Scheduling Variance Loss Using Population Level Annealing for Evolutionary Computation

Arnold L. Patton
pattona@cse.msu.edu

Erik D. Goodman
goodman@egr.msu.edu

William F. Punch III
punch@cse.msu.edu

Genetic Algorithms Research and Applications Group (GARAGe)
Dept. of Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
http://garage.cse.msu.edu

Abstract

Evolutionary Programming (EP) has historically used a number of approaches for selection of the mutation step size. Current EP implementations typically use self-adaptive meta-parameters for mutation step size selection. However, one of the potential drawbacks of this scheme is that it is not directly responsive to the variance reduction caused by selection. In this paper, we investigate an alternate method for mutative step size selection that reacts directly to the variance-reducing effects of selection.

1 Introduction

Evolutionary Programming (EP), when applied to the domain of real-valued function optimization typically creates new potential solutions through simultaneous mutation of the parameters of an individual solution. This mutation is often applied in the form of addition of a sample from some zero mean probability distribution (typically Gaussian or Cauchy) with a variance of σ.

EP implementers have historically used a number of techniques for selection of the mutation step size, σ, for a given parameter. Early EP efforts used the error value of a given solution to calculate σ; however, this implies knowledge of the optimal value or at least the theoretical limit for a given optimization problem.

[Schwefel 81] introduced a method of self-adaptation for selection of mutation step sizes. Under this technique, each solution vector encodes both a series of solution parameters and a series of meta-parameters that specify the step sizes. Each meta-parameter is also mutated during the mutation phase, according to the expression:

$$\sigma_i' = \sigma_i * e^{\left(\tau' * N(0,1) + \tau * N(0,1)\right)}$$

where:

$$\tau = \frac{1}{\sqrt{2\sqrt{n}}}, \tau' = \frac{1}{\sqrt{2n}}$$

More recent adaptations of this mutation scheme use a Cauchy distribution for allocation of mutation samples [Yao 96]. This shift has the overall effect of biasing the search process away from local search in favor of increased global search characteristics.

In general, a mutative operator should achieve a balance between localized search and global exploratory search. The more biased the operator is toward local search, the more likely the system is to become trapped at a local minimum. On the other hand, a large global bias tends to lengthen the required search time, and at worst degenerates to random search.

2 Variance Recapture

The mutation size must remain large enough to counterbalance the variance-reducing effects of selection. If the average mutation size becomes too small, it is possible for a single individual to quickly dominate an entire population. The resulting “premature” convergence is often difficult to overcome.

One of the potential drawbacks of using a self-adaptive technique such as those offered by [Schwefel 81] and [Yao 96] is that there is no direct feedback between the focusing effects of selection and the expanding effects of mutation. Thus, it is still possible to reach a point of premature convergence while using these algorithms. (However, it is theoretically possible to escape such a situation in this case since the mutation step size is not directly dependent on the diversity of the population.)

On a population level, it may be possible to directly counterbalance the variance-reducing effects of selection by choosing an appropriate mutation size. However, this is only possible if the effects of the mutation operator on the population variance are predictable based solely on the mutation size.

As an example, consider a population which currently has a certain level of variance for one solution parameter across the current population. After the selection process, we can measure the variance for that parameter across all surviving solutions. Further, since we can calculate how much variance a given mutation size adds to the population, we can reverse the calculation and select the mutation size which allows us to *recapture* a given percentage of the variance loss caused by selection. Such an algorithm is labeled a *variance recapture* (VR) algorithm. Note that a VR algorithm is self-adaptive at a system level – that is, it adapts to changes in the overall system as measured by changes to the population.

The creation of a VR mutation operator adds an additional control parameter to the search system, namely the targeted percentage of recapture. This parameter allows additional user control in that dynamic adjustment of its value modifies the global vs. local search characteristics of the present search. In fact, it may be possible to place an artificial “annealing schedule” on this parameter, allowing the search designer to guide how resources should be allocated as the search progresses. Further, it may be advantageous to cycle through phases of *variance addition* by targeting greater than 100% variance recapture. (In this sense, variance recapture becomes somewhat of a misnomer; however, ,sustained targeting of greater than 100% of the variance eventually leads to system divergence, or the equivalent of random search.)

One possible difficulty with this approach is its dependency on the initial variance of the population, which in turn is dependent on the selected initializing ranges. Once the initial variance has been selected, it may take some time for the VR algorithm to allow sufficient variance loss before the system begins to locate useful or ”interesting” search areas. However, most evolutionary search algorithms are somewhat dependent on population initialization, though perhaps to a lesser degree. Further, such difficulties may be allowed for by adjusting the recapture target to a lower value for the initial phase of the search. Deciding when to make such adjustments and understanding their general net effects are the targets of further research.

3 Experimental Design

Empirical testing of the VR algorithm allows for direct comparison of the relative merits of this approach to standard EP mutation. However, empirical tests produce difficulties in that the results: are often difficult to interpret, are specific to the functions used for testing, and rely upon the precision of the implementation. Nonetheless, empirical testing may give us a concrete "proof of principle" as to whether the VR concept is worthy of further study.

1.1 Tested Algorithms

Three systems are tested: VR mutation, VR mutation in combination with lognormal mutation, and a standard, self-adaptive EP. The EP system using VR mutation (VREP) is in all ways identical to standard EP except for the mutation operator. Mutation values are drawn from a Gaussian distribution. The mutation step size for a given parameter is selected as follows:

1. $\mathbf{v_i}$ is the average measured variance of parameter **i** across the full population (both parents and children)
2. $\mathbf{w_i}$ is the average measured variance of parameter **i** across the selected parents for the following generation
3. Given a recapture target ratio **p** (where **p**=1.0 equals 100% recapture), calculate $\mathbf{t_i}$ as $t_i = p(v_i - w_i) * \frac{s}{c}$, where **s** is the full population size (parents and children), and **c** is the total number of children produced through VR mutation.
4. Select the mutation size as the lesser of $\sqrt{t_i}$ or 0.

Note that if we assume the values of parameter **i** are normally distributed in the current population, the resulting average variance of the next generation will be $w_i + p(v_i - w_i)$, which is the desired result.

It may be useful to use the VR mutation algorithm in combination with other forms of mutation, such as standard lognormal EP mutation. In order to achieve such a mixture, the VR+LN EP system applies VR mutation to half of the children for a given parent, and standard lognormal mutation to the other children. Children which are produced via VR mutation directly inherit the lognormal meta-parameters from their parent. Both parents and children produced via lognormal mutation are used to calculate $\mathbf{w_i}$.

The third system selected for testing is a standard EP using self-adaptive mutation size selection with lognormal updates for the self-adaptive parameter. The results from this system should help to provide a baseline for comparison with the VR variants.

2.1 Test Functions

We selected a number of well regarded optimization test functions from the literature (e.g. [Yang 97], [Savaranan 95], [Salomon 96]) as well as an original function (Sphere-Hull) intended to be difficult for algorithms which tend toward the population mean or have difficulty following non-linear surfaces (originally presented in [Patton 98]). All functions were redefined to be oriented toward minimization and to allow scaling to any number of dimensions. The details of the functions used for this evaluation are outlined in Table 1.

Table 1. Test Functions and Initial Parameter Ranges

Function	Name	Initial Range
$\sum x_i^2$	Sphere	$-100 \le x_i \le 100$
$\sum \lfloor \lvert x_i \rvert \rfloor$	Dejong f3	$-5.12 \le x_i \le 5.12$ †
$\sum_{i=1}^{n} i * x_i^4 + N(0,1)$	Dejong f4	$-1.28 \le x_i \le 1.28$ †
$\left(\sqrt{\sum x_i^2} - \sqrt{nc^2}\right)^2 + \sqrt{\sum (x_i - t_i)^2}$ where $\sqrt{\sum t_i^2} - \sqrt{nc^2} = 0$	Sphere-Hull	$-500 \le x_i \le 500$
$\sum - x_i \sin(\sqrt{\lvert x_i \rvert}$	Schwefel 1	$-500 \le x_i \le 500$
$-20e^{\left(-0.2*\sqrt{\frac{1}{n}\sum x_i^2}\right)} - e^{\frac{1}{n}\sum \cos(2\pi x_i)} + 20 + e$	Ackley	$-30 \le x_i \le 30$
$\sum_{i=1}^{n-1} (x_i^2 - 2x_{i+1}^2 - 0.3\cos(3\pi x_i) - 0.4\cos(4\pi x_{i+1}) + 0.7)$	Bohachevsky	$-15.12 \le x_i \le 15.12$
$10n + \sum (x_i^2 - 10\cos(2\pi x_i))$	Rastrigrin	$-15.12 \le x_i \le 15.12$
$\sum_{i=1}^{n-1} (x_i^2 + x_{i+1}^2)^{0.25} [\sin^2(50 * (x_i^2 + x_{i+1}^2)^{0.1}) + 1.0]$	Schaffer	$-100 \le x_i \le 100$
$\sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j^2 \right)$	Schwefel 2	$-500 \le x_i \le 500$
$1 + \sum \frac{x_i^2}{4000} - \prod \left[\cos\left(\frac{x_i}{\sqrt{i}} \right) \right]$	Griewangk	$-600 \le x_i \le 600$
$\sum_{i=1}^{n-1} [100 * (x_i^2 - x_{i+1}) + (1 - x_i)^2]$	Rosenbrock	$-20 \le x_i \le 20$
$\sum_{i=1}^{n-1} \frac{x_i^2 + x_{i+1}^2}{2} - (\cos(20\pi * x_i) * \cos(20\pi * x_{i-1})) + 1$	Yip & Pao* [Yip 95]	$-15 \le x_i \le 15$
$\sum_{i=0}^{99} (e(i) - t(i))^2$ where $e(i) = x_1 \sin\left(\frac{2i\pi}{100} x_2 + x_3 \sin\left(\frac{2i\pi}{100} x_4 + x_5 \sin\left(\frac{2i\pi}{100} x_6\right)\right)\right)$, $t(i) = c_1 \sin\left(\frac{2i\pi}{100} c_2 + c_3 \sin\left(\frac{2i\pi}{100} c_4 + c_5 \sin\left(\frac{2i\pi}{100} c_6\right)\right)\right)$, $c_1 = 1.0, c_2 = 5.0, c_3 = 1.5, c_4 = 4.8, c_5 = 2.0, c_6 = 4.9$	FM Matching [Tsutsui 97]	$-6.4 \le x_i \le 6.4$ †

† Note that with origin offset of 10.0 in each dimensions, global optimum is effectively outside of initial range.

*Modified slightly from original to allow 0 global optima value

These test functions were selected because of their ease of computation and widespread use, which should facilitate evaluation of the results and comparison to similar work. Several of these functions are highly multi-modal; and many are known to be difficult for other search algorithms, especially under high levels of dimensionality (i.e. in a *black box* form where the search algorithm should not necessarily assume independence of dimensions).

In order to avoid any bias introduced by axial symmetry, all test problems were translated away from the origin by a fixed amount in all dimensions (e.g. a solution of (0,0) is moved to (10,10), etc.) Note that the given initialization ranges are in terms of the translated origin, not the origin of the original function.

3.1 Test Conditions

Other than the mutation algorithms employed, all parameters remained constant across the three systems. Each parent contributed 6 children to the selection pool for the following generation. Standard EP tournament selection with a tournament size of 10 was used. All mutations were based on a Gaussian distribution, and all used the same algorithm for providing Gaussian samples. Population sizes were fixed at 50 parent solutions producing 300 (50 * 6) children each generation for a total of 350 potential solutions input to each tournament selection. All tests were continued for a total of 1000 generations, and the fitness of the best individual in each generation was recorded. For all VR mutations, the recapture ratio was fixed at 0.98 throughout each test.

4 Results

Each system was applied to each of the 14 different test functions for a total of 119 times. The $\log_{10}$ of the average best solution value is plotted against number of evaluations for selected test problems in Figures 1 through 6. Average final best values and the standard deviation of the final best values, as well as the best (minimum) and worst (maximum) final best values for all test functions and algorithms are listed in Tables 2-15.

Wilcoxon rank-sum 2-sample location statistics [McClave 85] were computed between the VREP and classic LN-EP algorithms, and also between the VR-LN-EP and classic LN-EP. This statistic is useful in that it can be used to calculate the probability that the samples are drawn from distributions with separate means without making assumptions about the underlying distributions (except that they be somewhat similar). To calculate the rank-sum, the conjoined list of final best results for two algorithms is sorted in increasing order. Each item on the list is assigned the value of its position (ties share their assigned values equally). These values are tallied for each algorithm. The result is two rank-sum scores which add to $\frac{n(n+1)}{2}$, where n is the total number of samples in the list. A z statistic may then be calculated from these rank-sums which may be used to test the hypothesis that the two distributions are indeed distinct. Tables 16 and 17 report the rank-sum and z statistic for the VREP and VR+LN EP, when compared to the classic LN-EP for each test problem. Also, the minimum α level required for acceptance of the hypothesis that the distribution of the given algorithm is shifted to the left of (i.e. is on average better than) the LN-EP is listed.

Figure 1. : Griewangk function (10 Dimensions)

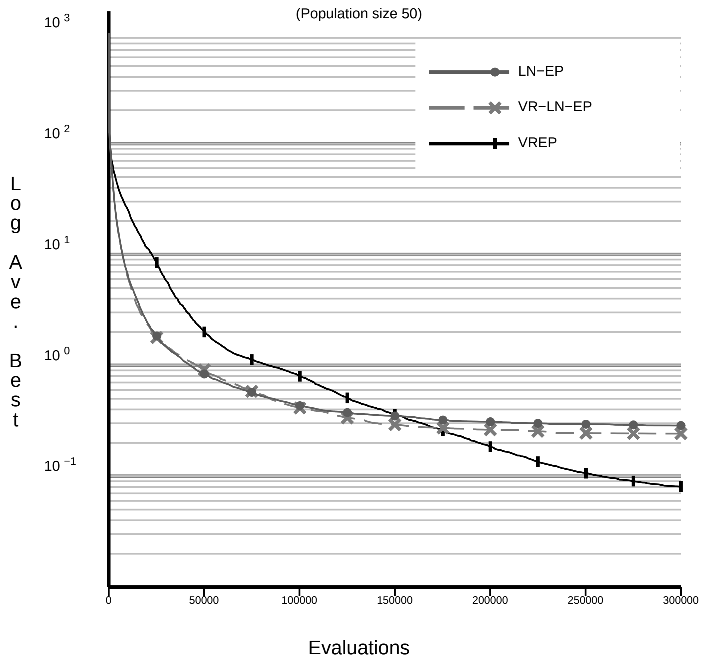


Figure 2. : Rosenbrock function (10 Dimensions)

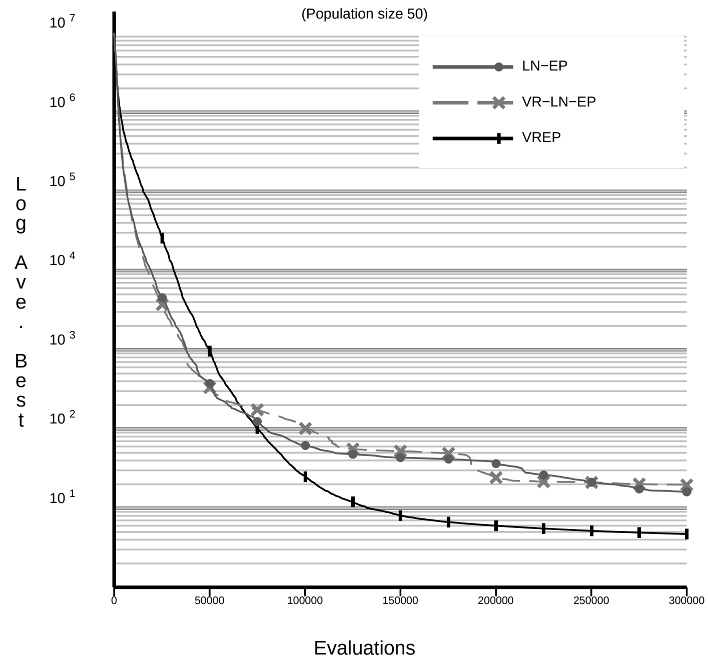


Figure 3. : Ackley function (10 Dimensions)

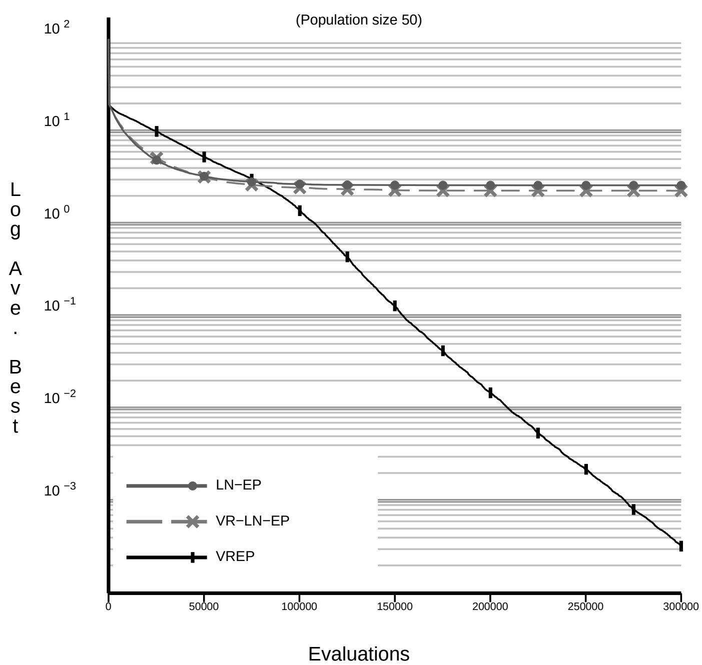


Figure 4. : Schaffer function (10 Dimensions)

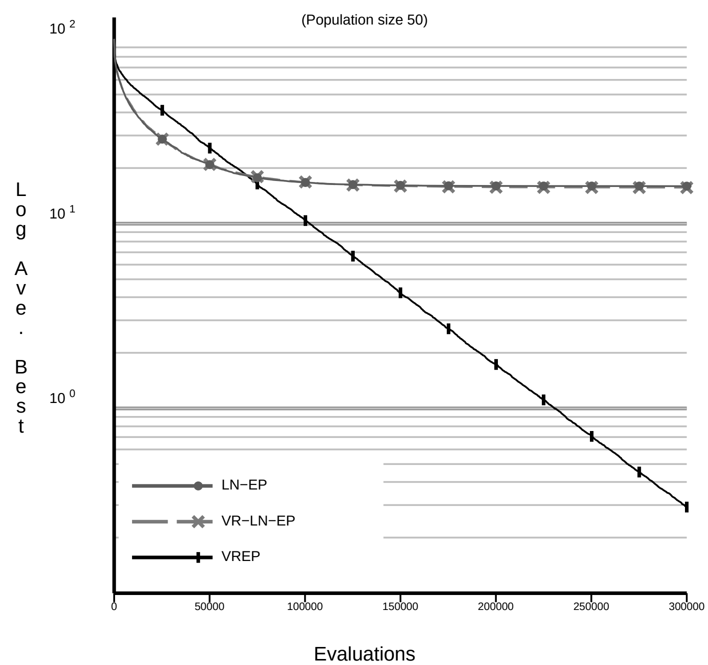


Figure 5. : Sphere hull function (10 Dimensions)

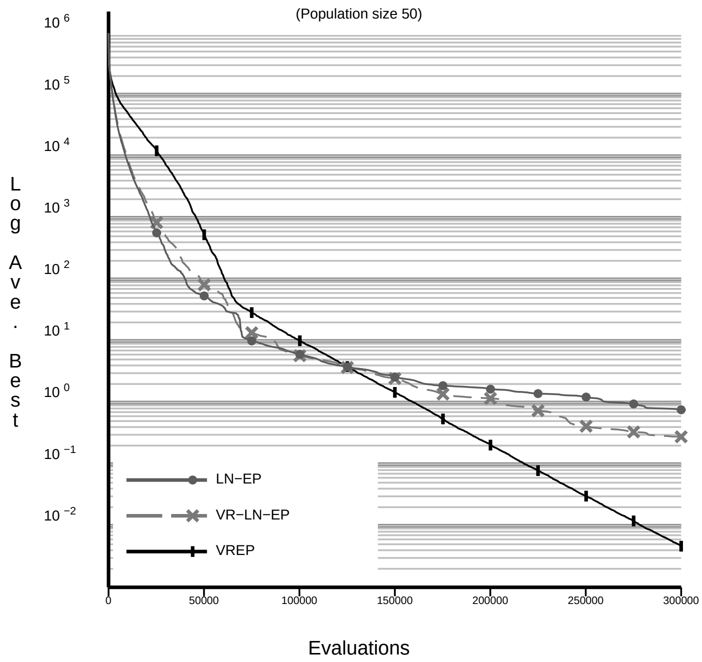
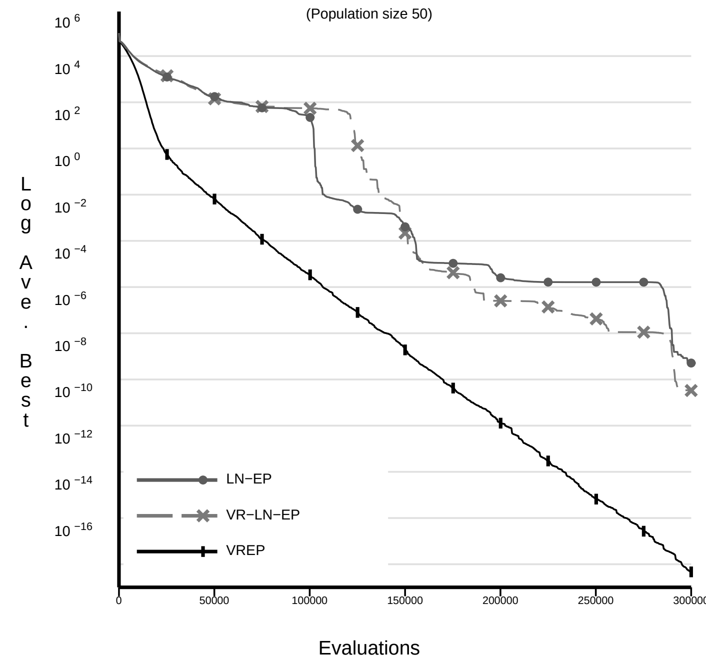


Figure 6. : Dejong function 4 (10 Dimensions)



Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	8.06E-07	1.21E-06	5.58E-08	1.17E-05
VR-LN	5.80E-06	2.60E-05	1.09E-18	2.31E-04
LN-EP	2.63E-06	1.60E-05	2.05E-19	1.53E-04

Table 2. Best Values for Generation 1000/1000 for Sphere Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	4.71E+00	1.09E+00	2.37E+00	1.12E+01
VR-LN	1.96E+01	3.98E+01	1.28E-02	2.97E+02
LN-EP	1.60E+01	3.66E+01	5.06E-03	2.31E+02

Table 3. Best Values for Generation 1000/1000 for Rosenbrock Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	1.28E-02	1.05E-01	9.00E-08	1.05E+00
VR-LN	1.36E+00	1.06E+00	6.95E-16	4.39E+00
LN-EP	1.53E+00	1.10E+00	-5.00E-16	5.02E+00

Table 4. Best Values for Generation 1000/1000 for Bohachevsky Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	3.23E-04	1.59E-04	8.89E-05	1.17E-03
VR-LN	2.27E+00	2.31E+00	1.26E-09	1.21E+01
LN-EP	2.60E+00	2.26E+00	1.03E-09	1.14E+01

Table 5. Best Values for Generation 1000/1000 for Ackley Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	5.37E+00	2.22E+00	1.34E-03	1.02E+01
VR-LN	2.21E+01	1.41E+01	2.98E+00	8.56E+01
LN-EP	2.44E+01	1.34E+01	2.98E+00	7.16E+01

Table 6. Best Values for Generation 1000/1000 for Rastrigin Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	4.65E-03	1.97E-03	1.04E-03	1.14E-02
VR-LN	2.78E-01	1.34E+00	5.24E-08	1.34E+01
LN-EP	7.63E-01	2.69E+00	1.97E-09	2.02E+01

Table 7. Best Values for Generation 1000/1000 for Sphere hull Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	0.00E+00	0.00E+00	0.00E+00	0.00E+00
VR-LN	7.56E-02	3.22E-01	0.00E+00	2.00E+00
LN-EP	7.56E-02	3.22E-01	0.00E+00	2.00E+00

Table 8. Best Values for Generation 1000/1000 for Dejong f3Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	4.66E-18	1.07E-17	2.72E-20	7.95E-17
VR-LN	3.32E-10	3.58E-09	7.76E-34	3.93E-08
LN-EP	5.13E-09	5.57E-08	1.01E-33	6.10E-07

Table 9. Best Values for Generation 1000/1000 for Dejong f4Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	2.93E-01	7.08E-02	1.59E-01	6.04E-01
VR-LN	1.57E+01	9.21E+00	9.00E-01	4.54E+01
LN-EP	1.60E+01	1.01E+01	1.29E+00	5.12E+01

Table 10. Best Values for Generation 1000/1000 for Schaffer Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	2.39E+00	6.40E-01	7.85E-01	4.00E+00
VR-LN	3.90E+00	4.88E+00	3.23E-01	3.89E+01
LN-EP	4.01E+00	5.02E+00	4.35E-01	4.98E+01

Table 11. Best Values for Generation 1000/1000 for Yip & Pao Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	9.35E-05	8.33E-05	2.68E-06	5.68E-04
VR-LN	1.04E+01	1.12E+02	9.91E-16	1.23E+03
LN-EP	1.52E-01	1.53E+00	1.09E-19	1.67E+01

Table 12. Best Values for Generation 1000/1000 for Schwefel 1 Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	5.05E+02	2.13E+02	1.20E-04	1.05E+03
VR-LN	9.62E+02	2.97E+02	1.18E+02	1.66E+03
LN-EP	9.20E+02	3.04E+02	1.18E+02	1.72E+03

Table 13. Best Values for Generation 1000/1000 for Schwefel 2 Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	8.06E-02	5.03E-02	2.43E-03	2.76E-01
VR-LN	2.42E-01	2.85E-01	9.86E-03	2.23E+00
LN-EP	2.87E-01	3.75E-01	2.21E-02	2.24E+00

Table 14. Best Values for Generation 1000/1000 for Griewangk Function

Algorithm	Ave. Best	S.Dev. Best	Min Best	Max Best
VREP	6.84E+00	4.29E+00	3.65E-07	1.80E+01
VR-LN	1.25E+01	7.00E+00	1.11E-23	3.75E+01
LN-EP	1.32E+01	6.43E+00	1.45E-25	3.19E+01

Table 15. Best Values for Generation 1000/1000 for FM matching Function

Function	Rank Sum of VREP	Z statistic	Minimum α for $\mu_{VREP}>\mu_{LN-EP}$
Schaffer	6105	-15.2813	0
Dejong f3	6441	-14.6486	0
Rastrigin	7297	-13.0368	0
Bohachevsky	8533	-10.7094	0
FM matching	9133	-9.57964	0
Griewangk	9594	-8.71159	0
Schwaefel 2	9819	-8.28792	1.11E-16
Ackley	10083	-7.79081	3.33E-15
Rosenbrock	10351	-7.28617	1.61E-13
Yip & Pao	11997	-4.1868	1.42E-05
Sphere hull	15662	2.714309	0.996679
Dejong f4	18583	8.214481	1
Sphere	20235	11.32516	1
Schwaefel 1	20849	12.4813	1

Table 16. Wilcoxon Rank-Sum 2-Sample Location Test Statistics for $\mu_{VREP}>\mu_{LN-EP}$

Function	Rank Sum of VR+LN	Z statistic	Minimum α for $\mu_{VR+LN}>\mu_{LN-EP}$
Dejong f3	8173	-11.3873	0
FM matching	12642	-2.97228	0.001478
Yip & Pao	12682	-2.89696	0.001884
Sphere hull	13125	-2.0628	0.019566
Ackley	13298	-1.73705	0.04119
Bohachevsky	13641	-1.09118	0.137596
Rastrigin	13784	-0.82192	0.205561
Rosenbrock	14186	-0.06496	0.474102
Schwaefel 1	14405	0.347409	0.635858
Schwaefel 2	15124	1.701269	0.955554
Griewangk	15205	1.85379	0.968115
Sphere	15835	3.040064	0.998817
Schaffer	15952	3.260372	0.999444
Dejong f4	17221	5.649869	1

Table 17. Wilcoxon Rank-Sum 2-Sample Location Test Statistics for $\mu_{VR+LN}>\mu_{LN-EP}$

5 Discussion

In looking at the data in Tables 2 through 15, we note first that the VREP system consistently displays the lowest standard deviation among final best solution values. Upon initial review, this result appears somewhat unremarkable, given that the variance of the VREP system is being strictly scheduled. However, this increased stability implies that the operation of selection in combination with the reactive mutation operator in VREP is extremely consistent for a given landscape, even across the inconsistencies of population initialization.

The apparent overlapping nature of the distributions of final best solution fitnesses as indicated by the mean and standard deviation measurements of these final best solutions as well as the minimum and maximum values, causes difficulty in reaching any reasonable conclusions as to the relative utility of these two approaches except that they appear somewhat equipollent. This overlap of distributions also renders graphic displays such as those in Figures 1 through 6 suspect, since the arithmetic mean may easily be skewed by a few outliers which converged prematurely. For this reason we have elected to use the Wilcoxon rank-sum 2-sampled location test to further analyze the data.

The Wilcoxon rank-sum statistics give a much clearer picture of the relative strengths of the three algorithms. For example, consider the mean and standard deviation values for the final best values for the Dejong f4 function. VREP appears to have a clear advantage over classic LN-EP, and the VR+LN system seems to have a potential advantage as well. However, upon inspecting the minimum and maximum values, this advantage appears less certain since the minimum value obtained using LN-EP is on par with that obtained by the VR+LN system, and several orders of magnitude better than that obtained with VREP. Finally, if we inspect the Wilcoxon rank-sum comparison statistics, we observe that LN-EP consistently outperformed both competing algorithms on the Dejong f4 function. Thus, while VREP shows a lower mean due, in part, to its reduced standard deviation, LN-EP consistently obtained better results; however, being less consistent, occasionally the LN-EP system obtained results several orders of magnitude worse which skewed the mean for LN-EP.

In view of the Wilcoxon test data in Table 16, VREP clearly outperformed LN-EP on a number of test functions, while the VR+LN system showed only marginal improvement for all but the Dejong f3 function. LN-EP performed better than VREP on the Sphere hull, Dejong f4, Sphere, and Schwefel functions.

Finally, we note that by using only a fixed schedule for the recapture target we may have unduly retarded the progress at the end of VREP search when we would expect the primary need to be for local hill-climbing behavior. Specifically, we note that the behavior of the VREP algorithm on the Schaffer, Griewangk, Ackley, Sphere Hull, and Schwefel 1 functions did not show any clear signs of having reached convergence, as is evidenced by the average performance. For example, consider the data presented in Figure 1 and Figures 3 through 6.

6 Conclusion

Modifying the mutation size selection algorithm in EP to react in proportion to the actions of selection produces a system which appears at least equipollent to classical LN-EP for the tested functions, and which clearly outperforms LN-EP for certain test functions. However, such modifications do not appear to work equally well when used in combination with other mutation operators, at least in the manner in which such combination was tested here. Further, the results indicate that the operation of this reactive mutation strategy within an EP framework produces extremely consistent search characteristics.

The ability to dynamically adjust the search characteristics of the system, through modification of the recapture target, may be of great practical importance. Dramatic increases of the performance potential of the VREP algorithm may be possible by judicious use of more complex annealing schedules for the recapture target.

Like the standard EP mutation size selection algorithm, the VR mutation algorithm has the potential for use with mutation distributions other than Gaussian. The only requirement is the ability to predict the projected impact on the variance of the population for a given mutation size. Hence, it may be possible to also apply the VR algorithm to fast EP (FEP) [Yao 96] style systems.

7 References

Bäck, T. and H.-P. Schwefel, (1993). "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, Vol. 1., No. 1., pp. 1-23.

McClave, J. T. and F. H. Dietrich, II (1985). *Statistics*, Third Ed., Dellen Publishing Company.

Patton, A., T. Dexter, E. D. Goodman, and W. F. Punch III, (1998). "On the Application of Cohort-Driver operators to continuous Optimization Problems Using Evolutionary Computation," *Evolutionary Programming VII*, Springer-Verlag, pp. 671-681.

Salomon, R. (1996). "Performance Degradation of Genetic Algorithms under Coordinate Rotation," *Proceedings of the Fifth Annual Conference on Evolutionary Programming*, MIT Press., pp. 153-161.

Saravanan, N., D.B. Fogel and K.M. Nelson (1995). "A Comparison of Methods for Self-Adaptation in Evolutionary Algorithms," *BioSystems*, **36**, pp 157-166.

Schwaefel, H.-P., (1981). *Numerical Optimization of Computer Models*, John Wiley, Chichester, UK.

Tsutsui, S., A. Ghosh, D. Corne, and Y. Fujimoto, (1997). "A Real Coded Genetic Algorithm with an Explorer and an Exploiter Populations," *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan-Kaufmann., pp. 230-237.

Yang, J., J. Horng, and C. Kao (1997), "A Continuous Genetic Algorithm for Global Optimization," *Proceedings of the Seventh International Conference on Genetic Algorithms*, Morgan-Kaufmann., pp. 230-237.

Yao, X., G. Lin and Y Liu, (1996). "An analysis of evolutionary algorithms based on neighborhood and step sizes", *Evolutionary Programming VI*, Springer-Verlag, pp. 297-307.

Yip, D. and Y. H. Pao, (1995). "Combinatorial Optimization with Use of Guided Evolutionary Simulated Annealing," *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, pp. 239-246.