A Synergistic Approach for Evolutionary Optimization

Maumita Bhattacharya
SOBIT, Charles Sturt University
Australia - 2640
maumita.bhattacharya@ieee.org

ABSTRACT
One of the major causes of premature convergence in Evolutionary Algorithm (EA) is loss of population diversity, which pushes the search space to a homogeneous or a near-homogeneous configuration. In particular, this can be a more complicated issue in case of high dimensional complex problem domains. In [13, 14], we presented two novel EA frameworks to curb premature convergence by maintaining constructive diversity in the population. The COMMUNITY_GA or COUNTER_NICHING_GA in [13] uses an informed exploration technique to maintain constructive diversity. In addition to this, the POPULATION_GA model in [14] balances exploration and exploitation using a hierarchical multi-population approach. The current research presents further investigation on the later model which synergistically uses an exploration controlling mechanism through informed genetic operators along with a multi-tier hierarchical dynamic population architecture, which allows initially less fit individuals a fair chance to survive and evolve. Simulations using a set of popular benchmark test functions showed promising results.

Categories and Subject Descriptors
Computing Methodologies [I.2 **Artificial Intelligence**]: I.2.8 Problem Solving, Control Methods, and Search.

General Terms
Algorithms, Design, Performance.

Keywords
Premature convergence, evolutionary algorithm, population.

1. INTRODUCTION AND BACKGROUND
Many real life problems often involve non-linear, high-dimensional, complex search space that may be riddled with many local minima or maxima. Stochastic methods, such as evolutionary algorithms perform a more exhaustive search of the model space as compared to deterministic search techniques. However, they are not as good at exploiting the early results of the search. When dealing with high dimensionality problems, it may be difficult or too time consuming for all the model parameters or variables to converge within a given margin of error. This renders crucial and typical EA problems such as *premature convergence* even harder to deal with. Algorithmic features like *high selection pressure* and *very high gene flow* among population members are primary causes of premature convergence [6]. As the evolutionary search process progresses, population diversity declines, as a canonical GA tends to concentrate more and more of its search effort near the already discovered "peaks" or "attractors", converging towards similar points or even single points in the search space, gradually reaching a near homogeneous state. High *gene flow* on the other hand, pushes the population towards a homogeneous state by spreading genetic material across the population by means of unrestricted mating or crossover. Escaping the local optimum will be difficult, as genetic operators can no longer produce offspring that are even different, leave alone superior, compared to their parents. Maintaining constructive population diversity, besides helping against entrapment in local optima, also leads to increase in exploration, so that a *good* single solution can be found, and also multiple solutions can be located when there is more than one optimum [9].

In case of high dimensional problems, as the number of model parameters increases, so does the required population size. Large population sizes imply large number of cost function evaluations. Certain micro-genetic algorithm has been suggested [16], which evolve very small populations that are claimed to be efficient in locating promising areas of the search space. Obviously, these small populations are unable to maintain diversity for many generations, but the population can be restarted whenever diversity is lost, keeping only the very best fit individuals (usually just the best one according to elitism of one individual). Restarting the population several times during the run of the genetic algorithm is also meant to prevent the risk of premature convergence due to the presence of a particularly fit individual, which has the potential to prevent further exploration of the search space and so may make the program converge to a local minimum. However, such methods overlook the fact that without required balance between exploitation and exploration it can be impossible to reach a result close to the true optimum or optima.

The major researches devoted to trying to maintain or introduce population diversity can be broadly categorized as one of the following [6]:

1) Controlled gene flow using complex population structures. The diffusion model [10], the island model [10], the multinational EA [7] and the religion model [8] are few examples of this method.
2) Controlled and assisted selection by means of specialized operators. This technique has been implemented in crowding [4], deterministic crowding [9], and sharing [1] and is believed to maintain diversity in the population.
3) Reintroduction of genetic material. The random immigrants [3], mass extinction models [8], [2], and [11] are few examples of this technique and are aimed at reintroducing diversity in the population.
4) Dynamic parameter encoding (DPE), which dynamically resizes the available range of each parameter by expanding [5] or reducing the search window.
5) Diversity guided/ controlled genetic algorithms that use a diversity measure to assess and control the survival probability

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-131-6/08/07.

of individuals and the process of *exploration* and *exploitation* [6].

6) Specialized operators to introduce diversity while getting rid of redundant genetic material. This has been implemented in [13, 14].

The EA framework [14] investigated in this paper uses a *synergistic approach* by combining the benefits of the complex population structure as in *item 1* and the specialized operator as in *item 6* above. While the informed mutation operator promotes exploitation, the multi-tier hierarchical population structure confronts genetic drift.

Rest of the paper is organized as follows: The synergistic EA framework is outlined in Section 2. Simulation details, results and discussions are presented in Section 3. Finally, conclusions are drawn in Section 4.

2. THE GA WITH A SYNERGISTIC APPROACH

Algorithm 1: *Procedure POPULATION_GA*

```
begin
t = 0
Initialize population P(t)
Evaluate population P(t)
while (not<termination condition>)
begin
    t = t + 1
    (* Perform pseudo-niching of the population*)
    Call Procedure GRID_NICHING
    (* Perform informed genetic operations *)
    Call Procedure INFORMED_OP
    Create new population using an elitist selection
     mechanism
Evaluate P(t)
end while
for (each individual in the population)
begin
Determine migration based on spatial and/or fitness information
end for
for (each one of n pre-sampled sub-populations)
begin
Evolve population as per canonical GA
Determine migration of individuals
end
end
```

The synergistic POPULATION_GA algorithm is as described in Algorithm 1. This model comprises of multiple co-existing and co-operative functional schemes, which work in tandem. Firstly, the preferential pseudo-genetic operation based on population characteristics and, secondly, the co-evolving hierarchical population structure. In two different levels, the proposed technique employs the COUNTER_NICHING_EA or the COMMUNITY_GA for the main population, while maintaining the hierarchical, co-evolving sub-populations with less fit individuals. The major building blocks of this model are presented in subsequent sections (Section 2.1 and Section 2.2). The mechanisms of procedure GRID_NICHING and procedure INFORMED_OP are as described in our earlier research [13, 14]. The technique also ensures quicker convergence as large populations are not evolved and also less memory is required to store the population.

2.1 Exploration with Informed Operator

The parent population of POPULATION_GA model evolves the population with COMMUNITY_GA technique as described in [13, 14]. The evolutionary mechanism here first extracts information about the population landscape before deciding on introduction of diversity through informed mutation. The aim is to identify locally converging regions or *donor* communities in the landscape that could spare less fit individuals those could be replaced by more promising members sampled in un-explored or under-explored sections of the decision space. The existence of such communities is purely based on the position and spread of individuals in the decision space at a given point in time. Once such regions are identified, random sampling is done on yet to be explored sections of the landscape. Best representatives replace the worst members of the identified regions. Regular mutation and recombination takes place in the population as a whole. The COMMUNITY_GA algorithm probabilistically and randomly samples the global search space and explores for promising regions while concentrating search on the hyperplanes that are likely to contain good solutions.

GRID_NICHING (see [13, 14]) is the mechanism used to extract information about the search space. Here, we have used the term *niching*, simply to identify *environments* of individuals in the population, based on their spatial information. In other words, we try to identify the gross individual clusters in the decision space based on their genotypic proximity. This method organizes the space around the patterns and not the patterns themselves. It finds the spatial distribution information of the individuals in the decision space. First, in each generation, the population members are placed on a multidimensional grid data structure.

This block partitioning of the decision space is done to identify initial signs of cluster formation. Once adequate number of individuals is found in any block, it is considered to be part of a potential cluster. Bounds of the identified blocks are expanded in all N dimensions in order to find a community or cluster.

GRID-NICHING returns information about community or cluster formation in the population for the current generation.

The INFORMED_OP algorithm identifies locally converging communities with too many members of similar fitness. The idea is to explore greater part of the solution space at the expense of these *extra* members. This algorithm uses the pseudo-niching information obtained from GRID-NICHING to guide the genetic operators as above.

However, INFORMED_OP operates on selected communities only. Regular mutation and recombination is performed as usual on the entire population.

Table 1. Description of Test Functions.

Function	Type	Global Minimum
$f_{ack}(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi \cdot x_i)\right)$ where $-30 \le x_i \le 30$	Multimodal	$f_{ack}(x=0)=0$
$f_{gri}(x) = \frac{1}{4000}\sum_{i=1}^{n}(x_i - 100)^2 -$ $\prod_{i=1}^{n}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$ where $-600 \le x_i \le 600$	Multimodal, Medium epistasis	$f_{gri}(x=0)=0$
$f_{rtg}(x) = \sum_{i=1}^{n}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right)$ where $-5.12 \le x_i \le 5.12$	Multimodal, No epistasis	$f_{rtg}(x=0)=0$
$f_{ros}(x) = \sum_{i=1}^{n-1}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right)$ where $-100 \le x_i \le 100$	Unimodal, High epistasis	$f_{ros}(x=1)=0$
$f_{elp}(x) = \sum_{i=1}^{M} i x_i^2$ where $-5.12 \le x_i \le 5.12$	Unimodal	$f_{elp}(x=0)=0$
$f_{sch-1.2}(x) = \sum_{i=1}^{M}\left(\sum_{k=1}^{i} x_k\right)^2$ where $-564 \le x_i \le 64$	Unimodal, High epistasis	$f_{sch-1.2}(x=0)=0$
$f_{rrtg}(x) = 10M + \sum_{i=1}^{M}\left(y_i^2 - 10\cos(2\pi y_i)\right)$ where $y = \boldsymbol{A}x$ with $A_{i,i} = 4/5$, $A_{i,i+1} = 3/5$ (i odd), $A_{i,i-1} = -3/5$ (i even), $A_{i,k} = 0$ (the rest)	Multimodal	$f_{rrtg}(x=0)=0$

2.2 Exploitation with Hierarchical Population Structure

While the COMMUNITY_GA guides EA search based on the information on current population, the hierarchical population scheme allow the apparently under-performing individuals to grow. Although the COMMUNITY_GA tries to confront crowding around fitter individuals, selection pressure will eventually force the less-fit individuals in other parts of the decision space to die. Hence, in the multi-tier population scheme, a set of sub-populations exist alongside the main population. Individuals are moved into these sub-populations according to their fitness level. These co-evolving subpopulations hold individuals within specific fitness ranges and thus prevent unfair competition among individuals belonging to different fitness levels. See [14] for further details on hierarchical population structure.

3. SIMULATION DETAILS AND DISCUSSIONS

Simulations were carried out to apply the proposed POPULATION_GA with the following set up: real-valued encoding for the candidate solutions in the population; parameters N (*population size*) =200, p_m (*mutation probability*) =0.01 and p_r (*recombination probability*) =0.9, to the following benchmark function optimization problems as described in Table 1: Ackley's Path Function ($f_{ack}(x)$), Griewank's Function $f_{gri}(x)$, Rastrigin's Function $f_{rtg}(x)$, Generalized Rosenbrock's function $f_{ros}(x)$, Axis parallel Hyper-Ellipsoidal Function or Weighted Sphere Model $f_{elp}(x)$, Schwefel Function 1.2 $f_{sch-1.2}(x)$ and a rotated Rastrigin Function $f_{rrtg}(x)$.

Three variants of the above problems were considered: 20 dimensional, 50 dimensional and 100 dimensional. Reported results were averaged over 30 independent runs. The maximum number of generations in each run are only 500, 1000 and 2000 for the 20, 50 and 100 dimensional problem variants respectively, as against 1000, 2500 and 5000 generations in [6] for the same set of test cases.

Table 2 presents the error values, ($f(x) - f(x)^*$) where, $f(x)^*$ is the optimum. Each column corresponds to a test function. The error values have been presented for the three dimensions of the problems considered, namely 20, 50 and 100. As each test problem was simulated over 30 independent runs, we have recorded results from each run and sorted the results in ascending order. Table 2 presents results from the representative runs: 1st (Best), 7th, 15th (Median), 22nd and 30th (Worst), Mean and Standard Deviation (Std). The interesting, observation here is that POPULATION_GA tends to have rather steady performance across different dimension ranges and also across the various simulation runs. This is an indicator of reliability of performance of the algorithm.

The observed results have been compared with the following techniques as reported by Ursem in [6]: (i) SEA (Standard EA), (ii) SOCEA (Self-organized criticality EA), (iii) CEA (The Cellular EA), and (iv) DGEA (Diversity guided EA). Simulation results ascertain POPULATION_GA's superior performance as regards solution precision for all four test cases, across their three variants, as can be observed from Table 3. This may be attributed to POPULATION_GA's ability to strike a better balance between *exploration* and *exploitation*. Interestingly the performance of POPULATION_GA [14] is very close to that of COMMUNITY_GA [13] for the test cases in their 20 dimensional and 50 dimensional variants. However, the POPULATION_GA outperforms COMMUNITY_GA when tested on the 100 dimensional variants of all the four test cases compared.

4. CONCLUSIONS

The POPULATION_GA algorithm investigated in this paper basically incorporates three key processes. Firstly, the population's spatial information is obtained with a computationally inexpensive GRID_NICHING algorithm. Secondly, the information is used to identify potential local convergence and community formations and then diversity is intelligently introduced with informed genetic operations, aiming at two objectives: (a) Promising samples from unexplored regions are introduced replacing *redundant* less fit members of over-populated communities. (b) While local entrapment is discouraged, representative members are still preserved to encourage *exploitation.* While the current focus of the research was to introduce and to maintain population diversity to avoid local entrapment, this community-based algorithm can also be adapted to serve as an inexpensive alternative for *niching* GA, to identify multiple solutions in multimodal problems as well as to suit the diversity requirements of a dynamic environment.

Finally, a multi-tier hierarchical population scheme has been used to prevent premature death of promising but initially under-performing individuals. The strong emphasis on exploration of the COMMUNITY_GA has been balanced in the POPULATION_GA with the help of the multi-population structure.

The empirical results obtained from the simulation runs revealed some interesting trends. While there was no significant differnce in the performance of the COMMUNITY_GA and the POPULATION_GA in the 20 dimensional and 50 dimensional test cases, POPULATION_GA showed superior performance in the higher dimensional test case (100 dimensional in the current research). One drawback of the proposed POPULATION_GA is the additional computational overhead of maintaining multiple populations. However this can be easily taken care of by parallel implementation of the framework. Also the synergistic GA framework i.e. the POPULATION_GA has delivered a relatively steady performance across different dimension ranges, including the higher dimensional test cases. This is a very welcome trait as most real life optimization problems tend to be complex and high dimensional.

Table 2. Error Values Achieved for the Test Problems with POPULATION_GA**.

		$f_{ack}(x)$	$f_{gri}(x)$	$f_{rtg}(x)$	$f_{ros}(x)$	$f_{elp}(x)$	$f_{sch-1.2}(x)$	$f_{rrtg}(x)$
20D	**1st (Best)**	1.00E-6	4.0E-10	1.01E-8	0.5E-10	1.01E-20	1.05E-4	1.03E-1
	7th	1.00E-6	4.01E-10	1.911E-8	1.05E-10	1.01E-20	1.11E-4	1.59E-1
	15th (Median)	1.01E-6	4.91E-10	1.950E-8	1.11E-10	1.11E-20	1.15E-4	2.01E-1
	22nd	1.90E-6	5.01E-10	2.001E-8	1.91E-10	1.90E-20	2.01E-4	2.90E-1
	30th (Worst)	3.01E-6	8.11E-10	3.00E-8	1.92E-10	2.00E-20	2.29E-4	4.01E-1
	Mean	1.11E-6	5.01E-10	1.99E-8	1.51E-10	1.29E-20	1.29E-4	3.00E-1
	Std.	5.04E-7	9.20E-11	4.09E-09	5.07E-11	4.07E-21	4.05E-05	0.07
50D	**1st (Best)**	0.59E-5	4.73E-7	1.1E-7	1.11E-4	1.11E-20	1.03E-2	1.01
	7th	0.60E-5	4.81E-7	1.1E-7	1.12E-4	1.11E-20	1.04E-2	1.01
	15th (Median)	0.61E-5	4.81E-7	1.1E-7	1.14E-4	1.11E-20	1.04E-2	1.11
	22nd	0.69E-5	4.85E-7	1.19E-7	1.14E-4	1.12E-20	1.09E-2	1.11
	30th (Worst)	0.99E-5	5.1E-7	1.3E-7	1.20E-4	1.21E-20	1.19E-2	1.19
	Mean	0.65E-5	4.82E-7	1.17E-7	1.14E-4	1.12E-20	1.05E-2	1.12
	Std.	4.01E-07	4.01E-09	4.05E-09	1.03E-06	4.05E-23	0.00023	0.049
100D	**1st (Best)**	1.99E-5	8.92E-6	1.21E-6	1.2E-2	1.15E-19	1.09E-2	1.50
	7th	1.99E-5	8.93E-6	1.21E-6	1.2E-2	1.15E-19	1.09E-2	1.50
	15th (Median)	1.99E-5	8.93E-6	1.21E-6	1.2E-2	1.16E-19	1.09E-2	1.51
	22nd	1.99E-5	8.94E-6	1.21E-6	1.2E-2	1.16E-19	1.09E-2	1.52
	30th (Worst)	2.00E-5	8.95E-6	1.22E-6	1.21E-2	1.18E-19	1.09E-2	1.53
	Mean	1.99E-5	8.93E-6	1.21E-6	1.2E-2	1.16E-19	1.09E-2	1.51
	Std.	1.70E-06	7.09E-08	2.09E-08	2.10E-05	5.02E-22	0.0002	0.00095

5. REFERENCES

[1] D. E. Goldberg and J. Richardson, "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Genetic Algorithms and their Applications* (ICGA'87), Grefenstette, J.J. (ed.), Lawrence Erlbaum Associates, Publishers, 1987, PP. 41–49.

[2] G. W. Greenwood, G. B. Fogel and M. Ciobanu, "Emphasizing Extinction in Evolutionary Programming", *Proceedings of the Congress of Evolutionary Computation*, Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A. (eds.), Vol. 1., 1999, pp. 666–671.

[3] H. G. Cobb and J. F. Grefenstette, "Genetic Algorithms for Tracking Changing Environments", *Proceedings of the 5th International Conference on Genetic Algorithms*, 1993, pp. 523–530.

[4] K. A. De Jong, "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", PhD thesis, University of Michigan, Ann Arbor, MI, Dissertation Abstracts International 36(10), 5140B, University Microfilms Number 76-9381, 1975.

[5] N. N. Schraudolph and R. K. Belew, "Dynamic parameter encoding for genetic algorithms", *Machine Learning*, 9(1), 1992, pp. 9—21.

[6] R. K. Ursem, "Diversity-Guided Evolutionary Algorithms", *Proceedings of Parallel Problem Solving from Nature VII (PPSN-2002)*, 2002, pp. 462-471.

[7] R. K. Ursem, "Multinational Evolutionary Algorithms", *Proceedings of the Congress of Evolutionary Computation (CEC-99)*, Angeline, P.J., Michalewicz, Z., Schoenauer, M., Yao, X., and Zalzala, A. (eds.), Vol. 3. , 1999, pp. 1633–1640.

[8] R. Thomsen and P. Rickers, "Introducing Spatial Agent-Based Models and Self-Organised Criticality to Evolutionary Algorithms" Master's thesis, University of Aarhus, Denmark, 2000.

[9] S. Mahfoud, "Crowding and preselection revisited.", Technical Report 92004, Illinois Genetic Algorithms Laboratory (IlliGAL),, 1992.

[10] T. B¨ack, D. B. Fogel, Z. Michalewicz, and others, (eds.), *Handbook on Evolutionary Computation*, IOP Publishing Ltd and Oxford University Press, 1997.

[11] T. Krink, R. Thomsen and P. Rickers, "Applying Self-Organised Criticality to Evolutionary Algorithms", *Parallel Problem Solving from Nature – PPSN VI*, Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J.J., and Schwefel, H.P. (eds.), Vol. 1. , 2000, pp. 375–384.

[12] H. B. Amor, A. Rettinger, "Intelligent exploration for genetic algorithms: using self-organizing maps in evolutionary computation", *Proceedings of GECCO 2005*, pp. 1531-1538.

[13] M. Bhattacharya, "An Informed Operator Approach to Tackle Diversity Constraints in Evolutionary Search", *Proceedings of The International Conference on Information Technology, ITCC 2004*, Vol. 2, IEEE Computer Society Press, ISBN 0-7695-2108-8,pp. 326-330.

[14] M. Bhattacharya, "Exploiting Landscape Information to Avoid Premature Convergence in Evolutionary Search", *Proceedings of The 2006 IEEE Congress on Evolutionary Computation, (CEC 2006),* Canada, 0-7803-9487-9/06, 2006 IEEE Press, pp. 2575-2579.

[15] J. Hu, E.Goodman, K.Seo, Z. Fan, R. Rosenberg, "The Hierarchical Fair Competition (HFC) Framework for Sustainable Evolutionary Algorithms". *Evolutionary Computation*, 13(1), 2005.

[16] K. Krishnakumar, "Micro-genetic algorithms for stationary and non-stationary function optimization". *SPIE: Intelligent control and adaptive systems*, 1196, 1989, pp. 289–296.

Table 3. Average Fitness Comparison For SEA, SOCEA, The CEA, DGEA, COMMUNITY_GA* And POPULATION_GA. Dimension of Each Function considered are 20, 50 and 100.**

Function	SEA	SOCEA	CEA	DGEA	P_GA**	C_GA*
$f_{ack}(x)$ 20D	2.494	0.633	0.239	3.36E-5	1.00E-6	1.08E-6
$f_{ack}(x)$ 50D	2.870	1.525	0.651	2.52E-4	0.59E-5	1.07E-5
$f_{ack}(x)$ 100D	2.893	2.220	1.140	9.80E-4	1.99E-5	1.08E-4
$f_{gri}(x)$ 20D	1.171	0.930	0.642	7.88E-8	4.0E-10	4.6E-10
$f_{gri}(x)$ 50D	1.616	1.147	1.032	1.19E-3	4.73E-7	3.31E-6
$f_{gri}(x)$ 100D	2.250	1.629	1.179	3.24E-3	8.92E-6	6.71E-4
$f_{rtg}(x)$ 20D	11.12	2.875	1.250	3.37E-8	1.01E-8	1.21E-5
$f_{rtg}(x)$ 50D	44.674	22.460	14.224	1.97E-6	1.1E-7	7.5E-5
$f_{rtg}(x)$ 100D	106.212	86.364	58.380	6.56E-5	1.21E-6	9.99E-5
$f_{ros}(x)$ 20D	8292.32	406.490	149.056	8.127	0.5E-10	1.0E-10
$f_{ros}(x)$ 50D	41425.674	4783.246	1160.078	59.789	1.11E-4	0.01E-3
$f_{ros}(x)$ 100D	91251.300	30427.63	6053.870	880.324	1.2E-2	1.032