Asset Allocation with Swarm/Human Blended Intelligence

Lauren Christopher, William Boler, Calvin Wieczorek,
Jonah Crespo, Paul Witcher
Purdue School of Engineering and Technology
IUPUI
Indianapolis, IN, USA
lauchris@iupui.edu

Scot A. Hawkins, James Stewart
Naval Surface Warfare Center, Crane Division
Crane, IN, USA

***Abstract*— PSO has been used to demonstrate the near-real-time optimization of frequency allocations and spatial positions for receiver assets in highly complex Electronic Warfare (EW) environments. The PSO algorithm computes optimal or near-optimal solutions so rapidly that multiple assets can be exploited in real-time and re-optimized on the fly as the situation changes. The allocation of assets in 3D space requires a blend of human intelligence and computational optimization. This paper advances the research on the tough problem of how humans interface to the swarm for directing the solution. The human intelligence places new pheromone-inspired spheres of influence to direct the final solution. The swarm can then react to the new input from the human intelligence. Our results indicate that this method can maintain the speed goal of less than 1 second, even with multiple spheres of pheromone influence in the solution space.**

Keywords—human in the swarm; blended intelligence, pheromones; particle swarm optimization; terrain; 3D graphics

I. Introduction

This paper introduces a method for optimizing the placement of limited receiver assets in 3D space, utilizing particle swarm optimization (PSO). The transmitters and receivers are used to simulate an Electronic Warfare (EW) battlefield where the optimized limited receiver assets need to be spatially positioned and frequency agile to enable either communication to, or jamming of, the transmitters. A graphical user interface (GUI) is developed to illustrate the placement of receivers in relation to randomized transmitter locations in regards to signal strength, communication priority, receiver placement spread, distance from the transmitter cluster, placement outside of keep away zones, and attraction to or deflection from pheromone zones. The PSO is also used to optimize the allocation of frequencies to receiver assets, which will not be covered in this paper.

This work is a continuation of [7], where a baseline 2D GUI was implemented to optimize receiver asset placements based on signal strength, communication priority, receiver placement spread, and distance from the transmitter cluster. This extension introduces the use of the Qt Data Visualization package to present transmitter and receiver placement against a 3D scatter plot and a 3D surface plot using real terrain elevation data from NASA's Shuttle Radar Topography Mission (SRTM). New pheromone and keep away zones in 3D are introduced, providing the capability for a user to establish zones that either attract or repel receiver placements. Developing the fitness functions for the pheromone zones is ongoing and examined in this paper.

II. Particle Swarm Optimization

A. *PSO Function*

The use of optimization is appealing in the sense of providing a series of solutions for a decision problem. Certain problems, such as "travelling salesman" problems, require an exhaustive search through every possible combination of parameters or a heuristic to find the best solution and is not expected to be calculated in deterministic polynomial time. This computation grows exponentially with increasing number of parameters; and the problem space itself can be highly nonlinear. Optimization algorithms, such as gradient descent, can provide accurate estimations of solutions to the decision problems, but have the pitfall of converging on a neighborhood best solution and possibly miss a global best solution.

PSO is a form of evolutionary computation which provides the capability to decrease the time required to calculate both neighborhood best and global best solutions to decision problems. PSO is based on the swarm behavior of social groups, such as schools of fish or flocks of birds, hunting for food and avoiding predators. Particles are randomized in placement and velocity, and are flown in N-dimensional hyperspace, where N is the number of parameters to be optimized. The particle, neighborhood, or global bests among each particle can be applied to a weighted fitness function of the PSO algorithm for each iteration. The PSO algorithm uses a simple weighted inertia and velocity to fly the particles through hyperspace for each iteration and converges on neighborhood or global best solutions [1].

B. *Fitness Function*

PSO is dependent on the complexity of the weighted fitness function used to determine optimal fitness of a solution. The fitness function does not need to be a direct calculation of an exact solution to a given component of the problem, but does need to be representative of a continuing trend which scores the general goodness of the component in hyperspace [4]. A linear combination of weighted fitness functions for each component can be used to generate an overall fitness, which is further used

This research is sponsored by the Naval Engineering Education Consortium (NAVSEA) contract N00174-16-C-0024, with support of NSWC Crane Division in Crane, Indiana.

978-1-5090-3502-1/16/$31.00 ©2016 IEEE

to calculate the goodness of the particular solution. Equation (1) illustrates this, where F is the overall fitness, C are the fitness components, and W are the weights [6].

$$F = \sum_{i \epsilon C} C_{(i)} \times W_{(i)} \quad (1)$$

Weights are attributed to each fitness component in an attempt to balance the overall fitness of the solution. Improperly weighted fitness components will result in the dominance of fitness components, which becomes apparent in the comparison of nonlinear fitness functions. In the case of nonlinear fitness functions, linearity can be assumed for a given range or a linear approximation may be used. Weights are used to reduce the dominance of the nonlinear fitness functions and require a balance to ensure that components do not fall out of influence. Methods are used to gather statistics, such as mean, variance and standard deviation, on the output of fitness components, yet much of the weights are selected through trial and error. Optimal weights are determined based on minimal variance across repeated solution calculations. Our previous paper [7] discusses the fitness components implemented for priority, power, spread, and distance. We will continue the discussion with the contributions made to terrain adjustments and human in the swarm.

III. Human in the Swarm

Human in the Swarm allows a user to input restrictions and modifications between iterations of optimized solutions. These restrictions and modifications can be changed to allow dynamic motivation of assets in regards to the final optimization. These restrictions may be areas where it is desirable to **Keep Away** from or **Stay In** zones for assets to remain. We consider two types of human in the swarm implementations during the optimization process: “keep away” zones and pheromone placements.

A. Keep Away Region

Keep away zones are simple penalties that are applied to a solution that crosses into the region, covered in [7]. There are two types of zones: linear and circular. The linear boundary is simply a line that we do not want the assets to cross. This implementation could be an example of a line between friendlies and enemies; we don’t want the assets to go across this line into enemy territory. Linear boundaries are determined by defining the x-axis position for the line to be drawn.

Similar to the linear boundary, the circular boundary is circular region that we want the assets to stay out of. This allows for the assets to keep a defined distance away from the central area, where the transmitters are randomly populated. The circular region is defined by a radius and is drawn at the origin of the plot.

For each receiver found within the keep away zone, the overall fitness is multiplied by 0.5. This ensures that a strict penalty is placed on keep away regions. The penalty is calculated by first determining the type of boundary. If the boundary is linear, the penalty is determined by verifying that the location of the particle is less than the “enemy line”. If the boundary is circular, the penalty is determined by verifying that the distance from the origin is less than the radius of influence.

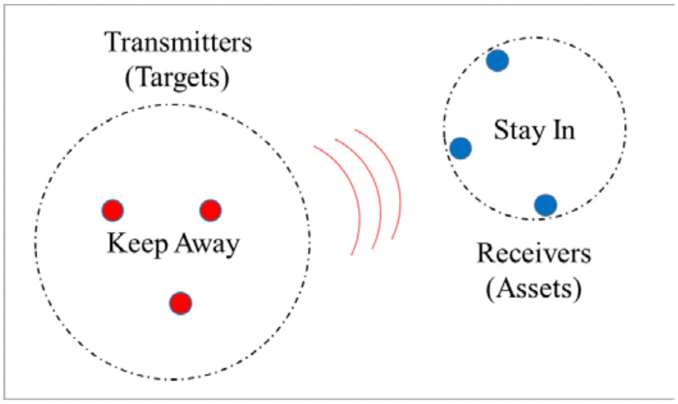
Preventing the assets from staying above the terrain is handled by using a large keep away zone. Solutions that generate assets below the elevation of the terrain matrix at the (x,y) coordinate will be heavily penalized by a scale of 0.5 for each particle found beneath the zone. This also allows the use of setting upper elevation limits for flying assets, by asserting that assets do not pass below a certain elevation above ground.

Although these zones are shown to work, their implementation is limited. Only one zone can be drawn at a time by the human and the placement of the zones is restricted to specific directions. The next section describes the treatment of pheromone zones, which uses a similar penalty system to determine keep away and stay in zones.

B. Pheromones

Pheromones are chemical secretions made by members of the same species that are used to communicate a social response, such as sex, danger, territory, or food. Typically, the animal or insect, such as an ant, will secrete a “good” pheromone in the finding of food, or a “bad” pheromone in a dangerous situation. The “good” pheromones would attract ants to follow a trail, whereas the “bad” pheromones would alert the ants to avoid or attack an enemy. Over time, pheromones will evaporate, allowing cleanup of expired signals and requiring the continuous spread by society members to provide a network of good areas and bad areas.

Figure 1 Concept of human-defined zones



Our new research adopts the concept of pheromones as a means to guide the receiver assets by the human by dropping pheromone zones in 3D space. Fig. 1 illustrates this concept. The user has the capability to mark the cylindrical pheromone as either “attract” or “avoid”. By specifying “attract”, the assets will attempt to remain in the applied zone; by “avoid”, the assets will attempt to avoid the applied zone. The user can specify the type of zone as either “direct” or “radiant”. “Direct zones” lure the assets towards or away from the zones, but once inside or outside the radius of influence, a penalty is no longer applied. This specification is optimal for a situation where placement anywhere in the zone is deemed safe or dangerous, with no consideration for the specific position inside the zone. “Radiant” pheromones provide “beacons” that are continuous penalties applied to the assets based on distance from the center of the pheromone and do not take a radius of influence into consideration. Rather, radiant pheromones rely on the strength of the pheromone to dictate how strongly the receivers should move towards or away from the origin of ambience, determined by strength levels defined by the user.

Figure 2 Optimized solution with a “direct” 25 km radius pheromone placed at an expected optimization point as a “keep away” zone (yellow sphere).

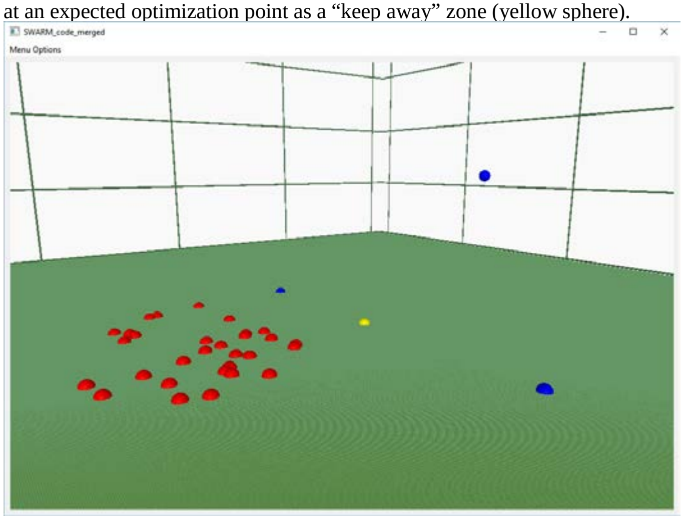


Fig. 2 illustrates the 3D output of a solution with the use of a “direct” 25 km radius pheromone set as a “keep away” zone. The radius of influence is not plotted in this 3D plot and will be available in future versions of the program. The yellow sphere represents the placed pheromone, the red spheres indicate the randomized placement of the transmitters and the blue spheres indicate the optimized solution of the receiver assets. The expected optimization without the pheromone would be the three blue receivers formed as a line or orthogonal pyramid, rather than a slanted pyramid. The top blue transceiver is fighting to stay out of the “keep away” zone.

Figure 3 Optimized solution with a “direct” 5 km radius pheromone placed at an expected optimization point as a “stay in” zone.

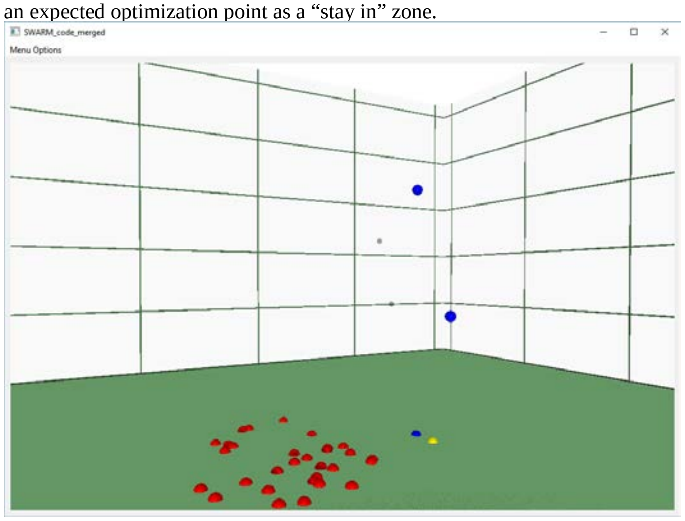


Fig. 3 illustrates the same pheromone being defined as a “stay in” zone and a reduction of radius to 5 km. It is shown that the receiver placements are currently confined within the 5 km radius of the pheromone placement and are vertically spreading to maintain a proper spread fitness.

C. Fitness Calculation Method

Two methods are being tested for the optimal fitness calculations and penalty assignments. The first method uses similar methods as the “keep away” zones by multiplying a penalty to the overall fitness. The second method is to apply the fitness of the pheromone regions as a weighted linear component of the overall fitness. Each method is under continuing research to validate the proper application of the pheromone fitness.

$$F_{Pheromone} = \beta \prod_{p=1}^{P} \prod_{a=1}^{A} f_1(p,a) \qquad (2)$$

Application of a stiff penalty by the pheromones through multiplication showed that receivers would either stay in or stay away from their respected defined zones, as shown in (2). The function f_1 indicates the penalty to be multiplied across P pheromones for A assets and β weight. Penalty infliction was a good method for ensuring that areas marked as dangerous would be avoided, but the zones dominated other fitness components such as spread, power, and priority. One example is when too small of a pheromone radius is used, the small radius produced a cluster of tightly packed receivers. Further tuning of the penalty to assign, as well as the weight of the component, will need to be made.

$$F_{Pheromone} = \beta \sum_{p=1}^{P} \sum_{a=1}^{A} f_2(p,a) \qquad (3)$$

In the application of the weighted linear component, the fitness of the pheromones is dependent on a sum of functions across all receivers and all pheromone zones. So far, this method has shown to be more ideal than penalty infliction. Equation (3) provides the fitness component f_2 for P total pheromones across A total assets, with β weight applied. Each function is dependent on the strength, distance, radius of influence, and type of pheromone zone. For each function, it is determined if the pheromone is attracting or deflecting, as well as the distance from the center of the pheromone. Figures 2 and 3 were generated based off of (3).

D. Direct Pheromones

For “direct” pheromones, the distance of the receiver is compared to the radius of influence for that particular pheromone. If the receiver is inside the radius for attracting or outside the radius for deflecting, the function will add the strength of the pheromone, which is normally a value of 1, to the fitness component. If the receiver is outside the radius for attracting or inside the radius for deflecting, the function will add a reduced fitness to the fitness component scaled by a second order polynomial function of distance. This allows for a penalty to be applied for assets located on the wrong side of the radius, and constant fitness for assets on the correct side of the radius. The coefficients for the second order polynomial equation are calculated to ensure that there is a steady decrease from the radius edge to the maximum distance from 1 to 0 on attracting pheromones, and a decrease from the radius edge to the center on deflecting pheromones. The penalty for deflecting is shown in (4) and the coefficients for the polynomial are shown in (5), (6) and (7). The strength is the individual strength assigned to the pheromone. The penalty applies only to the polynomial and is used to scale it down between 0.25 and 0. D is the distance from the radius to the asset. D_{max} for attracting is the expected maximum distance at $\frac{200}{\sqrt{2}} \cong 283\ km$ and for deflecting is the radius. The direct attracting function negates the polynomial portion and sets c to 0 to reverse the effect. For any value that falls outside of D_{max}, the fitness is floored to 0.

$$F_{Direct} = strength \times penalty \times (aD^2 + bD + c) \quad (4)$$

$$a = -\frac{(1+b\times D_{max})}{D_{max}^2} \quad (5)$$

$$\text{b} = -\frac{2}{D_{max}} \quad (6)$$

$$\text{c} = 1 \quad (7)$$

E. Radiant Pheromones

For "radiant" pheromones, several functions were considered for measuring the fitness of a particular solution. Initially, an inverse relation of the distance was considered for either applying a penalty or increasing the fitness. This resulted in the fitness function becoming unstable and difficult to manage, particularly for multiple numbers of pheromones. An alternative was to model the distance based on the step response of a first order system, which would nominally produce fitness values between 0 and 1 within the 200 x 200 km^2 space. For the attracting pheromone, the fitness value in (8) would gradually reduce from 1 to 0 as the distance increases. For the deflecting pheromone, the fitness value in (9) would gradually increase from 0 to 1 as the distance increases.

$$F_{Radiant,Attract} = e^{-\frac{x}{\tau}} \quad (8)$$

$$F_{Radiant,Deflect} = 1 - e^{-\frac{x}{\tau}} \quad (9)$$

Equations 8 and 9 show the theoretical fitness functions for the radiant pheromones, where x is the positive distance from the asset to the pheromone and τ is a time constant equal to 5 times the maximum distance. Maximum distance is approximated at 283 km for the diagonal across the x-y plane, so the function has a limited range of consideration.

Due to the computational complexity of implementing a sum of the exponential functions across all pheromones and receivers, the time intensity of this sort of function would result in diminished performance of the program. As a trade-off between computational complexity and precision, a 4th order polynomial approximation of the step response resulted in an ideal fitness function for the "radiant" pheromones. Equation 10 provides the appropriate formula used for evaluating the fitness of the distance between the pheromones and assets, where the coefficients differ depending on attracting or deflecting.

$$F_{Radient} = strength \times [ax^4 + bx^3 + cx^2 + dx + e] \quad (10)$$

Due to the modelling of an arbitrary step response, the fitness of the "radiant" pheromones can be summed quickly without introducing an imbalance in the overall fitness function. The result of the fitness component for the "radiant" pheromones can be divided by the number of pheromones and assets to ensure that the result is a normalized value between 0 and 1. Furthermore, a strength can be applied to each pheromone to indicate if certain pheromones are to provide a stronger or weaker influence.

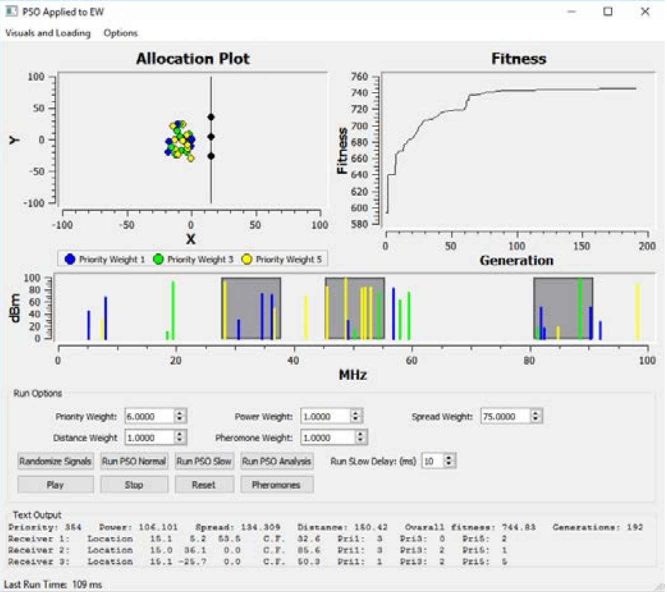
IV. Graphical User Interface

A GUI interface was constructed using Qt, a cross-platform framework for User Interface development in C++. There is a 2D and 3D representation of the PSO data. The 2D plot shows locations of the receivers and transmitters in a restricted 200 x 200 km^2 space, the fitness of our function over generations, and a display of signals in the frequency domain. The 3D plot, as shown in figures 2 and 3, illustrate a 200 x 200 x 20 km^3 space of the same receivers and transmitters.

A. 2D Interface

The 2D Interface has 4 main points of interest: Allocation Plot, Fitness, Frequency Graph, and Run Options.

Figure 4 2D GUI



Priority Weight: 6.0000
Power Weight: 1.0000
Spread Weight: 75.0000
Distance Weight 1.0000
Pheromone Weight: 1.0000
Run SLow Delay: (ms) 10

```
Priority: 364   Power: 106.101   Spread: 134.309   Distance: 160.42   Overall fitness: 744.83   Generations: 192
Receiver 1:   Location   15.1   5.2  53.5   C.F.  32.6   Pri1:  3   Pri3:  0   Pri5:  2
Receiver 2:   Location   15.0  36.1   0.0   C.F.  85.6   Pri1:  3   Pri3:  2   Pri5:  1
Receiver 3:   Location   15.1 -25.7   0.0   C.F.  50.3   Pri1:  1   Pri3:  2   Pri5:  5
```

Last Run Time: 109 ms

The representation of the 2D GUI (Figure 4) provides the user feedback on location of receivers/assets, fitness value, frequency representation of the assets, adjust parameter values, move around transceivers with the Play and Stop buttons, modify pheromones locations and properties with the pheromones button, and give text feedback in the Text Output section. The GUI also has two menu options: "Visual and Loading" and "Options". The Visual and Loading menu allows the user to load terrain values and display 3D surface and point graphs of the receivers/transmitters. The Options menu allows the user to change various initialization parameters.

The Allocation Plot is a representation of the assets, transceivers, and Keep Away Penalties and updates dynamically when the PSO is running. Each transmitter also has a weight priority, where the higher priority transmitters (yellow 5) are more important than the lower priority transmitters (blue 1). The Fitness graph shows the current fitness of our PSO function, as well as the number of generations the fitness function iterated during optimization. The Frequency Graphs display the transceiver frequencies (in MHz) and power (in dBm). It highlights bands of frequencies that the assets optimized to along with the transmitters and receivers associated. Run Options allows the user to adjust weights, randomized the signals, use Play to simulate random movement of the transceivers, and gives options for giving properties and properties to pheromones.

B. Migration from 2D to 3D

This project originally contained a 2D GUI. Modifications were made to move the project to a 3D space. This was accomplish through the use of Qt's Data Visualization's Plugin. This plugin allowed for the plotting of 3D objects using 3D coordinates. The ability to import terrain data was added to allow the plotting of the data as a 3D surface. Receivers, transmitters, and pheromones were loaded as custom mesh objects to the surface plot. The user is able to load new terrain files and change perspectives of the 3D plots.

V. RESULTS

The Swarm/Human Blended program described in this paper convergences in approximately one second with three (3) receivers and thirty (30) transmitters without compiler optimizations. With the introduction of pheromones, this convergence takes a little over a second and increases over added pheromones and receivers. With full compiler optimizations set by compiler flags, the time of convergence is reduced to several hundred milliseconds.

VI. CONCLUSIONS AND FUTURE WORK

This research has advanced the human / swarm blended intelligence using the pheromone concept, allowing 3D human directed optimization of the swarm. The PSO algorithm continues to converge rapidly (less than 1 second) after the human input. The combination of **Keep Away** and **Stay In** zones for assets allows for flexibility in the human interface, without a heavy burden on the human.

The future work will include refining and testing the pheromone zones, exploring various human input techniques in 3D, and putting the transmitters and receiver assets in motion, and interacting over time. Parallelization of the fitness functions will also be evaluated for enhancing the performance of the optimization.

ACKNOWLEDGMENT

The authors would like to acknowledge the support from the Expeditionary Electronic Warfare Division, Spectrum Warfare Systems Department and Joshua Reynolds for his contributions. This research is sponsored by the US Navy (Office of Naval Research) grant N00174, with the support of NSWC-Crane Division in Crane, Indiana.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on,, 1995.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in Neural Networks, 1995. Proceedings, IEEE International Conference on, 1995.

[3] R. C. Eberhart, P. K. Simpson and R. W. Dobbins, Computational Intelligence PC tools, 1st ed., Boston, MA: Academic press professional, 1996.

[4] R. C. Eberhart and Y. Shi, Computational Intelligence: Concepts to Implementations, San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 2007.

[5] F. Johansson and G. Falkman, "Real-time allocation of defensive resources to rockets, artillery, and mortars," in Information Fusion (FUSION), 2010 13th Conference on, 2010.

[6] J. Reynolds, L. Christopher, R. Eberhart and P. Shaffer, "Using computational swarm intelligence for real-time asset allocation," in Computational Intelligence for Security and Defense Applications (CISDA), 2015 IEEE Symposium on, Verona, New York, 2015.

[7] L. Christopher, J. Reynolds, J. Crespo, R. Eberhart and P. Shaffer, “Human Fitness Functions,” in Swarm/Human Blended Intelligence Workshop (SHBI), 2015, Cleveland, OH, 2015, pp. 1-4.

[8] P. Walker, S. Nunnally, M. Lewis, A. Kolling, N. Chakraborty and K. Sycara, "Neglect benevolence in human-swarm interaction with communication latency," in Swarm, Evolutionary, and Memetic Computing, Berlin Heidelberg, Germany, Springer, 2012, pp. 662-669.

[9] A. Kolling, K. Sycara, S. Nunnally and M. Lewis, "Human swarm interaction: an experimental study of two types of interaction with foraging swarms," Journal of Human-Robot Interaction, vol. 1, no. 1, pp. 78-95, 2013

[10] J. Reynolds, “Particle swarm optimization applied to real-time asset allocation,” Purdue University, Indianapolis. 2015.

[11] J. Crespo, “Asset allocation in frequency and in 3 spatial dimensions,” Purdue University, Indianapolis. 2016.