# Entropy-Driven Parameter Control for Evolutionary Algorithms

Shih-Hsi Liu
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294, USA
liush@cis.uab.edu, http://www.cis.uab.edu/liush

Marjan Mernik
Faculty of Electrical Engineering and Computer Science
University of Maribor
2000 Maribor, Slovenia
marjan.mernik@uni-mb.si, http://lpm.uni-mb.si/mernik

Barrett R. Bryant
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL 35294, USA
bryant@cis.uab.edu, http://www.cis.uab.edu/bryant

*Every evolutionary algorithm needs to address two important facets: exploration and exploitation of a search space. Evolutionary search must combine exploration of the new regions of the space with exploitation of the potential solutions already identified. The necessity of balancing exploration with exploitation needs to be intelligent. This paper introduces an entropy-driven parameter control approach for exploring and exploiting evolutionary algorithms. Entropy represents the amount of disorder of the population, where an increase in entropy represents an increase in diversity. Four kinds of entropy to express diversity and to control the entropy-driven approach are discussed. The experimental results of a unimodal, a multimodal with many local minima, and a multimodal with only a few local minima functions show that the entropy-driven approach achieves good and explicit balance between exploration and exploitation.*

*Povzetek: V članku je opisan adaptiven način krmiljenja raziskovanja in izkoriščanja v evolucijskih algoritmih, voden s pomočjo entropije.*

## 1 Introduction

Evolutionary Algorithms (EAs) [2, 12] are a common term for solving problems with computers that uses models and mechanisms from biological evolution. Such nature inspired EAs simulate evolution and its mechanisms such as selection, crossover, and mutation. Most well known examples of EAs are Genetic Algorithms (GAs), Evolution Strategies (ESs), Evolutionary Programming (EP), and Genetic Programming (GP) [12]. They have been used successfully for planning, design, simulation and identification, controlling, classification, and for solving many other hard optimization problems. EAs are general purpose search methods with good yet implicit balance between exploration and exploitation. Exploration is a process of visiting entirely new regions of a search space and seeing if anything promising may be found in the regions. Exploita-

tion is a process of using information gathered from the previously visited points in the search space to determine which regions might be profitable to be visited next. Additionally, exploitation techniques are good at finding local optima. However, how is the balance between exploration and exploitation achieved in EAs? More importantly, how can the balance be controlled?

In EAs, the selection process, operators (e.g., crossover and mutation), and population size establish a balance between the exploration and exploitation of the search space [6]. The selection process drives search towards the regions of the best individuals. Hence, exploitation is done by selection. However, Bäck [1] showed that the selection processes can control the level of exploration or exploitation by varying selection pressure. Higher selection pressure pushes the search towards more exploitation and lower selection pressure urges the search towards more exploration.

A mutation operator randomly modifies individuals, with a given probability, and thus increases the structural diversity of the population. From this point of view, the mutation operator is more an exploration operator. Such an operator helps to recover the genetic diversity lost during the selection phase and to explore new solutions avoiding premature convergence. Conversely, mutation can also be seen as an exploitation operator, because most of the genetic material is preserved. However, note that in some EAs (e.g., evolution strategies) mutation has a much bigger exploration role than in genetic algorithms. The crossover operator combines two or more parents to generate better offspring. Such a combination can be derived from the idea that the exchange of information between good individuals will generate even better offspring. From this point of view, the crossover operator is more an exploitation operator. However, a good crossover operator should also generate individuals in the exploration zone. Directing the evolutionary process towards exploration or exploitation is also possible by population resizing [9]. With bigger population size, the search space is more explored than with smaller population size. Therefore, good balance between exploration and exploitation in EAs is achieved by selection, good mutation and crossover operators and by determining parameters (e.g., $p_m$, $p_c$, tournament size, population size), which control mutation, crossover, and selection, respectively.

There have been a variety of studies on determining the best control parameter values [4, 5]. The main problem is to find a set of control parameters, which optimally balances exploration and exploitation: if crossover and mutation rates are very high, much of the space will be explored, but there is a high probability of losing good solutions and of failing to exploit existing schema. If crossover and mutation rates are low, the search space is not explored. The population diversity is therefore rapidly decreasing and ending up in a premature convergence to a non-optimal solution. Despite that, many researchers believed that EAs are effective because of a good ratio between exploration and exploitation. In EAs, however, this ratio is implicitly controlled. In some other search techniques such as reinforcement learning [18], one has explicit control over exploration and exploitation. In EAs, one no longer has explicit and respective control over exploitation and exploration, because it is difficult to delimit exploration from exploitation.

In this paper, an entropy-driven exploration and exploitation approach is presented. The exploration/exploitation of the search space is adapted on-line based on the current status of the evolutionary process. The on-line adaptation mechanism involves a decision process as to whether more exploitation or exploration is needed depending on the current progress of the algorithm and on the current estimated potential of discovering better solutions. This decision process is described in a metaprogramming fashion using a domain-specific language, PPCEA (Programmable Parameter Control for Evolutionary Algorithms) [10]. Because of space consideration, the paper only presents the experimental results using genetic algorithms. Experimenting the mutation role for balancing between exploration and exploitation in evolution strategies is our future work.

The paper is organized as follows. Section 2 describes the related work. In Section 3, four kinds of entropy are introduced to control exploration and exploitation. Section 4 shows the experimental results on the benchmark functions. Finally, Section 5 presents the conclusion.

## 2 Related Work

Optimal balance between exploration and exploitation has been mainly controlled by determining the best control parameter values. There are a variety of studies on this topic [5, 8, 10]. Recommendations on control parameters for a particular set of problems can be found in [4, 15]. In [5], an overview of this problem has been given, where the authors distinguish between parameter tuning and parameter control. Furthermore, methods for parameter control have been classified into deterministic, adaptive, and self-adaptive categories: the deterministic category adjusts parameters by deterministic rules; the adaptive category utilizes the feedback of the evolutionary process to control the direction and magnitude of parameters; and the self-adaptive category encodes parameters into individuals and undergoes mutation and recombination. An example of how to balance between exploration and exploitation by parameter control is described as follows. As soon as an algorithm approaches the optimum, the mutation step size must be decreased to balance the probability of generating a new successful point. A simple idea is to decrease the mutation step size $s$ by a deterministic schedule such as $s_t = s_0/t$ or $s_t = \beta^t \cdot s_0$, where $\beta \in (0, 1)$.

One of the earliest researchers that investigated entropy in EAs was Rosca [14], whose experiments showed that populations appeared to be stuck in local optima when entropy did not change or decrease monotonically in successive generations. Rosca used fitness values in a population to define entropy and free energy measure. Our work extends Rosca's in trying to find different ways to compute entropy in EAs. Moreover, using entropy as a diversity measure and metaprogramming parameter control by PPCEA [10], we are able to control exploration and exploitation in an adaptable manner.

The Diversity-Guided Evolutionary Algorithm (DGEA) [17] uses a distance-to-average-point measure to alternate between phases of exploration and exploitation. It can be expressed easily as a PPCEA program. Moreover, DGEA does not use entropy as a measure for diversity.

In [11], entropy is introduced into EAs for determining the optimal number of clusters. However, in this case the fitness function is entropy-based.

# 3  Entropy in Evolutionary Algorithms

Entropy is a concept in thermodynamics, information theory, and statistical mechanics. The thermodynamic entropy $S$ is a measure of the amount of energy in a physical system that cannot be used to do work. As such, it is also a measure of the disorder and randomness presented in a system. The entropy depends not only on the current state of the system, but also its history. Therefore, it is a state function of the parameters (e.g., pressure and temperature), which describe the observable macroscopic properties of the system. The macroscopic state of the system is defined by a distribution on the microstates that are accessible to a system in the course of its thermal fluctuations. Entropy $S$ of the system is defined as:

$$S = -k_B \sum_i p_i \ln p_i \qquad (1)$$

where $k_B$ is a physical constant known as Boltzmann's constant, $i$ is the energy of microstate, and $p_i$ is the probability that it occurs during the system's fluctuations.

The basic concept of entropy in information theory has to do with how much randomness there is in a signal or random event. Shannon [16] defines entropy in terms of a discrete random event $x$, with possible states $1..n$ as:

$$H(x) = \sum_i^n p_i \log_2(\frac{1}{p_i}) = -\sum_i^n p_i \log_2 p_i \qquad (2)$$

Statistical mechanics explains entropy as the amount of uncertainty which remains about a system, after its observable macroscopic properties have been taken into account. For a given set of macroscopic quantities, such as temperature and volume, entropy is a function of the probability that the system is in various quantum states. The more states available to the system with higher probability, the greater the disorder and thus, the greater the entropy. If the system has only one possible state, there is no uncertainty, and the entropy of the system is zero. If the system has $n$ possible states which are equiprobable ($p_i = \frac{1}{n}$), the entropy is the highest:

$$H = -n\frac{1}{n}\log_2(\frac{1}{n}) = \log_2 n \qquad (3)$$

As such, entropy represents also a succinct measure of diversity. Biological diversity refers to the differences between individuals in a population, which in nature imply structural (genotype) and behavioral (phenotype) differences. In EAs, identical genotypes produce the same fitness. Thus, a decrease in genotype diversity will necessarily cause a decrease in phenotype diversity. Hence, to measure entropy/diversity, one needs to define some structural measures. Such measures, however, might be computationally intensive in some instances of EAs (e.g., genetic programming) [3]. Fortunately, based on the described entropy/diversity relationship between genotype and phenotype, such measures at the phenotype level are sufficient.
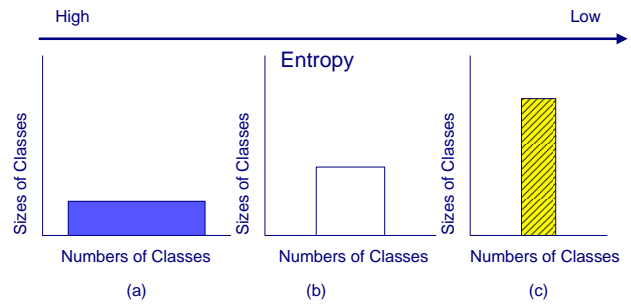


Figure 1: The relationship between entropy and the numbers and sizes of classes

Figure 1 shows how the numbers and sizes of classes of a population affect entropy. High entropy in EAs reveals the presence of many unique fitness values, where the population is evenly distributed over those values, as shown in Figure 1 (a). Figure 1 (c) represents low entropy computed from a population which contains fewer unique fitness values as many individuals have the same fitness.

Rosca [14] calculates entropy for a population by first placing fitness values into fitness classes $p_i$ and counting the size of each fitness class. $p_i$ is the proportion of the population occupied by the population partition $i$. Entropy is then defined as:

$$-\sum_i p_i \log_2 p_i \qquad (4)$$

This paper extends [14] to experiment with entropy, using different flexible cases of fitness classes, to facilitate explicit balance between exploration and exploitation.
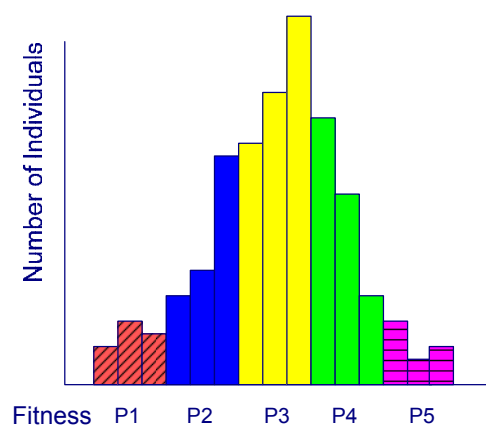


Figure 2: Fitness classes of linear entropy

Figures 2, 3, and 4 show three new cases for defining fitness classes:

– Linear: Assign a predefined yet changeable value to the number of fitness classes, $n$. For each generation, the interval between the best and worst fitness values is evenly partitioned into $n$ sub-intervals as fitness
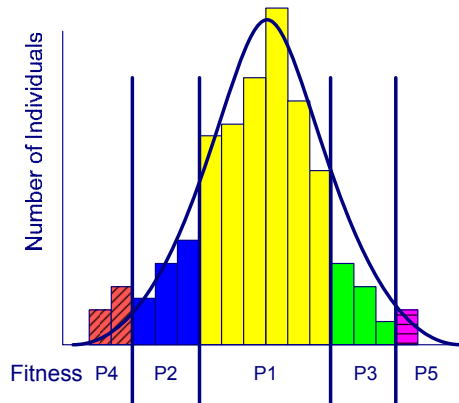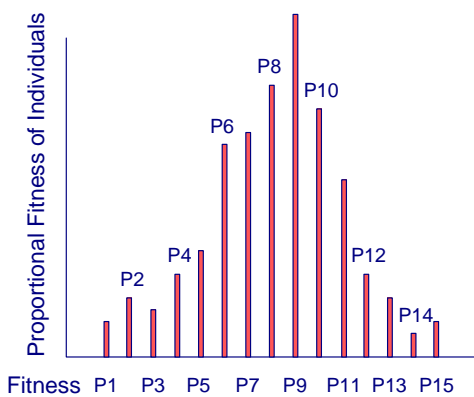
Figure 3: Fitness classes of Gaussian entropy



Figure 4: Fitness classes of fitness proportional entropy

classes (Figure 2). An individual whose fitness value is occupied in a specific sub-interval is classified into the corresponding fitness class. The concept of linear fitness classes is adapted from [14]. Changeable $n$ and various upper and lower bounds of each generation (i.e., the best and worst fitness values) are the two key differences between our approach and Rosca's.

– Gaussian: The partition of fitness classes in this case is derived from Gaussian distribution, as shown in Figure 3. For each generation, fitness classes are "spread out" from the average fitness value ($average$) with the standard deviation ($\sigma$). For example, the upper/lower bound of the first fitness class (P1 in Figure 3) is computed as $average$ +/- $\sigma$. The boundaries of the successive classes (P2 - P5) can be generalized as $average$ +/- $i*\sigma$, where $i \in Z_+$ and $i \leq n/2$. For each generation, the lower bound of the leftmost fitness class is less than or equal to the smallest fitness value, and the upper bound of the rightmost fitness class is larger than or equal to the biggest fitness value.

– Fitness proportional: The fitness proportional approach is a variation of Rosca' approach [14]. Rosca's

fitness classes are partitioned by individuals having the same phenotypes. $p_i$ is the proportion of a population occupied in the $i^{th}$ partition. In our approach, $p_i$ is formalized as $f_i / \sum_i^{Popsize} f_i$, where $f_i$ is the fitness value of an individual. $p_i$ is the criterion for categorizing fitness classes. As all individuals of a population have different $p_i$ (namely, different fitness values), the number of fitness classes $n$ equals the population size ($Popsize$). If more than one individual has the same fitness value (i.e., $p_i = p_j$, where $i \neq j$), $p_j \cdot \log_2 p_j$ is eliminated in the Equation (1) and $n < Popsize$. It is because two identical fitness classes are not needed, and the elimination complies with the definition of diversity. Figure 4 shows 15 fitness classes sorted by $p_i$, each of which has one or more individuals occupied.

The next section exercises linear, Gaussian, fitness proportional and Rosca entropies for the entropy-driven approach and compares the experimental results with the Fogarty [7], Schaffer [15], and 1/5 success rule [12] approaches.

## 4 Experiments

Entropy-driven exploration and exploitation have been experimented with on the suite of test functions presented in [19]. Due to lack of space, only examples of the *Sphere Model* ($f_1$), *generalized Rastrigin's function* ($f_9$), and *Branin function* ($f_{17}$) are presented in this section. To illustrate all the experiments easily, Best fitness value (B), Average fitness value (A), Worst fitness value (W), Population Diversity (D), Standard Deviation (S), Linear Entropy (E), Gaussian Entropy (G), Fitness Proportional Entropy (P), and Rosca Entropy (R) with respect to a population from generations 0 to maximum generation (X-axis) are included in the following figures. Curves B, A, and W use the same definitions as all other EAs; curves E, G, and P are defined in Section 3; curve S is the standard deviation of the fitness values of all individuals; curve D is the Euclidean distance between all individuals; and curve R is the entropy defined in [14]. All but entropy curves (E, G, P, and R) use the left Y-axis as the coordinate. Table 4 shows the initial values setup (we used the same setting as in [19]) for the following experiments: $f_1$, $f_9$, and $f_{17}$ have different maximum generation ($Maxgen$) settings; $Popsize$ is the population size; $p_m$ and $p_c$ are mutation and crossover rates; $Epoch$ is the stride of parameter adjustments during the evolutionary process; and $Round$ is the number of experiments for each example, and the experimental results in subsequent figures are the average values out of 50 rounds.

Sections 4.1, 4.2 and 4.3 respectively present $f_1$, $f_9$, and $f_{17}$ with their experimental results of the Fogarty [7], Schaffer [15], 1/5 success rule [12], and entropy-driven approaches. Only two figures of each function are selected in the paper. All of the experimental results with the corresponding figures may be found at the PPCEA website [13].
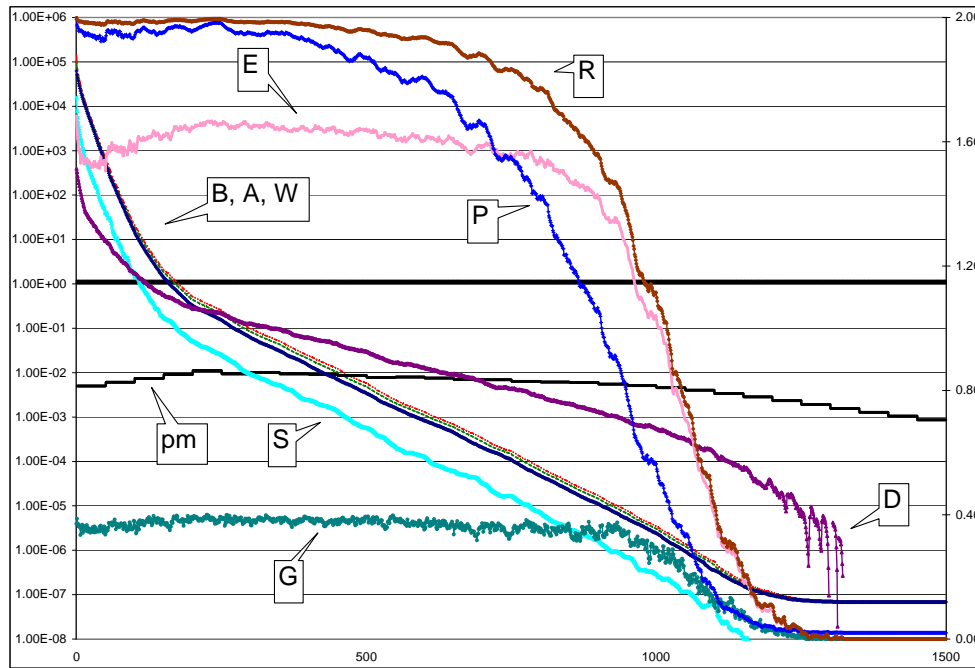
Figure 5: 1/5 success rule approach for $f_1$

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| Maxgen ($f_1$) | 1500 | Maxgen ($f_9$) | 5000 |
| Maxgen ($f_{17}$) | 100 | Popsize | 100 |
| $p_m$ | 0.005 | $p_c$ | 0.75 |
| Epoch | 50 | Round | 50 |

Table 1: Initial values of parameters in experiments on functions $f_1$, $f_9$, and $f_{17}$

## 4.1 The Sphere Model

The Sphere Model ($f_1$) is a unimodal function as shown in Equation (5).

$$f_1(x) = \sum_i^d x_i^2 \qquad (5)$$

where $x_i \in [-100, 100]$, $d$ (dimension) = 30, and min($f_1$) = $f_1(0,...,0)$ = 0.

The first presented experiment is the parameter tuning approach using the Schaffer parameter setting ($p_m = 0.005$ and $p_c = 0.75$). The mean best value and convergence rate[1] are $6.82 \cdot 10^{-8}$ and 830, respectively. The Fogarty approach is a deterministic one that initializes $p_m = 0.11375$ and adjusts the value using the Fogarty formula [7]. The mean best value is $2.13 \cdot 10^{-5}$ at generation 765. Figure 5 presents the results using the 1/5 success rule [12]. Such a rule determines $p_m$ to be increased when the successful permutation rate ($\varphi$) is greater than 1/5, and to be decreased when $\varphi$ is less than 1/5. In Figure 5, a good balance between exploration and exploitation (yet still more on ex-

---

[1]The point that curve Best becomes flat in the figure.

ploration) can be found before generation 900: curves E and R are stable in the ranges between 1.4 and 1.65 and between 1.55 to 2.00, respectively; curves B, A, W, S, and D are smoothly decreased; and $p_m$ is changed every 50 generations to adjust the mutation step. From generations 900 to 1220, curves E and R steeply decline, and curve G has downhill move. Such curves show that the evolutionary process is inclined from exploring to exploiting the current regions with relatively small mutation steps. From generations 1220 to 1320, all entropy curves are getting flat and curve D has a "saw-toothed" shape. Such curves imply that the searching process is in the exploitation phase and is not stuck in local optima. The best value found using the 1/5 success rule approach is $6.82 \cdot 10^{-8}$ at generation 1274.

Before examining the last chart, an entropy-driven approach written in PPCEA is shown in Figure 6. When entropy is greater than 0.5, $p_m$ is decreased to facilitate the exploitation phase. Smaller mutation steps avoid the increase of population diversity. As entropy is smaller than 0.5, more exploration is required to avoid local optima. Therefore, $p_m$ is increased to diversify the search regions. Such an example shows that balance between exploration and exploitation can be adjusted in synergy of entropy and $p_m$. Figure 7 shows the result using this source code.

In Figure 7, curves E, P, and R steeply decline between generations 0 and 450. Curves B, A, W, S, and D also diagonally traverse the plane. When curve E is between its midpoint (at generation 350) and upper bound (0.74 to 1.68), $p_m$ is decreased (line 9 of the PPCEA code) to balance exploitation against exploration. As curve E is between its lower bound and midpoint (0 to 0.74), exploration outperforms exploitation by raising $p_m$. This phenomenon can be observed from curve D that declines more steeply and has a
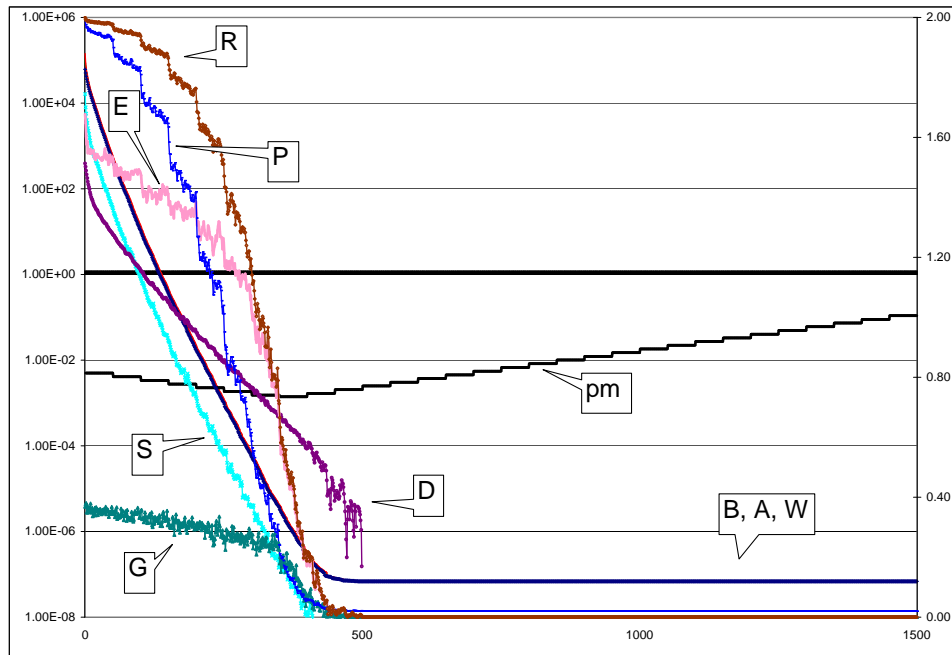
Figure 7: Entropy-driven approach for $f_1$

```
1  genetic
2     g := 0;
3     while ( g < Round ) do
4        t:=0;
5        init;
6        while ( t < Maxgen ) do
7           callGA;
8           if ( Entropy > 0.5 ) then
9              pm := pm * 0.82
10          fi;
11          if ( Entropy < 0.5 ) then
12             pm := pm * 1.22
13          fi;
14          t := t + Epoch
15       end;
16    g:= g + 1
17    end
18 end genetic
```

Figure 6: Entropy-driven parameter control written in $PPC_{EA}$

drastic "saw-toothed" shape from generations 400 to 500. Curve R is similar to curve E in terms of the shapes and the balance between exploration and exploitation. The best value found is the same as in the 1/5 success rule. However, please note that the convergence is much faster in the entropy-driven approach (at generation 467). Hence, many fitness evaluations after 467 generations can be skipped.

## 4.2   Generalized Rastrigin's Function

Generalized Rastrigin's Function ($f_9$) is a multimodal function with many local minima as shown in Equation (6).

$$f_9(x) = \sum_{i}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10] \qquad (6)$$

where $x_i \in [-5.12, 5.12]$, $d$ (dimension) = 30, and $\min(f_9)$ = $f_9(0,...,0)$ = 0.

For the Schaffer approach for $f_9$ (figure in [13]), exploration is still carried out energetically after generation 1000 in the search space comprising many local minima. Because of the late vivid exploration, the best optimal solution is still improved slightly (20.86) until the evolutionary process converges at generation 3988. Figure 8 (i.e., the experimental results of the Fogarty approach) is a good example to represent that the process is stuck at the local minima. The figure shows that $p_m$ may decrease too fast to perform enough exploration. After generation 400, entropy curves (i.e., curves E, G, P, and R) and fitness curves (i.e., curves B, A, and W) are nearly static, yet diversity curves (i.e., curves D and S) exhibit extreme shakiness. This phenomenon implies that even though the exploration is still active, the relatively small $p_m$ does not provide enough exploration power to assist the evolutionary process to jump out the local optima. Hence, the experimental results of the Fogarty approach are the worst among the four approaches (40.55 at generation 4079).

The characteristic of exploiting many local minima can be also examined in the results of the 1/5 success rule (figure in [13]). However, because of the inefficient exploration power determined by the small $p_m$ value at the later stage, there is no exploration or exploitation activity
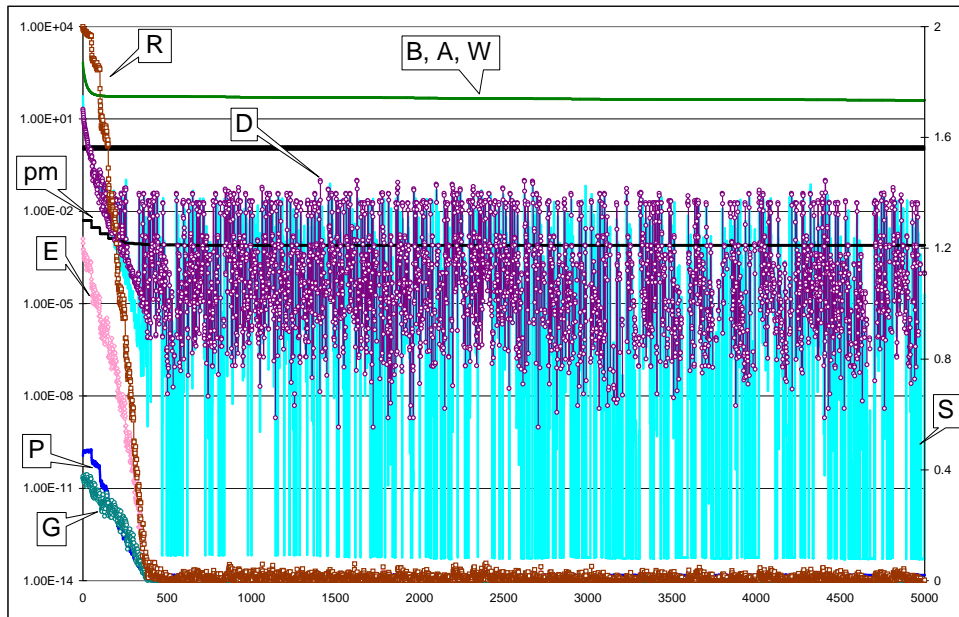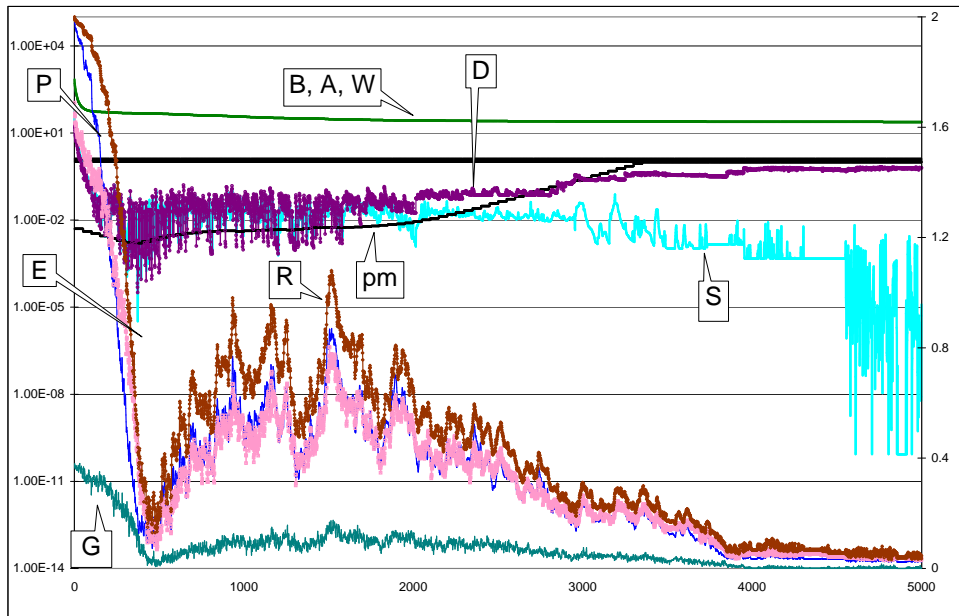
Figure 8: Fogarty approach for $f_9$



Figure 9: Entropy-driven approach for $f_9$

observed. The mean best value and convergence rate are 25.24 and 1265, respectively. Figure 9 shows a good case of balance between exploration and exploitation using the entropy-driven approach in the case of multimodal function. In the chart, the evolutionary process starts at inclining from more exploration toward more exploitation driven by declining $p_m$ before generation 320. From generation 300 to 850, the rising $p_m$ facilitates more exploration to discover many local minima. In this phase, the same or better values may be found and selected. Entropy curves and diversity curves are therefore updated drastically. Most importantly, because of the exploration on the search space of local minima, fitness values are still slightly improved (23.99) at the very late phase (generation 3023).

## 4.3 Branin Function

The Branin Function ($f_{17}$) is a multimodal function with only a few local minima as shown in the following equation.

$$\begin{aligned} f_{17}(x) &= [x_2 - (5.1x_1^2)/(4\pi^2 + 5x_1/\pi - 6]^2 \\ &\quad + 10[1 - 1/(8\pi)]cos x_1 + 10 \end{aligned} \tag{7}$$

where $x_1 \in [-5, 10]$ and $x_2 \in [0, 15]$, $d$ (dimension) = 30, and $\min(f_{17}) = f_{17}(0,...,0) = 0.398$.

Because there are only a few local minima in $f_{17}$ given a small maximum generation number, the evolutionary process cannot be guaranteed to discover all of the local optima using the Schaffer approach (i.e., parameter tuning problem). In Figure 10, diversity curves appearing again after generations 30, 66, 76, 83 and 90 show that a few local optima are found in this phase. Fortunately, the evolutionary process still possesses enough exploration power to improve the value of mean best value (0.421 at generation 90). Similar to the Schaffer results, the Fogarty approach for $f_{17}$ also generates small refinements for the mean best value (0.432) at the late stage (generation 80). However, the slightly different results between the two approaches may be derived from the early decreasing $p_m$ in the Fogarty approach. Please refer to [13] for the enlarged Figure 10 and the numerical improvement of mean best value that may not be observed in Figure 10. For the 1/5 success rule, because the success mutation ratio is always below an ideal value, 0.2, the entire process inclines towards exploitation by reducing $p_m$. The mean best value (0.434) is close to the Fogarty approach. However, because of different formulae for adjusting $p_m$, the 1/5 success rule converges at generation 59, which is much earlier than the Fogarty approach.

Although $f_{17}$ has a few local maxima, the entropy-driven approach still performs a good balance between exploration and exploitation as well as finding even better solutions at the end of the evolutionary process. Figure 11 presents similar characteristics (i.e., rising $p_m$, drastic changing entropy curves, and decreasing fitness value curves) as Figure 10. The mean best value is 0.398 at generation 100.

The experimental results on all benchmark functions indicate that the linear and Rosca approaches for defining fitness classes are superior to Gaussian and fitness proportional ones in terms of providing the information for balancing exploration and exploitation.

## 5 Conclusion and Future Work

The opinions on the research on exploration and exploitation are still widely divided [5]. In this paper, we introduce a novel entropy-driven exploration and exploitation approach. The balance between exploration and exploitation is fulfilled by the synergy of $p_m$, $p_c$ and entropy on-line. The on-line adaptation mechanism involves PPCEA as to whether more exploitation or exploration is needed depending on the current progress of the algorithm and on the current estimated potential of discovering better solutions. The experimental results in all figures show that our approach can easily interpret the influence of exploration and exploitation using curve E and auxiliary curves.

Experiments with the entropy-driven exploration and exploitation approach for evolution strategies [12] are planned. Additionally, a more generic PPCEA that manipulates more similar related work (e.g., Harik's parameterless genetic algorithm [9]) will benefit the community of evolutionary computation.

## References

[1] T. Bäck. *Selective Pressure in Evolutionary Algorithms: A Characterization of Selection Mechanisms.* Proc. 1$^{st}$ IEEE Conf. on Evolutionary Computing, pages 57-62, 1994.

[2] T. Bäck, D.B. Fogel, and Z. Michalewicz. *Handbook of Evolutionary Computation.* University of Oxford Press, 1996.

[3] E. Burke, S. Gustafson, G. Kendall, and N. Krasnogor. *Advanced Population Diversity Measures in Genetic Programming.* Parallel Problem Solving from Nature - PPSN VII, Springer-Verlag LNCS, No. 2439, pages 341-350, 2002.

[4] K. De Jong. *The Analysis of the Behavior of a Class of Genetic Adaptive Systems.* Ph.D. thesis, Department of Computer Science, University of Michigan, Ann Arbor, Michigan, 1975.

[5] A. Eiben, R. Hinterding, and Z. Michalewicz. *Parameter Control in Evolutionary Algorithms.* IEEE Trans. on Evolutionary Computation, Vol. 3, No. 2, pages 124-141, 1999.

[6] A. Eiben and C. Schippers. *On Evolutionary Exploration and Exploitation.* Fundamenta Informaticae, No. 35, pages 35-50, 1998.
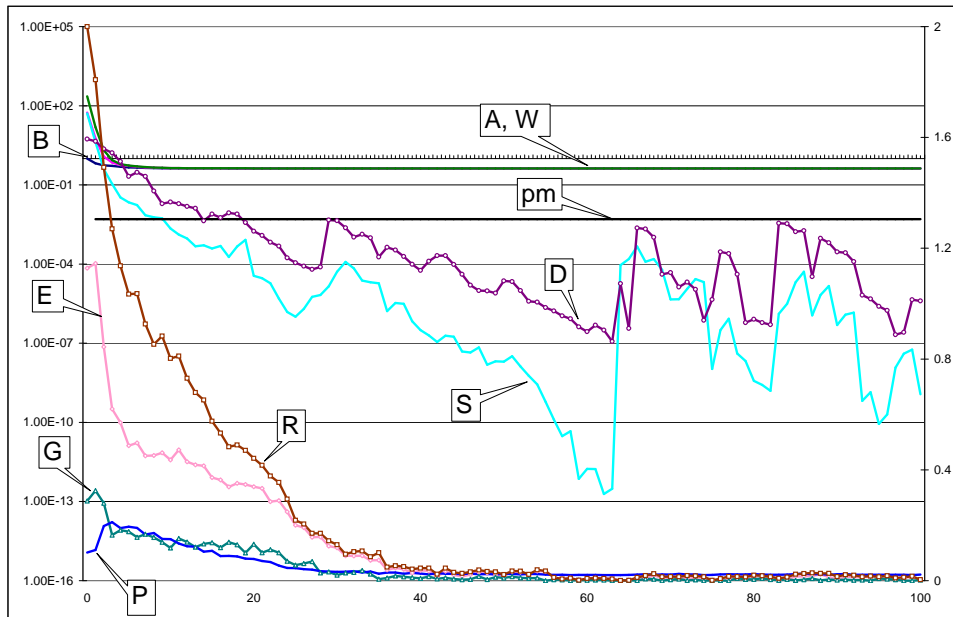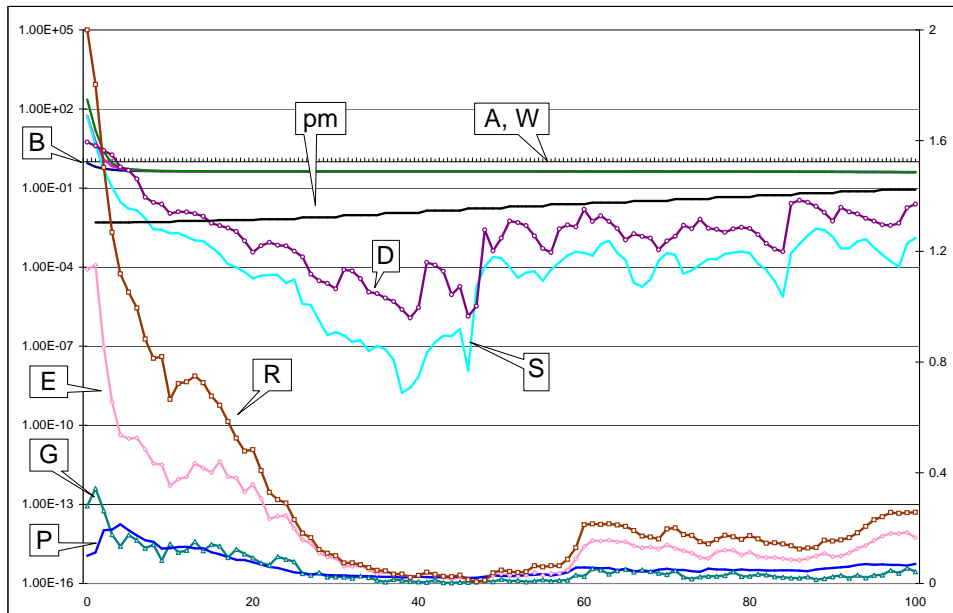
Figure 10: Schaffer approach for $f_{17}$



Figure 11: Entropy-driven approach for $f_{17}$

[7] T.C. Fogarty. *Varying the Probability of Mutation in the Genetic Algorithm.* Proc. $3^{rd}$ Intl. Conf. on Genetic Algorithms, pages 104-109, 1989.

[8] J.J. Grefenstette. *Optimization of Control Parameters for Genetic Algorithms.* IEEE Trans. on Systems, Man & Cybernetics SMC-16, No. 1, pages 122-128, 1986.

[9] G. Harik and F. Lobo. *A Parameter-less Genetic Algorithm.* Technical Report IlliGAL 9900, Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign, 1999.

[10] S.-H. Liu, M. Mernik, and B.R. Bryant. *Parameter Control in Evolutionary Algorithms by Domain-Specific Scripting Language PPCEA.* Proc. $1^{st}$ Intl. Conf. on Bioinspired Optimization Methods and their Applications, pages 41-50, 2004.

[11] W. Lu, and I. Traoré. *A New Evolutionary Algorithm for Determining the Optimal Number of Clusters.* Proc. $17^{th}$ Intl. Conf. on Tools with Artificial Intelligence, pages 712-713, 2005.

[12] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* $3^{rd}$ ed., Springer-Verlag, 1996.

[13] PPCEA: A Domain-Specific Language for Evolutionary Algorithms.
http://www.cis.uab.edu/liush/PPCea.htm

[14] J. Rosca. *Entropy-Driven Adaptive Representation.* Proc. of the Workshop on Genetic Programming: From Theory to Real-World Applications, pages 23-32, 1995.

[15] J.D. Schaffer et al. *A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization.* Proc. $3^{rd}$ Intl. Conf. on Genetic Algorithms, pages 51-60, 1989.

[16] C. Shannon. *A Mathematical Theory of Communication.* Bell Systems Technical Journal, Vol. 27, pages 379-423, 623-656, 1948.

[17] R. Ursem. *Diversity-Guided Evolutionary Algorithms.* Parallel Problem Solving from Nature - PPSN VII, Springer-Verlag LNCS, No. 2439, pages 462-471, 2002.

[18] S. Whitehead. *Learning from Delayed Rewards.* Ph.D. thesis, King's College, Cambridge University, England, 1992.

[19] X. Yao, Y. Liu, and G. Lin. *Evolutionary Programming Made Faster.* IEEE Trans. on Evolutionary Computation, Vol. 3, No. 2, pages 82-102, 1999.