10.1109/ICIINFS.2009.5429852

A New Biologically Inspired Optimization Algorithm

Upeka Premaratne
Department of Electronic and Telecommunication Engineering
University of Moratuwa
Moratuwa, Sri Lanka
Email:upeka@ent.mrt.ac.lk

Jagath Samarabandu, Tarlochan Sidhu
Department of Electrical and Computer Engineering
University of Western Ontario
London, Ontario, Canada
Email:jagath@uwo.ca,sidhu@eng.uwo.ca

***Abstract*—This paper proposes a new biologically inspired algorithm for optimization. The algorithm, called the Paddy Field Algorithm (PFA) operates by initially scattering seeds at random in the parameter space. The number of seeds of each plant depend on the function value such that a plant closer to the optimum solution produces the most seeds. Out of these, depending on the number of neighbors of the plant, only a percentage will become viable due to pollination. In order to prevent getting stuck in local minima, the seeds of each plant are dispersed. This algorithm is tested on four sample functions along side other conventional algorithms. The effect of various parameters on the performance of the algorithm is also investigated. Its performance is also tested with a hybrid algorithm. The results show that the algorithm performs well.**

***Index Terms*—Optimization, biologically inspired optimization algorithm, plant seed dispersion.**

I. Introduction

Optimization is a commonly encountered mathematical problem with a complexity affected by the curse of dimensionality. In order to find the global optimum solution of a particular problem, the one and only method with a guaranteed 100% success rate is the exhaustive search. However, for most problems, the cost of an exhaustive search in terms of time and computational power is prohibitively high. Hence, commonly used optimization algorithms do not search the entire parameter space. This leads to the high possibility of the optimization algorithm thinking that a local optimum is in fact the best solution.

A. Overview of Optimization Algorithms

The Hill Climbing Algorithms (HCA) were amongst the earliest algorithms for optimization and are heavily dependant on the initial values [1]. If the initial value is found in the vicinity of a local solution, the algorithm is most likely to get stuck at this undesired solution. Similarly, such algorithms are also likely to get confused and fail on plains, plateaux or ridges. Methods devised to overcome these issues include stochastic, first choice and random-restart hill climbing where the algorithm starts again if it appears to get stuck in a local solution according to a set of rules; or beam searches where the algorithm starts from multiple states selected according to a particular criteria [2]. Another method devised to overcome this shortcoming is the Simplex or Nelder-Mead Algorithm [3] which uses a N+1 simplex to find the optimum of an N dimensional problem instead of a single point.

Optimization is a common occurrence in nature where the best solution to a problem is the one that survives. Hence, most optimization algorithms are inspired from nature itself. These include algorithms based upon random physical phenomena such as Simulated Annealing (SA) [4]. SA is less likely to reach a local solution but reaches its solution at a considerable cost in terms of time and computation.

Other biologically inspired optimization algorithms include, Evolutionary Algorithms (EA) [5] [6] and Artificial Immune Systems (AIS) which are modeled on the immune system of a vertebrate [7]. Both of these depend on random mutations with crossover in the case of EA's and clonal selection in the case of AIS [8]. There are also swarm optimization algorithms based upon the combination of behaviors of different individuals governed by a set of principles. These include ant colony optimization [9] [10], bee colony optimization [11] and particle swarm optimization [12].

B. Contribution

This paper proposes a new such biologically inspired algorithm which does not involve crossover between individuals as in an evolutionary algorithm nor combined behaviors. It operates on a reproductive principle dependant on proximity to the global solution and population density similar to plant populations and differs from clonal selection [8] where reproduction (hypermutation) depends only on the proximity to the target.

II. The Paddy Field Algorithm

A. Biological Inspiration

When seeds are sown in an uneven field, the seeds that fall into places with the most favorable places (most fertile soil, best drainage, soil moisture etc.) tend to grow to become the best plants. Such plants tend to grow taller and capable of producing more seeds than less fortunate individuals. From this, the tallest plant of the population would correspond to the location with the optimum conditions. In this algorithm, the fitness of the plant is determined by a fitness function (Figure 1).

When it comes to plant reproduction, the other main factor that would affect reproduction is pollination. In the absence of animal pollinators, pollen would be carried by the wind. In such a case, a high population density would increase the chance of pollination by wind. Therefore, the higher population

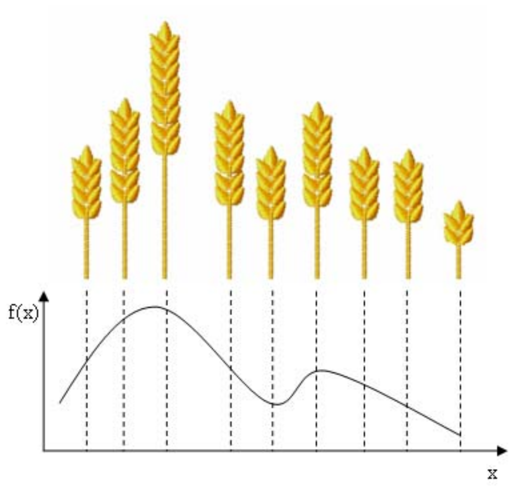

Fig. 1. Plant Fitness

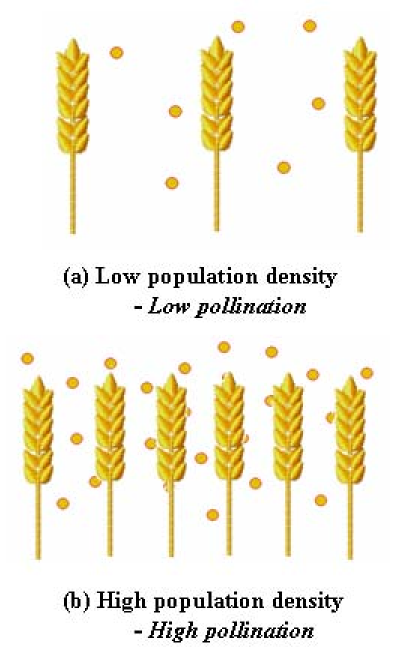

Fig. 2. Plant Pollination

of density of plants the more likely the chance of proper pollination (Figure 2). Thus, the plants producing the most *viable seeds* would be the healthy individuals living in a high population density. The seeds of these plants once scattered will fall on fresh ground and depending on the state of the ground become new plants and continue the cycle.

B. Theory

Let $f(x)$ be an arbitrary function of n variables such that,

$$y = f(\mathbf{x}) \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]$. A particular value of vector $\mathbf{x} = x^i$ is known as a seed or plant. The fitness of the plant would

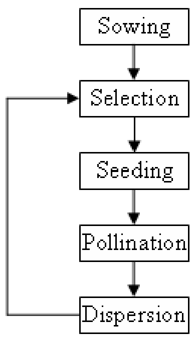

Fig. 3. Stages of the Paddy Field Algorithm

correspond to y. Depending on the nature of the parameter space each dimension of the seed can be bounded such that,

$$x_j \in [(x_j)_{min}, (x_j)_{max}] \tag{2}$$

The Paddy Field Algorithm (PFA) consists of five basic stages (Figure 3).

1) Sowing: In this stage, an initial population of (p_0) seeds are sown at random in the parameter space. In order to ensure optimum dispersion within the parameter space, the values of seed dimensions are uniformly distributed depending on the bounds of the parameter space.

2) Selection: After the seeds are sown into the field and produce plants, the best plants are selected depending on a threshold method. This is to selectively weed out unfavorable solutions. The reason for having a threshold is to ensure that the plant population does not grow explosively.

The threshold operator (H) is used to select a number of individuals from the population for the next iteration. The main purpose of this is to control the population of the system. A plant will be selected to the next iteration only if its fitness is greater than the threshold (y_t).

$$y \geq y_t \tag{3}$$

The threshold used is a constant selection scheme. In it the plants are first ranked for fitness and the n_0^{th} fittest individuals of the population are selected for seeding. This threshold would be given by:

$$\begin{aligned} y_t &= n_0^{th} \text{ fittest individual} \\ &= y_{n_0^{th}} \end{aligned} \tag{4}$$

3) Seeding: In this stage each plant develops a number of seeds proportional to its health. This takes place such that the healthiest plant (best solution) produces the maximum number of seeds (q_{max}). The total number of seeds produced by a plant (s) would be a function (ϕ) of the health of the plant and the maximum number of seeds (q_{max}). Thus:

$$s = \phi[f(\mathbf{x}), q_{max}] \tag{5}$$

The threshold (y_t) can be used to determine the number of seeds of a plant (Equation 5) depending on its fitness in proportion to the fittest plant of the population (y_{max}):

$$s = q_{max}\left[\frac{y - y_t}{y_{max} - y_t}\right] \tag{6}$$

4) Pollination: Out of the seeds produced only a percentage will become viable due to pollination. Pollination depends on the number of neighbors of a particular plant. The percentage of seeds pollinated to become viable seeds will depend on a neighborhood function (U) such that a plant with more neighbors will be better pollinated. Hence, the number of viable seeds per plant would therefore become,

$$S_v = U\phi\left[f(\mathbf{x}), q_{max}\right] \tag{7}$$

The function U should have a range given by,

$$0 \le U \le 1 \tag{8}$$

In order to satisfy this condition, a sphere of radius a is used. For two plants $\mathbf{x}_j$ and $\mathbf{x}_k$, the perimeter formula of,

$$u(\mathbf{x}_j, \mathbf{x}_k) = ||\mathbf{x}_j - \mathbf{x}_k|| - a \tag{9}$$

is used. If the two are within the sphere, then $u < 0$. From this for a particular plant, the number of neighbors (v_j) can be determined. Once this is done, the pollination factor for that plant can be obtained from,

$$U_j = e^{\left[\frac{v_j}{v_{max}} - 1\right]} \tag{10}$$

where v_{max} is the number of neighbors for the plant with the most neighbors within the population. According to this formula, the plant with the maximum number of neighbors will have perfect pollination.

5) Dispersion: The seeds of each plant are then dispersed and the cycle starts again from the selection stage. When dispersing, the dimension values take a Gaussian distribution such that the new seed will land on a location in the parameter space given by,

$$X^{i+1}_{seed} = N(x^i, \sigma) \tag{11}$$

The parameter σ (dispersion spread) will determine the spread of the dispersion. The spreading of seeds within the parameter space will ensure that if the healthiest plant of a particular iteration corresponds to a local solution, the dispersing seeds may fall in the parameter space corresponding to the global optimum. The term generation is avoided for a cycle because the population would consist of plants from the seeds of the current cycle as well as older plants from previous cycles which have survived the selection process. Therefore the term iteration is used. This cycle will continue until the stopping criterion has been reached. This can be a maximum number of iterations, lack of improvement over a specified number of iterations, population reaching a maximum value etc.

III. Comparison of Algorithms

The PFA is tested against other common optimization algorithms for the following four functions (Figure 4) for $x, y \in [0, 1]$,

$$\begin{aligned} f_1(x,y) &= cos^2(n\pi r)\, e^{-\left[\frac{r^2}{\sigma^2}\right]} \\ \text{where} \quad & r = \sqrt{(x-0.5)^2 + (y-0.5)^2} \\ & \sigma = 0.15 \\ & n = 9 \end{aligned} \tag{12}$$

$$\begin{aligned} f_2(x,y) &= 0.80e^{-\left[\frac{r_1^2}{\sigma_1^2}\right]} + 0.88e^{-\left[\frac{r_2^2}{\sigma_2^2}\right]} \\ \text{where} \quad & r_1 = (x-0.5)^2 + (y-0.5)^2 \\ & r_2 = (x-0.6)^2 + (y-0.1)^2 \\ & \sigma_1 = 0.3 \\ & \sigma_2 = 0.03 \end{aligned} \tag{13}$$

$$\begin{aligned} f_3(x,y) &= -a[(x-0.5)^2 + (y-0.5)^2] \\ & -b[cos(2\pi m x) + cos(2\pi m y)] \\ \text{where} \quad & m = 3 \\ & a = 3 \\ & b = 0.3 \end{aligned} \tag{14}$$

$$f_4(x,y) = -(x-0.5)^2 - (x-y^2)^2 + 1 \tag{15}$$

The functions used for testing have different characteristics

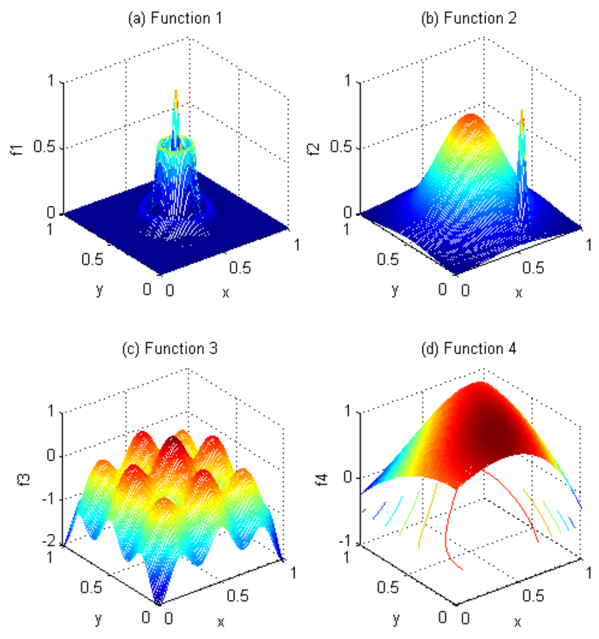

Fig. 4. Test Functions

that challenge the optimization algorithm. Equation 12 has its global maximum at (0.5,0.5) but has a series of local maxima ridges that form concentric circles. Equation 13 has two optimum solutions, the global at (0.6,0.1) and a local maximum at (0.5,0.5). The global maximum is confined to a very small region of the parameter space in order to fool an optimization algorithm. The third function (Equation 14) has eight local maxima and the global maximum at (0.5,0.5). It is an adaptation from the Rastigrin function. The fourth on (Equation 15) is an adaptation from the Rosenbrock function which has a nearly flat region in the vicinity of the global maximum at (0.5,0.25).

Table I shows a comparison of the results of simulated annealing, a genetic algorithm (GA) for different populations and the PFA for different values for the maximum seeds per plant. A total of 121 trials were carried out to evenly distribute the initial values within the sample space according to a 11x11 grid with a spacing of 0.1. This is necessary for algorithms that require an initial value.

For the PFA the remaining parameters are given by,

- *Initial Seeds* = 20
- *Number of Iterations* = 10
- The fittest individuals equal to the number of initial seeds (20) of an iteration are selected.
- *Dispersion Spread (Static)* (σ) = 0.2
- The neighborhood function of Equations 9 and 10 with a radius of 0.02.

TABLE I
RESULTS OF OPTIMIZATION ALGORITHMS

Algorithm	Success Rate (%)			
	f_1	f_2	f_3	f_4
Simulated Annealing	92.56	45.45	100.00	66.94
GA (Population - 20)	56.20	4.96	100.00	90.91
GA (Population - 50)	62.81	5.79	100.00	100.00
GA (Population - 100)	83.47	4.13	100.00	100.00
PFA (Max Seeds - 20)	63.64	25.62	100.00	73.55
PFA (Max Seeds - 50)	80.17	42.98	100.00	99.17
PFA (Max Seeds - 100)	100.00	69.42	100.00	100.00

IV. Parameter Analysis

The next stage of analysis involves the exploration of the effects of each parameter on the performance of the algorithm. For this purpose, 100 trials are conducted while varying the parameter being investigated. Unless otherwise stated, the default values used for each experiment are given by,

- *Initial Seeds* = 10
- *Maximum Seeds per Plant* = 20
- *Number of Iterations* = 10
- The fittest individuals equal to the number of initial seeds of an iteration are selected for the next one.
- *Dispersion Spread (Static)* (σ) = 0.2
- The neighborhood function of Equations 9 and 10 with a radius of 0.02.

A. Number of Initial Seeds

The main effects of increasing the number of initial seeds would be to increase the change of rapid convergence of the algorithm by having a seed fall near the optimum solution. A higher number of initial seeds would translate into a better exploration of the parameter space. Hence, it would improve the performance of the algorithm as shown in Table II. Table III shows the effect it has when the threshold operator selects the n_0 fittest individuals for the next iteration.

TABLE II
EFFECT OF NUMBER OF INITIAL SEEDS ON PERFORMANCE

Initial Seeds	Success Rate (%)			
	f_1	f_2	f_3	f_4
5	45	11	100	47
10	52	10	100	61
20	64	29	100	80
50	79	38	100	96
100	95	64	100	100

TABLE III
EFFECT OF NUMBER OF INITIAL SEEDS ON AVERAGE POPULATION

Initial Seeds	Average Population			
	f_1	f_2	f_3	f_4
5	41.21	53.14	53.57	53.95
10	51.91	81.34	87.92	87.77
20	73.18	129.38	164.66	166.50
50	145.11	236.91	410.79	419.26
100	283.82	304.94	857.19	847.49

B. Maximum Seeds per Plant

The maximum number of seeds per plant will improve the exploration capability of the algorithm by increasing the number of new sampling points in the parameter space. The results of Table IV confirm this. As shown in Table V, it also has an effect of increasing the overall population since a larger number of seeds produced by each plant would add up to a larger population. This parameter is closely coupled with both dispersion and the neighborhood function.

TABLE IV
EFFECT OF MAXIMUM SEEDS PER PLANT ON PERFORMANCE

Max. Seeds	Success Rate (%)			
	f_1	f_2	f_3	f_4
5	23	4	90	100
10	39	5	100	35
20	50	15	100	55
50	76	32	100	93
100	93	48	100	97

TABLE V
EFFECT OF MAXIMUM SEEDS ON POPULATION

Max. Seeds	Population			
	f_1	f_2	f_3	f_4
5	27.84	28.84	27.24	29.04
10	35.14	48.13	48.31	48.10
20	51.11	79.59	93.09	92.94
50	150.67	158.17	234.77	227.57
100	428.89	253.70	485.03	463

C. *Number of Iterations*

Figure 5 shows the general convergence behavior of the algorithm. The system converges within 5 iterations. The system population, initially spikes during this time and then decreases. After convergence the population slowly increases until it stabilizes. By increasing the number of iterations, the system is likely to converge on the global solution even after being initially being stuck in a local solution (Figure 6). In such a case the population appears to oscillate significantly but ultimately converges (Figure 7). In all cases, the average fitness of the population, appears to remain highly static and decrease after convergence.

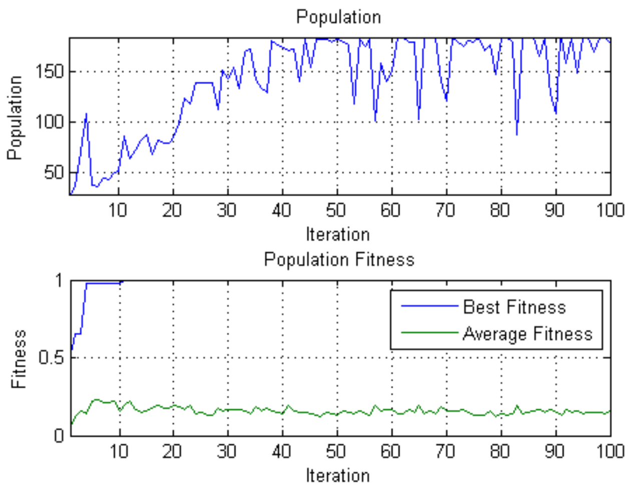

Fig. 5. System Population for Rapid Convergence (100 Iterations)

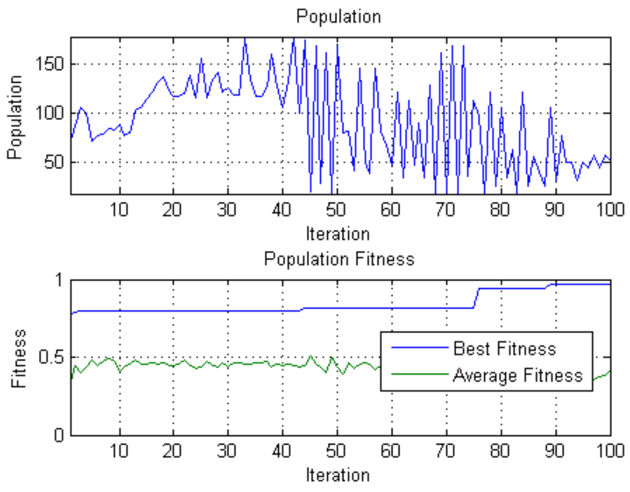

Fig. 6. System Population for Delayed Convergence (100 Iterations)

Table VI gives the performance of the algorithm for different numbers of iterations. It becomes clear that the improved performance by increasing the number of iterations, demonstrates the capability of the algorithm in coming out of a local solution. This is due to the dispersion of new seeds. It is especially apparent in the case of Equation 13 where most algorithms are likely to get stuck in the local solution. However, increasing the number of iterations also makes it equivalent to an exhaustive search.

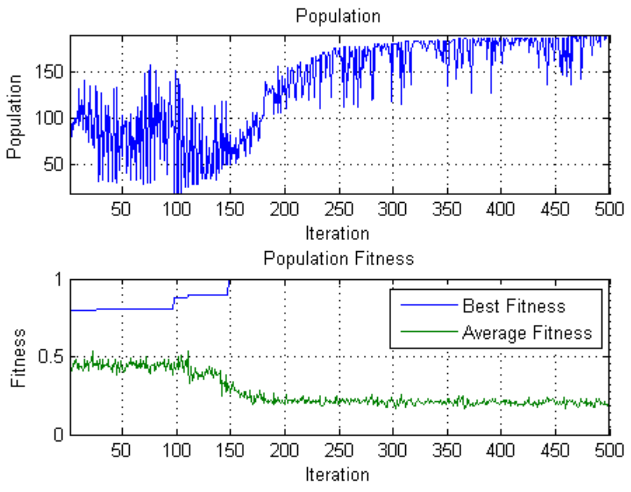

Fig. 7. Population Behavior for Delayed Convergence (500 Iterations)

TABLE VI
EFFECT OF NUMBER OF ITERATIONS ON PERFORMANCE

Iterations	Average Population			
	f_1	f_2	f_3	f_4
5	37	6	100	30
10	55	11	100	52
20	68	24	100	85
50	100	64	100	100
100	100	92	100	100
200	100	100	100	100

The asymptotic behavior would occur when the rate of new individuals created through seeding and pollination (Equation 7) stabilizes with the rate in which individuals are removed by selection threshold operator (H).

$$\sum_{j=1}^{p_i} U_j \phi \left[f(\mathbf{x}_j), q_{max} \right] \approx n_{i-1} - H[n_{i-1}] \qquad (16)$$

V. Hybrid Algorithms

This section investigates the performance of the PFA when coupled with a HCA. It is a common practice to form a hybrid algorithm by coupling a HCA and a stochastic algorithm [13]. Usually stochastic algorithms are good at getting into the region of the optimum solution but not that good in navigating to the exact location of the optimum. Since HCA's require an initial value and are capable of coming to very close to the optimum. Therefore, the approximate solution obtained from the stochastic algorithm can be used to initialize the HCA and reach the optimum. In essence, the strengths of both types algorithms are synergetically combined.

Unlike a GA the PFA does not have any form of crossover, hence the optimum solution is not capable of migrating in a manner similar to that of a GA. Therefore, coupling of a HCA with the PFA will enhance its performance. Table VII shows the results of a hybrid algorithm. The PFA is first run for 5 or 10 iterations, and then either the Nelder-Mead (NM) algorithm or Newton (Nw) algorithm is run. The results show that the performance is improved significantly due to the hybrid algorithm. It also shows that the number of iterations of the PFA can be reduced while still obtaining a satisfactory improvement. It should be noted that for the hybrid algorithm, a successful hit is taken as reaching to within 99.99% of the optimal value.

TABLE VII
PERFORMANCE OF A HYBRID PFA+HCA

Iterations	Algorithm	Average Population			
		f_1	f_2	f_3	f_4
5	PFA only	31	7	100	35
	PFA+NM	89	9	100	100
	PFA+Nw	92	9	100	100
10	PFA only	58	19	100	54
	PFA+NM	100	28	100	100
	PFA+Nw	100	28	100	100

Figure 8 shows a typical plot of the scatter of the results of the PFA (a) algorithm followed the scatter after the PFA results are fed to the NM (b) or Nw (c) algorithms. It clearly reinforces the performance indicated in Table VII.

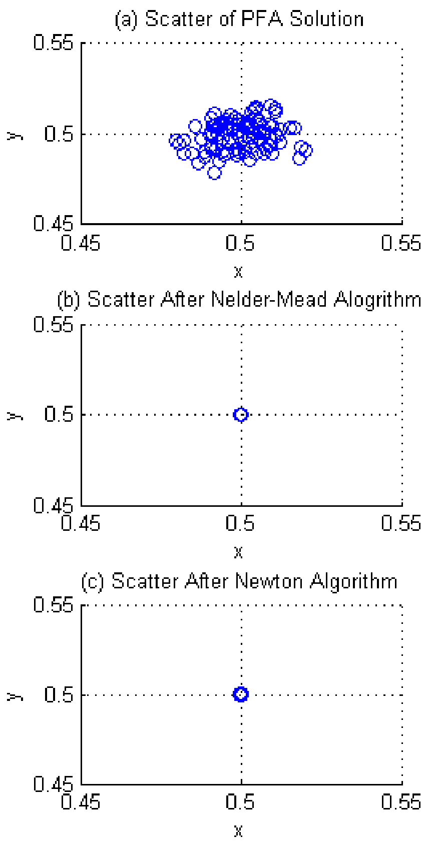

Fig. 8. Scatter Plot of Results of Hybrid Algorithm (10 Iterations)

VI. CONCLUSIONS

An optimization algorithm depends heavily on its initial values and the problem at hand which is a main problem of HCA's [1]. The PFA solves this by selecting random sampling points within the parameter space. It is done in a manner such that the fittest individuals in high population densities produce the most new sampling points. This ensures that the population aggregates towards regions of both globally and locally optimum solutions and finally finds the global optimum. The dispersion of these sampling points within the parameter space ensures that the algorithm is capable of breaking out of a local solution.

The capability of an optimization algorithm is also bound by the no free lunch rule [14]. This can be interpreted as the performance of an algorithm being achieved at a cost in terms of time and computation. The results of the PFA show that its performance is at par to that of SA but it is capable of achieving its results at a much lower computational cost. Since the PFA does not have crossover like that of a GA, the optimum solution is incapable of migrating. Therefore instead of using it inefficiently for final convergence, a HCA can be used to take over and reach the optimum solution once the PFA comes to the vicinity of it combining the strengths of both algorithms.

REFERENCES

[1] A. Pragel-Bennet, "When a genetic algorithm outperforms hill-climbing," *Theoretical Computer Science*, vol. 320, no. 1, pp. 135–153, June 2004.
[2] S. J. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2002.
[3] J. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.
[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
[5] L. J. Fogel, A. J. Owens, and M. J. Walsh, *Artificial Intelligence through Simulated Evolution*. New York: John Wiley, 1966.
[6] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.
[7] J. Farmer, N. Packard, and A. Perelson, "The immune system, adaptation and machine learning," *Physica D*, vol. 2, no. 1-3, pp. 187–204, October-November 1986.
[8] L. N. de Castro and F. J. V. Zuben, "Learning and optimization using the clonal selection principle," *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 3, pp. 239–251, June 2002.
[9] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Milan, Italy, 1992.
[10] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and CyberneticsPart B*, vol. 26, no. 1, pp. 29–41, February 1996.
[11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, 2005.
[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, Piscataway, NJ, 27 November-1 December 1995, pp. 1942–1948.
[13] A. D. J. Cross, R. Myers, and E. R. Hancock, "Convergence of a hill-climbing genetic algorithm for graph matching," *Pattern Recognition*, vol. 33, no. 11, pp. 1863–1880, November 2000.
[14] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, April 1997.