**Effects of combining Particle Swarm Optimization with Genetic Algorithms to Minimize Deceptive Cases**

**Daniel N. Khalil**

*Computer Science*

*Artificial Intelligence Club*

Daytona Beach, Florida

600 South Clyde Morris Blvd.

**Luke R. Crump**

*Software Engineering*

*Artificial Intelligence Club*

Daytona Beach, Florida

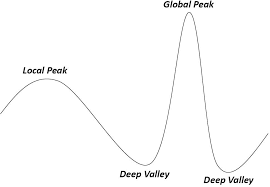
600 South Clyde Morris Blvd.

**Abstract**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam iaculis orci eget ex placerat congue. In hac habitasse platea dictumst. Praesent sed scelerisque elit, a fermentum velit. Morbi ultrices ipsum vitae placerat maximus. Donec a ultrices lacus. Maecenas non dignissim nibh, in ullamcorper est. Nunc commodo malesuada leo, sed blandit est dictum vel. Phasellus lobortis pharetra magna non molestie. Sed eu gravida leo.

**Introduction**

Heuristic optimization techniques have become a massive part of machine learning. Whether it be updating weights or topologies of Neural Networks or approximating solutions to NP-Complete problems, Heuristic evolutionary optimization algorithms, like genetic algorithms and particle swarm optimization, are handy. Although most heuristic optimization algorithm’s drawback is that in searching for an optimal solution, an inaccurate solution can be converged upon because the algorithm ran on a deceptive solution space. This problem can cause unwanted behaviors in machine learning or undesirable solutions.

****

Our goal in writing this paper is to test whether hybridizing the heuristic evolutionary optimization algorithm’s Genetic Algorithm and Particle Swarm Optimization together may have a positive effect on convergence speed, and accuracy in searching through deceptive solution spaces.

**Definitions**

**Heuristic:** a problem dependent algorithm that is supposedly faster and more efficient than traditional methods by sacrificing accuracy, or completeness.

**Metaheuristic**: a high-level problem independent algorithm that uses heuristic methods to allow application to a broad variety of problems.

**Optimization**: the process of finding the best solution or solutions out of a range of possible solutions

**Solution** **Space**: the set of all possible solutions to a specific problem

**Search** **Space**: the set of all possible solutions

**Genetic Algorithm**

Genetic Algorithms (GA) created in 1960 by John Holland are an evolutionary metaheuristic optimization algorithm based on Darwin's theory of evolution and survival of the fittest.

The Algorithm initially constructs a population of random solutions in a solution space, encodes them, then will run through the following steps:

1. evaluate solutions through defining some fitness function
2. select the solutions based off of those evaluations
3. breed the solutions together using one of an assortment of crossover methods
4. mutating each solution based on a defined mutation rate

Repeating these steps until an optimal solution has been found, Genetic Algorithms eventually converges the entire population to a solution inside the solution space.

The critical aspects of how GAs find a solution is in how they explore and converge. Exploration is affected by the selection and mutation methods of GA and is essential to help the population not get stuck in deceptive cases. Convergence is affected by the selection and crossover methods of GA and is essential to moving the entire population towards a specific solution. In some cases, if these deceptive cases contain too much noise, GAs would have a much lower chance of converging to the most optimal solution.

**Particle Swarm Optimization**

Particle Swarm Optimization (PSO) is a metaheuristic optimization algorithm based off of the social behavior of birds. PSO instead of using the practice of selection and crossover, uses a mathematical vector equation that uses the position of the specific particle, the position of the best particle, and randomness to explore and converge on a solution

The algorithm initially constructs a population of particles that contain a random solution between the upper and lower bounds of the solution space, assigns each particle a random velocity vector **vi** within the bounds of the solution space. Then runs through these steps.

ω – inertia parameter (normally between 0.6-0.9)

φp – social component parameter (normally 2)

φg – cognitive component parameter (normally = φp = 2)

r – random number between 0-1

1. Evaluates the particle’s solution: **x**i
2. If the particle’s solution is better, then it’s personal best it updates that particles personal best: **p**i
3. If the particle’s solution is the best solution out of the entire swarm then the global best solution is updated **g**d
4. Updates the particle’s velocity using  
   **v**i,d ← ω **v**i,d + φp *r*p (**p**i,d-**x**i,d) + φg *r*g (**g**d-**x**i,d)
5. Updates the particle’s position  
   **p**i ← **p**i+ **v**i,d

These steps repeat for each particle for each epoch. To promote convergence the velocity vector equation uses the social component to move each of the particles towards the global best solution, to promote exploration of the solution space the random parameter **r** is multiplied to each component of the equation, as well as the inertia parameter ω.

**Hybridization**

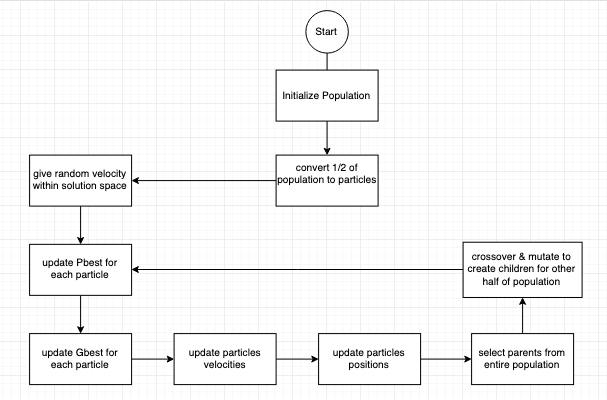
To test whether the hybridization of PSO and GA is viable. GA, and PSO will be tested against 3 different hybrid algorithms. If these algorithms are awarded a higher fitness, in a shorter amount of iterations it will be accepted that these algorithms perform better than the nonhybrid versions of GA or PSO.

**H1: Tandem GAPSO**

The Tandem GAPSO Algorithm will first construct a population of random solutions within the bounds of a solution space. Then encapsulate 1/2 of those individual solutions within a particle object that contains a pBest and gBest as well as a velocity vector.

1. pBest updates based on the current particles position.
2. gBest updates based off of the entire population’s global best solution.
3. Particles update velocities of the first ½ of the population
4. Particles update position the first ½ of the population
5. GA will select parents from the entire population
6. GA performs crossover and mutation to fill up the other ½ of the population with children.

These steps will repeat to converge to a solution.



**Test Cases**

**Results**

**Conclusion**

**References**

PSO-GSA (Particle Swarm Optimization, Gravitational Search Algorithm Hybrid

<https://ieeexplore.ieee.org/abstract/document/6141614>

Gene selection in cancer classification using PSO/SVM and GA/SVM hybrid algorithms

<https://ieeexplore.ieee.org/abstract/document/4424483>

GA and a novel PSO-GA-based hybrid algorithm

<https://www.sciencedirect.com/science/article/pii/S0020019004003254>

Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification

<https://ieeexplore.ieee.org/document/8458226>

How efficient are genetic algorithms to solve high epistasis deceptive problems?  
<https://ieeexplore.ieee.org/abstract/document/4630806>