EA Stability Visualization: Perturbations, Metrics and Performance

Matthew J. Craven
SoCM, University of Plymouth
Drakes Circus, Plymouth
PL4 8AA, UK
matthew.craven@plymouth.ac.uk

Henri C. Jimbo
RISE, Waseda University
3-4-1 Okhubo, Shinjuku-Ku
Tokyo 165-8555, Japan
jimbo_maths@yahoo.com

ABSTRACT

It is well-known that Evolutionary Algorithms (EAs) are sensitive to changes in their control parameters, and it is generally agreed that too large a change may turn the EA from being successful to unsuccessful. This work reports on an experimental hybrid visualization scheme for the determination of EA stability according to perturbation of EA parameters. The scheme gives a visual representation of local neighborhoods of the parameter space according to a choice of two perturbation metrics, relating perturbations to EA performance as a variant of Kolmogorov distance. Through visualization and analysis of twelve thousand case study EA runs, we illustrate that we are able to distinguish between EA stability and instability depending upon perturbation and performance metrics. Finally we use what we have learnt in the case study to provide a methodology for more general EAs.

Categories and Subject Descriptors

D.2.8 [**Software Engineering**]: Metrics—*complexity measures, performance measures*; G.1.6 [**Optimization**]; I.4.10 [**Image Representation**]: *multidimensional*

General Terms

Algorithms, Experimentation, Performance

Keywords

Evolutionary Algorithm; performance; perturbation; stability; visualization

1. INTRODUCTION

Visualization exists to communicate data in a form understandable (by, or pleasing to) the eye, enabling the understanding of that data in such a way that the user is able to strike conclusions or make decisions which would be more difficult to make by other methods of communication. It is well-known that visualization has many applications, including assisting the understanding of Evolutionary Algorithm (EA) output, runs and states (which are all referred to as *data*). EA data in practice tends to be vast and multidimensional [13].

There are common techniques for static visualization of EA data in at most four dimensions, such as contour maps or heat maps, but for dimension numbers beyond four the problem becomes difficult. One school of thought is in favor of dimensionality reduction through statistical techniques. For example, linear techniques such as Principal Component Analysis (PCA) or Canonical Correlation Analysis, or nonlinear techniques such as Kernel PCA or Local Linear Embedding may be used (see [15] for a comparative review of the above techniques). Alternatively, just part of the information may be presented through a simple projection. Another school of thought favors judicious presentation of statistics or consequences of the data. The popular taxonomy of this type of visualization is due to [12], which considers that the course and state of the EA may be separately visualized.

Past work on visualization focussed mainly on summarizing the EA course and state, where parameters and other EA settings were assumed to be constants. In particular, there seems to be an absence of work on visualizing what happens to an EA when its parameters are changed, at random or otherwise. The common EA paradigm of sensitivity to parameter change is well known [4], it being generally agreed that "bad" values of control parameters may render the EA ineffective. Hence for effective EA performance it is intuitively useful to find some way of measuring long-term EA behavior given this sensitivity. Broadly this is referred to as measuring EA stability. In this work we report on a preliminary study concerning EA stability visualization.

There are various kinds of stability which may be measured, such as Lyapunov stability or Input-Output stability for dynamical systems [14] which have well-known mathematical treatments. However, there is a paucity of work dealing with stability of EAs (for example, [9]) in general, and via visualization in particular. We seek to characterize the long-run stability of EAs under perturbation of their control parameters by using a parametric visualization, giving some examples of the visualization in action on twelve thousand runs of a case study EA from previous work [3]. Specifically, we shall answer the following question:

> If the control parameters are changed from a "standard" set then how does the EA performance change, visually speaking?

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.

http://dx.doi.org/10.1145/2598394.2610549.

Our visualization scheme provides information to determine whether a given EA is stable under change of control parameters and, in particular, a boundary where the phase transition between stability and instability occurs. This is not always a simple transition, however, because "bad" parameter values may not be located on or close to a clear boundary (that is, such values may occur in unexpected locations in parameter space).

This paper is organized as follows. In Section 2, we review relevant work in EA visualization before specializing context and preparing our case study in Section 3. In Section 4 we give further details of the visualization and some results, before concluding the work in Section 5.

2. PREVIOUS WORK IN EA VISUALIZATION

In this section we review relevant work in EA visualization. There are generally two separate well-known methods of processing visualizations: online and offline. Online visualization refers to visualization while the EA is running and is more immediate. It may be useful in order to track the behavior of runs, leading to possible conclusions before the end of a run. Offline visualization is based on a summary of EA run information, or statistics collected after completion of the run. Some popular visualization works are as follows.

The work of [12] gives many examples of online EA visualization and also details GEATbx, a GA toolbox for MATLAB. Another work is that of [7], where in their visualization package EAVis the authors used a so-called evolution map that "holds various information about the evolution of a whole system over the time". The map is comprised of lines (representing mutations), boxes (inversion) and circles (crossover) positioned at random. The work of [6] introduced GAVEL, an offline Java package that assists in reverse tracking GA runs in order to uncover the lineage of good solutions. This enables generation of displays such as "number of chromosomes in the ancestry tree per generation" and "origin of genes in final solution".

The work of [2] exemplified the popular EA visualization tool Gonzo which enables the generation of most common visualizations (fitness-time graphs, basic search space visualization, schema highlighting). Other visualizations of this type have been performed, most notably those of [8] for GraphDice which visualized EA solution space and [10] for high-dimension fitness landscape analysis. The work of [8] was later built upon, combining stochastic optimization with GraphDice to produce EvoGraphDice [1]. EvoGraphDice combines dimensions in an evolutionarily learned manner, with user input, to facilitate dimensionality reduction.

In the work of [16], the authors visualized "mutually nondominating solutions" coming from a large number of dimensions in the Multi-Objective Evolutionary Algorithm context. Parameter space and objective space were visualized with the help of heatmaps and the RadViz mechanism. Their visualization was in the form of an n-vertex polygon, each vertex corresponding to an objective of the problem at hand. Finally, the work of [11] also considered visualizations of clusters of EA outputs. The authors attempted to measure solution sensitivity in an EA through a variation of standard deviation, but control parameters were not considered.

The above visualizations aim to tell the user more about the processes of the EA in an attempt to understand results. Distinct to the above works, we shall treat the EA as a black box and thus will not consider the visualization of process or objectives. We simply wish to determine EA performance stability, through visualization, under control parameter perturbation. Our visualization is presented as a "fusion" of parameter and performance metrics, and was borne out of an investigative approach to the mathematical aspects of EA stability. We wish to use the visualization to analyze the parameter space landscape, uncovering not only whether a particular EA is stable or not but also at what magnitude of parameter perturbation the EA makes a transition from stability to instability.

3. CASE STUDY AND EA SETUP

In this section we consider the EA presented in [3] with the intention of using it as a case study. The mentioned work details an EA to solve a cryptographic problem in groups. We shall not present a treatment of the mathematical problem here, but do direct the interested reader to the above work for further details. The EA has input of a pair of n-ary strings, output a choice of a pair of n-ary strings or "no solution reached" and is set according to the values of its control parameters. Each string represents a word in a particular group, G. To solve the problem, an exact solution is required. If the problem is not solved within s generations then the EA terminates.

Starting with a population size of q (in this work we take $q = 200$) these parameters control the numbers of individuals produced from the following operations (in this order): crossover, mutation (of which there are three types), selection and random immigration. This yields a six-dimensional parameter space, $S = \left\{(p_i)_{i=1}^{6} : \sum p_i = q, p_i \geq 0\right\}$, although for EAs with $n \geq 6$ parameters our resulting visualization may, of course, be generalized. Below we summarize the main notations, which are described in more depth in the article [5], first covering the parameter space and the metrics over it and, second, the EA performance measure.

3.1 Parameter Space

We may represent a parameter set as an ordered vector $\underline{p} = (p_1, p_2, \ldots, p_6) \in S$. For example, the parameter vector $\underline{p} = (22, 40, 21, 54, 39, 24)$ denotes that, out of each population, twenty-two individuals were produced from the previous population by crossover, forty by the first type of mutation, and so on. Given a parameter vector $\underline{p}$ (the *base vector*) for which the EA achieves a given level of performance we may define a perturbation in two ways. First we may define the vector $\underline{p}'$ that is produced from $\underline{p}$. As we assume the population size is constant, the sum of the vector $\underline{p}'$ will be preserved. Second, we may define the perturbation itself as $\underline{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_6)$ such that $\underline{p} + \underline{\varepsilon} = \underline{p}'$ by elementwise addition. It is easy to observe that $\sum_{i=1}^{6} \varepsilon_i = 0$. We now may define metrics over S. Two examples of such metrics producing non-negative integer outputs are the l_1- and l_∞-norms over S. As perturbation metrics, these may be written as

$$d_1\left(\underline{\varepsilon}\right) = \sum_{i=1}^{6} \left|\varepsilon_i\right|, \tag{1}$$

$$d_\infty\left(\underline{\varepsilon}\right) = \max_{i=1,\ldots,6} \left|\varepsilon\right|. \tag{2}$$

We naturally define the parameter vector $b-$neighborhoods

$$N_{1,b}\left(\underline{p}\right) = \left\{\underline{p}' \in S : \underline{p} + \underline{\varepsilon} = \underline{p}', d_1\left(\varepsilon\right) \leq b\right\} \quad (3)$$

$$N_{\infty,b}\left(\underline{p}\right) = \left\{\underline{p}' \in S : \underline{p} + \underline{\varepsilon} = \underline{p}', d_\infty\left(\varepsilon\right) \leq b\right\} \quad (4)$$

to denote the set of all parameter vectors up to and including a distance of b from the given base parameter vector $\underline{p}$ and metric. These neighborhoods will be our search spaces.

3.2 EA Performance Metric

There are several well-known performance metrics for EA runs, depending upon the structure of the problem at hand. For example, we may wish to find the average objective value, a solution within a specified tolerance of the optimum or a solution achieving a given objective value. In our work we take the EA performance to be the *generation count*, which is simply the number of required generations to find an exact solution to a given problem instance. From the generation count, we define the EA performance metric as an analogue of Kolmogorov distance

$$d_K\left(\underline{p}, \underline{p}'\right) = \left(E_r\left(\underline{p}\right) - E_r\left(\underline{p}'\right)\right)^2, \quad (5)$$

where $\underline{p}' \in N_{\bullet,b}\left(\underline{p}\right)$ is a parameter vector chosen from the b-neighborhood of $\underline{p}$. The mean generation count from r independent trials of the EA with the given parameter vector is denoted $E_r\left(\cdot\right)$ [4], with r the *resolution*. Given a bound $b > 1$ and a parameter vector $\underline{p}$, it is clear that the sizes of the above neighborhoods become large relatively quickly. For instance, for $b = 30$ we have $\left|N_{1,30}\left(\underline{p}\right)\right| = 1894007$ and $\left|N_{\infty,30}\left(\underline{p}\right)\right| = 844596302$, making sampling of whole neighborhoods inefficient and thus justifying the sampling strategy used in Section 4.1.

3.3 Setup

We used a short instance (instance (I1) of [3]), fixed for all experiments. This particular instance was used for efficiency purposes, as each figure in this paper is comprised of $1000r$ EA runs. To provide a comparison, we used two base parameter vectors for our experiments (more details are given in Section 4.2). In the next section we detail the visualization of EA performance and give experimental results.

4. VISUALIZATION AND RESULTS

In this section we first give details of the parameter space projection and describe the two distinct types of visualization that will be performed using that projection. We follow this with results of the EA runs, contrasted by parameter base vector and including visual EA stability analysis.

4.1 Explanation of Visualization

In [5] we introduced a mathematical approach to EA stability determination, and in this work, focus upon the visualization and evaluation of it. The codes to produce the visualization are implemented in MATLAB. As the means of evaluation of the visualization we introduce a visual notion of EA stability, judging stability by sight rather than by statistical methods. Our visualization is one of the EA performance landscape rather than of the solution space or objective functions, and is achieved through a projection

$$\pi : G^2 \times S \to \mathbb{R}^+ \times C, \quad (6)$$

where S is the parameter space, G^2 is the direct product of two copies of the group G (each containing one word of the instance) and C is the color space. The color space in MATLAB is a collection of 64 colors mapped to values in the range 0,1,...,63. We elected to use the standard heatmap, with small values in the range mapped to blue colors and large values to red colors.

We take the codomain of the projection π to be a section of EA output. From this we term our visualization to be one of *parameter-output space*. The visualization is with respect to distance of perturbation (rather than the perturbation itself, which would not be representable) and size of performance effect, combined in a 2-D representation. The first factor of the codomain of π is associated to the distance from the origin of the plot. We have the two choices of either mapping $d_\bullet\left(\varepsilon\right)$ or mapping $d_K\left(\underline{p}, \underline{p}'\right)$ onto $\mathbb{R}^+$. In the former case its image is restricted to the set of all positive integers. The second factor of the codomain of π, the color space, has the alternate metric mapped to it. In other words, we have two types of visualization, where each given parameter vector is represented by a colored dot.

Type 1: The distance of the dot from the center represents the size of perturbation under the parameter metric d_1 or d_∞. The Kolmogorov distance is represented by a color.

Type 2: The distance of the dot from the center represents the scaled Kolmogorov distance, and the color represents the size of perturbation with respect to the given parameter metric.

On the visualization there are concentric circles provided for ease of visualization and for convenience (the circles are not intended to convey a sense of location in the neighborhood). The samples depicted in each visualization are produced by repeated EA trials with the sampling being uniformly at random, without replacement, in the neighborhood. We fixed the sample size to $N = 1000$ output distance measurements (and so $1000r$ EA runs per visualization). To produce each diagram we fixed a base parameter vector $\underline{p}$ and a perturbation bound b and then sampled $N-1$ parameter vectors $\left\{\underline{p}_i\right\}_{i=1}^{N-1}$ in the neighborhood $N_{\bullet,b}\left(\underline{p}\right)$. As above, we may use either metric such that the metric is fixed throughout the runs that produce the visualization. The value of the Kolmogorov (output) metric $d_K\left(\underline{p}, \underline{p}_i\right)$ was then calculated via performing EA runs as described.

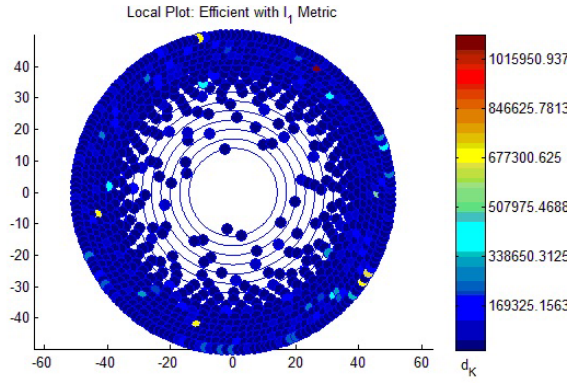
4.2 Results of Runs

The following two subsections give examples of visualizations under both the l_1 and l_∞ metrics, comparing the type 1 and type 2 outputs under an example of a, so-called, efficient parameter base vector and an inefficient parameter base vector. Both vectors were chosen by experimentation on account of encouraging distinct EA performance in terms of mean generation count and the standard deviation (see subsections 4.2.1 and 4.2.2). In all figures we chose the bound $b = 50$ and resolution $r = 3$, omitting other plots for smaller bounds and smaller resolutions. We shall observe that changing the base parameter vector affects the output produced, and by extension, the visualization.

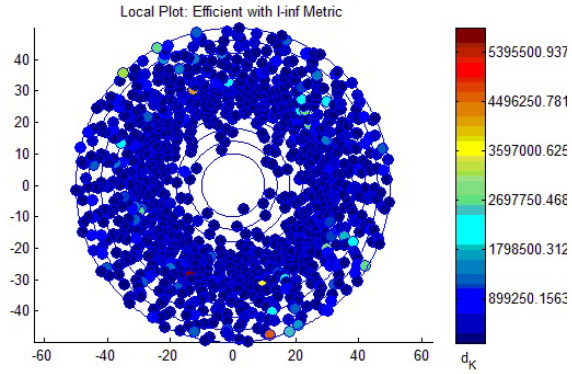
4.2.1 An Efficient Parameter Base Vector

The "efficient" parameter vector was chosen to be $\underline{p} = (5, 33, 4, 128, 30, 0)$, giving a mean generation count over ten independent EA runs of 179 for the instance, and for which

an EA termination constant of $s = 1500$ was used. This was chosen as the EA performance was relatively consistent; over ten runs the standard deviation was 51.8 generations [5].



(a) EA performance differences under the l_1 metric as the radius.

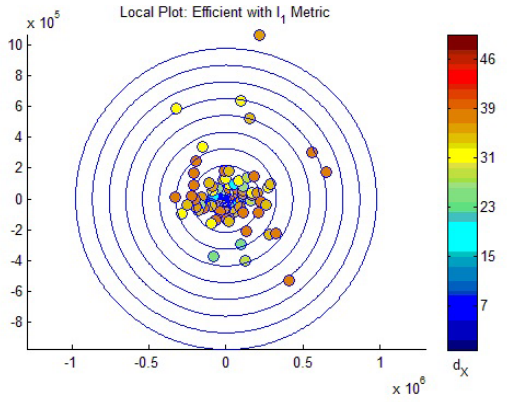


(b) EA performance differences under the l_∞ metric as the radius.

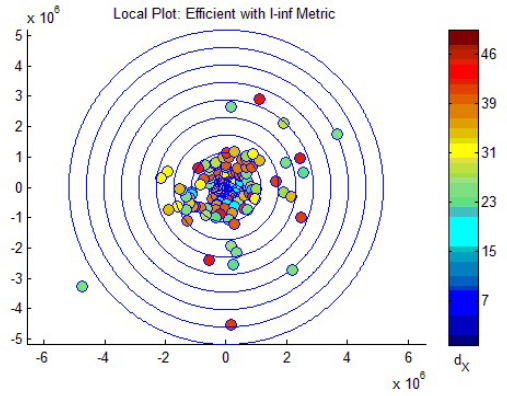
Figure 1: Typical type 1 plots for an efficient parameter vector, showing colored EA performance differences under the l_1 metric and l_∞ metric. Each plot uses a distinct set of EA runs.

Fig. 1 presents two type 1 plots. In this figure we observe that the smaller perturbations encourage only small performance differences (blue colors) in the EA when compared to the base performance using parameter vector $\underline{p}$. As the perturbation size (radius) increases, other colors begin to appear. This is more noticeable on the l_∞ metric plot (Fig. 1(b)). The different clustering behaviors between Figs 1(a) and 1(b) correspond to the distinct definitions of the l_1 and l_∞ metrics. For example, if we take the parameter vector $\underline{p}' = (5, 23, 7, 135, 25, 5)$ we see that $\underline{\varepsilon} = \underline{p}' - \underline{p} = (0, -10, 3, 7, -5, 5)$, and so $d_1(\underline{\varepsilon}) = 30$ while $d_\infty(\underline{\varepsilon}) = 10$. We may see this difference as greater information loss under the l_∞ than the l_1 metric. That is, under the l_∞ metric the dots have a closer clustering behavior towards the center of the plot, as a large value of the l_1 metric is likely to be much smaller under the l_∞ metric.

For the l_1 metric (Fig. 1(a)) we notice that non-blue dots begin to appear at around radius 30-40. Correspondingly on the l_∞ metric (Fig. 1(b)), non-blue dots appear to occur appreciably earlier, at around radius 20. This shows that outside the given bounds for each metric the EA begins to exhibit instability (with respect to the base parameter vector). Thus the visualization shows where the zone of EA stability ends, something difficult to predict from repeated EA runs alone. We include only the resolution $r = 3$ but note that, for $r = 1$, fewer dots are in the blue range (that is, there is more performance variation, which should be expected). This indicates that as r increases the EA performance is less variable under perturbation of the base parameter vector $\underline{p}$. Examples of type 2 plots are given in Fig. 2.



(a) EA performance differences under the l_1 metric as the color.



(b) EA performance differences under the l_∞ metric as the color.

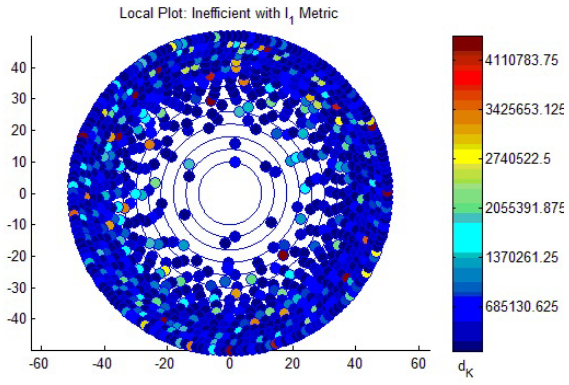
Figure 2: Typical type 2 plots for the same sets of runs as Fig. 1 under an efficient parameter vector. Now, the radius denotes a performance measure rather than perturbation size (denoted by color).

The type 2 visualization displays a much more intuitive picture of stability. The values given on the axes may be ignored (we clearly do not have negative output distance d_K) on the grounds of it being purely a diagrammatic representation. In Fig. 2 there is clear clustering behavior around

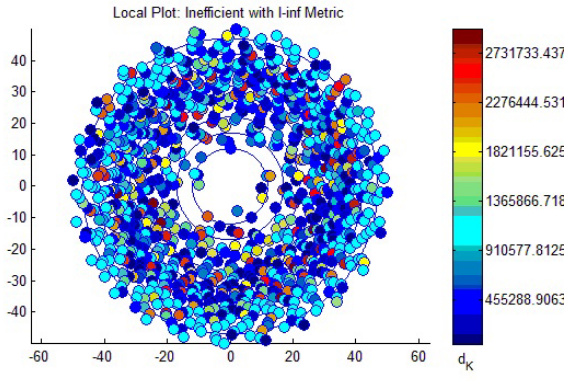
the center for each metric, which is expected and this result is in accordance with those of Fig. 1. The majority of the dots are within the circle denoting a Kolmogorov distance of approximately 5×10^5. However, for the l_∞ metric (Fig. 2(b)) there seems to be a greater variety of colors within that clustering, showing that there is a smaller correlation between performance and perturbation size with that metric. There is a wider spread of the dots, the majority of which are within a Kolmogorov distance of approximately 3.2×10^6, approximately one order of magnitude larger than the spread with the l_1 metric. Next we consider visualizations of the EA with an inefficient parameter vector.

4.2.2 An Inefficient Parameter Base Vector

Our second base vector was $\underline{p}' = (22, 40, 21, 54, 39, 24)$, giving a mean generation count of 1060 over ten independent EA runs. This was chosen because the EA performance was rather variable: the standard deviation over those runs was 480.9 generations. Because of the relatively high standard deviation we increased the termination constant to $s = 3000$.



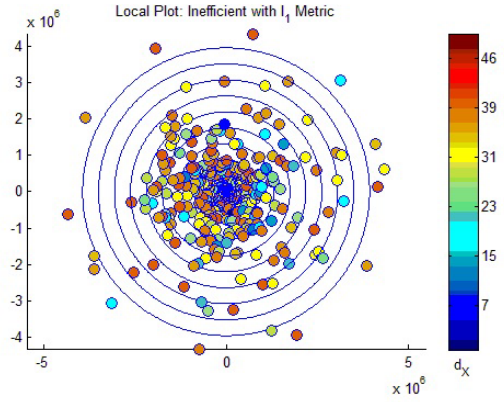
(a) EA performance differences under the l_1 metric as the radius.



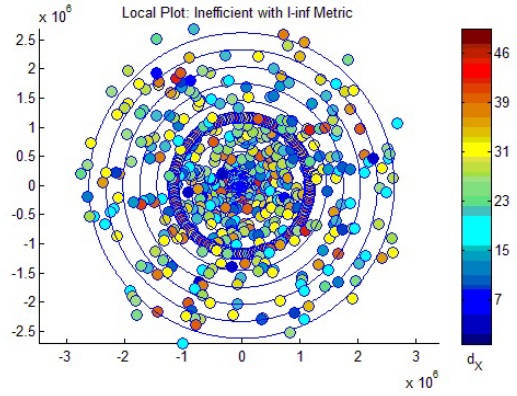
(b) EA performance differences under the l_∞ metric as the radius.

Figure 3: Type 1 plots for an inefficient parameter vector. The upper subfigure shows the distribution under the l_1 metric, and the lower the l_∞ metric.

In Fig. 3, it is clear there is far wider variation in EA performance for both parameter metrics than in Fig. 1. For the l_1 metric (Fig. 3(a)) there are already many more lighter colors compared to Fig. 1(a), which had an efficient base parameter vector. There is still a reasonable radius of stability, although this does not seem to be clear (due to the color variation even within a small radius). However, for the l_∞ metric (Fig. 3(b)) the effect is starkly apparent. There seems to be a small zone of stability (for example, to radius 10 at most) but this contains many perturbations that produce large changes in EA performance (light blue colors). There are also a preponderance of red colors and thus large EA performance differences. As the radius increases, large changes in EA performance become apparent. Hence the radius of stability is smaller than that of an efficient base parameter vector, if such a radius exists at all.



(a) EA performance differences under the l_1 metric as the color.



(b) EA performance differences under the l_∞ metric as the color.

Figure 4: Type 2 plots for an inefficient parameter vector, corresponding to the type 1 plots of Fig. 3.

Fig. 4 displays type 2 plots for an inefficient parameter vector. Compared to Fig. 2, on each subfigure there is a much wider spread of dots (as the EA performance differs more). An intuitive conclusion from this figure is that an inefficient parameter vector is likely to induce a type of less

controllable randomness in the EA. This is supported by the counterpart plots of Fig 4, where this increased randomness is illustrated by a large variety of colors.

For the l_1 metric plot of Fig. 4(a), even the outer circle denoting a Kolmogorov distance of approximately 4×10^6 has a large number of dots on or outside it. As we should expect in the l_∞ metric case (Fig. 4(b)), the visualization seemingly has no central clustering with respect to Kolmogorov distance, a conclusion also supported by its counterpart Fig. 3(b). In the next section we summarize our findings and conclude the work.

5. CONCLUSIONS

In this work we presented a novel experimental visualization of EAs under control parameter perturbation, using a case study of a published EA in cryptography. The visualization was a hybrid, representing a combination of projections of two metrics. The first metric was on the parameter space and the second was an EA performance metric based upon Kolmogorov distance. This enabled us to achieve a more substantial picture of EA stability than would be possible by basic statistical methods.

Our visualization captures the radius of stability under a typical efficient parameter vector and also the step change which occurs when the parameter vector is changed to an inefficient one. We observed when running the EA repeatedly with an efficient parameter vector that there was a clear radius of stability. This is given by type 1 plots where we have a preponderance of dark blue colors and by type 2 plots where there was an identifiable radius with the majority of dots inside it. This was certainly true for the l_1 metric, albeit the conclusion was weaker for the l_∞ metric. Contrary to the above findings are the conclusions for when an inefficient parameter vector is used. In this case there does not seem to exist such a clear radius of stability for type 1 visualizations under the l_1 metric. For the l_∞ metric there is no identifiable radius of stability. These conclusions are reinforced by the type 2 visualizations of Fig. 4. This provides a positive evaluation of the visualization; that is, we may distinguish between EA stability and instability depending upon perturbation and performance metrics.

We may thus answer the question posed in the introduction as follows. If the standard set of control parameters is efficient then there is a clear bound on EA performance change (in terms of the defined output metric) given by an identifiable radius of stability with little variation outside this radius. This is illustrated by both type 1 and type 2 visualizations. If the standard set of control parameters is inefficient then there often does not exist a clear bound on EA performance change. The key part of our visualization is that we may visualize why exactly this does not occur and also the performance distribution with respect to perturbation size. This performance distribution assists in the identification of parameter vectors which encourage atypical EA behavior and are worth investigating further. This further investigation may entail approximating the objective landscape and runtime analysis calculations. It may also assist as a method of nonlinear stochastic parameter optimization, with the identification of "sweet spots" (and the converse!) in the parameter distribution.

Turning to generalized EAs, our visualization may be easily generalized. If we have a general EA with parameter vector $\underline{p} = (p_1, p_2, \ldots, p_n)$ for $n > 1$ then the metrics we used for the case study are still suitable. Our treatment makes the assumption that the sum of a parameter vector is constant, using that assumption in the definition of the neighborhoods. As the visualization is created through projection of only part of the EA output, it may offer an insight into parameter spaces of arbitrary finite dimension.

Our visualization has the feature that there is no information loss. The visualization is created from a log file generated by the repeated EA runs, which contains the pertinent information. Thus the original information may be recovered. The advantage of our offline visualization is that it may be customized to any number of EA runs. However, the visualization also has the clear disadvantage that conclusions are difficult to reach until the end of a set of runs. This gives several sources of further work which we intend to pursue:

1. We wish to extend the visualization to an online visualization in order to enable further human intervention after a comparatively small number of runs. The update speed of visualization would depend upon the instance and resolution chosen.

2. This paper chronicles just two of the possible parameter metrics that may be used. It is also clear that the chosen parameter space metric dramatically affects the visualization. Hence we should like to attempt other parameter and performance metrics in an integrated approach.

3. Finally, we intend to pursue research in other types of visualization, including those giving some indication of the "direction" of the perturbation in the parameter space.

Due to time constraints other EA instances were not considered, but the visualizations (and the conclusions on EA stability therein) are not expected to radically change. From experience [4], the majority of runs under our setup are on the same performance spectrum of parameters (from which we have chosen two parameter vectors at opposite ends of the spectrum and so have presented two opposing visualizations). Overall, then, we have shown an interesting and novel visualization which proposes some interesting and potentially fruitful directions for EA performance analysis and research.

Acknowledgements

We thank the University of Plymouth and RISE, Waseda University for their generous research support during preparation of this paper. We also thank the anonymous referees for their helpful comments.

6. REFERENCES

[1] W. Cancino, N. Boukhelifa, A. Bezerianos, and E. Lutton. Evolutionary visual exploration: Experimental analysis of algorithm behaviour. In *VizGEC '13 (Proc. GECCO'13)*, pages 1373–1380. ACM, 2013.

[2] T. D. Collins. Applying software visualization technology to support the use of evolutionary algorithms. *J. Visual Lang. and Comp.*, 14:123–150, 2003.

[3] M. J. Craven. An evolutionary algorithm for the solution of two-variable word equations in partially commutative groups. *Studies in Comp. Intel.*, 153:3–19, 2008.

[4] M. J. Craven and H. C. Jimbo. A kolmogorov-type stability measure for evolutionary algorithms. In *"EvoCOP 2011" (P. Merz and J. -K. Hao, eds.), LNCS 6622*, pages 26–37. Springer, 2011.

[5] M. J. Craven and H. C. Jimbo. Stability of evolutionary optimization algorithms: Comparison and visualization of perturbation metrics. *Submitted to Proc. NACA2013*, 2014.

[6] E. Hart and P. Ross. GAVEL - a new tool for genetic algorithm visualization. *IEEE Trans. Evol. Comp*, 5(4):335–348, 2005.

[7] A. Kerren and T. Egger. EAVis: A visualization tool for evolutionary algorithms. In *"Proc. VL/HCC 2005"*, pages 299–301. IEEE, 2005.

[8] E. Lutton and J. D. Fekete. Visual analytics and experimental analysis of evolutionary algorithms. *INRIA Rapport de Recherche*, 7605, 2011.

[9] J. Ombach. Stability of evolutionary algorithms. *J. Math. Analysis and Appl.*, 342:326–333, 2008.

[10] B. Østman and C. Adami. Predicting evolution and visualizing high-dimensional fitness landscapes. *arXiv:1302.2906v2*, 2013.

[11] I. Packham and I. Parmee. Data analysis and visualisation of cluster-orientted genetic algorithm output. In *Proc. IEEE Intl. Conf. on Info. Visualization*, pages 173–178. IEEE, 2000.

[12] H. Pohlheim. Visualization of evolutionary algorithms - set of standard techniques and multidimensional visualization. In *Proc. GECCO'99*, pages 533–540. ACM, 1999.

[13] G. Romero, J. J. Merelo, P. A. Castillo, J. G. Castellano, and M. G. Arenas. Genetic algorithm visualization using self-organizing maps [sect. 2 and 3]. In *"PPSN VII" (J. J. Merelo Guervós et al., eds.), LNCS 2439*, pages 442–451. Springer, 2002.

[14] E. D. Sontag and Y. Wang. Notions of input to output stability. *Systems & Control Letters*, 38:236–248, 1999.

[15] L. J. van der Maaten, E. O. Postma, and H. J. van den Herik. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, 10(1-41):66–71, 2009.

[16] D. J. Walker, R. M. Everson, and J. E. Fieldsend. Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Trans. Evol. Comp.*, 17(2):165–184, 2013.