Publication P7

D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska
"A general framework for statistical performance comparison of evolutionary computation algorithms"
in
Proc. of the IASTED International Conference on Artificial Intelligence and Applications
Innsbruck, Austria, 2006, pp. 7-12.

© 2006 ACTA Press

Reprinted with kind permission of ACTA Media.

A GENERAL FRAMEWORK FOR STATISTICAL PERFORMANCE COMPARISON OF EVOLUTIONARY COMPUTATION ALGORITHMS

David Shilane
University of California,
Berkeley
Division of Biostatistics
140 Warren Hall
Berkeley, CA 94720
USA
E-mail:
dshilane@berkeley.edu

Jarno Martikainen
Helsinki University of
Technology
Institute of Intelligent
Power Electronics
P. O. Box 3000
FI-02015 HUT
FINLAND
E-mail: jkmartik@cc.hut.fi

Sandrine Dudoit
University of California,
Berkeley
Division of Biostatistics
140 Warren Hall
Berkeley, CA 94720
USA
E-mail:
sandrine@stat.berkeley.edu

Seppo J. Ovaska
Helsinki University of
Technology
Institute of Intelligent
Power Electronics
P. O. Box 3000
FI-02015 HUT
FINLAND
E-mail: ovaska@ieee.org

ABSTRACT

This paper proposes a statistical methodology for comparing the performance of evolutionary computation algorithms. A two-fold sampling scheme for collecting performance data is introduced, and this data is assessed using a multiple hypothesis testing framework relying on a bootstrap resampling procedure. The proposed method offers a convenient, flexible, and reliable approach to comparing algorithms in a wide variety of applications.

KEY WORDS

Evolutionary computation, statistics, performance comparison, two-fold sampling, bootstrap, multiple hypothesis testing.

1. Introduction

Evolutionary algorithms (EAs) [1,2] are used to estimate the solution to difficult optimization problems. An EA's result is determined by a stochastic process with two sources of variation: the fitness of the input or initial population (used interchangeably), and the algorithm's improvements on this fitness produced via the mechanisms of selection, reproduction, and mutation. EAs are often hand-crafted to meet the requirements of a particular problem because no single optimization algorithm can solve all optimization problems competitively [3]. When alternative algorithms are proposed, their relative efficacies should be assessed. This paper seeks to provide a general methodology for comparing the performance of evolutionary algorithms based on statistical sampling and hypothesis testing.

In [4], Christenssen and Wineberg explain the use of appropriate statistics in artificial intelligence and propose non-parametric tests to verify the data's output distribution. In [5], Flexer proposes many guidelines for statistical evaluation of neural networks that can also be applied to evolutionary computation. Czarn et. al [6] discuss the use of the Analysis of Variance (ANOVA) in comparing the performance of EAs. However, such procedures rely upon distributional assumptions that are not necessarily valid and limit the class of performance metrics that can be used.

An EA's initial population consists of a set of starting values for the evolution process. Most previous EA performance comparisons have only considered results for a single initial population or even provided different inputs for each algorithm studied. Supplying different single inputs to each EA may result in a *founder effect,* in which a population's initial advantage is continually propagated to successive generations. Furthermore, relying upon a single input can at best determine the plausibility of preferring one candidate EA to another given suitable initial conditions. We can alleviate these issues by assessing relative performance over each of a representative sample of initial populations.

For each particular sampled input, performance differences can be assessed using a hypothesis test. Student's *t*-statistics [7] are commonly used to test the equality of two population means. However, the parametric *t*-test assumes that the data are normally distributed. If this assumption is not valid, the resulting inference may not be meaningful. Therefore, we require a more general and objective framework for statistical comparison of evolutionary computation algorithms.

This paper proposes a two-fold sampling scheme to perform repeated EA trials at each of a representative sample of possible inputs. The candidate EAs' efficacies will be assessed in a multiple hypothesis testing framework that relies upon bootstrap resampling [8] to estimate the joint distribution of the test statistics. This methodology will establish a procedure for fair comparison of EAs that can be considered general in the following aspects: first, the results do not rely heavily on a single advantageous input. Second, the bootstrap-based testing procedure is applicable to any data-generating distribution and requires no *a priori* model assumptions. Finally, this methodology can be applied to essentially any function of the data collected, so the researcher is free to choose how performance should be evaluated.

The paper is organized as follows: Section 2 describes the two-fold sampling scheme for data collection. Section 3 introduces performance comparison in a multiple hypothesis testing framework. Section 4 shows how to use the bootstrap to estimate the test statistic's underlying distribution. Section 5 introduces a variety of multiple testing procedures. Section 6 provides an example comparing the performance of two EAs seeking to minimize Ackley's function. Section 7 discusses further applications of statistics in EAs and concludes the paper.

2. Data Collection Using Two-Fold Sampling

An EA's *initial population* or *input* is a set of individuals that serve as starting values for the algorithm. Because an EA's result depends both on its input fitness and its efficacy given this initial population, data must be collected in a *two-fold sampling* scheme. We will first generate a representative sample of initial populations, and then, for each of these inputs, we will perform a number of trials of each candidate EA. If we specify g, the number of generations each EA is allowed to evolve, the data are collected via the following algorithm:

1. Generate M initial populations of H individuals. Each individual is described by a D-dimensional vector of traits (or *genes*). The value of D corresponds to the dimension of the domain of the function to be optimized. The value of the d^{th} gene of the h^{th} individual of the m^{th} population is labeled $y_{mh,d}$. (When referring to an overall population y_m or single individual y_{mh} within that population, the unnecessary indices will be dropped.) Populations of individuals are constructed from genes randomly generated from an associated D-dimensional distribution function P. When all genes of all individuals are independent and equally likely, a uniform distribution is used. The resulting $M\times H$ individuals (and hence the M populations) are independent and identically distributed (*i.i.d.*).

2. Because an EA with a particular input follows a stochastic process, we will sample results for each of the inputs generated in Step 1. For each initial population y_m, perform n_a, $a \in \{1,2\}$, trials of algorithm a. Save the optimal result observed after g generations as the $[m,i]^{th}$ entry of an $M\times n_a$ data matrix $X_a(g)$.

   The values n_a specify the sample size, and M represents the number of hypotheses, each of which correspond to an initial population. In general, one should collect as much data as possible given the computational constraints of the problem.

3. Multiple Hypothesis Testing

For any comparison, we must first select the theoretical *parameter of interest* $\mu_a(y_m,g)$, which in this setting is an EA's measure of performance given initial population y_m and the number of generations g. A typical choice for $\mu_a(y_m,g)$ is the expected optimum fitness after g generations. This parameter will be estimated by a *statistic* $\hat{\mu}_a(y_m,g)$, which is just a function of the observed data $X_a(g)$. When the expected optimum fitness is the parameter of interest, the corresponding statistic is the sample mean:

$$\hat{\mu}_a(y_m,g)=\frac{1}{n_a}\sum_{i=1}^{n_a} X_a\left(g\right)_{[m,i]} \text{ , } m=1,\dots, M;\ a=1,2 \qquad (1)$$

and the corresponding estimate of the data's variance is

$$\hat{\sigma}^2_a(y_m,g)=\frac{1}{n_a}\sum_{i=1}^{n_a}\left(X_a(g)_{[m,i]}-\hat{\mu}_a(y_m,g)\right)^2,\ m=1,\dots, M;\ a=1,2 \quad (2)$$

A multiple hypothesis testing framework is needed to compare algorithmic performance based on the data collected in Section 2. Typically we wish to demonstrate that the EAs differ significantly in performance given an initial population, so a skeptical null hypothesis would assume for each input that no difference in performance exists between the two algorithms. This corresponds to the multiple null hypotheses

$$H_m : \mu_1(y_m,g)-\mu_2(y_m,g)=0\ ;\ m=1,\dots,\ M \qquad (3)$$

We will test (3) at multiple significance level α (e.g. FWER 0.05 – Section 5). The null hypothesis can take many forms depending on the researcher's priorities. For example, one may wish to show that a new algorithm's expected optimal fitness after g generations is greater than that of an established standard or that its performance falls in a particular range.

To test (3), we must construct test statistics and corresponding decision rules that reject the null hypotheses when the test statistics exceed a to-be-determined cut-off. We will test each component null hypothesis using a two-sample t-statistic:

$$t_m=\frac{\hat{\mu}_1(y_m,g)-\hat{\mu}_2(y_m,g)}{\sqrt{\dfrac{\hat{\sigma}^2{}_1(y_m,g)}{n_1}+\dfrac{\hat{\sigma}^2{}_2(y_m,g)}{n_2}}}\ ,\ m=1,\ \dots,\ M \qquad (4)$$

In order to specify cut-offs that probabilistically control a suitably defined *Type I error rate* (Section 5), we must estimate the underlying distribution of (4). When the data are assumed to follow a Normal distribution, Student's t-distribution is appropriate. However, if this assumption is not valid, the test statistics may not follow any mathematically simple distribution. Under any of these circumstances, the distribution of (4) can be estimated using the bootstrap.

4. Using the Bootstrap in Hypothesis Testing

The bootstrap is a simulation-based resampling method that uses the data collected to derive a statistic's estimated distribution in a mathematically simple but computationally intensive way. This estimate is consistent, asymptotically efficient, and does not rely upon parametric assumptions, so it is widely applicable to many problems in statistical inference [8]. In a hypothesis testing environment, we can estimate the underlying joint distribution of (4) via the following algorithm [9]:

1. Specify a number B (typically at least 10,000 for multiple hypothesis testing) of bootstrap iterations.

2. Let $n=n_1+n_2$. Concatenate the columns of $X_1(g)$ and $X_2(g)$ to form an $M\times n$ data matrix $X(g)$. For each $b\in 1,\ldots, B$, sample n columns at random with replacement from $X(g)$ and store this resampling in an $M\times n$ matrix $X^{\#b}(g)$. The first n_1 columns of $X^{\#b}(g)$ are considered the bootstrap resampled data for $a=1$ at this iteration, and the final n_2 columns correspond to $a=2$.

3. For $b=1,\ldots,B$, compute statistics from the resampled data $X^{\#b}(g)$ of Step 2 that correspond to the test statistics (4). Store these values in an $M\times B$ matrix $T(g)$.

4. Obtain an $M\times B$ matrix $Z(g)$ by shifting $T(g)$ about its row means and scaling by its row standard deviations.

$$Z(g)[m,b]=\sqrt{\min\left[1,\frac{1}{\frac{1}{B}\sum_{b=1}^{B}\left(T[m,b]-\frac{1}{B}\sum_{b=1}^{B}T[m,b]\right)^2}\right]}\left(T[m,b]-\frac{1}{B}\sum_{b=1}^{B}T[m,b]\right)$$

$m=1,\ldots, M$; $b=1,\ldots,B$. (5)

The estimate of (4)'s joint distribution is given by the empirical distribution function of the columns of (5). For hypothesis testing applications, the bootstrap is implemented in the ***MTP*** function of the **R** statistical programming environment's **multtest** package [10,11].

5. Test Results and Statistical Inferences

The significance level α, the observed test statistics (4), and the bootstrap test statistic matrix $Z(g)$ constitute the input to a Multiple Testing Procedure (MTP). In this setting, a variety of methods that reflect a diversity of attitudes toward risk are available. Statistical tests can generate two types of errors: a Type I (or *false positive*) error occurs when a true null hypothesis is incorrectly rejected, and a Type II (*false negative*) error occurs when a false null is not rejected. When testing M hypotheses simultaneously, as in (3), we define the random variables:

V: The total of Type I errors (not observed) (6)
R: The number of rejected hypotheses (observed) (7)

Classical MTPs sought to control the Family-Wise Error Rate (FWER). More recent research has been developed to control the generalized Family-Wise Error Rate (gFWER), False Discovery Rate (FDR), and the Tail Probability for the Proportion of False Positives (TPPFP), which are defined in Table 1.

Table 1: Type I Error Rates.

Type I Error Rate	Parameter	Quantity Controlled
FWER	-	$Pr(V > 0)$
gFWER	k (*int*)	$Pr(V > k)$
FDR	-	$E[V / R]$
TPPFP	q (%)	$Pr(V/R > q)$

As summarized in [9,11,12,13,14,15], Table 2 lists a selection of available MTPs for each Type I error rate. The results of a multiple hypothesis test can be summarized in terms of rejection regions for the test statistics, confidence regions for the parameters of interest, and *adjusted* p-values [12]. The rejection region provides a set of values for which each hypothesis H_m of (3) is rejected while controlling the desired Type I error rate at level α. A corresponding set of 1-α confidence regions may also be constructed.

Table 2: MTPs by Type I Error Rate.

Type I Error Rate	Multiple Testing Procedures
FWER	SS maxT, SS minP, SD maxT, SD minP, Bonferroni, Holm, Hochberg, SS Sidak, SD Sidak
gFWER	Augmentation Procedure
FDR	Conservative Augmentation, Restrictive Augmentation, BY, BH
TPPFP	Augmentation Procedure

Adjusted p-values define the minimum value of α necessary to reject each hypothesis H_m of (3). Adjusted p-values from different testing procedures controlling the same Type I error rate may be directly compared, with smaller values reflecting a less conservative test [14]. The multiple testing procedures above are automated in the ***MTP*** function of the **R multtest** package [10,11]. The user needs only supply the data, the value of α, the form of the null hypothesis, the test statistic, the Type I error rate to control, and the MTP.

6. Example: Ackley's Function Minimization

6.1. Defining Ackley's Function

We seek to compare two candidate EAs that approximate the minimum of a D=10-dimensional Ackley function [2]. With $y_{mh}=(y_{mh,1},\ldots,y_{mh,D})$ as in Section 2, Ackley's multimodal function, which achieves a known minimum at the origin, is defined as:

$$fit\,(y_{mh}) = -c_1 \cdot \exp\left(-c_2\sqrt{\frac{1}{D}\sum_{d=1}^{D} y_{mh,d}^2}\right) - \exp\left(\frac{1}{D}\cdot\sum_{d=1}^{D}\cos\left(c_3 \cdot y_{mh,d}\right)\right) + c_1 + \exp(1) \quad (8)$$

with the following parameters supplied for this example: $c_1 = 20,\ c_2 = 0.2, c_3 = 2\pi, D = 10, -20 \le y_{mhd} \le 30.$

6.2. Candidate EAs Ack1 and Ack2

The algorithms *Ack1* and *Ack2* were devised to estimate the minimum of (8). Each EA takes an input population y_m as described in Section 2. Each individual of this population has associated fitness $fit(y_{mh})$ given by (8). At each generation, the algorithm includes a selection, reproduction, and mutation phase:

Selection: Sort and re-label the H individuals in order of increasing $fit(y_{mh})$, h=1,...,H. The $H/2$ best-fit individuals – those with the smallest values of (8) – are selected for breeding, while the other members will not reproduce. (*Floor*(H/2) may be used for odd values of H.)

Reproduction: For h=1,…,H/4, pair individuals $y_{m,2h-1}$ and $y_{m,2h}$ for mating. Each pair produces two offspring to replace individuals not selected. For the first child (c=1), a uniform random variable $weight_1$ is selected on (0,1), and the second child (c=2) receives $weight_2$ = 1 - $weight_1$. Traits are inherited (vector-wise) by the weighted average

$$y_{m,\,H/2+2(h-1)+c} = weight_c \cdot y_{m,\,2h-1} + \left(1 - weight_c\right)\cdot y_{m,\,2h} \quad (9)$$

Mutation: Each offspring $y_{H/2+1},...,y_H$ will mutate independently in a single gene at birth with probability mut_a. When mutation occurs, the gene is selected from a uniform random variable on 1,..., D, and this trait is assigned a uniform random variable on (-20, 30). In this example, mutation probabilities for *Ack1* and *Ack2* were mut_1=0.1 and mut_2=0.8, respectively. Because at most one of the D=10 genes may mutate, the population's expected proportion of mutating genes is therefore 1% and 8% for the two algorithms, respectively. Otherwise, *Ack1* and *Ack2* were identical.

The initial population is considered the completion of the reproduction and mutation phases for the 0th generation. The process of selection, reproduction, and mutation repeats a total of g generations, and the estimate to the minimum of (8) is given in the resulting population by

$$\text{RESULT} = min\left[fit\left(y_{mh}\right)\right], h \in \left[1, \ldots, H\right] \quad (10)$$

The value of (10) observed for EA a on the i^{th} trial given initial population y_m is stored as the $[m,i]^{th}$ entry of $X_a(g)$. Because the reproduction and mutation phases introduce variability at each generations, the value $X_a(g)[m,i]$ is a random variable.

It should be noted that *Ack1* and *Ack2* were designed solely to provide an illustrative example of our comparison methodology. Different population sizes, reproduction schemes, or mutation rates may lead to improved estimates of (8)'s minimum.

6.3. Study Design and Results

Using the two-fold sampling approach of Section 2, we generated M=100 initial populations $y_1,...,y_M$, each consisting of H=100 individuals of D=10 dimensions. Each individual's traits were initialized using pseudo-random number generation from a uniform distribution on the interval (-20, 30). The function (8) was used to assess each individual's fitness. Then, for each initial population m=1,..., M, we collected optimum fitness data on n_1=n_2=50 trials of the EAs. On each trial, both *Ack1* and *Ack2* were allowed to reproduce for g=10000 generations. The resulting data are displayed in Fig. 1.

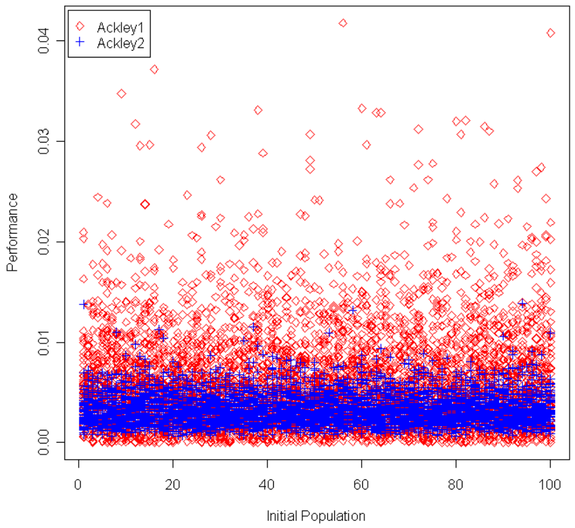

Fig. 1: Performance data for *Ack1* and *Ack2* by initial population.

Figure 2 shows the average performance of the algorithms for each initial population. Though *Ack2* produces a better mean value of (10) than *Ack1* at each initial population, Fig. 1 shows that *Ack1* is capable of producing competitive results for some trials across all inputs. Furthermore, *Ack1* appears to exhibit greater variance than *Ack2* in its estimates. Therefore, it is not immediately clear that *Ack2* performs better than *Ack1*.

We performed two-sided tests of the multiple hypotheses (3) corresponding to no difference in performance between *Ack1* and *Ack2* at each given input versus the alternative of unequal performance. Note that one could also perform one-sided tests that designate one candidate EA as superior to the other.

The hypotheses (3) were tested using the **multtest** package [10,11] of **R** at FWER level α=0.05 based on the data collected and the statistic (4). We first employed the FWER-controlling SS maxT MTP at nominal level α=0.05. Figure 3 shows several summary plots of the SS maxT results. The first plot shows how R (7) grows as a function of α. The second plot shows the ordered SS maxT adjusted p-values. This curve indicates that 92 hypotheses are rejected at level α=0.05. The third plot shows how the SS maxT adjusted p-value for a hypothesis decreases with the absolute value of the test statistic. Here the adjusted p-value value approaches 0.05 as the test statistic increases toward -2.75. The final plot of Fig. 3 shows the unordered adjusted SS maxT p-values, which allow one to identify the initial populations that result in significant ($p.adj$<0.05) performance differences.

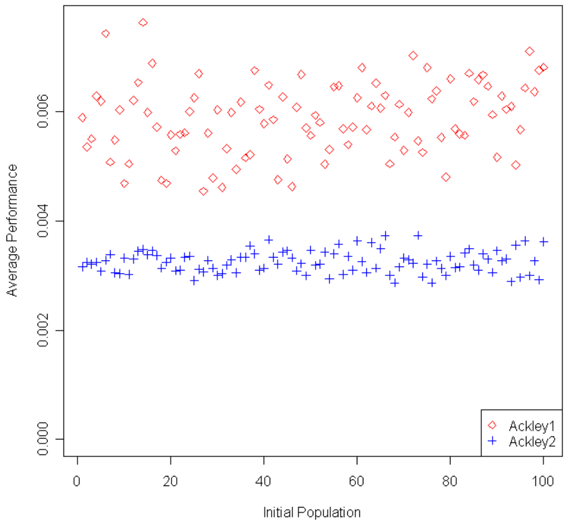

Fig. 2: Average Performance by Initial Population.

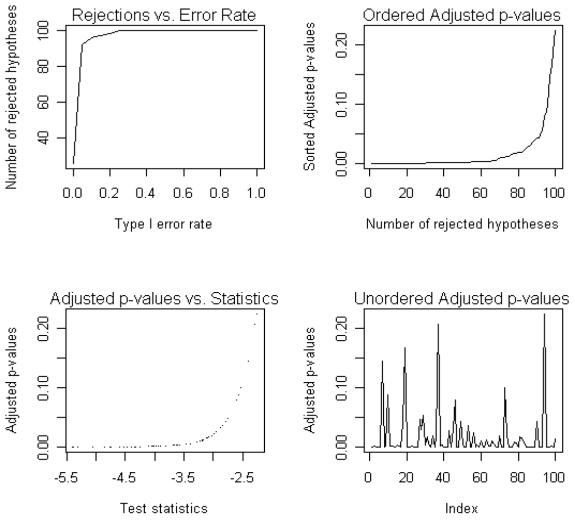

Fig. 3: Summary displays for SS maxT testing.

We then implemented a selection of the MTPs listed in Table 2 to test (3) under different Type I error rates. Table 3 displays the number of hypotheses rejected by each MTP at varying levels of α. The following procedures reject all 100 hypotheses at level α=0.05: Holm, Hochberg, SidakSD, BY, and BH.

For the gFWER and TPPFP augmentation procedures, the question remains whether 8 false positives or an 8% rate of false positives, respectively, is tolerable in testing for EA performance differences. This question is epistemological in nature and must be decided by the scientific community. In practice, a maximum value for these parameters should be established before comparison takes place. Although the particular benchmark is somewhat arbitrary (much like the choice of α=0.05 in hypothesis testing), establishing a uniform standard is necessary for future studies.

Table 3: The number of rejected hypotheses as a function of α for a selection of MTPs.

Rate	MTP	α=0.01	α=0.03	α=0.05	α=0.07	α=0.10
FWER	SS maxT	72	86	92	93	96
	Bonf.	75	90	93	95	95
	Holm	100	100	100	100	100
	Hochberg	100	100	100	100	100
	SS Sidak	83	90	94	95	95
	SD Sidak	100	100	100	100	100
GFWER	gFWER (5)	72	86	92	93	96
	gFWER (10)	77	91	97	98	100
FDR	AMTP Conserv.	66	77	86	90	96
	AMTP Rest.	66	77	86	90	96
	BY	99	100	100	100	100
	BH	100	100	100	100	100
TPPFP	Tppfp (.07)	72	86	92	93	96
	Tppfp (.10)	77	92	99	100	100

The results of the MTPs suggest a performance difference between *Ack1* and *Ack2*. On each of M=100 sample input populations, *Ack2* achieved a smaller average observed minimum. All MTPs rejected at least 86 of the M=100 hypotheses at level α=0.05, and a number of procedures rejected all hypotheses at level α=0.01. Therefore, based upon the data collected, we conclude that *Ack2* significantly outperforms *Ack1* in estimating the minimum of (8) when the expected optimum fitness is the parameter of interest. Because the two algorithms only differed in their mutation probabilities, it appears that increased mutation is beneficial in this application.

7. Conclusion

Although this paper's methodology provides a general approach to EA performance comparison, the reader should be cautioned that issues of sample size cannot be neglected. In particular, the bootstrap approximation of (4)'s joint distribution grows more accurate as the values B and n_a increase. In practice, researchers may choose to collect as much data as a pre-specified time limit will allow. Data-adaptive study designs may also be implemented to halt data collection once a pre-specified level of statistical power is achieved.

The framework proposed in this paper allows the researcher to choose the parameter of interest in an EA comparison. When parameters other than the expected optimum fitness are used (such as the median, 75^{th} percentile, or other quantile estimates), our methodology is applicable provided that the necessary data are collected and appropriate estimating statistics (1), null hypotheses (3), and test statistics (4) are chosen. In crafting an EA for a particular optimization problem, this paper's methodology can be used iteratively to select the best among a set of candidate parameter values for quantities such as the mutation rate, population size, and selection proportion. When three or more EAs are simultaneously compared, a null hypothesis of equality in means may be tested using F statistics.

If competing algorithms draw from different input sets, then testing average results from representative input samples in a single (M=1) test may be considered. When the input sets are identical, an alternative to the approach of this paper may choose to average all trials in a single hypothesis test provided that all inputs are *i.i.d*. The choice of which approach to use is philosophical: this paper assumes that EAs should be compared using the same input sample. In this setting, the parameter of interest is the expected performance given the initial population. This allows the algorithm to be assessed solely on its own merits without any possibility of a founder effect. However, if one views the input generation and resulting evolution as inextricably linked in the same algorithm, then a single hypothesis testing framework may be more appropriate, and this paper's methodology is otherwise applicable. In this scenario, the parameter of interest shifts to the unconditional expectation of performance, and each EA should generate inputs independently for comparison trials. Though a single test may simplify the interpretation of performance differences, this approach is not applicable when inputs are dependently generated and lacks the appeal of direct performance comparison on the same trial inputs.

The researcher may also wish to compare EAs as a function of time by collecting data at regular generational intervals. Displaying performance curves and confidence regions graphically may allow one to quickly determine decision criteria and search for clues about an algorithm's rate of convergence and asymptotic result. Finally, an EA's efficacy should be considered in terms of both performance and computational complexity. Researchers may consider performing a comparison in which each candidate algorithm is allowed to run for the same amount of time instead of the same number of generations to satisfy both objectives simultaneously.

Acknowledgment

The authors gratefully acknowledge Frances Tong for her helpful comments and suggestions during the process of preparing this manuscript.

References

[1] D. B. Fogel, *Evolutionary computation: Toward a new philosophy of machine intelligence* (Piscataway, NJ: IEEE Press, 2000).

[2] T. Bäck, *Evolutionary algorithms in theory and practice* (New York, NY: Oxford University Press, 1996).

[3] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, 1(1), 1997, 67-82.

[4] S. Christensen and M. Wineberg, Using appropriate statistics – Statistics for artificial intelligence, *Tutorial Program of the 2004 Genetic and Evolutionary Computation Conference*, Seattle, WA, 2004, 544-564.

[5] A. Flexer, Statistical evaluation of neural network experiments: Minimum requirements and current practise, *Proceedings of the 13th European Meeting on Cybernetics and Systems Research. Austrian Society for Cybernetic Studies*, Vienna, Austria, 1996, vol. 2, 1005-1008.

[6] A. Czarn, C. MacNish, K. Vijayan, B. Turlach, and R. Gupta, Statistical exploratory analysis of genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 8(4), 2004, 405-421.

[7] J. S. Milton and J. C. Arnold, *Introduction to probability and statistics: Principles and applications for engineering and the computing science* (New York, NY: McGraw-Hill, 1990).

[8] B. Efron and R. Tibshirani, *An introduction to the bootstrap* (Boca Raton, FL: Chapman and Hall, 1994).

[9] K. S. Pollard et. al, Test statistics null distributions in multiple testing: Simulation studies and applications to genomics, Technical Report #184, *U.C. Berkeley Division of Biostatistics Working Paper Series*. (Submitted, *Journal de la Société Francaise de Statistique*.)

[10] WWW-pages of the R project for statistical computing. [Cited 1.9.2005]. Available at <http:// r-project.org>.

[11] K. S. Pollard, S. Dudoit, and M. J. van der Laan, Multiple testing procedures: The Mulltest package and applications to genomics, R. C. Gentleman, V. J. Carey, W. Huber, R. Irizarry, and S. Dudoit (eds), *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* (New York:, NY: Springer-Verlag, 2005), Chapter 15, 249-271.

[12] K. S. Pollard and M. J. van der Laan, Resampling-based multiple testing: Asymptotic control of type I error and applications to gene expression data, *U.C. Berkeley Division of Biostatistics Working Paper Series*. Working Paper 121, 2003.

[13] S. Dudoit, M. J. van der Laan, and K. S. Pollard, Multiple testing. Part I. Single-step procedures for control of general type I error rates, *Statistical Applications in Genetics and Molecular Biology*, 3(1), Article 13, 2004.

[14] S. Dudoit and M. J. van der Laan (In preparation), *Multiple testing procedures and applications to genomics*, Springer Series in Statistics.

[15] M. J. van der Laan, S. Dudoit, and K. S. Pollard, Augmentation procedures for control of the generalized family-wise error rate and tail probabilities for the proportion of false positives, *Statistical Applications in Genetics and Molecular Biology*, 3(1), Article 15, 2004.