzy rules that tune the gravitational constant [4], and combined with a k-means algorithm to generate the initial population [3]. In various independent studies, GSA has been shown to outperform other well known algorithms, such as genetic algorithms [2].

GSA takes the idea of the gravitational attraction between massive bodies in space and applies it to a population of candidate solutions. By setting the masses of each of the candidates to be proportional to their fitness relative to the rest of the population, more fit candidate solutions will exert stronger gravitational force on the rest of the population than less fit candidates. This results in a convergence towards the more fit positions.

In the calculation of the gravitational force, each solution has to determine the force component from every other solution. This is computationally expensive, and incurs $O(n^2)$ complexity, where $n$ is the number of solutions. In this work, we introduce a method that reduces the complexity of GSA to $O(n \log(n))$. The experimental evaluation presented here assesses the efficiency of the proposed method on continuous function optimisation problems.

## 2 BARNES-HUT GSA

GSA is based on the physical law of gravity and the interactions between massive bodies. Solutions are considered as objects and their mass is measured by their quality, where solutions with better fitness are heavier objects. The gravitational force pulls solutions towards 'heavier' solutions. Solutions with better fitness are also slower in changing their position, ensuring the exploitation of the promising candidates. The gravitational force of solution $s_i$ is directly proportional to the active mass of solution $s_i$ and passive mass of solution $s_j$, and indirectly proportional to the square distance between the two solutions $r_{ij}$.

Barnes-Hut algorithm speeds up the calculation of the force applied to each solution [1]. Barnes-Hut gravitational search algorithm (BH-GSA) approximately computes the gravitational force at all particle positions $s_i$ by treating a group of solutions as a single centre-of-mass particle. This is achieved through recursively splitting the space into k-dimensional cells with a side-length half of the previous space, and storing the cells in a tree, where each node has $2^k$ children.

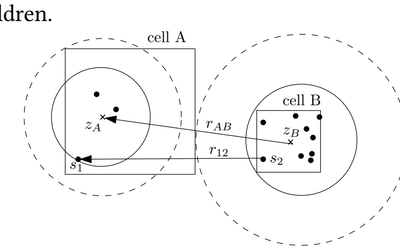


**Figure 1: Splitting of the space into cells. The crosses represent the centres of mass.**

Consider the particles depicted in fig. 1, and the two cells $A$ and $B$, with centres of mass $z_A$ and $z_B$ respectively. Only the interaction with the centre of mass of cell B is considered in the calculation of the gravitational force of particles in cell A. The cells are stored in a hierarchical tree structure with each internal node having the cells within it as children and the root node containing the entire space. To calculate the forces, the tree must first be constructed by inserting each body that will be exerting a force into the tree such that each external node represents either an empty cell or a single particle which belongs in the cell. Additionally, each internal

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
*GECCO '17 Companion, Berlin, Germany*
© 2017 Copyright held by the owner/author(s). 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3067720

particle stores the total mass and centre of mass location for the bodies stored in its subtree. To calculate the force on each particle the following if statement is evaluated, starting with the root node, until all necessary nodes are considered:

- If the current node holds a single particle and is an external node, calculate the force as the exact force exerted by that particle.
- Else, if the currently considered node's side length divided by the distance to it from the particle is less than the approximation extent constant, $\theta$, then it is sufficiently far enough away to be considered as a single particle. The force component for all bodies in this subtree are then calculated together as the centre-of-mass.
- Otherwise consider the child nodes.

## 3 EXPERIMENTS

To test the quality of the solutions that BH-GSA generates, and its time efficiency, both BH-GSA and the original GSA were implemented in Python, and used to solve the unimodal high-dimensional, multimodal high-dimensional, and multimodal low dimensional functions used in the original work [5]. Extra functions from CEC 2013 benchmark were included for a more complete analysis [**?** ].

BH-GSA was run with four different values for $\theta$: $10^5$, 0.5, $10^{-5}$ and 0 in order to test the impact that the approximation extent has. Note that $\theta = 0$ corresponds to running BH-GSA with no force calculation approximation extent. All optimisation schemes were run for a fixed number of iterations to compare the quality achieved by each of the methods. Here, the number of iterations was set to 1,000. The population size of GSA and BH-GSA was initially set to 100. All trials were repeated 30 times.

The cost effectiveness of the algorithms is reported as 1/(quality× time). This also measures the efficiency of BH-GSA in approximating the force component. The median of the cost effectiveness over the 30 trials for all optimisation schemes and problems are shown in table 1.

Results show that BH-GSA outperforms the original GSA in the majority of the problems independent of the $\theta$ value (force approximation value). The best performing variant, however, is BH-1e5 with $\theta$ equal to $10^5$ (the largest approximation value tested). The superior performance of the BH-GSA is more evident in the more complex and multidimensional test functions. In this problem category, all BH-GSA variant perform better than the original method.

We performed a two tailed t-test analysis, with a null hypothesis of no significant difference in the performances of the GSA and the BH-GSA variants. Statistically significant results are marked with an asterisk (*). The statistical analysis confirms that the superior performance of the BH-GSA is statistically significant at a 95% confidence in the majority of the problem instances, rejecting the null hypothesis.

**Table 1: Medians and statistical significance.**

| | | | | | |
|---|---|---|---|---|---|
| | **Unimodal high-dimensional test functions** | | | | |
| | GSA | BH-0 | BH-1e-5 | BH-0.5 | BH-1e5 |
| $F_1$ | 3.00e+15 | 1.24e+15$^*$ | 1.25e+15$^*$ | 1.23e+15$^*$ | **1.14e+16**$^*$ |
| $F_2$ | 5.80e+05 | 4.66e+05$^*$ | 4.83e+05$^*$ | 4.55e+05$^*$ | **3.20e+06**$^*$ |
| $F_3$ | 4.30e-05 | 4.20e-05 | 2.60e-05$^*$ | 3.11e-05 | **1.17e-04**$^*$ |
| $F_4$ | 4.87e+06 | 4.62e+06 | 4.45e+06$^*$ | 4.63e+06 | **3.26e+07**$^*$ |
| $F_5$ | 2.12e-04 | 3.33e-04$^*$ | 3.68e-04$^*$ | 3.49e-04$^*$ | **9.67e-04**$^*$ |
| $F_6$ | inf | inf | inf | inf | inf |
| $F_7$ | 2.09e-01 | 1.88e-01 | 1.55e-01$^*$ | 1.92e-01 | **7.19e-01**$^*$ |
| | **Multimodal high-dimensional test functions** | | | | |
| | GSA | BH-0 | BH-1e-5 | BH-0.5 | BH-1e5 |
| $F_1$ | 4.54e-07 | 8.53e-07$^*$ | 8.50e-07$^*$ | 8.56e-07$^*$ | **1.48e-06**$^*$ |
| $F_2$ | 1.05e-03 | 6.74e-04$^*$ | 8.06e-04$^*$ | 5.69e-04$^*$ | **2.09e-03**$^*$ |
| $F_3$ | 7.14e+06 | 5.75e+06$^*$ | 5.86e+06$^*$ | 5.75e+06$^*$ | **3.72e+07**$^*$ |
| $F_4$ | 3.76e-03 | inf | inf | inf | inf |
| $F_5$ | 5.63e-03 | 6.72e-03$^*$ | 6.70e-03 | 6.77e-03 | **1.82e-02**$^*$ |
| $F_6$ | 3.89e+15 | 1.99e+15$^*$ | 2.00e+15$^*$ | 2.29e+15$^*$ | **4.62e+15**$^*$ |
| | **Multimodal low-dimensional test functions** | | | | |
| | GSA | BH-0 | BH-1e-5 | BH-0.5 | BH-1e5 |
| $F_1$ | 4.44e-04 | 1.51e-03 | 1.23e-03 | 1.28e-03 | 2.16e-03 |
| $F_2$ | 1.95e-01 | 2.51e-01$^*$ | 2.66e-01$^*$ | 2.96e-01$^*$ | **1.15e+00**$^*$ |
| $F_3$ | 2.34e+01 | 6.67e+01$^*$ | 6.69e+01$^*$ | 6.90e+01$^*$ | **1.31e+02**$^*$ |
| $F_4$ | 4.91e+01 | 1.31e+02$^*$ | 1.32e+02$^*$ | 1.35e+02$^*$ | **2.41e+02**$^*$ |
| $F_5$ | inf | inf | inf | inf | inf |
| $F_6$ | 1.84e+00 | 4.70e+00$^*$ | 4.76e+00$^*$ | * 4.80e+00$^*$ | **8.02e+00**$^*$ |
| $F_7$ | 2.55e+00 | 7.15e+00$^*$ | 7.19e+00$^*$ | 7.18e+00$^*$ | **1.14e+01**$^*$ |
| $F_8$ | 1.55e-04 | 1.61e-03$^*$ | 1.61e-03 | 1.61e-03$^*$ | **2.54e-03**$^*$ |
| $F_9$ | 3.48e+01 | 8.04e+01$^*$ | 8.11e+01$^*$ | 8.05e+01$^*$ | **1.24e+02**$^*$ |
| $F_{10}$ | 4.29e+01 | 9.34e+01$^*$ | 9.43e+01$^*$ | 9.44e+01$^*$ | **1.37e+02** |
| | **CEC 2013 test functions** | | | | |
| | GSA | BH-0 | BH-1e-5 | BH-0.5 | BH-1e5 |
| $F_1$ | 1.35e-14 | 1.20e-14$^*$ | 1.19e-14$^*$ | 1.19e-14$^*$ | **1.56e-14**$^*$ |
| $F_2$ | 2.51e-11 | 1.82e-11$^*$ | 1.83e-11$^*$ | 1.83e-11$^*$ | 2.09e-11$^*$ |
| $F_3$ | 4.73e-19 | 1.84e-18$^*$ | 2.07e-18$^*$ | **2.42e-18**$^*$ | 1.54e-18$^*$ |
| $F_4$ | 1.70e-15 | 1.36e-15$^*$ | 1.35e-15$^*$ | 1.36e-15$^*$ | 1.70e-15$^*$ |
| $F_5$ | 1.99e-19 | 1.37e-18$^*$ | 1.43e-18$^*$ | 1.79e-18$^*$ | **2.29e-18**$^*$ |
| $F_6$ | 5.86e-20 | 3.38e-19$^*$ | **3.85e-19**$^*$ | 3.64e-19$^*$ | 2.19e-19$^*$ |
| $F_7$ | 2.31e-18 | 5.69e-18$^*$ | 5.55e-18$^*$ | 5.17e-18$^*$ | **8.47e-18**$^*$ |
| $F_8$ | 5.69e-09 | 6.75e-09$^*$ | 6.66e-09$^*$ | 6.79e-09$^*$ | **9.10e-09**$^*$ |
| $F_9$ | 1.11e-04 | 7.82e-05 | 7.80e-05 | 7.81e-05 | **8.97e-05**$^*$ |
| $F_{10}$ | 1.42e-16 | 6.59e-16$^*$ | 6.32e-16$^*$ | 6.37e-16$^*$ | **8.71e-16**$^*$ |
| $F_{11}$ | 1.47e-11 | 2.68e-11 | 3.09e-11 | 3.31e-11 | **4.37e-11** |
| $F_{12}$ | **2.30e-09** | 1.66e-09$^*$ | 1.66e-09$^*$ | 1.66e-09$^*$ | 1.91e-09$^*$ |
| $F_{13}$ | 1.58e-17 | **1.24e-16**$^*$ | 8.39e-17$^*$ | 9.70e-17$^*$ | 1.01e-16$^*$ |
| $F_{14}$ | 2.80e-21 | 1.20e-20$^*$ | 1.32e-20$^*$ | 1.28e-20$^*$ | **1.64e-20**$^*$ |
| $F_{15}$ | 1.08e-12 | 1.03e-12$^*$ | 1.08e-12$^*$ | 1.11e-12$^*$ | **1.37e-12**$^*$ |

## 4 CONCLUSION

An improvement over the Gravitational Search Algorithm through replacing the force calculation with the Barnes-Hut algorithm was introduced in this work. The new method was found to significantly improve the computational cost of the original GSA. The experimental evaluation over 38 benchmark fitness functions showed that the performance of the new approach is superior. As such, it would be beneficial for further research to be conducted into the performance of other population based methods such as Particle Swarm Optimisation when combined with the Barnes hut algorithm to reduce the complexity in calculating the velocity and position of the particles.

## REFERENCES


[1] Josh Barnes and Piet Hut. 1986. A hierarchical O(N log N) force-calculation algorithm. *Nature* 324, 6096 (Dec. 1986), 446–449. DOI: http://dx.doi.org/10.1038/324446a0

[2] Taisir Eldos and Rose Al Qasim. 2013. On the performance of the Gravitational Search Algorithm. *International Journal of Advanced Computer Science and Applications* 4, 8 (2013).

[3] Abdolreza Hatamlou, Salwani Abdullah, and Hossein Nezamabadi-pour. 2012. A combined approach for clustering based on K-means and gravitational search algorithms. *Swarm and Evolutionary Computation* 6 (2012), 47–52.

[4] J Vijaya Kumar, DM Vinod Kumar, and K Edukondalu. 2013. Strategic bidding using fuzzy adaptive gravitational search algorithm in a pool based electricity market. *Applied Soft Computing* 13, 5 (2013), 2445–2455.

[5] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. 2009. GSA: a gravitational search algorithm. *Information sciences* 179, 13 (2009), 2232–2248.

[6] Filipp Seljanko. 2011. Hexapod walking robot gait generation using genetic-gravitational hybrid algorithm. In *Advanced Robotics (ICAR), 2011 15th International Conference on*. IEEE, 253–258.