

A Most Irrational Foraging Algorithm

Abhinav Aggarwal
Department of Computer Science
University of New Mexico
Albuquerque, USA
abhiag@unm.edu

William F. Vining
Moses Biological Computation Lab
University of New Mexico
Albuquerque, USA
wfvining@cs.unm.edu

Diksha Gupta
Department of Computer Science
University of New Mexico
Albuquerque, USA
dgupta@unm.edu

Jared Saia
Department of Computer Science
University of New Mexico
Albuquerque, USA
saia@cs.unm.edu

Melanie E. Moses
Moses Biological Computation Lab
University of New Mexico
Albuquerque, USA
melaniem@unm.edu

ABSTRACT

We present a foraging algorithm, GOLDENFA, in which search direction is chosen based on the Golden Ratio. We show both theoretically and empirically that GOLDENFA is more efficient for a single searcher than a comparable algorithm where search direction is chosen uniformly at random. Moreover, we give a variant of our algorithm that parallelizes linearly with the number of searchers.

CCS CONCEPTS

• **Computing methodologies** → **Bio-inspired approaches**; *Distributed algorithms*;

KEYWORDS

Fibonacci, Golden Ratio, collective search, swarm robotics, foraging.

1 INTRODUCTION

"Golden Ratio is a powerful mathematical constant woven into the very fabric of biology. It is the unique visual tension between comforting symmetry and compelling asymmetry, and its thoughtful application can bring beauty and harmony and intrigue to all manner of designed things." - Darrin Crescenzi

The Fibonacci sequence 1, 1, 2, 3, 5, 8, 13, ... is created by the recurrence $F_n = F_{n-1} + F_{n-2}$ and is one of the most famous biologically-inspired mathematical sequences. The Golden Ratio, denoted ϕ , is the limit of the ratio of consecutive numbers in this sequence. Fibonacci generated the sequence as an idealized model of a reproducing rabbit population assuming overlapping generations [2]. It was documented in India many centuries earlier, and has been observed in numerous biological systems including the

Permission to make digital or hard copies of all or part 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 components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
BDA'19, July 29, 2019, Toronto, CA.

© 2019 Association for Computing Machinery.
ACM ISBN\$15.00
[https://doi.org/...](https://doi.org/)

arrangement of pine cones, unfurling of fern leaves, and the arrangement of sunflower seeds that optimally fills the circular area of the flower [17]. The Golden ratio and Fibonacci numbers have been used in computer science for various applications like obtaining optimal schedules for security games [10], Fibonacci hashing [14], bandwidth sharing [8], data structures [4] and game theoretic models for blocking-resistant communication [13].

In this paper, we use the Golden Angle created by arcs that form the Golden Ratio to develop a collective foraging algorithm that reduces the time to first discovery of clustered targets. In particular, we show that our new foraging algorithm, GOLDENFA, performs better in theory and practice than a previous algorithm that chooses search directions uniformly at random [1].

Motivation for using the "most irrational" number. Any number can be written as an integer plus 1 over another number. The larger the denominator in the fractional part, the better the integer part is as an approximation. For example, $\pi = 3 + 1/x$, for $x \geq 7$, and so π is fairly-well approximated by 3. Thinking recursively, the number x in the denominator can itself be written as an integer plus 1 over another number. Thus, we can write any number as a (possibly infinite) continued fraction [11]

$$x_1 + \frac{1}{x_2 + \frac{1}{x_3 + \dots}}$$

where the x_i values are all integers for $i \geq 1$.

The degree to which the original number is well-approximated by a finite continued fraction depends on how large the x_i values are. When $x_i = 1$ for all $i \geq 1$, we obtain an irrational number that is most difficult to approximate. To find this most difficult irrational number, we set $y = 1 + \frac{1}{y}$, and solve the resulting quadratic equation to obtain a solution $y = \frac{1+\sqrt{5}}{2}$, which is the celebrated Golden ratio ϕ .

The fact that ϕ is difficult to approximate with a rational number has useful implications in ensuring angles are "well-spread". For example, if we start at the point 0 on a unit circle, and iteratively add points by moving clockwise by distance ϕ , then we will end up with points that spread out with ample distance between points.¹ Interestingly, this approximates how plants add florets, leaves and

¹In contrast, if we use π instead of ϕ , the points will cluster around 3 spokes since π is fairly well-approximated by 3. See [18] for a fascinating simulation and discussion of these facts.

petals as they grow. If the next leaf is added by moving distance ϕ along a unit circle, this ensures that leaves are well-spread in order to increase their total intersection with sunlight without interference.

In this paper, we make use of the “most irrational” property of ϕ to design a foraging algorithm. A searcher first forages in an initial direction from a nest site to a boundary of the search space, and then returns to the nest site. The angle for the next spoke from the nest site is chosen by moving a distance of ϕ along a unit circle. In this way, we can ensure that our foraging spokes are “well-spread”, thus minimizing overlap at the circle center while maximizing the probability of the first discovery of a cluster of resources somewhere in the circle area. If there are multiple searchers, it is straightforward to parallelize this process.

The rest of the paper is organized as follows: We first define our formal model and problem statement. Then we introduce GOLDENFA and explain the theoretical upper bound on finding a single target of a given diameter for a single searcher and then multiple searchers. We then compare theoretical predictions to simulated searchers.

2 OUR MODEL AND PROBLEM STATEMENT

We assume a circular arena of radius R with the *nest* at the centre². A *cluster*³ of targets is placed at a distance D from this collection zone and has a diameter Δ . We assume an obstacle-free arena in which the cluster does not move or regenerate as targets are collected.

We assume $N \geq 1$ searchers. Each searcher has limited memory and can complete straight line motion in a specified direction from the nest to the edge of the search arena. Searchers can detect if they have encountered a target. Moreover, they know the direction in which they are currently moving, but no information from the past can be stored and or communicated.

Overview of our Approach. Previous work shows that foraging for a single target in an arena without any knowledge of the arena parameters requires time that is proportional to the area of the arena [3]. We seek to reduce time to discover a cluster with large diameter. In particular, when cluster diameter is any increasing function of arena diameter our foraging strategy can locate the cluster in sub-quadratic time.

We state our main result in the theorem below and defer its proof to the full version of our paper.

THEOREM 2.1. *The number of time steps taken by GOLDENFA before the cluster is located for the first time is $O\left(\left(\frac{R}{N\Delta} + 1\right)D\right)$.*

Again, as long as Δ is some increasing function of D , GOLDENFA has improved performance. Additionally, when $N = R$ searchers work in parallel, the distance travelled is $O(D)$, which is asymptotically optimal.

To prove this theorem, we compute the minimum number of spokes needed before the maximum arc length between any two consecutive spokes becomes sufficiently small. In particular, we determine when this maximum arc length become smaller than

²Similar to existing work, this arena can be modelled as a discrete grid with the Manhattan distance as the metric, however, since working in the continuous Euclidean space introduces only a constant multiplicative blowup, we state our algorithm and the results for a continuous arena for mathematical simplicity.

³We assume that this cluster is circular in shape for mathematical simplicity, however, our results apply to all cluster shapes that can be circumscribed by a circle.

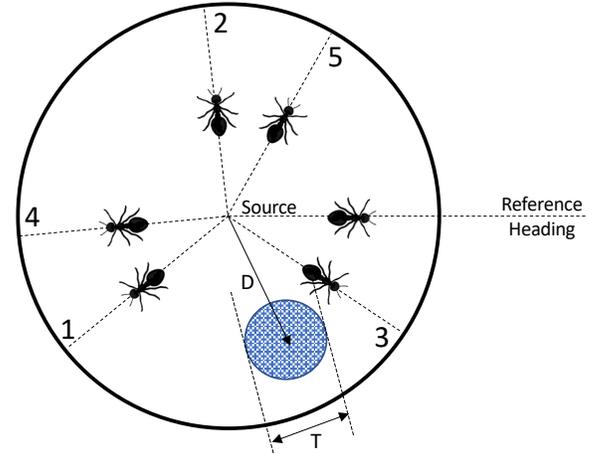


Figure 1: A schematic of the GOLDENFA algorithm for a single cluster (shown as the shaded area) of diameter Δ located at a distance D from the source.

the cluster diameter. No matter where the cluster is located in the arena, we require $O(D/\Delta)$ spokes before a spoke will intersect the cluster.

3 THE GOLDEN FORAGING ALGORITHM

We now describe our algorithm, GOLDENFA.

Single-searcher case: The searcher starts from the reference heading, moves along a spoke to the end of the arena and then returns to the nest. Next, the searcher turns an arc distance of ϕ along a unit circle centered at the nest, and moves along a spoke in that direction to the end of the arena. This process continues until the searcher locates the cluster.

Multiple searchers case: When $N \geq 1$ searchers search for the cluster together, searcher i starts the search at an arc distance of $(i - 1)\phi$ from the reference heading and searches along spokes that are $N\phi$ arc distance apart (see Figure 1).

For this to work, each searcher must have a unique identifier and know its relative order in the sorted sequence of these identifiers. For simplicity, we assume that the search stops as soon as any searcher discovers a target.

4 EXPERIMENTAL EVALUATION

The theoretical analysis in the previous section makes many assumptions that do not hold in a robotics setting. These include the lack of collisions between robots, which is a cause of significant slow-down in central place foraging [16]; the use of an exact value of ϕ instead of an approximation; and imprecise robot motion. To test whether the analysis holds under more realistic constraints, we implemented the multiple-searcher GOLDENFA in the ARGOS swarm robotics simulator [19], and evaluated the time required to discover a single square cluster of resources placed uniformly at random in a 100×100 meter square arena.

The resources in the cluster are placed 0.15 meters apart such that a square cluster with k resources on a side has diameter $\Delta \approx 0.15k$

meters. We conducted experiments with square clusters of 8×8 , 16×16 , 32×32 , 64×64 , and 128×128 resources. For each cluster and value of N , we computed the average time over 100 experiments.

Time to Discover versus Δ and N . Figure 2 shows how the time to discover decreases as Δ increases. There seems to be a nice linear decrease, as predicted in Theorem 2.1, for $N = 1$ and $N = 10$, with a slope of 32. However, for $N = 100$, the slope is larger, probably because of congestion effects. We note that congestion happens when searchers are not infinitely small, as is assumed by the theory.

Figure 3 shows how the time to find the cluster depends on the number of searchers. For increasing N , we evaluated square clusters of size 16×16 , 32×32 and 64×64 in a 100×100 meter arena. The figure shows how additional searchers reduce the search time, but how adding too many searchers causes increased congestion, which results in increased search times.

Comparison to Ballistic Algorithm. We compared our algorithm with the Ballistic Central Place Foraging Algorithm (bCPFA) [1], for the case $N = 10$. In bCPFA, the searchers traverse spokes in directions chosen uniformly at random, until the cluster is located. It can be shown that the number of spokes required by this algorithm is $O\left(\frac{D}{\Delta} \log \frac{D}{\Delta}\right)$ with high probability, using the result about maximum distance between random points on an arc from [12]. This is asymptotically worse than GOLDENFA for all values of D and Δ .

Figure 4 shows that the searchers using GOLDENFA are able to find the cluster faster than those using the Ballistic CPFA (bCPFA). The results for varying cluster diameter show that the search time using bCPFA is both longer and more variable than the search time using GOLDENFA. Ballistic CPFA searches have a large number of outliers that take considerably longer than average time to find the cluster.

5 RELATED WORK

Search is a fundamental problem in biology, where survival depends on search for mates, prey and other resources. It is also a common problem in robotics and mobile computing. Collective search, where multiple searchers must coordinate, is a key problem in computer science, robotics and in social insects. Ant- and bee-inspired algorithms have been particularly influential in swarm robotics research [9, 15, 20]. In prior work, we have used algorithms inspired by foraging behaviours of desert seed-harvesting ants. These ants forage collectively as follows: each ant leaves a central nest, travels in a relatively straight line in an apparently randomly chosen direction. Upon finding food, the ant determines whether to remember and return to that location or communicate the location to nest mates by laying a pheromone trail.

We mimic this behaviour in robots with a generic Central Place Foraging Algorithm (CPFA) which is effective at finding nearby resources quickly, particularly when resources are distributed in multiple clusters of unknown diameter [7]. However, a simpler interlocking spiral algorithm finds targets faster than the bio-inspired CPFA, and the spiral is particularly fast at completely collecting all targets [1, 5, 6].

Our prior work [1] shows that the most efficient foraging algorithms that completely retrieve all items in a search arena in

minimal time use geometric patterns that fill space with minimal traversal of already searched territory. In contrast, in this work, we seek foraging patterns that minimize the time to first discovery when resources are clustered in a single pile of unknown diameter.

6 DISCUSSION

Foraging for Multiple Clusters When multiple clusters are placed in the arena, time for first detection will be determined by the largest cluster. For discover of all clusters, the number of spokes must be large enough to find the cluster with smallest diameter. Note that if the smallest cluster has a diameter that is constant, then the asymptotic performance of GOLDENFA is no better than exhaustive search, which has quadratic cost.

Conclusions and Future Work: We have described an algorithm, GOLDENFA that can locate a cluster efficiently. Our algorithm has search time that asymptotically decreases linearly with cluster diameter and the number of searchers. Moreover, our algorithm performs well in practice, with search times that generally match our theoretical predictions. Further, our algorithm empirically outperforms another common search algorithm in both mean time to discovery and variance. We believe our algorithm is a first step toward minimizing foraging time when resources are distributed in an unknown number of clusters of unknown and variable diameter.

Next steps of this work include the following. First, implementing GOLDENFA in real, rather than simulated, robots to determine how well it performs with real-world noise and error. Second, a theoretical analysis of resilience of GOLDENFA. Finally, measuring collection time of multiple clusters by GOLDENFA and comparing to other foraging algorithms.

REFERENCES

- [1] AGGARWAL, A., GUPTA, D., VINING, W. F., FRICKE, G. M., AND MOSES, M. E. Ignorance is not bliss: An analysis of central-place foraging algorithms. *Manuscript under submission*. (2019).
- [2] DUNLAP, R. A. *The golden ratio and Fibonacci numbers*. World Scientific, 1997.
- [3] FEINERMAN, O., AND KORMAN, A. The ants problem. *Distributed Computing* 30, 3 (2017), 149–168.
- [4] FREDMAN, M. L., AND TARJAN, R. E. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)* 34, 3 (1987), 596–615.
- [5] FRICKE, G. M., JOSHUA, P. H., ANTONIO, D. G., LINH, T. T., AND MELANIE, E. M. A Distributed Deterministic Spiral Search Algorithm for Swarms. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on* (2016), Copyright 2016, IEEE reprinted with permission from the authors., pp. 4430–4436.
- [6] HECKER, J. P., CARMICHAEL, J. C., AND MOSES, M. E. Exploiting clusters for complete resource collection in biologically-inspired robot swarms. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2015* (2015), pp. 434–440.
- [7] HECKER, J. P., AND MOSES, M. E. Beyond pheromones: evolving error-tolerant, flexible, and scalable ant-inspired robot swarms. *Swarm Intelligence* 9, 1 (2015), 43–70.
- [8] ITAI, A., AND ROSBERG, Z. A golden ratio control policy for a multiple-access channel. *IEEE Transactions on Automatic Control* 29, 8 (1984), 712–718.
- [9] KARABOGA, D., AND AKAY, B. A survey: algorithms simulating bee swarm intelligence. *Artificial intelligence review* 31, 1–4 (2009), 61–85.
- [10] KEMPE, D., SCHULMAN, L. J., AND TAMUZ, O. Quasi-regular sequences and optimal schedules for security games. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms* (2018), Society for Industrial and Applied Mathematics, pp. 1625–1644.
- [11] KHINCHIN, A. I. *Continued fractions*, vol. 525. P. Noordhoff, 1963.
- [12] KING, V., AND SAHA, J. Choosing a random peer. In *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing* (2004), ACM, pp. 125–130.

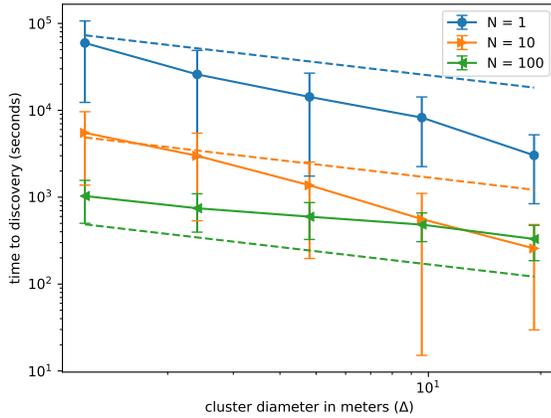


Figure 2: Mean time to discover the single cluster in a 100×100 meter arena. The dashed lines show the predicted time to discover the cluster $f(x) = 32(\frac{RD}{N\Delta})$ (from Theorem 2.1) where $D = \frac{2}{3}R$ (the expected distance of a randomly placed cluster from the nest). Error bars show plus or minus one standard deviation. As the swarm size increases clusters are found more quickly; however, for $N = 100$ the effects of congestion near the nest slow down the search and lead to longer than the predicted search time.

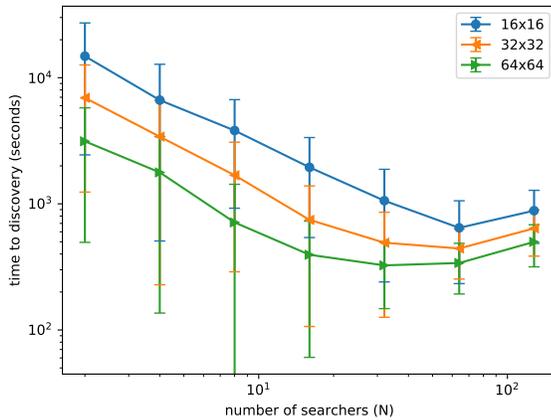


Figure 3: Mean time to discover the cluster as the number of searchers increases for three cluster diameters. The effect of increased congestion is clearly visible as an increase in time to discovery in the largest swarms.

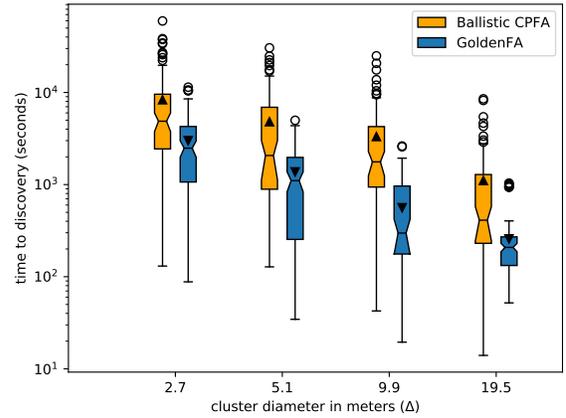


Figure 4: Comparison of bCPFA and GOLDENFA. The time for 10 searchers to find a single cluster in a 100×100 meter arena as the cluster diameter increases. Black triangles indicate the mean. Not only does it take longer to find the cluster using the ballistic CPFA, but the search time is more variable compared to GOLDENFA.

algorithm for biologically-inspired robot swarms. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct 2016), pp. 3815–3821.

[17] NAYLOR, M. Golden, and π flowers: A spiral story. *Mathematics Magazine* 75, 3 (2002), 163–172.

[18] NUMBERPHILE. The golden ratio (why it is so irrational) - numberphile. <https://www.youtube.com/watch?v=sj8Sg8qjOg&t=711s>.

[19] PINCIROLI, C., TRIANNI, V., O'GRADY, R., PINI, G., BRUTSCHY, A., BRAMBILLA, M., MATHEWS, N., FERRANTE, E., DI CARO, G., DUCATELLE, F., BIRATTARI, M., GAMBARDILLA, L. M., AND DORIGO, M. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence* 6, 4 (2012), 271–295.

[20] ŞAHİN, E. Swarm robotics: From sources of inspiration to domains of application. In *International workshop on swarm robotics* (2004), Springer, pp. 10–20.

[13] KING, V., SAIA, J., AND YOUNG, M. Conflict on a communication channel. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing* (2011), ACM, pp. 277–286.

[14] KNUTH, D. E. The art of computer programming, vol. 3: Searching and sorting. Reading MA: Addison-Wisley (1973).

[15] KRIEGER, M. J., BILLETER, J.-B., AND KELLER, L. Ant-like task allocation and recruitment in cooperative robots. *Nature* 406, 6799 (2000), 992.

[16] LU, Q., HECKER, J. P., AND MOSES, M. E. The mpfa: A multiple-place foraging