# Mitigating Premature Convergence with Hybrid GA-Boids

**Mohamed Affan Dhankwala** mdhankw1@jh.edu
*Masters of Artificial Intelligence*
*Johns Hopkins University*
*Baltimore MA, USA*

**Abstract**

Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are widely accepted due to their swarm mechanics and global communication. They, however, falter with premature convergence as particles group too early around promising results or evolve too similarly. Pulled from the realm of computer graphics, Boids model is used to animate flocking and swarming behavior of birds, fish and other herd animals. Each particle, boid, is aware of itself, the swarm, and its neighbors. Combining this model with GAs creates a hybrid-GA model that should mitigate premature convergence due to additional diversity spurred by local exploration. The performance of this hybrid shall be benchmarked on continueous optimization functions, feature selection, and scheduling tasks and statistically compared to the performance of PSOs and GAs.

## 1 Introduction

Evolutionary and swarm intelligence problems have been widely used to solve complex optimization problems. Two algorithms, Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are commonly accepted and utilized due to their efficiency. PSOs consist of particles, solutions, that navigate a state space and adjust their movement in accordance to the optimization problem. GAs likewise consist of individuals, solutions, that evolve over generations to best fit the overarching optimization problem. While both can be tuned and have demonstrated flexible and robust qualities, they both suffer from premature convergence as solutions begin to gather in one area or all evolve into some dominant species. An unrelated model, Boids model, proposed by Reynolds in 1987 in the realm of computer graphics to mimic flocking behavior of birds may offer some assistance due to it's focus on individual, local, and global interactions. By combining GA and Boids models, we result in a hybrid GA-Boids model that combines GA's global information and diversity with Boids model's local interactions and swarm dynamics to prolong exploration and mitigate premature convergence. We hypothesize that a hybrid GA-Boids model will perform statistically better on contineous functional optimization, feature selection, and scheduling tasks than PSOs or GAs due to its more robust avoidance of premature convergence.

## 2 Literature and Background Review

PSOs and GAs have, as previously mentioned, be used widly in the field of optimziation problems due to their evolutionary and swarm intelligent behavior. Boids model is relatively new in the field of optimization with utilization only in computer graphics. In addition to that, while hybrid models such as GA-PSO exist, no hybrid between GA and Boids exists to confirm its expected efficiency. Below are brief histories, summaries and equations for the models we shall investigate in this paper.

### 2.1 PSO Models

Inspired by flocks of birds and swarms of fish, particle swarm optimization (PSO) is a population-based stochastic optimization method introduced by Kennedy and Eberhart in 1995. It operates by simulating a swarm of candidate particles that independently move around the state space and adjust their velocity and acceleration based on their saved best position, best positions founds by the other particles–the swarm, or based on their own inertia. A 'good' position is based on the optimization definition of a position that provides high objective value. In other words, a minimization problem would consider positions where $f(x)$ is small to be good positions. The equation that dictates the velocity of each singular particle as proposed by Kennedy and Eberhart is as follows:

$$v_i(t+1) = wv_i(t) + c_1 r_1[p_{best} - x_i(t)] + c_2 r_2[g_{best} - x_i(t)]$$

In the above equation, $w$ is the inertia weight-controlling momentum introduced by Shi and Eberhart. $c_1$ is the cognitive acceleration coefficient which is representative of how much of a particle's past experience impacts its future movements. $c_2$ is the social cognitive acceleration coefficient which is how much of the group's collective knowledge impacts the particle's future movements. $r_1$ and $r_2$ are random variables uniformly distributed in [0, 1]. These random variables are each multiplied to the cognitive and social acceleration coefficients to randomly weigh how much each coefficient impacts the next action. These random particles serve in allowing some exploration and mitigate some premature convergence. $p_{best}$ is the particle's best saved position while $g_{best}$ is the swarm's best saved position. $x_i(t)$ is the current position of the particle while $v_i(t)$ is the current velocity of the particle.

PSOs have proven to be quite useful in continuous optimization problems where objective functions are noisy or information is limited or unavailable. Their utilization ranges in fields from neural networks to engineering design (Bonyadi and Michalewicz). The velocity equation governing each particle allows a balanced exploration and exploitation rate and the communication between each particle and the rest of the swarm allows for spreading of information and converging on optimal solutions.

However, as noted in the introduction, this model is prone to premature convergence in multimodal landscapes. This can be quite disadvantageous as it beats the purpose of the optimization problem by providing a less than optimum solution. Hybrid and adaptive methods have been proposed by Bonyadi and Michalewicz and others but we shall not cover those models.

### 2.2 GA Models

Genetic algorithms are population-based stochastic search models inspired by real organisms and the process of natural selection and biological evolution (Holland). Each individual is a solution that is encoded as a string of genes that are carried between generations–similar to how real genes are mapped in accordance to an individual's ancestors. This algorithm evolves a population while relying on selection, crossover, and mutation equations (Goldberg).

The evolution of each population occurs in generations. At each generation, individuals are evolved by selection, crossover, and mutation methods. Selection methods mimic real world natural laws where well-fit individuals are selected for breeding and less-fit individuals are ignored or discarded (Blickle and Thiele). Fitness, as in PSO, has to do with how well an individual conforms to the opitmization goal. A minimizing optimization problem would consider individuals with lower $f(x)$ as more fit with those with a higher value. Crossover is the process of combining two individuals's genomes via either single-point or uniform crossover to create an offspring chromosome that is a product of the two parents. Mutation refers to the random chance of alteration in the genes of the offspring. This mutation element is to encourage exploration and prevent premature convergence.

GA have excelled in diverse environments such as function optimization (De Jong), feature selection (Siedlecki and Sklansky), and machine learning model optimization (Whitley). This is due to the intrinsic global search capability that GAs can enforce due to their population-based models. Each individual solution can be encoded as binary strings, integers, or vectors which allow for a flexible representation for diverse problems. GAs also do not need much problem information as their individuals require little information to explore the state space–functionally allowing them to be inserted in many optimization problems.

However, GAs do have some limitations that include premature convergence due to loss of genetic diversity (Mahfoud) which arises when one species begins to dominate the rest and future generations all share this individual's gene. Another issue is slower convergence speed in high dimensional or continueous settings (Haupt and Haupt) and sensitivity to crossover and mutation rates (Eiben and Smith). For the sake of this research project, we shall focus on the premature convergence deficiency. As previously noted in the PSO model, hybrid and adaptive solutions have been proposed by researchers included Talbi to balance exploration and exploitation rates via a GA-PSO model, but we shall not focus on those hybrids over the course of this paper.

### 2.3 Boids Models

Unlike the previous PSO and GA models introduced, Boids models are not an optimization model but rather one that simulates the flocking pattern of birds, fish, and other swarming creatures. Introduced by Craig Reynolds, this model consists of particles known as boids–a contraction of bird-oids–that abide by a set of laws. Reynolds aimed at defining the laws that govern animals that operate as a swarm. He noticed that birds and fish that move as a swarm all begin acting as one entity rather than individuals. This allows them to stick together while avoiding adversaries. He proposed three rules that these animals operate by: separation, alignment, and cohesion.

Separation is the avoidance of crowding neighbors. In other words, do not overlap or intersect other nearby particles. It is defined as

$$v_i^{sep} = \sum_{j \in N_i} \frac{r_i - r_j}{|r_i - r_j|}^2$$

Where $r_i$, $r_j$ are positions of nearby boids $i$, $j$ and $N_i$ is the set of neighbors of the boid at some prespecified radius.

Alignment is the action of steering towards the average heading of neighbors. Attempt to move towards wherever one's neighbors are moving. It is defined as follows:

$$v_i^{align} = \frac{1}{|N_i|} \sum_{j \in N_i} v_j$$

Where $v_j$ is the velocity of neighboring boid $j$.

Cohesion is steering towards the average position of neighbors. This allows particles to stay close to one another. It is defined as:

$$v_i^{cohesion} = \frac{1}{|N_i|} \sum_{j \in N_i} (r_j - r_i)$$

When combined, the three laws contribute as velocity vectors to the overall velocity of a particle:

$$v_i^{total} = v_i^{sep} + v_i^{align} + v_i^{cohesion}$$

As previously stated, this model is not for optimization but rather has been used in computer graphics to simulate realistic flocking behaviors in films and games. It can also be placed in predator-prey models to observe animal behaviors.

### 2.4 Hybrid GA-Boids

Combining GA and Boids results in a hybrid GA-Boids model consisting of particles that behave according to the global search capabilities of GA and according to local interactions as defined in Reynolds Boid model. We propose this model to counteract PSOs and GAs largest limitation of premature convergence. Since both of the previously mentioned models operate on particles that have some understanding of themselves and the rest of their swarm, it may be worth including additional information into each particle about its neighbors. We anticipate this allows for maintenance of population diversity and encouragement of exploration.

In light of this paper, we propose to introduce hybrid GA-Boids models in the realm of continuous function optimization, feature selection, and scheduling tasks. This may result in a more computational complex model and one that converges slowly but this should be offset by one that is more robust than both PSO and GAs individually per Talbi and Michalewicz. Note that the previously mentioned authors focused on other hybrid models not including the hybrid GA-Boids model.

## 3 Benchmark functions and metrics

Since we wish to determine how well a hybrid GA-Boids model mitigates the issues faced by PSOs and GAs, it is worth comparing these models in the realms of problems that PSOs and GAs face. This includes continuous function optimization, feature selection, and scheduling tasks.

### 3.1 Continuous Function Optimization

Continuous function optimization are mathematical test functions with known optima. The goal is to align particles within each model to either find the maximum or minimum of a real-valued function over a continuous domain. They also test an algorithms ability to avoid local optima and adapt to multimodal landscapes (De Jong). We shall test all models on the following continuous function optimization tests and measure performance based on mean fitness scores:

#### 3.1.1 Sphere

This is a uni-modal function that simply tests the algorithms ability to converge on a global optima. It is a smooth convex function $f(x) = \sum_{i=1}^{n} x_i^2$ proposed by De Jong.

#### 3.1.2 Rosenbrock

A more complex function that offers a narrow valley quality. This means that the global optima is difficult to find and tests a models ability to converge efficiently. The banana-shaped function is $f(x) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ proposed by Rosenbrock.

#### 3.1.3 Rastrigin

Rastrigin is a highly multimodal function that tests a models ability to ignore local optima and perform global optimization. The function is defined as $f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10cos(2\pi x_i)]$ proposed by Rastrigin.

### 3.2 Feature Selection

Feature selection ties to the reduction of the feature spaces of datasets while still maintaining high predictive accuracy. The performance is measured by the fitness equation: $Fitness = \alpha \times (1 - Accuracy) + (1 - \alpha) \times \frac{|S|}{|F|}$ where $|S|$ is the number of selected features while $|F|$ is the original number of features and $\alpha$ is a weighting parameter that balanced the two objectives of classification accuracy and subset size, respectively. We shall test our models on the following settings:

#### 3.2.1 Iris

The Iris dataset is a simple baseline dataset with four features and three classes proposed by Fisher. This baseline is just to test how well each model can handle basic feature discrimination.

#### 3.2.2 Breast Cancer Wisconsin

This dataset is utilized in a wide variety of applications including neural networks, reinforcement learning, and decision trees. It is composed of thirty numerical features describing symptoms or conditions of tumors with a final binary classification of either benign or malignant. Proposed by Street et al., this dataset will allow us to measure the accuracy and robustness of our models.

#### 3.2.3 Sonar

A high dimensional and noisy dataset composed of sixty frequency features, Sonar presents a difficult challenge to measure the exploration and feature reduction capabilities of our models and was proposed by Gorman and Senjowski.

### 3.3 Scheduling Tasks

Scheduling problems are those where models are given tasks that need resources attached to them over time. The overal goal is to minimize total task time but additional goals may include satisfying prerequisites or other constraints. They test the global search and constraint-handling capabilities of swarm based algorithms. Since computer runtime is not a reliable metric, we shall use makespan to measure average job completion time. Makespan is the sum of total job time with respect to each job's prespecified duration. The benchmarks we wish to use are as follows:

#### 3.3.1 Job Shop Scheduling Problem (JSSP)

This baseline benchmark allocates a set of tasks that must be completed on a specific set of machines in a specified sequence while minimizing total expended time. Proposed by Lawler et al., this benchmark tests basic scheduling tasks optimization.

#### 3.3.2 Flexible Job Shop Scheduling (FJSP)

This is an extension of the previously introduced JSSP with loosening on the machine specific constraint. Now tasks may be completed by a larger set of machines rather than exactly one. Proposed by Brandimarte, this setting allows us to see how well each model handles additional flexibility.

#### 3.3.3 Resource-Constrained Project Scheduling problem (RCPSP)

Another extension of the JSSP but by changing the type of resource from machines to renewable resources. These include workers, equipment, money, space, etc. Proposed by Kolisch and Sprecher, this benchmark tests adaptability of models.

### 3.4 Statistical tests

To answer our hypothesis on whether hybrid GA-Boids models are more efficient than PSOs and GAs, we need statistical evidence. For this, we shall use the Friedman test proposed by J.Demšar which is a non-parametric statistical test to compare our three models on multiple benchmark functions. It works based on ranks and does not expect a standardized distribution. Additional details on this test have been omitted for space sake.

## References

T. Blickle and L. Thiele. A comparison of selection schemes used in evolutionary algorithms. *Evolutionary Computation*, 4(4):361–394, 1995.

M. R. Bonyadi and Z. Michalewicz. Particle swarm optimization for single objective contineous space problems: A review. *Evolution Computation*, 1(25):1–54, 2017.

P. Brandimarte. Routing and scheduling in a flexible job shop. *International Journal of Production Research*, 31(10):2901–2918, 1993.

K. A. De Jong. An analysis of the behavior of a class of genetic adaptive systems. *Ph.D. Dissertation, University of Michigan*, 1975.

A. E. Eiben and J. E. Smith. Introduction to evolutionary computing. *Springer*, 2003.

R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

D. E. Goldberg. Genetic algorithms in search, optimization, and machine learning. *Addison-Wesley*, 1989.

R. P. Gorman and T. J. Senjowski. Analysis of hidden units in a layered network trained to classify sonar targets. *Neural Networks*, 1(1):75–89, 1988.

R. L. Haupt and S. E. Haupt. Practical genetic algorithms. *Wiley*, 2004.

J. H. Holland. Adaptation in natural and artificial systems. *University of Michigan Press*, 1975.

J.Demšar. Statistical comparisons of classifiers over multiple datasets. *Journal of Machine Learning Research*, 7:1–30, 2006.

J. Kennedy and R. Eberhart. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, (4):1942–1948, 1995.

R. Kolisch and A. Sprecher. Psplib - a project scheduling problem library. *European Journal of Operational Research*, 96(1):205–216, 1996.

E. L. Lawler, X, and X. The traveling salesman problem: A guided tour of combinatorial optimization. *Wiley*, 1993.

S. W. Mahfoud. Niching methods for genetic algorithms. *Ph.D. Dissertation, University of Illinois*, 1995.

Z. Michalewicz. Genetic algorithms + data structures = evolutionary programs. *Springer*, 1996.

L. A. Rastrigin. Systems of external control. *Mir, Moscow*, 1974.

C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, pages 25–34, 1987.

H. Rosenbrock. An automatic method of finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, 1960.

Y. Shi and R. C. Eberhart. A modified particle swarm optimizer. *Proceedings of IEEE International Conference on Evolutionary Computation*, pages 69–73, 1998.

W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.

W. N. Street, X, and X. Nuclear feature extraction for breast tumor diagnosis. *Analytical and Quantitative Cytology and Histology*, 15(2):77–87, 1993.

E. G. Talbi. A taxonomy of hybrid metaheuristics. *Journal of Heuristics*, 8(5):541–564, 2002.

D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.