Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, 2-5 November 2003

CONTROLLING DIVERSITY OF EVOLUTIONARY ALGORITHMS

D. H. M. Nguyen, K. P. Wong
Department of Electrical Engineering, The Hong Kong Polytechnic University
E-MAIL: eenguyen@polyu.edu.hk

Abstract:

This paper presents a control system based method for adapting the mutation step–size in order to control the diversity of the genome population. Population diversity is controlled so that its decreases exponentially with time in order to facilitates the linear order convergence that evolutionary algorithms are capable of. The paper restricts its attention to the application of unimodal search since linear order convergence of evolutionary algorithms have only been established analytically for unimodal and not multimodal search. The case of multimodal search is left as an exercise in implementation of sub–population schemes. The paper also highlights the subtle but important difference between setting of EAs parameters and control of EAs performance.

Keywords:

Evolutionary algorithms;Feedback control;Population diversity;Mutation step–size

1 Introduction

The application of Evolutionary Algorithms (EAs) to solve a given problem is essentially an interrelated three stage process: The first being the selection on an appropriate EA with the required search properties and corresponding population schemes and genetic and selection operators. Next, comes the mapping of problem variables to the genespace, including the incorporation of problem constraints and heuristics, and the problem objectives to the fitness space.

Finally, the EA parameters are required to be set for optimal solution convergence, a complex and resource intensive task which generally results sub–optimum outcomes. With the performance of implemented EAs strongly dependent on the optimality of parameters settings, and the task of finding such optimal settings difficult. This leads to parameters settings being considered a fundamental practical limitation of EAs [1].

To overcome this limitation numerous methods have been developed for setting EA parameters, these have been categorized and given an overview most recently in [2], which identify two principle means of parameter settings that of parameter tuning and parameter control. Further [2] argues that the parameter control schemes where EA parameters are dynamically varied during runtime is superior to static parameter settings obtained from parameter tuning.

The current paper presents a new scheme for automatically setting mutation step–size via a self tuning feedback control system in response to how the EA is evolving. Thus, implementation wise it belongs to the subs class of adaptive control within the class of parameter control. However, the aim of the new automatic mutation step–size setting is to dynamically obtain a desired population diversity, in order to enable the EA to achieve linear order convergence. Consequently, the new scheme is in effect a control of population diversity via adaptive mutation step–size setting, and not a control of mutation step–size as such.

2 Previous Approaches to Mutation Step–size Setting

Previous approaches to effect mutation step–size to improve EA convergence is outlined in [2] which classifies three types of mutation step–size controls 'deterministic', 'adaptive' and 'self–adaptive'. Essentially the deterministic control schemes assumes that mutation step–size σ is strictly a function of iteration time t per (1), which is predetermined based on some heuristics. Typically these heuristics states that σ should decrease monotonically, or that $s_D \in (0;1]$, as this would allow exploration in the initial stages and exploitation in the final stages.

$$\sigma(t+1) = \sigma(t) \cdot s_D(t) \qquad (1)$$

The limitations of the 'deterministic' σ setting schemes is two fold, firstly appropriate general heuristic are difficulty to formulate since the effect of σ on convergence is coupled in a complex way to the stochastic dynamics of the EA and the fitness landscape induced by the specific underlying application. Secondly, having s_D monotonically decreasing means that if σ becomes too small on a particular run it can not be corrected, yet it is difficult to justify any general form of s_D a prior which is not monotonically decreasing.

The 'adaptive' σ setting schemes, do not pre– suppose a reduction schedule, instead adjustments are made to

0-7803-7865-2/03/$17.00 ©2003 IEEE

σ based on how previous values performed. That is there is feedback from the EA as to the appropriateness of the current values allowing it to be change to a more suitable value per (2), where $\dot{T} = \{1,2,\cdots t\}$ is the set of iterations up to the current time, $\dot{X}$ is the set of genomes and $\dot{Y}$ is the set of corresponding fitness of the population. Like the case with s_D heuristics are required to determine the form of s_A however feedback allows positive as well as negative adjustments to be made thus $s_A \in \mathsf{R}^+$.

$$\sigma(t+1) = \sigma(t) \cdot s_A\left(\dot{X}(\dot{T}), \dot{Y}(\dot{T})\right) \qquad (2)$$

The 'adaptive' σ setting schemes improve upon the 'deterministic' schemes in that σ can be change by increasing or decreasing from the current value if it is found by observations made from the population that current value is inappropriate, instead of blindly following a predefined schedule with deterministic σ. However existing 'adaptive' schemes are still inherently limited by the need have heuristic that relate how σ should be change from its currents value. Here again the complex response EA's performance due to σ makes it difficult to formulate the required incremental model.

The 'self–adaptive' σ setting schemes is similar to the 'adaptive' schemes in that there is feedback from the population. The mechanism for this feedback is that of evolution itself, that is σ is evolved as part of the genome and is subjected to genetic variations and natural selection. Thus a genome becomes $x = (x_1, x_2, \cdots, x_K, \sigma)$ where K is the dimension of the genespace X. In these schemes fitness is defined only on sub–vector $(x_1, x_2, \cdots, x_K)$ and genetic variation on the additional gene(s) takes the form of (3), where $s_S \in \mathsf{R}^+$ and R is a random variable. Furthermore, generally better results are obtained if σ is modified first before it is used to generate new sub–vector(s).

$$\sigma(t+1) = \sigma(t) \cdot s_S(\mathsf{R}) \qquad (3)$$

The 'self–adaptive' σ setting schemes are an improvement on the 'adaptive' schemes since no heuristics is required to be formulated as the relationship between σ and EA performance is to be discovered through evolution. Apart from susceptibility to premature convergence arising from 'loss of step–size control' [4], 'self–adaptive' EA are also inherently limited by the evolutionary mechanisms used to control σ, specifically any improvement in convergence rate is limited by the rate at which EA can discover appropriate values for σ.

3 New Approach to Mutation Step–size Setting, Controlling Diversity

To have more direct and more timely σ setting than that possible with the 'self–adaptive' schemes, this paper advances a new adaptive/deterministic hybrid scheme for setting σ. The new hybrid 'control–system' scheme essentially establishes an automatic incremental relationship between the mutation step–size parameter and the state of genetic diversity by using the current genome population $\dot{X}(t)$ as feedback. Heuristics are then defined for what the state of diversity 'should' be as a deterministic function of t, consequently the new 'control–system' scheme for setting σ can be described by (4).

$$\sigma(t+1) = \sigma(t) \cdot s_C\left(\dot{X}(t), t\right) \qquad (4)$$

The advantage of the new 'control–system' scheme is that it has a feedback mechanism that is not limited by the response or learning rate of the evolutionary mechanism of 'self–adaptive' schemes. Although heuristics are required to be defined, unlike the case with existing 'deterministic' and 'adaptive' schemes these heuristics can be formulated at the level of EA's system states and not EA's systems parameters. Free from considerations of parameter effects on system states, more far reaching hypothesis or even analytical results can be implemented to improve EAs performance.

3.1 Convergence and Genetic Diversity Hypothesis

It has been established analytically that EAs can have linear order convergence, or converges exponentially with iteration time, for unimodal search if parameters are optimised [5]. Although exponential progress has been observed for some EAs on multimodal search, the current paper limit discussion to unimodal search as the analytical results of linear order convergence are only given for this case. However, the new scheme is not limited to unimodal search, multimodal search simply require application of the scheme to a sub–population assigned to search a particular solution mode.

From the result in [5] it is hypothesised that EAs can be induced to have linear order convergence for unimodal search if the population diversity can be controlled to decreased at a matching exponential rate with t, or that the desired diversity $\bar{\varsigma}$ should be of the form (5), where $\bar{\varsigma}(0) > 0$ and $\tau > 0$ are constants defining the initial value of desired diversity and decay rate respectively. This heuristic is motivated by the simple observations that for unimodal search convergence necessary imply a corresponding reduction in diversity and that an

exponential rate of convergence would need to be accompanied by an exponential reduction in diversity.

$$\vec{\varsigma}(t) = \vec{\varsigma}(0) \cdot e^{-\frac{t}{\tau}} \tag{5}$$

The initial diversity parameter $\vec{\varsigma}(0)$ can be estimated from the boundaries of the genespace $\mathcal{X}$. Assuming that diversity is measured as some deviation from the center of the genome population, then a heuristical value for $\vec{\varsigma}(0)$ is that it should be half the maximal extent L of $\mathcal{X}$. With non–spherically bounded $\mathcal{X}$, the maximal extent is directionally dependent and a representative value is required. For example the rectangularly bounded $\mathcal{X} = \{x : x_{k\in\mathcal{K}} \in [\breve{x}_k; \hat{x}_k]\}$, where $\mathcal{K} = \{1,2,\cdots K\}$ enumerates the K dimensions of $\mathcal{X}$, under consideration, a representative L can be given by the geometric average of the maximal extents along the K axes, consequently $\vec{\varsigma}(0)$ can be given by (6).

$$\vec{\varsigma}(0) = \frac{L}{2} \text{ where } L = \left[\prod_{\forall k\in\mathcal{K}} \hat{x}_k - \breve{x}_k\right]^{\frac{1}{K}} \tag{6}$$

With a deviation based measure for diversity, the final diversity $\vec{\varsigma}(T)$ where T is the final iteration, represents the desired level of accuracy $\breve{\rho}$. Consequently, T can be determined for any specified $\breve{\rho}$ per (7). However, $\breve{\rho}$ limited below by (8) since this is the smallest value that still allow each of the N genomes to *approximately* occupy a distinct state different from the solution x^*, assuming the finite numeric representation uniformly samples the K components of x in the vicinity of x^*. That is, $\breve{\rho}$ is the radius of the hyper–sphere centered around x^* who's volume must contain $N+1$ distinct states, or that $\kappa(K)\cdot\breve{\rho}^K = (N+1)\cdot\eta^K$ where $\kappa(K)$ is the hyper–sphere volume constant and the sampling interval $\eta = 2.2204\times10^{-16}$ is taken as the machine accuracy.

$$T = \left\lceil \tau \cdot \ln\left(\frac{\vec{\varsigma}(0)}{\breve{\rho}}\right)\right\rceil \tag{7}$$

$$\breve{\rho} = \left[\frac{N+1}{\kappa(K)}\right]^{\frac{1}{K}} \cdot \eta \text{ where } \kappa(K) = \frac{\left[\Gamma\left(\frac{1}{2}\right)\right]^K}{\Gamma\left(1+\frac{K}{2}\right)} \tag{8}$$

Explicit equation for τ is more difficult to arrive at as it is more problem specific, however larger values are more appropriate for harder problems and smaller population size N and vice versa, since larger τ implies that the EA would be converging a slower rate. Thus τ can be considered as a speed setting for the EA, as such appropriate values are not only dependent on the EA 'vehicle', but it is also dependent on the 'terrain' or fitness landscape in which it operates. For the paper τ is determined by observing the decay rate of diversity of the first 10 trials of the self–adaptive EP. The lowest decay rate is chosen for τ since lower decay rates means higher diversity and hence lower chance of local convergence.

3.2 Controlling Diversity with Mutation Step–size

To bring the actual diversity ς so that it is equal to the desired diversity $\vec{\varsigma}$ within some tolerance, a self–tuning proportional control system [6] is employed. This approach of using a self–tuning control system allows for arbitrary $\vec{\varsigma}(t)$ to be implemented since the control system would not only adapt itself to the dynamics of the EA but also to the form of $\vec{\varsigma}(t)$ as well. Consequently, any hypothesis about how diversity should be as a function of time such as those of [7] can be implemented and not just the current hypothesis of (5).

The proposed control system, illustrated in Figure 1, operates as follows: Given the current mutation step–size $\sigma(t)$ the EA E would produce a distribution of genomes $\dot{X}(t)$. Sensor S then takes $\dot{X}(t)$ and generates the logarithmic of the current diversity $\ln\varsigma(t)$. The $\ln\varsigma(t)$ is then compared with $\ln\vec{\varsigma}(t)$ the desired current value, to generate an error signal $\varepsilon(t)$ for iteration t. Controller C then use $\varepsilon(t)$ to generate $\Delta(t)$, the required adjustment for the current mutation step–size, on a logarithmic scale. Actuator A, would then use $\Delta(t)$ to generate $\sigma(t+1)$ the required mutation step–size for the next iteration.

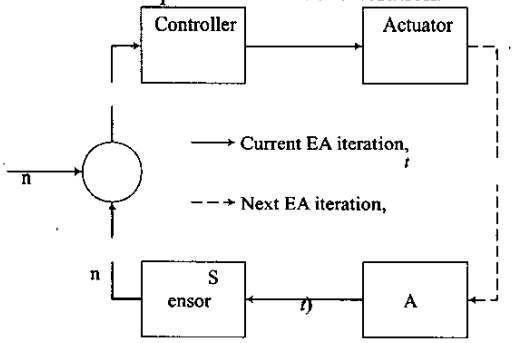

Figure 1: Feedback Control System for Population Diversity with Mutation Step–size

From Figure 1 it can be seen that the control system operates in the logarithmic domain of the process variable

$\varsigma \in \mathsf{R}^+$ and the control variable $\sigma \in \mathsf{R}^+$. The logarithmic scale is chosen to allow the domains of the process variable and the control variable to be extended to R. This done so that the Controller C, an 'additive' incremental module, can operate in an unbiased manner given the positive range of the process and control variables. In other words operating on the logarithmic scale allows for symmetrical changes in σ and ς towards the lower bound of 0 and the upper bound of ∞.

3.3 Operation of the Sensor Module

Sensor module S measures the current genetic diversity $\varsigma(t)$ on a logarithmic scale so that it can be compared to the desired value $\bar{\varsigma}(t)$. In essence the current genetic diversity describes the distribution of genome population $\dot{X} = \{x(t) : {}^{n\in\mathcal{N}} x(t) \in X\}$, where $\mathcal{N} = \{1,2,\cdots N\}$ enumerates the genomes, about its' centre. Various statistical metrics can be used to measure this distribution and its center, the one adopted for this paper is based on the radial deviation from the mean value over $\forall n \in \mathcal{N}$ per (9).

$$\varsigma(t) = \left\langle \left\| {}^{n}x(t) - \left\langle {}^{n}x(t) \right\rangle_{\mathcal{N}} \right\| \right\rangle_{\mathcal{N}} \tag{9}$$

3.4 Operation of the Controller Module

Control module C generates the required incremental change $\Delta(t)$ in mutation step–size on a logarithmic scale by specifying that this change should be proportional to the error $\varepsilon(t)$ or discrepancy in $\ln\varsigma(t)$ and $\ln\bar{\varsigma}(t)$, the measured and desired diversity on a logarithmic scale, per (10). Proportional gain γ is required to be set to appropriate values in order for the system to function. Thus it is proposed that γ is set so that it normalises $\varepsilon(t)$ by its current root mean square value $\tilde{\varepsilon}(t)$ over $\forall t \in \dot{T}$ per (11).

$$\Delta(t) = \gamma(t)\cdot\varepsilon(t) \quad \text{where} \quad \varepsilon(t) = \ln\bar{\varsigma}(t) - \ln\varsigma(t) \tag{10}$$

$$\gamma(t) = \frac{1}{\tilde{\varepsilon}(t)} \text{ where } \tilde{\varepsilon}(t) = \sqrt{\left\langle [\varepsilon(t)]^2 \right\rangle_{\dot{T}}} \tag{11}$$

3.5 Operation of the Actuator Module

Actuator module A producers the next mutation step–size $\sigma(t+1)$ by incrementing $\ln\sigma(t)$ the current mutation step–size on a logarithmic scale by $\Delta(t)$ per (12). By substituting (11) into (10) and (10) into (12), the incremental transformation of $\sigma(t)$ to $\sigma(t+1)$ by the control system evaluates to (13). With (13) the functional form of s_C in (4) can be identified as $s_C\left(\dot{X}(t),t\right) = e^{\varepsilon(t)/\tilde{\varepsilon}(t)}$ by noting the implicit dependence of $\varepsilon(t)$ on $\dot{X}$ and the dependence of $\tilde{\varepsilon}(t)$ on $\dot{T}$ and hence t.

$$\ln\sigma(t+1) = \ln\sigma(t) + \Delta(t) \tag{12}$$

$$\sigma(t+1) = \sigma(t)\cdot e^{\frac{\varepsilon(t)}{\tilde{\varepsilon}(t)}} \tag{13}$$

3.6 Initial Mutation Step–size

The initial mutation step–size $\sigma(0)$ is arbitrary as the system should adjust $\sigma(t)$ to the required value, however to not waste computation resources with unnecessary adjustments, $\sigma(0)$ should be chosen so that initially the EA can generate diversity $\varsigma(0) \approx \bar{\varsigma}(0)$. With $\bar{\varsigma}(0)$ set to span X an appropriate setting for $\sigma(0)$ is to have each genome initially cover the fractional volume V/N where V is the total volume of X. For radially symmetric mutation the volume covered by a genome under mutation is a hyper–sphere with volume $\kappa(K)\cdot\sigma(0)^K$, thus $\sigma(0)$ can be evaluated per (14).

$$\sigma(0) = \left[\frac{V}{N\cdot\kappa(K)}\right]^{\frac{1}{K}} \text{ where } V = \prod_{\forall k\in\mathcal{K}} \hat{x}_k - \check{x}_k \tag{14}$$

The 'covering' of a volume of the search space is to be understood in a probabilistic sense, since given mutation action (15), then it is with some finite probability ω that $\|x(t+1) - x(t)\| \le R$ for some radial distance $R > 0$. To make precise the concept of 'covering' the paper pre–scale the random mutation vector $\mathbf{X}$ so that when $\sigma = \sigma(0)$, for $R = 1$ the probability of confinement is $\omega = 0.955$, or that initially ≈ 2 standard deviations are contained in a radius of one unit. Thus if radially symmetric or multivariate Gaussian mutation is used then $\mathbf{X}$ would be set per (16).

$$x(t+1) = x(t+1) + \sigma(t)\cdot\mathbf{X} \text{ where } \langle \mathbf{X} \rangle = \mathbf{0} \tag{15}$$

$$\mathsf{X}_{k\in\mathcal{K}} = \mathsf{N}\ (0,2) \tag{16}$$

4 Evaluation of New Control Scheme's Performance

The performance of the new control scheme with

$s_C = e^{\varepsilon(t)/\tilde{\varepsilon}(t)}$ is compared to the self-adaptive scheme with $s_S = e^R$ where $R = N(0,1)$. The EA chosen is an EP with radially symmetric Gaussian mutation per (16), unitary reproduction where one new offspring is generated from each existing parent, and tournament selection is applied to the combined parent and offspring population, with tournament size of 2. Population size is set at $N = 100$, and the EPs are run for $T = 662$ iterations per (7). For the self-adaptive EP $M_S = 100$ trials are conducted where as for the control system EP $M_C = M_S - 10 = 90$ were conducted since 10 trials were use to determine τ.

4.1 Test Function

Both EPs are applied to find the maximum of function (17) where the length of $\boldsymbol{x}$ is $K = 3$. In essence f_0 is a unimodal function with 1 as the only axis of symmetry, and an infinite number of analytical local optima in any finite neighbourhood of the global optimum $f_0(\mathbf{0}) = \mathbf{0}$. Function f_0 is chosen since the low symmetry would limit the exploitation of problem specific information but more importantly the increasing number of local optima would make the search prone to premature convergence. This characteristic is important since it is necessary to demonstrate that the gain in accelerated convergence is not at the cost of premature convergence.

$$f(\boldsymbol{x}) = f_0(\boldsymbol{A}\cdot\boldsymbol{x} + \boldsymbol{b}), \quad \forall \boldsymbol{x} \in [-2\cdot K; 2\cdot K]^K$$
$$f(\boldsymbol{x}) = -r_x \cdot e^{\cos\left(\frac{\phi_{+x}}{r_x}\right)\cos\left(\frac{\phi_{-x}}{r_x}\right)\cos(r_x\cdot\phi_{+x})\cos(r_x\cdot\phi_{-x})} \tag{17}$$
$$\text{where } r_x^2 = \boldsymbol{x}\cdot\boldsymbol{x} \text{ and } \cos(\phi_{\pm x}) = \frac{\pm \boldsymbol{x}\cdot\mathbf{1}}{r_x \cdot r_1}$$

In (17) f is simply f_0 evaluated under affine transformation defined by (18) and (19), where linear transformation $\boldsymbol{A}$ consists of axes scaling by $\boldsymbol{S}$ and axes rotation by $\boldsymbol{R}$, and $\boldsymbol{b}$ represent a translation. In essence the transformation states that firstly, each of the $K\cdot(K-1)/2$ distinct axes pairs are rotated by planar angle $\theta_{i,j}$ from $x_i \to x_j$ so that $\tan(\theta_{i,j}) = j/i$. Next, each of the k^{th} axis is scaled by k so that $x_k \mapsto k\cdot x_k$. Finally, each of the k^{th} axis is then translated in the positive direction by k so that $x_k \mapsto x_k - k$. The affine transformation is employed to remove orientation bias, scaling bias and positional bias that may be exploited by the algorithms. Figure 2 illustrates function f for the case of $K = 2$.

$$\boldsymbol{A} = \boldsymbol{S}\cdot\boldsymbol{R} \text{ where } \boldsymbol{S}: s_{i,j} = \begin{cases} k & ,i = j = k \\ 0 & ,i \neq j \end{cases}$$
$$\text{and} \quad \boldsymbol{R} = \prod_{\substack{\forall i\in\{1,2,\cdots K-1\} \\ \forall j\in\{i,i+1,\cdots K\}}}{}^{i,j}\boldsymbol{P} \quad \text{with}$$
$${}^{i,j}\boldsymbol{P}: {}^{i,j}p_{i',j'\in K} = \begin{cases} 1 & ,i' = j' \neq i \text{ or } j \\ \frac{i}{\sqrt{i^2+j^2}} & ,i' = j' = i \\ \frac{+i}{\sqrt{i^2+j^2}} & ,i' = j' = j \\ 0 & ,i' \neq j' \neq i \text{ or } j \\ \frac{j}{\sqrt{i^2+j^2}} & ,i' \neq j' = i \\ \frac{-j}{\sqrt{i^2+j^2}} & ,i' \neq j' = j \end{cases} \tag{18}$$

$$\boldsymbol{b}: b_{k\in K} = -k \tag{19}$$

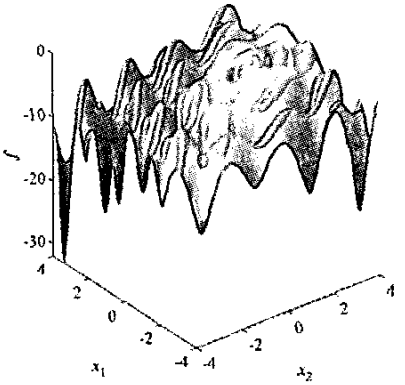

Figure 2: Two dimensional version of test function $f(\boldsymbol{x})$

4.2 Results and Discussion

The measure adopted for gauging the convergence at each iteration t of each trial m is the residual $\rho(t)$ or geometric mean separation of the current genome population from the solution genome $\boldsymbol{x}^*$ per (20). To assess the stochastic performance three statistics of $\rho(t)$, taken over all trials $m \in \mathcal{M}$, are employed. These are defined in (21) with $\mu(t)$ being the geometric mean of $\rho(t)$, $\mu^-(t)$ the geometric mean of all $\rho(t) < \mu(t)$, and $\mu^+(t)$ the geometric mean of all $\rho(t) > \mu(t)$. The μ^- and μ^+ statistics are the equivalent to the first and third quartile statistics with the geometric mean replacing the median statistic.

$$\rho(t) = \left\langle \left\langle \left\| x(t) - x^* \right\| \right\rangle_{\mathcal{M}} \right\rangle_{\mathcal{N}} \tag{20}$$

$$\begin{aligned} \mu(t) &= \left\langle \left\langle \rho(t) \right\rangle_{\mathcal{M}} \right\rangle_{\mathcal{N}} \\ \mu^+(t) &= \left\langle \left\langle \rho(t) > \left\langle \left\langle \rho(t) \right\rangle_{\mathcal{M}} \right\rangle \right\rangle_{\mathcal{M}} \right\rangle_{\mathcal{N}} \\ \mu^-(t) &= \left\langle \left\langle \rho(t) < \left\langle \left\langle \rho(t) \right\rangle_{\mathcal{M}} \right\rangle \right\rangle_{\mathcal{M}} \right\rangle_{\mathcal{N}} \end{aligned} \tag{21}$$

The μ^-, μ and μ^+ statistics for 'C' the EP with diversity control and 'S' the EP with self–adaptive mutation, are illustrated in Figure 3. In general it can be seen that initially all of the geometric mean residuals of the 'S' are less then those of the 'C', however after approximately $t \approx 50$, $t \approx 100$ and $t \approx 200$ iterations the geometric mean residuals μ^+, μ and μ^- of 'C' are respectively smaller, and remain smaller then those of 'S'. Thus it can be seen that 'C' converges faster then 'S' and that it is able to get closer to the global optimum then 'S'. In fact μ_C^- indicates that convergence of 'C' consistently approaches the numerical accuracy of the machine which is of order $\eta \approx 10^{-16}$.

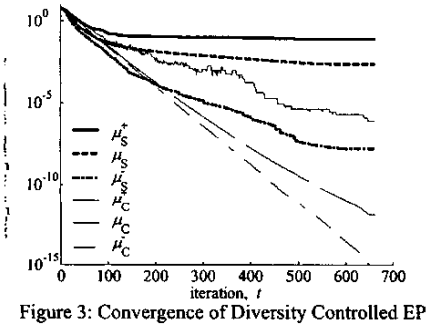

Figure 3: Convergence of Diversity Controlled EP 'C' and Self–adaptive Mutation Step–size EP 'S'

The greater consistency by which 'C' globally converges is evident by the smaller separation between μ_C and μ_C^- then between μ_C and μ_C^+. When μ_C is closer to μ_C^- then μ_C^+, this indicates that the geometric distribution of the mean residual for 'C' is skewed so that the average is close to the minimum value attainable by 'C'. In contrast, the opposite situation exists for 'S' with μ_S being closer to μ_S^+ then μ_S^-, consequently it can be expected that the average geometric mean residual is generally larger than that attainable by 'S'.

5 Summary and Conclusions

In summary, a new mutation step–size setting scheme is presented that seeks to control the diversity of an evolutionary algorithm. A hypothesis of the required diversity setting is quantitatively postulated together with the specification of a fixed termination criteria. A complete description of the sensor, actuator and controller modules of the new control system is presented together with appropriate initialisation values. A new test function with the required properties for assessing global convergence of the new scheme is also given. The performance of the new scheme is demonstrated to be statistically superior to the self–adaptive mutation scheme.

In conclusion, this paper illustrates that linear order global convergence of unimodal optimisation can be consistently achieved with control of population diversity. And that population diversity control can be accomplished using a proportional control system with automatic adaptive gain when control actions acts on the logarithmic domain of mutation step–size and deviation based diversity measure. More generally the paper demonstrates a new approach to evolutionary computation which proceeds from a hypothesis of how the state of an EA can be modulated to improve its performance by providing the mean for this modulation via a simple control system.

Acknowledgments

The work is supported by Research Grants Council, Hong Kong (PolyU 5214/03E) and The Hong Kong Polytechnic University.

References

[1] K. Deb, "Limitations of evolutionary computation methods', eds T. Back, D.B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation*, chapt B.2.9, pp B2.9:1-B2.9:2, pub Oxford University Press, 1997.

[2] A.E. Eiben, R. Hinterding, "Parameter control in evolutionary algorithms", *IEEE Transactions on Evolutionary Computation*, vol 3, no 2, July 1999, pp 124-141.

[3] T. Back, "Self–adaptation', eds T. Back, D.B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation*, chapt C.7.1, pp C7.1:1-C7.1:15, pub Oxford University Press, 1997.

[4] G. Rudolph, "Self–adaptive mutations may lead to premature convergence', *IEEE Transactions on Evolutionary Computation*, vol 5, no 4, August 2001, pp 410-414.

[5] H.G. Beyer, G. Rudolph, "Local performance Measures", eds T. Back, D.B. Fogel and Z. Michalewicz, *Handbook of Evolutionary Computation*, chapt B.2.4, pp B2.4:1-B2.4:27, pub Oxford University Press, 1997.

[6] K. Ogata, *Modern Control Engineering*, 2nd Eds, chapt 7, pp 536-644, pub Prentice-Hall, 1990.

[7] H.G. Beyer and K. Deb, "On self–adaptive features in real– parameter evolutionary algorithms', *IEEE Transactions on Evolutionary Computation*, vol 5, no 3, June 2001, pp 250-270.