Studying and Tackling Noisy Fitness in Evolutionary Design of Game Characters

J. J. Merelo[1], Pedro A. Castillo[1], Antonio Mora[1], Antonio Fernández-Ares[1],
Anna I. Esparcia-Alcázar[2], Carlos Cotta[3] and Nuria Rico[4]
[1] *University of Granada, Department of Computer Architecture and Technology,*
ETSIIT and CITIC, 18071 Granada, Spain
[2] *S2Grupo, Valencia, Spain*
[3] *Universidad de Málaga, Departamento de Lenguajes y Sistemas Informáticos, Málaga, Spain*
[4] *Universidad de Granada, Depto. Estadística e Investigación Operativa, Granada, Spain*

Keywords: Evolutionary Algorithms, Noisy Optimization Problems, Games, Strategy Games.

Abstract: In most computer games as in life, the outcome of a match is uncertain due to several reasons: the characters or assets appear in different initial positions or the response of the player, even if programmed, is not deterministic; different matches will yield different scores. That is a problem when optimizing a game-playing engine: its fitness will be noisy, and if we use an evolutionary algorithm it will have to deal with it. This is not straightforward since there is an inherent uncertainty in the true value of the fitness of an individual, or rather whether one chromosome is better than another, thus making it preferable for selection. Several methods based on implicit or explicit average or changes in the selection of individuals for the next generation have been proposed in the past, but they involve a substantial redesign of the algorithm and the software used to solve the problem. In this paper we propose new methods based on incremental computation (memory-based) or fitness average or, additionally, using statistical tests to impose a partial order on the population; this partial order is considered to assign a fitness value to every individual which can be used straightforwardly in any selection function. Tests using several hard combinatorial optimization problems show that, despite an increased computation time with respect to the other methods, both memory-based methods have a higher success rate than *implicit* averaging methods that do not use memory; however, there is not a clear advantage in success rate or algorithmic terms of one method over the other.

1 INTRODUCTION

In our research on the optimization of the behavior of bots or game strategies, we have frequently found that the fitness of a bot is *noisy*, in the sense that repeated evaluations will yield different values (Mora et al., 2012) which is a problem since fitness is the measure used to select individuals for reproduction. If we look at in in a more general setting, noise in the fitness of individuals in the context of an evolutionary algorithm has different origins. It can be inherent to the individual that is evaluated; for instance, in (Mora et al., 2012) a game-playing bot (autonomous agent) that includes a set of application rates is optimized. This results in different actions in different runs, and obviously different success rates and then fitness. Even comparisons with other individuals can be affected: given exactly the same pair of individuals, the chance of one beating the other can vary in a wide range. In other cases like the one presented in the MADE environment, where whole worlds are evolved (García-Ortega et al., 2014) the same kind of noisy environment will happen. When using evolutionary algorithms to optimize stochastic methods such as neural networks (Castillo et al., 1999) using evolutionary algorithms the measure that is usually taken as fitness, the success rate, will also be noisy since different training schedules will result in slightly different success rates.

The examples mentioned above are included actually in one or the four categories where uncertainties in fitness are found, fitness functions with intrinsic noise. These four types include also, according to (Jin and Branke, 2005) approximated fitness functions (originated by, for instance, surrogate models); robust functions, where the main focus is in finding

Merelo J., Castillo P., Mora A., Fernández-Ares A., Esparcia-Alcázar A., Cotta C. and Rico N..
Studying and Tackling Noisy Fitness in Evolutionary Design of Game Characters.
DOI: 10.5220/0005085700760085
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2014), pages 76-85
ISBN: 978-989-758-052-9

Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

values with high tolerance to change in initial evaluation conditions, and finally dynamic fitness functions, where the *inherent* value of the function changes with time. Our main interest will be in the first type, since it is the one that we have actually met in the past and which has led to the development of this work.

At any rate, in this paper we will not be dealing with actual problems; we will try to simulate the effect of noise by adding to the fitness function Gaussian noise centered in 0 and $\sigma = 1, 2, 4$. We will deal mainly with combinatorial optimization functions with noise added having the same shape and amplitude, that we actually have found in problems so far. In fact, from the point of view of dealing with fitness, these are the main features of noise we will be interested in.

The rest of the paper is organized as follows: next we describe the state of the art in the treatment of noise in fitness functions. The method we propose in this paper, called Wilcoxon Tournament, will be shown in Section 4; experiments are described and results shown in Section 5; finally its implications are discussed in the last section of the paper.

2 STATE OF THE ART

The most comprehensive review of the state of the art in evolutionary algorithms in *uncertain* environments was done by Jin and Branke in 2005 (Jin and Branke, 2005), although recent papers such as (Qian et al., 2013) include a brief update of the state of the art. In that first survey of evolutionary optimization by Jin and Branke in uncertain environments this uncertainty is categorized into noise, robustness, fitness approximation and time-varying fitness functions, and then, different options for dealing with it are proposed. In principle, the approach presented in this paper was designed to deal with the first kind of uncertainty, noise in fitness evaluation, although it could be argued that there is uncertainty in the true fitness as in the third category. In any case it could be applied to other types of noise. In this situation, several solutions were been proposed and explained in the survey (Jin and Branke, 2005). These will be explained next.

An usual approach is just disregard the fact that the fitness is noisy and using whatever value is returned a single time or after re-evaluation each generation. This was the option in our previous research in games and evolution of neural networks (Castillo et al., 1999; Mora et al., 2010; Merelo-Guervós et al., 2001) and leads, if the population is large enough, to an *implicit averaging* as mentioned in (Jin and Branke, 2005). In fact, evolutionary algorithm selection is also stochastic, so noise in fitness evaluation will have the same effect as randomness in selection or a higher mutation rate, which might make the evolution process easier and not harder in some particular cases (Qian et al., 2013). In fact, Miller and Goldberg proved that an infinite population would not be affected by noise (Miller and Goldberg, 1996) and Jun-Hua and Ming studied the effect of noise in convergence rates (Jun-hua and Ming, 2013), proving that an elitist genetic algorithm finds at least one solution with a lowered convergence rate. But populations are finite, so the usual approach is to increase the population size to a value bigger than would be needed in a non-noisy environment. This has also the advantage that no special provision or change to the implementation has to be made; but a different value of a single parameter.

Another more theoretically sound way is using a statistical central tendency indicator, which is usually the *average*. This strategy is called *explicit averaging* by Jin and Branke and is used, for instance, in (Jun-hua and Ming, 2013). Averaging decreases the variance of fitness but the problem is that it is not clear in advance what would be the sample size used for averaging (Aizawa and Wah, 1994). Most authors use several measures of fitness for each new individual (Costa et al., 2013), although other averaging strategies have also been proposed, like averaging over the neighbourhood of the individual or using resampling, that is, more measures of fitness in a number which is decided heuristically (Liu et al., 2014). This assumes that there is, effectively, an average of the fitness values which is true for Gaussian random noise and other distributions such as Gamma or Cauchy but not necessarily for all distributions. To the best of our knowledge, other measures like the median which might be more adequate for certain noise models have not been tested; the median always exists, while the average might not exist for non-centrally distributed variables. Besides, most models keep the number of evaluations is fixed and independent of its value, which might result in bad individuals being evaluated many times before being discarded; some authors have proposed *resampling*, that is, re-evaluate the individuals a number of times to increase the precision in fitness (Rada-Vilela et al., 2014), which will effectively increase the number of evaluations and thus slow down the search. In any case, using average is also a small change to the overall algorithm framework, requiring only using as new fitness function the average of several evaluations. We will try to address this in the model presented in this paper.

These two approaches that are focused on the evaluation process might be complemented with changes to the selection process. For instance, using a threshold (Rudolph, 2001) that is related to the noise characteristics to avoid making comparisons of individuals that might, in fact, be very similar or statistically the same; this is usually called *threshold selection* and can be applied either to explicit or implicit averaging fitness functions. The algorithms used for solution, themselves, can be also tested, with some authors proposing, instead of taking more measures, testing different solvers (Cauwet et al., 2014), some of which might be more affected by noise than others.

Any of these approaches do have the problem of statistical representation of the *true* fitness, even more so if there is not such a thing, but several measures that represent, *all of them* the fitness of an individual. This is what we are going to use in this paper, where we present a method that uses resampling via an individual memory and use either explicit averaging or statistical tests like the non-parametric Wilcoxon test. First we will examine and try to find the shape of the noise that actually appears in games; then we will check in this paper what is the influence on the quality of results of these two strategies and which one, if any, is the best when working in noisy environments.

3 NOISE IN GAMES: AN ANALYSIS

In order to measure the nature of noise, we are going to use the Planet Wars game, that has been used as a framework for evolving strategies, for instance, in (Mora et al., 2012). In this game, initial position of the players is random with the constraint that they should be far enough from each other; other than that, any planet in the game can be an initial position. Besides, in the strategy used in that game, actions are not deterministic, since every player is defined by a set of probabilities to take one course of action.

An evolutionary algorithm with standard parameters was run with the main objective of measuring the behavior of fitness. A sample of ten individuals from generation 1, and another 10 from generation 50 were extracted. The fitness of each individual was measured 100 times. The main intention was also to see how noise evolved with time. Intuitively we thought that, since the players become better with evolution, the noise and thus the standard deviation would decrease. However, what we found is plotted in Figure 1, which shows a plot of the standard deviation in both generations and illustrates the fact that the spread of fitness values is *bigger* as the evolution proceeds, going from around 0.15 to around 0.20. which might be a bit misleading since the average values of the fitness increase at the same time, but it implies that the noise level might be around 20% of the *signal* in these kind of problems.

But we were also interested in checking whether, in fact, the normal distribution is the best fit for the fitness measures. We tested three distributions: Gamma, Weibull and normal (Gaussian) distribution after doing an initial test that included Cauchy and Exponential. All this analysis was done using the library `fitrdist` in R, and data as well as scripts needed to do it are publicly available. After trying to fit these three distribution to data in generation 1 and 50, we analyzed goodness-of-fit using the same package and the `gofstat` function. This function yields several measures of goodness, including the Akaike Information Criterion and the Kolmogorov-Smirnov statistic.

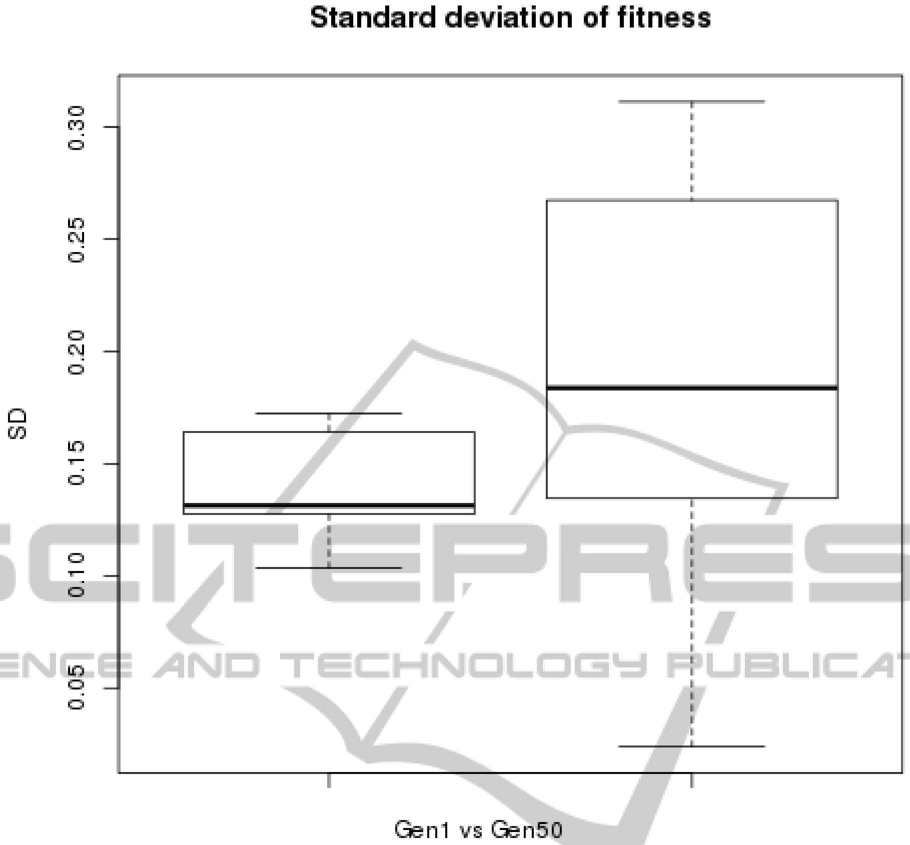
What we found was that, in all cases, Gamma is the distribution that better fits the data. That does not mean that the noise effectively follows this distribution, but that if it can be said to follow one, it's this one. In fact, just a few individuals have a good fit (to 95% accuracy using the Kolmogorov-Smirnov), and none of them in generation 50. The fit for an individual in the first generation that does follow that distribution (individual 8) is shown in figure 2.

This figure also shows that, even if it is skewed, its skewness is not too high which makes it close to the standard distribution (which is considered a good approximation if *k* ¿ 10).

Some interesting facts that can be deduced from these measures is that in general, fitness is skewed and has a high value. Besides, it follows a gamma distribution, which, if we wan to model noise accurately, should be the one used. However, we are rather interested in the overall shape of noise so since the skewness value of the gamma distribution is rather high we will use, in this paper, a Gaussian noise to simulate it. This will be used in the experiments shown below.

4 FITNESS MEMORY AND STATISTICAL SIGNIFICANT DIFFERENCES

As indicated in previous section, most explicit averaging methods use several measures to compute fitness as an average, with resampling, that is, additional measures, in case comparisons are not statistically significant. In this paper we will introduce a fitness *memory*, which amounts to a resampling every generation an individual survives. An individual is born with a fitness memory of a single value, with memory size increasing with *survival* time. This is actually a



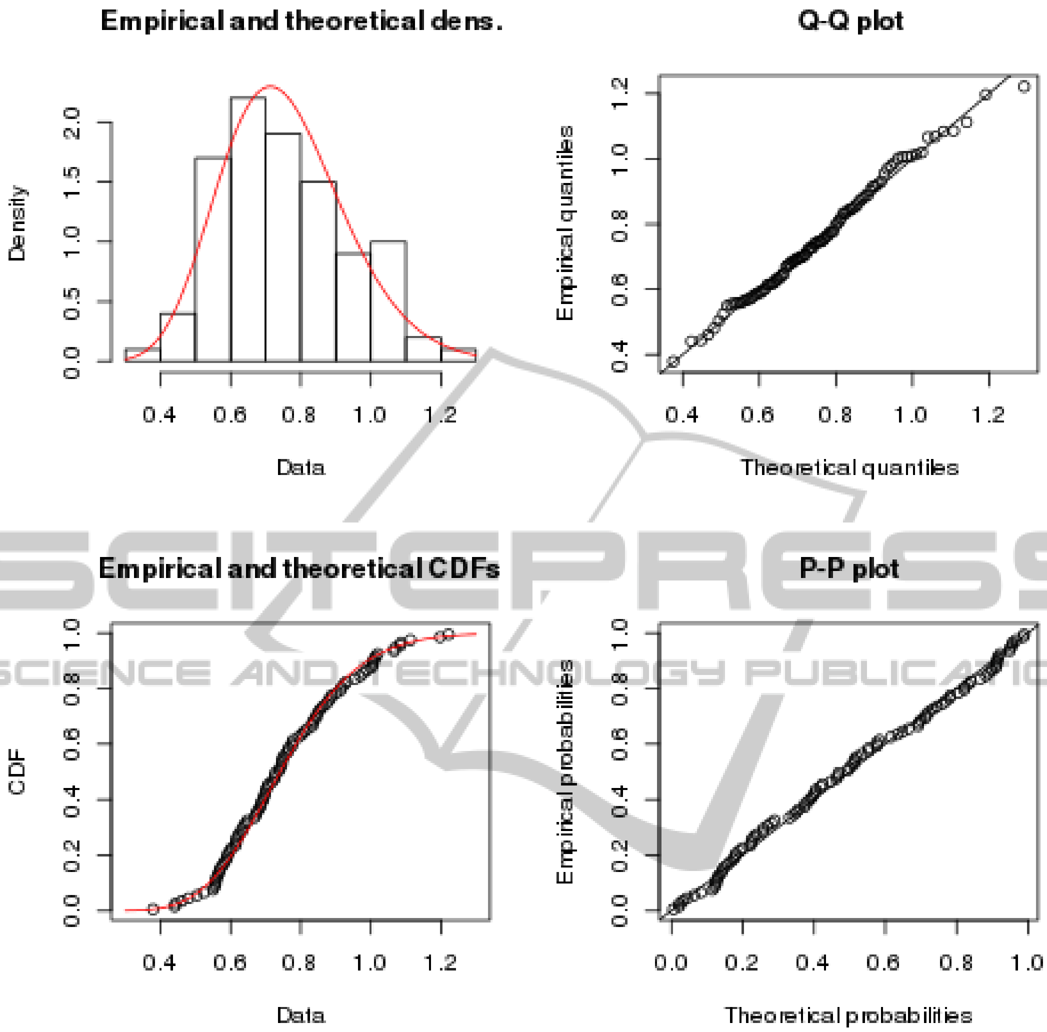
SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 1: Boxplot of standard deviation of noise fitted to a normal distribution, left for the first generation and right for the 50th. Fitness averages around 1.

combination of an implicit and an explicit evaluation strategy: *younger* individuals are rejected outright if their fitness computed after a single evaluation is not enough to participate in the pool, while *older* ones use several measures to compute average fitness, which means that averages will be a more precise representation of actual value. As evolution proceeds, the best individuals will, effectively, have an underlying non-noisy best value. We will call this method *incremental temporal average* or ITA.

However, since average is a single value, selection methods might, in fact, select as better individuals some that are not if the comparison is not statistically significant; this will happen mainly in the first and middle stages of search, which might effectively eliminate from the pool or not adequately represent individuals that constitute, in fact, good solutions. That is why we introduce an additional feature: using Wilcoxon test (Wilcoxon, 1945) for comparing not the average, but all fitness values attached to an individual. This second method introduces a partial order in the population pool: two individuals might be different (one better than the other) or not. There are many possible ways of introducing this partial order in the evolutionary algorithm; however, what we have done is to pair individuals a certain number of times (10, by default) and have every individual score a point every time it is better than the other in the couple; it will get a point less if it is the worse one. An individual that is better that all its couples will have a fitness of 20; one whose comparisons are never significant according to the Wilcoxon test will score exactly 10, the same as if it wins as many times as it loses, and the one that always loses will score 0. We will call this method Wilcoxon-test based partial order, or WPO for short.

Initial tests, programmed in Perl using `Algorithm::Evolutionary` (Merelo-Guervs et al., 2010) and available with an open source license at http://git.io/a-e were made with these two types of algorithms and the Trap function (Merelo-Guervs;, 2014), showing the best results for the WPO method and both of them being better that the implicit average method that uses a single evaluation per individual, although they needed more time and memory. Since it does not need to perform averages



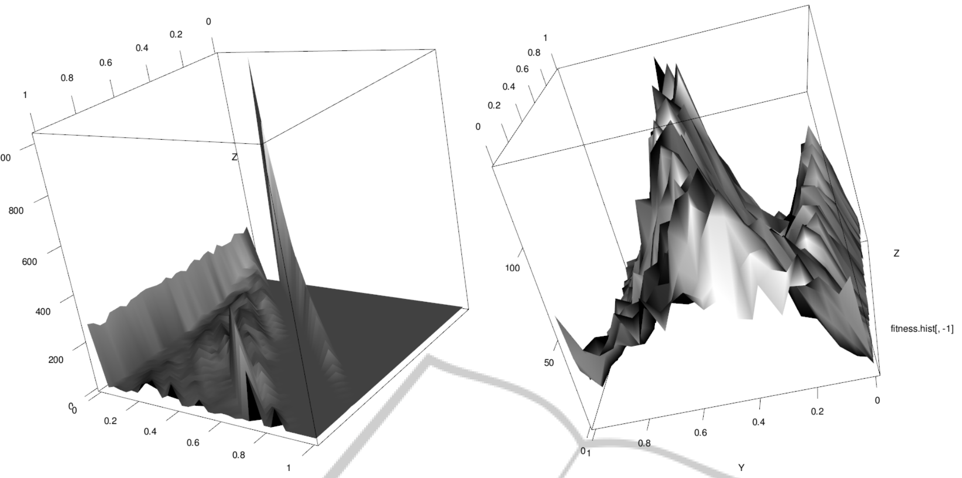
SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 2: Fit of the fitness value of an individual in the first generation of the evolutionary algorithm to a gamma distribution, showing an histogram in the top left corner, CDFs in the bottom left corner and quantile-quantile and percentile-percentile plots in the right hand side.

or make additional fitness measures every generation, it is twice as fast as the next method, the one that uses explicit average fitness. An exploration of memory sizes (published in http://jj.github.io/Algorithm-Evolutionary/graphs/memory/ and shown in Figure 3 for a typical run) showed that they are distributed unevenly but, in general, there is no single memory size overcoming all the population. Besides, distribution of fitness, published at http://jj.github.io/Algorithm-Evolutionary/graphs/fitness-histo/ shows a distribution with most values concentrated along the middle (that is, fitness equal to 10 or individuals that cannot be compared with any other, together with a few with the highest fitness and many with the lowest fitness. Besides showing that using the partial order for individual selection is a valid strategy, it also shows that a too greedy selection method would eliminate many individuals that might, in fact, have a high fitness. This will be taken into account when assigning parameter values to the evolutionary algorithm that will be presented next.

5 RESULTS

ITA and WPO have been tested using two well-known benchmarks, the deceptive bimodal Trap (Deb and Goldberg, 1992) function and the Massively Multimodal Deceptive Problem (Goldberg et al., 1992) (MMDP). We chose to use just these two functions were chosen because they have different fitness landscapes, are usually difficult for an evolutionary algorithm and have been extensively used for testing other



(a) Memory size

(b) Fitness distribution

Figure 3: (Left) 3D plot of the distribution of memory sizes for a single execution of the Wilcoxon-test based partial order. (Right) 3d plot of the distribution of fitness values along time for the WPO method on the Trap function.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Table 1: Common evolutionary algorithm parameters.

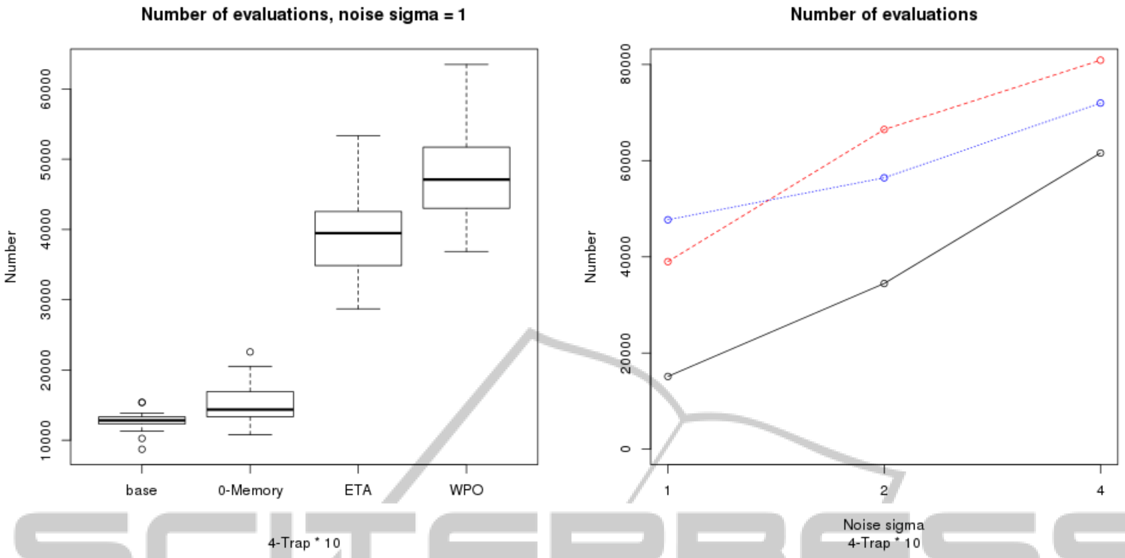
Parameter	Value
Chromosome length	40 (Trap) 60 (MMDP)
Population size	1024
Selection	2 tournament selection
Replacement rate	50%
Mutation rate	20%
Crossover rate	80%
Max evaluations	200K (Trap) 1 Million(MMDP)
Stopping criterion	Non-noisy best found or max evaluations reached

kind of operators and algorithms.

Several methods were tested: a baseline algorithm without noise to establish the time and number of evaluations needed to find the solution, a 0-memory (implicit average) method that uses noisy fitness without making any special arrangement, ITA and WPO. Evolutionary algorithm parameters and code for all tests were the same, except in one particular case: we used 2-tournament with 50% replacement, 20% mutation and 80% crossover, $p = 1024$ and stopping when the best was found or number of evaluations reached. This was 200K for the Trap, which used 40 as chromosome size, and 1M for MMDP, which used 60 as chromosome size; these parameters are shown in Table 1. We have also used an additive Gaussian noise centered in zero and different σ, which is independent of the range of variation of the fitness values. By default, noise will follow a normal distribution with center in 0 and $\sigma = 1$.

All tests use the `Algorithm::Evolutionary` library, and the scripts are published, as above, in the GitHub repository, together with raw and processed results. The evolutionary algorithm code used in all cases is exactly the same except for WPO, which, since it needs the whole population to evaluate fitness, needed a special reproduction and replacement library. This also means that the replacement method is not exactly the same: while WPO replaces every generation 50% of the individuals, the rest evaluate new individuals before replacement and eliminate the worst 512 (50% of the original population). Replacement is, thus, less greedy in the WPO case, but we do not think this will be a big influence on result (although it might account for the bigger number of evaluations obtained in some cases), besides, it just needed a small modification of code and was thus preferred for that reason. All values shown are the result of 30 independent runs.

The results for different noise levels are shown in Figure 4. The boxplot on the left hand side compares the number of evaluations for the baseline method and the three methods with $\sigma = 1$. The implicit average method (labelled as 0-memory) is only slightly worse than the baseline value of around 12K evalua-



(a) Memory size (b) Fitness distribution

Figure 4: (Left) Comparison of number of evaluations for the 4-Trap x 10 function and the rest of the algorithms with a noise σ equal to 1. (Right). Plot of average number of evaluations for different methods: 0 memory (black, solid), ITA (red, dashed), WPO (blue, dot-dashed).

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

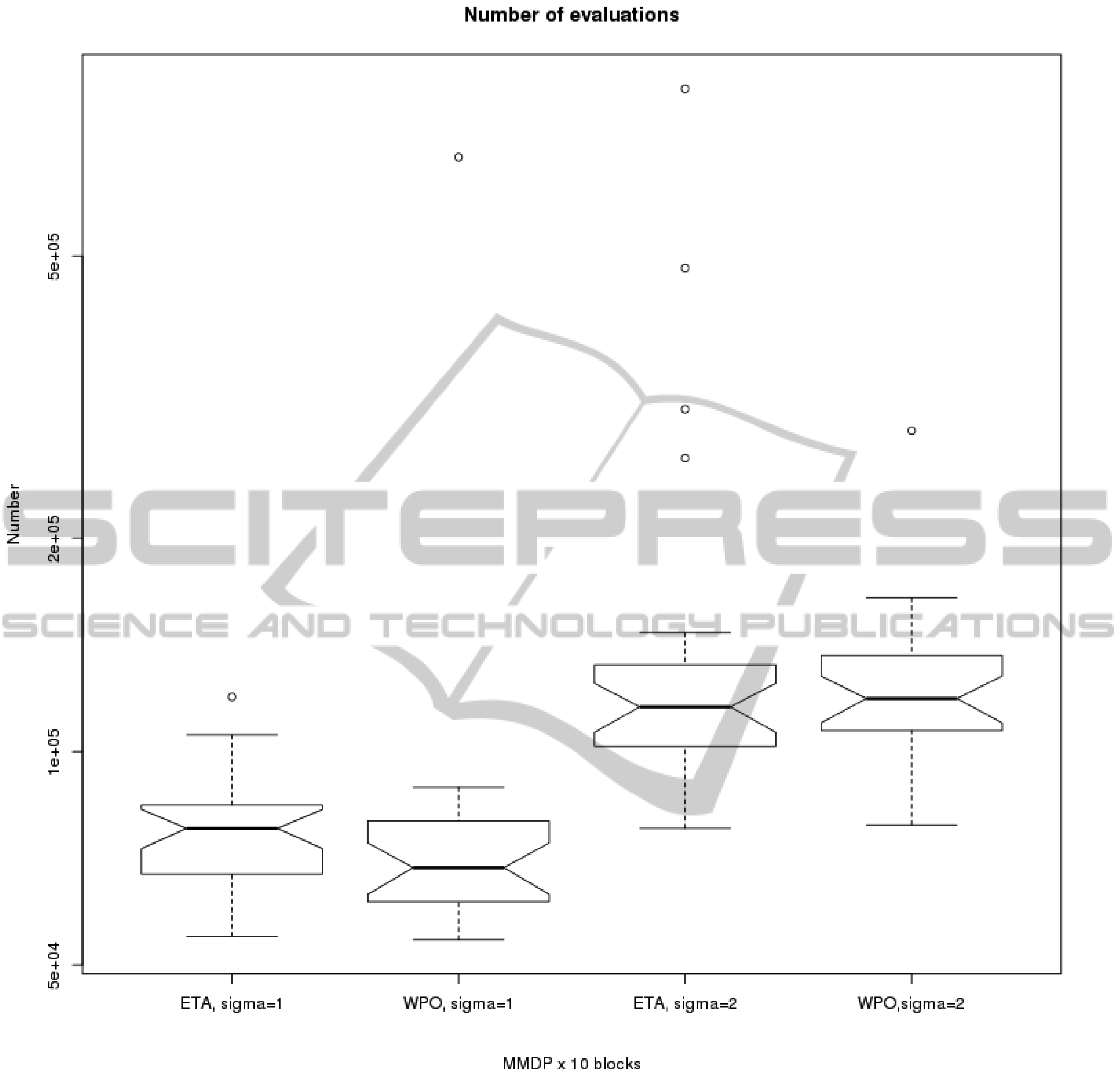
tions, with the ITA and WPO methods yielding very similar values which are actually worse than the 0-memory method. However, the scenario on the right, which shows how the number of evaluations scales with the noise level, is somewhat different. While the 0-memory method still has the smallest number of evaluations *for successful runs*, the success rate degrades very fast, with roughly the same and slightly less than 100% for $\sigma = 2$ but falling down to 63% for 0-memory and around 80% for ITA and WPO (86% and 80%). That is, best success rate is shown by the ITA method, but the best number of evaluations for roughly the same method is achieved by WPO.

These results also show that performance degrades quickly with problem difficulty and the degree of noise, that is why we discarded the 0-memory method due to its high degree of failure (a high percentage of the runs did not find the solution) with noise = 10% max fitness and evaluated ITA and WPO over another problem, MMDP with similar absolute σ, with the difference that, in this case, $\sigma = 2$ would be 20% of the max value, which is close to the one observed experimentally, as explained in the Section 3.

The evolutionary algorithm for MMDP used exactly the same parameters as for the Trap function above, except the max number of evaluations, which was boosted to one million. Initial tests with the 0-memory method yielded a very low degree of success, which left only the two methods analyzed in this paper for testing with MMDP. Success level was in all cases around 90% and very similar in all experiments; the number of evaluations is more affected by noise and shown in Figure 5. In fact, WPO and ITA have a very similar number of evaluations. It is statistically indistinguishable for $\sigma = 2$, and different only at the 10% level (p-value = 0.09668) for $\sigma = 1$, however, if we take the time needed to reach solutions into account, ITA is much faster since it does not apply 10*1024 statistical tests every generation. However, WPO is more robust, with a lower standard deviation, in general, at least for high noise levels. However, both methods obtain a good result with a much higher success rate than the implicit fitness (0-memory) method. Besides, ITA and APO incorporate explicit fitness evaluation naturally into the population *resampling* only surviving individuals. This accounts for a predictable behaviour of the algorithm, since the number of evaluations per generation is exactly the population size, which is important for optimization processes with a limited budget.

6 CONCLUSIONS

In this paper we have introduced two methods to deal with the problem of noisy fitness functions. The two methods, ITA, based on re-evaluation of surviving individuals and WPO, which uses the Wilcoxon test to compare a sample of individuals and partial-order



SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 5: Number of evaluations for successful runs ITA and WPO needed for solving the MMDP problem with 6 blocks and different noise levels, $\sigma = 1, 2$.

them within the population, have been tested over two different fitness functions and compared with implicit average (or 0-memory) methods, as well as among themselves. In general, memory-based methods have much higher success rate than 0-memory methods and the difference increases with the noise level, with 0-memory methods crashing at noise levels close to 20% while ITA and WPO maintain a high success level.

It is difficult to choose between the two proposed methods, ITA and WPO. However, ITA is much faster since it avoids costly comparisons. It also has a slightly higher success rate, and the number of evaluations it needs to find the solution is only slightly worse; even if from the point of view of the evolutionary algorithm it is slightly less robust and slightly worse, it compensates the time needed to make more evaluations with the fact that it does not need to perform statistical tests to select new individuals.

However, this research is in initial stages. The fact that we are using a centrally distributed noise gives ITA an advantage since, in fact, comparing the mean of two individuals will be essentially the same as doing a statistical comparison, since when the number of measures is enough, statistical significance will be reached. In fact, with a small difference ITA might select as better an individual whose fitness is actually the same, something that would be correctly spotted by WPO, but, in fact, since there is an average selective pressure this is not going to matter in the long run.

It might matter in different situations, for instance in numerical optimization problems and also when noise follows an uniform distribution; behavior might in this case be similar to when noise levels are higher. These are scenarios that are left for future research, and destined to find out in which situations WPO is better than ITA and the other way round.

Besides exploring noise in different problems and modelling its distribution, we will explore different parameters. The first one is the number of comparisons in WPO. Initial explorations have proved that changing it from 5 to 32 does not yield a significant difference. Looking for a way to speed up this method would also be important since it would make its performance closer to ITA. Memory size could also be explored. Right now evaluations are always performed, but in fact after a number of evaluations are done comparisons will be statistically significant; it is difficult to know, however, which is this number, but in long runs it would be interesting to cap fitness memory size to a sensible number, or, in any case, see the effect of doing it.

ACKNOWLEDGEMENTS

This work has been supported in part by project ANYSELF (TIN2011-28627-C04-02 and -01). The authors would like to thank the FEDER of European Union for financial support via project ”Sistema de Informacin y Prediccin de bajo coste y autnomo para conocer el Estado de las Carreteras en tiempo real mediante dispositivos distribuidos” (SIPEsCa) of the ”Programa Operativo FEDER de Andaluca 2007-2013”. We also thank all Agency of Public Works of Andalusia Regional Government staff and researchers for their dedication and professionalism.

Our research group is committed to Open Science and the writing and development of this paper has been carried out in GitHub at this address https://github.com/JJ/wilcoxon-ga-ecta. We encourage you to visit, comment and to all kind of suggestions.

JUNTA DE ANDALUCIA

Agencia de Obra Pública de la Junta de Andalucía
CONSEJERÍA DE FOMENTO Y VIVIENDA

Andalucía
se mueve con Europa

PROYECTO COFINANCIADO
EN 80% CON FEDER
POR LA UNIÓN EUROPEA

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

Aizawa, A. N. and Wah, B. W. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, 2(2):97–122.

Castillo, P. A., González, J., Merelo-Guervós, J.-J., Prieto, A., Rivas, V., and Romero, G. (1999). G-Prop-III: Global optimization of multilayer perceptrons using an evolutionary algorithm. In *GECCO-99: Proceedings Of The Genetic And Evolutionary Computation Conference*, page 942.

Cauwet, M.-L., Liu, J., Teytaud, O., et al. (2014). Algorithm portfolios for noisy optimization: Compare solvers early. In *Learning and Intelligent Optimization Conference*.

Costa, A., Vargas, P., and Tinós, R. (2013). Using explicit averaging fitness for studying the behaviour of rats in a maze. In *Advances in Artificial Life, ECAL*, volume 12, pages 940–946.

Deb, K. and Goldberg, D. E. (1992). Analyzing deception in trap functions. In *FOGA*, volume 2, pages 98–108.

García-Ortega, R. H., García-Sánchez, P., and Merelo, J. J. (2014). Emerging archetypes in massive artificial societies for literary purposes using genetic algorithms. *ArXiv e-prints*. Available at http://adsabs.harvard.edu/abs/2014arXiv1403.3084G.

Goldberg, D. E., Deb, K., and Horn, J. (1992). Massive multimodality, deception, and genetic algorithms. In R. Männer and Manderick, B., editors, *Parallel Problem Solving from Nature, 2*, pages 37–48, Amsterdam. Elsevier Science Publishers, B. V.

Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions on Evolutionary Computation*, 9(3):303–317. cited By (since 1996)576.

Jun-hua, L. and Ming, L. (2013). An analysis on convergence and convergence rate estimate of elitist genetic algorithms in noisy environments. *Optik - International Journal for Light and Electron Optics*, 124(24):6780 – 6785.

Liu, J., Saint-Pierre, D. L., Teytaud, O., et al. (2014). A mathematically derived number of resamplings for noisy optimization. In *Genetic and Evolutionary Computation Conference (GECCO 2014)*.

Merelo-Guervós, J.-J., Prieto, A., and Morán, F. (2001). *Optimization of classifiers using genetic algorithms*, chapter 4, pages 91–108. MIT press. ISBN: 0262162016; draft available from http://geneura.ugr.es/pub/papers/g-lvq-book.ps.gz.

Merelo-Guervs;, J.-J. (2014). Using a Wilcoxon-test based partial order for selection in evolutionary algorithms with noisy fitness. Technical report, GeNeura group, university of Granada. Available at http://dx.doi.org/10.6084/m9.figshare.974598.

Merelo-Guervs, J.-J., Castillo, P.-A., and Alba, E. (2010). `Algorithm::Evolutionary`, a flexible Perl module for evolutionary computation. *Soft Computing*, 14(10):1091–1109. Accesible at http://sl.ugr.es/000K.

Miller, B. L. and Goldberg, D. E. (1996). Genetic algorithms, selection schemes, and the varying effects of noise. *Evolutionary Computation*, 4(2):113–131.

Mora, A. M., Fernández-Ares, A., Guervós, J. J. M., García-Sánchez, P., and Fernandes, C. M. (2012). Effect of noisy fitness in real-time strategy games player behaviour optimisation using evolutionary algorithms. *J. Comput. Sci. Technol.*, 27(5):1007–1023.

Mora, A. M., Montoya, R., Merelo, J. J., Snchez, P. G., Castillo, P. A., Laredo, J. L. J., Martnez, A. I., and Espacia, A. (2010). Evolving botś ai in unreal. In di Chio et al., C., editor, *Applications of Evolutionary Computing, Part I*, volume 6024 of *Lecture Notes in Computer Science*, pages 170–179, Istanbul, Turkey. Springer-Verlag.

Qian, C., Yu, Y., and Zhou, Z.-H. (2013). Analyzing evolutionary optimization in noisy environments. *CoRR*, abs/1311.4987.

Rada-Vilela, J., Johnston, M., and Zhang, M. (2014). Population statistics for particle swarm optimization: Resampling methods in noisy optimization problems. *Swarm and Evolutionary Computation*, 0(0):–. In press.

Rudolph, G. (2001). A partial order approach to noisy fitness functions. In *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*, volume 1, pages 318–325.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS