WCCI 2010 IEEE World Congress on Computational Intelligence
July, 18-23, 2010 - CCIB, Barcelona, Spain CEC IEEE

Fitness Landscape Analysis for Evolutionary Non-Photorealistic Rendering

Jeff Riley and Vic Ciesielski

Abstract—**The best evolutionary approach can be a difficult problem. In this work we have investigated two evolutionary representations to evolve non-photorealistic renderings: a variable-length classic genetic algorithm representation, and a tree-based genetic algorithm representation. These representations exhibit very different convergence behaviour, and despite considerable exploration of parameters the classic genetic algorithm was not competitive with the tree-based approach for the problem studied in this work. The aim of the work presented in this paper was to investigate whether analysis of the fitness landscapes described by the different representations can explain the difference in performance. We used several current fitness landscape measures to analyse the fitness landscapes, and found that one of the measures suggests there is a correlation between search performance and the fitness landscape.**

I. INTRODUCTION

DETERMINING the best evolutionary approach to a new problem is difficult and is often more art than science. There is usually experience-based or anecdotal guidance, but basically it is necessary to try different alternatives and see which one works best. In previous work [2], we built an evolutionary Non-Photorealistic Rendering (NPR) system with what we thought was a sub-optimal representation: we used an existing genetic programming package for rapid prototyping. We subsequently recoded the system using what we believed to be a more appropriate representation, but to our great surprise the second system was unable to achieve the convergence behaviour of the first. Our experiments also indicate that the best results are obtained with the smallest populations. Usually in evolutionary systems larger populations give better convergence, but in this case the best results were obtained with a population of size two.

In our previous work our goal was to evolve the required straight-line “pencil” strokes to get an interesting rendering of a target image similar to that shown in Figure 4. Since the solution is a set of strokes, one would normally consider the “natural” representation to be a variable length genetic algorithm (GA) [9] representation, where each gene on the chromosome represents the parameters of a line (e.g. origin, angle, length, shade). However, in order to quickly determine feasibility, we decided to use an existing tree-based genetic programming system [14]. This system provides the necessary mechanism to implement a tree-based GP solution to a problem - the programmer needs to provide the terminals, functions and the fitness function; everything else is provided by the system. Our NPR application was implemented using a simple function to join two sub-trees, a function to place the line on the canvas, and floating point numbers for the parameters of the line-drawing function. The fitness function was a pixel-by-pixel summation of the differences between the evolved and the target images.

This initial system was very successful and generated still images and animations that excited our artist collaborators. The implementation was very slow however, with each evolutionary run taking many hours to complete, so we decided to rework it to execute faster while maintaining functionality. We changed the representation to a variable length GA but found the convergence behaviour of the GA was much worse than the original GP implementation. In fact, while the GA implementation completed in minutes rather than hours for the same number of evaluations, despite much experimentation with parameter values it proved impossible to find a GA configuration that was competitive with the GP implementation, with respect to likeness of the resultant image to the target image.

Many optimizations of our original GP implementation are possible, but the major limiting factor is the convergence behaviour of the evolutionary search, and this is the focus of the work in this paper. In particular, the primary goal of this work is to determine if the difference in performance of the tree-based method versus the variable-length GA can be explained by analysing the fitness landscapes described by the different representations.

The concept of fitness landscapes, and the idea that the process of evolution could be studied by visualising the distribution of fitness values across the population as a landscape, has been long-established in the field of evolutionary biology, having been first proposed by Wright in 1932 [31] and revived later in [8]. In previous work, fitness landscape analysis using various landscape measures has been used with mixed results to predict the performance of evolutionary algorithms [17 - 20, 25]. As noted by Merz, “*performance prediction is a complicated task and is only reasonable if all or most problem characteristics having influence on heuristic search are taken into account*”, and “*the local structure of the landscape - its ruggedness - and its global structure … appear to be key characteristics*” [18, pp 181-182]. It is our goal here to determine if some combination of fitness landscape characteristics can be used

Manuscript received February 7, 2010.

Jeff Riley is with Hewlett-Packard Australia, Melbourne, VIC 3131 Australia (phone: 61-3-8833-4779; e-mail: jeff.riley@hp.com).

Vic Ciesielski is with the School of Computer Science and Information Technology, RMIT University, Melbourne, VIC 3000 Australia (e-mail: vic.ciesielski@rmit.edu.au).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

to explain the difference in performance of the representations under investigation.

II. Representations

A. Variable-Length Classic GA

In this representation each member of the population is a bitmap image represented by a chromosome containing a variable number of genes. The genes on the chromosome are arranged in groups of four, with each group of four genes defining a line to be drawn on the image represented by the chromosome. The variable length chromosome is constructed by concatenating a number of line gene groupings (Figs. 1 and 2).

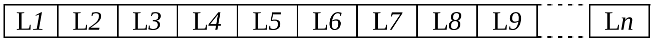


Fig. 1. Representation of the chromosome.

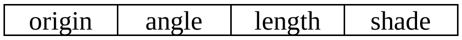


Fig. 2. Gene grouping representing a single line.

Each grouping of genes, defining a line to be drawn, is represented on the chromosome by four scaled integer numbers representing the line start point (origin), the angle at which the line is to be drawn, the length of the line (in pixels) and the grayscale shade of the line. Scaled integers are used for performance reasons.

Our tests showed that single-point crossover out-performed two-point crossover, so only single-point crossover is implemented for the variable-length classic GA. For this implementation the crossover point is at any gene boundary, not necessarily a gene grouping (or line) boundary.

The mutation scheme implemented for the GA representation changes each allele on a chromosome, with some given probability from *0.0* to *1.0*, to a randomly chosen value.

The crossover and mutation operators implemented guarantee there will no oversize chromosomes or out-of-bounds data in the resultant chromosomes.

B. Tree-based GA

Our original GP implementation is as described in [2]. In this representation each member of the population is a greyscale bitmap image represented by a GP program tree. There are two kinds of functions: *Draw* functions and *ProgN* functions. The *Draw* functions have as input four integer numbers representing the parameters of a line to be drawn. Execution of a *Draw* function causes pixel values on the canvas to be changed, resulting in the line being drawn. The *ProgN* functions have *N* input arguments and are used for sequential execution of their arguments as described in [14]. *ProgN* functions accept both *ProgN* and *Draw* functions as input arguments, while *Draw* function accepts only terminals.

As an example, programs which contain only *ProgN* nodes of arity *4* would be generated according to the following grammar:

S -> DRAW | PROG4
PROG4 -> Prog4(S, S, S, S)
DRAW -> Draw(T, T, T, T)
T -> float

Following from the understanding that a GP tree generated from the grammar described above and evaluated using pre-order traversal can be viewed as a linear sequence of pencil strokes, we developed a tree-based GA representation in which a chromosome is constructed by performing a left to right, depth-first search of the tree to discover the *Draw* nodes (lines to be drawn), and executing the *Draw* nodes as they are found (Fig. 3). It can be seen that this chromosome (and so line sequence) is the same as that described in Section IIA.

Using the chromosome shown in Fig. 3, crossover in the tree-based GA interpretation can be achieved by implementing GA two-point crossover, with the selection of the crossover points determining whether crossover takes place at a *ProgN* node, *Draw* node or terminal.

Similarly, mutation can occur at *ProgN* nodes, *Draw* nodes, or at individual terminals. Mutation at a *ProgN* or *Draw* node is the complete replacement of the genetic material between two randomly chosen genes on the chromosome. Mutation at a terminal is just the mutation of a single, randomly chosen gene. As with crossover, the selection of the mutation points determines whether mutation takes place at a *ProgN* node, *Draw* node or terminal. The crossover and mutation operators implemented guarantee there will no oversize chromosomes or out-of-bounds data in the resultant chromosomes.

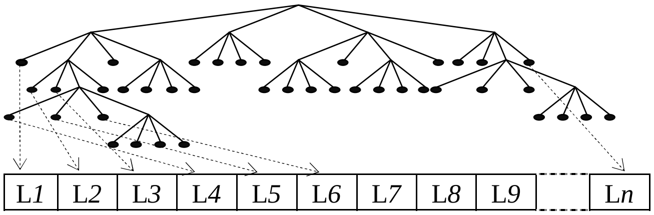


Fig.3. Chromosome construction for tree-based GA.

Analysis of a number of trees generated by the GP implementation from [2] indicates that the average probability of randomly choosing a *ProgN* node for crossover or mutation is *0.05*, the probability of choosing a *Draw* node is *0.20*, and the probability is *0.75* for a terminal. The tree-based GA is implemented using the crossover and mutation operators described above, with the crossover and mutation points chosen randomly according to the probabilities discussed. That is, single terminals (parameters of a line) will be chosen for (for example) crossover *75%* of the time, resulting in the crossover of a single gene; *20%* of the time a single, but complete, line will chosen for crossover, resulting in the crossover of a specific group of genes contiguous on the chromosome; and *5%* of the time a group of lines will be chosen, resulting in the crossover of a larger number of contiguous genes on the chromosome. Using this method, rather than a GP tree defining the chromosome and crossover points, the chromosome defines the *Draw* nodes of the tree, and the probability of crossover at a particular node or terminal and the crossover points chosen define the topology of the tree. Mutation is implemented similarly using the probabilities discussed.

Our tests show that the tree-based GA is a very good approximation of the tree-based GP implementation, with little or no difference in performance and operation characteristics between the two methods. In this work we study the difference in performance of the tree-based GA versus the variable-length classic GA.

III. THE FITNESS LANDSCAPE

A. *Fitness Landscape Definition*

Previous work involving fitness landscapes often avoids a rigorous definition of the landscape under analysis [11], and where it is mentioned or implied at all the landscape is usually assumed to be the *single-bit mutation* landscape: the landscape generated by arranging all single-bit mutations of a chromosome represented as a string of binary digits such that chromosomes that differ by only a single bit are neighbours. On such landscapes, genetic operators such as crossover are assumed to take *hypersteps* over the fitness landscape described by mutation.

There is no generally accepted definition of what constitutes a fitness landscape. Although there has been work to try to present a coherent, consistent view of fitness landscapes and the neighbourhood relations that define them, as well as the methods that are used to measure them, to date there is no complete agreement. A review of relevant literature [e.g. 3, 7, 10, 11, 13, 26 - 30] indicates there are several possible definitions of, and representations for, fitness landscapes, and choosing the definition and representation which best describes the combination of the problem being studied and the algorithm being used to study it is extremely important. A fitness landscape is most often defined by three basic attributes:

- a search space
- a relation that defines which points are neighbours in the search space
- a fitness function that assigns a fitness value to each point in the search space

The neighbourhood relation and its specification is extremely important because any discussion of landscapes invariably involves the terms “peaks” and “valleys”, and no peak or valley can exist without the notion of neighbourhood – a peak is only a peak because it is higher than its neighbours.

For this work the fitness landscape is considered to be defined by the overall operation of the genetic algorithm [21, 22]. Consider an observer watching a genetic algorithm searcher perform a random walk on a fitness landscape and assume that although the observer is able to discern the granularity of the search (the genetic algorithm’s single steps), the means by which the GA determines where each step takes it is hidden from the observer. The random walk is conducted as follows:

- An individual i_0 is randomly selected from the search space
- For each step *s, s = 1 .. maxsteps*
    - o Another individual i_s, $i_s \neq i_{s-1}$, is randomly selected from the search space.
    - o Crossover is performed between i_s and i_{s-1} with probability $P_{crossover}$, resulting in two new individuals i'_1 and i'_2, both of which are neighbours of (a single step from) $i_{s-1.}$ Set $i_s = i'_1$ and discard i'_2.
    - o i_s undergoes mutation with probability $P_{mutation}$
    - o Step to i_s

The observer sees the searcher walking randomly over the landscape and considers points on the landscape one step apart to be neighbours. The definition of the neighbourhood relation is of no consequence to, and is not required by, the observer since the searcher is defining neighbouring points by performing the walk. If the random walk performed by the genetic algorithm searcher was sufficiently long, and the “altitude” (fitness) at each step recorded for the observer, the entire fitness landscape would be determined by observation. The landscape so determined would be the precise fitness landscape defined by the search algorithm.

This “black box” view of the genetic algorithm operation and consequential determination of the neighbourhood relation and fitness landscape satisfies the requirement that the landscape neighbourhood relation be defined by the search algorithm, and is the definition used for this work.

B. *Fitness Landscape Measures*

Several methods for measuring and analysing landscapes for search algorithms have been proposed in previous work. The methods proposed can be categorised into two broad streams: statistical measures [1, 10, 12, 15, 16, 28, 29] and information measures [4 – 6, 26, 27]. Borenstein and Poli have extended the information measure to include a measure of the performance of the algorithm and so define a “performance landscape”. All the methods proposed have in common the notion that the points in the search space are arranged according to some neighbourhood relationship, and a measure of fitness, performance or information content associated with the points defines the ruggedness of the landscape.

The methods used to measure and analyse the structure of fitness landscapes in this work are the *autocorrelation* method suggested by Weingberger [28, 29], and the *information content* approach suggested by Vassilev [26, 27]. These methods were chosen because they are different methods of measuring the structure of landscapes and, while there seems to be no generally accepted standard approach, both methods have gained some favour and are commonly cited as reasonable landscape characterisation methods [10, 17, 23, 24]. Jones' *Fitness Distance Correlation* [11] is an interesting landscape measure but requires that the solution be known in order to calculate the metric, so is not applicable to the work presented here.

1) Autocorrelation and Correlation Length

Weinberger's autocorrelation definition [28, 29]:

Given measurements, $Y_1, Y_2, \ldots, Y_N$ at time $X_1, X_2, \ldots, X_N$, where N is the number of measurements, and

$$\overline{Y} = \frac{1}{N}\sum_{i=1}^{N} Y_i \quad , N > 0$$

the time lag k autocorrelation function r_k is defined as

$$r_k = \frac{\sum_{i=1}^{N-k}\left(Y_i - \overline{Y}\right)\left(Y_{i+k} - \overline{Y}\right)}{\sum_{i=1}^{N}\left(Y_i - \overline{Y}\right)^2} \quad , N > k$$

If $|r_k| \approx 1.0$ there is much correlation between the points k steps apart in the series, whereas if $|r_k| \approx 0.0$ there is little correlation.

Weinberger proposed that a random walk be generated on the fitness landscape, where each step on the walk is taken between neighbouring points, with the neighbour to which the step is taken selected randomly. The fitness values for the points visited during the random walk form a time series of numbers. The autocorrelation function can then be used as a measure of the ruggedness of the landscape described by the random walk.

The correlation length of a series of numbers is the largest distance, or time lag, between points for which some correlation exists. Hordijk [10] defines the correlation length of a time series as one less than the first time lag for which the autocorrelation falls inside the region bounded by the two-standard-error bound (i.e. one less than the first time lag at which the autocorrelation becomes statistically equal to zero, making the correlation length the largest time lag for which the correlation between two points is still statistically significant). This is the method used for calculating the correlation length in this work. The two-standard-error bound e is defined as

$$e = \pm\frac{2}{\sqrt{N}}$$

so the correlation length ℓ is defined in this work as the first lag k for which

$$|r_k| < \left|\frac{2}{\sqrt{N}}\right|$$

2) Information Content

Vassilev et al. propose three information measures that characterise the structure of a fitness landscape from a series of points generated by a random walk over the landscape [26, 27]:

- Information Content – characterises the ruggedness of the landscape.
- Partial Information Content – measures the modality of the landscape.
- Information Stability – the sensitivity of the information content measures.

These measures are calculated by generating a random walk of length ***n*** on the fitness landscape, with the aim being to extract information by characterising the series of points as an ensemble of objects. To calculate the information content, a string

$$S(\varepsilon) = s_1 s_2 \ldots s_n \quad , s_i \in \{\overline{1}, 0, 1\}$$

representing a group of objects generated from a random walk over the fitness landscape. $S(\varepsilon)$ is enumerated according to the function:

$$s_i = \psi_{f_t}(i, \varepsilon) \quad \text{,for } t = 1..n$$

and

$$\psi_{f_t}(i,\varepsilon) = \begin{cases} \overline{1} & if \quad f_i - f_{i-1} < -\varepsilon \\ 0 & if \quad |f_i - f_{i-1}| \le \varepsilon \\ 1 & if \quad f_i - f_{i-1} > \varepsilon \end{cases}$$

Thus the string $S(\varepsilon)$ defines a sequence of objects where each object is represented by a substring $S_i S_{i+1}$ being a sub-block of length two of the string $S(\varepsilon)$.

The parameter ε is a real number taken from the interval [*0.0, 1.0*] which defines neutral fitness and determines the accuracy with which the string $S(\varepsilon)$ is defined. If the absolute fitness difference between neighbouring points is less than ε the points are considered to be of equal fitness. This means that as ε increases from *0.0* to the maximum possible fitness difference between points along the walk (*1.0*), the amount of fitness change (entropy) and the sensitivity of ψ_{f_t} decrease to zero.

The *information content* is defined as the entropic measure of the group of sub-blocks of length two of string $S(\varepsilon)$, and is given by:

$$H(\varepsilon) = -\sum_{p \ne q} P_{[pq]} \log_6 P_{[pq]}$$

$P_{[pq]}$ are frequencies of the possible blocks *pq* of elements from the set $\{\bar{1},0,1\}$ given by

$$P_{[pq]} = \frac{n_{[pq]}}{n}$$

where $n_{[pq]}$ is the number of occurrences of *pq* in $S(\varepsilon)$ and *n* the length of $S(\varepsilon)$.

It can be seen from this that *Information Content* measures the frequency of *changes in direction* of the gradient between neighbouring points on the fitness landscape, thus giving an indication of the "choppiness" of the surface of the fitness landscape. There are two parts to the information content measure: its value at $\varepsilon = 0$ which indicates the frequency of change between neighbouring points on the fitness landscape ("choppiness"), and how quickly it drops to zero or near zero for $\varepsilon > 0$, providing an indication of the local "depth" of the "chop".

The *partial information content* is a measure of the modality of the landscape, and is calculated by filtering out elements of the string $S(\varepsilon)$ which are not essential for measuring modality to create a new string $S'(\varepsilon)$, then measuring the length μ of the new string $S'(\varepsilon)$. The string is defined as

$$S'(\varepsilon) = s_{i_1}s_{i_2}...s_{i_j} \ , s_{i_j} \neq 0 \ \ , s_{i_j} \neq s_{i_{j-1}} \ , j > 1$$

Thus the string $S'(\varepsilon)$ has the form $"\bar{1}1\bar{1}..."$, representing the non-zero slopes of the path taken by the random walk over the landscape, and so is empty if all the elements of $S(\varepsilon)$ are the same (i.e. all *0s*, all *1s* etc.). The *partial information content* is given by

$$M(\varepsilon) = \frac{\mu}{n}$$

Note that when $M(\varepsilon)$ is *1*, the path taken by the random walk over the landscape is considered to be maximally multimodal, and when $M(\varepsilon)$ is *0*, the path is a constant gradient (flat, increasing or decreasing at a constant rate).

Partial Information Content is an indication of the frequency of local maxima and minima on the landscape – whereas the information content measure counts changes of gradient direction and includes flat spots, the partial information content measure ignores zero gradients, so measures the *modality* of the landscape.

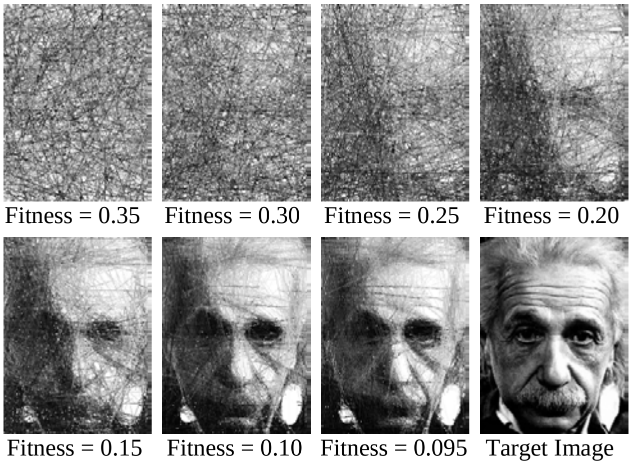
The *information stability* ε^* is defined as the smallest value of ε for which the landscape becomes flat (i.e. for which $S'(\varepsilon)$ is empty). Since ε governs the sensitivity of the information content and partial information content measures, ε^* is a measure of the largest difference in fitness between neighbouring points on the random walk.

IV. EXPERIMENTS

A number of experiments were conducted for each representation discussed above. Parameters were varied to get a good understanding of the performance of the different representations - a smaller population size was found to produce the best results: for variable-length classic GA a population size of *10* produced superior results, and for tree-based GA population sizes from *2* to *4* produced the best results. Population sizes of *10* for the classic GA and *4* for the tree-based GA were used for these experiments so that a combination of elitism, crossover and mutation could be tested. Experiments were conducted using different combinations of mutation and crossover rates for each of the representations.

In addition to ten evolutionary runs for each experiment, ten random walks were conducted for each experiment, each walk starting at a randomly selected point on the fitness landscape and continuing for a duration of *100,000* steps. The landscape statistics gathered during the walks are also analysed.

The task set for the representations being tested was to evolve a non-photorealistic representation of the target image shown in Fig. 4, using a stroke-based rendering technique. The parameters used for the experiments are shown in Table 1. Crossover and mutation rates varied (see results). The fitness function was a pixel-by-pixel summation of the differences between the evolved and the target images – a lower fitness indicates a smaller difference and hence a better match ("better" is subjective, but it meets our needs here).


Fitness = 0.35 Fitness = 0.30 Fitness = 0.25 Fitness = 0.20

Fitness = 0.15 Fitness = 0.10 Fitness = 0.095 Target Image

Fig. 4. Target image and evolved images of varying fitness.

The following information for each experiment is presented for analysis and comparison:

- Data from the evolutionary search:
    - Line graphs showing the best fitness averaged over the ten evolutionary runs.
- Data from the random walks over the fitness landscape:
    - A correlogram showing the autocorrelation data for steps *1* to *300* averaged over the ten random walks, and the two-standard error bounds.
    - Graphs showing the information content data and partial information content data for $0 <= \varepsilon <= 0.15$ averaged over the ten random walks.

Table1. Parameters used in experiments.

Parameter	Value
Population Size	10 for classic GA; 4 for tree-based GA
Selection Strategy	Roulette Wheel
Elitism	Enabled, count = 1
Replacement Strategy	Generational Replacement
Termination Policy	Fitness target or maximum generations
Minimum Line Length	2 pixels
Maximum Line Length	0.65 *x* image diagonal
Image Size	180 *x* 240 pixels
Pixel Update Method	Average grey level of existing pixel shade and shade of pixel to be painted
Chromosome Generation	Random Initialisation
Crossover Probability	Varies
Mutation Probability	Varies
Min Chromosome Size	2000 genes (500 lines)
Max Chromosome Size	6000 genes (1500 lines)
Runs per Experiment	10
Crossover Method	Single-Point for classic GA; GA Two-Point for tree-based GA

V. RESULTS

The results of the evolutionary runs for each representation are shown in Fig. 5. The legend (shown in Fig. 9) for Figures 5, 6 & 7 and Table 2 conveys the following for each line:

- representation (“GA” or “treeGA”)
- population size (“p*n*”)
- elitist count (“e*n*”)
- crossover probability for classic GA (“c*x.xx*”), or
- crossover count for tree-based GA (“c*n*”)
- mutation probability for classic GA (“m*x.xx*”), or
- mutation count for tree-based GA (“m*n*”)

The coloured shading on Fig. 5 highlights four distinct performance bands exhibited by the different representations tested. The same highlight colours are used to delineate the performance bands in Table 2.

Fig. 4 shows that a fitness of above *0.15* is not an acceptable result for our purposes here, between *0.10* and *0.15* is marginal, and below *0.10* is an acceptable result (again, these are subjective terms but they are sufficient for our needs here)..

Fig. 5 shows that all variations of tree-based GA outperformed all variations of variable-length classic GA, and in fact only the tree-based GA variations produced results in the acceptable range. For the tree-based GA trials, the mutation-only and mutation-plus-crossover implementations were virtually identical, and outperformed the crossover-only implementation. Given the small population size, with elitism = *1*, these results would tend to suggest that the problem is one most easily solved by a hill-climbing algorithm of some kind.

For the variable-length classic GA trials, in all cases for which crossover was implemented except the crossover-only implementation, a higher probability of crossover produces a worse result. This is consistent with the tree-based GA results.

The crossover-only results for the variable-length classic GA are interesting in that they suggest that with no mutation, a small crossover rate performs not much better than one would expect a random search to perform – which is somewhat intuitive since with no mutation a small crossover rate in a small population produces no change to the population most of the time. Higher rates of crossover for crossover-only produce better results – again, for the same reason, somewhat intuitive.

The results shown in Fig. 5 indicate that for the variable-length classic GA a smaller rate of mutation produces better results, suggesting that a constrained local search around locations visited by crossover produces better results than a larger local search.

It is interesting to note that the best results for the variable-length classic GA are produced by the mutation-only (rate *0.05*) and crossover-only (rate *0.95*) implementations. This is not the case for the tree-based GA trials – the crossover-only implementation was outperformed by the other implementations.

Weinberger’s *Autocorrelation* measures the correlation between neighbouring points on the fitness landscape. *Correlation Length* is an indication of the extent to which some correlation exists between more distant points on the landscape. An interesting result is that autocorrelation and correlation length seem not to be significant success factors for these experiments (see Fig. 6). Certainly the mostly highly correlated landscape - classic GA with no crossover and a mutation rate of just *0.05* - is not the one that produced the best search results. A reasonable assumption might have been that information about neighbouring points provided by such a highly correlated landscape would have helped the search algorithm, but the data presented indicates otherwise. This result could be explained by the information content and partial information content figures for this experiment – both are low, indicating that while the landscape may be well correlated, it is very smooth and so doesn’t convey much usable information to the search algorithm.

For the purpose of this analysis, information content (x_i) and partial information content (x_p) values are arbitrarily categorised in Table 2 as:

$$category = \begin{cases} very\ low, & 0.0 \le x_i, x_p < 0.2 \\ low, & 0.2 \le x_i, x_p < 0.4 \\ medium, & 0.4 \le x_i, x_p < 0.6 \\ high, & 0.6 \le x_i, x_p < 0.8 \\ very\ high, & 0.8 \le x_i, x_p \le 1.0 \end{cases}$$

From the data presented in Figures 5, 7 and 8 and Table 2 it can immediately be seen that a *low* value for information content at $\varepsilon = 0$ is suggestive of poor performance. It can also be seen that the two best results, and the only results which are considered acceptable (*fitness* $\leq$ *0.1*), were attained by representations which exhibit a *medium* value for information content at $\varepsilon = 0$, and for which information content drops off quickly to a *very low* value for $\varepsilon > 0$. The third best result - just a little outside acceptable with a best fitness of *0.105* - exhibits a *high* value for information content at $\varepsilon = 0$, and drops off quickly to *very low* for $\varepsilon > 0$.

Most experiments (*15* of *18*) exhibited a *medium* value for partial information content, with the remaining being *low* or *very low*. Similarly, most (*14* of *18*) exhibited a quick drop off to *very low*. This would tend to indicate that partial information content, at least when taken in isolation, is not a significant success factor.

As for autocorrelation and partial information content, *Information Stability* (ε^*), on its own, would seem not to be a significant indicator of performance.

These results tend to suggest that, for this problem, the most significant fitness landscape measure for predicting performance is the *Information Content* measure, both at $\varepsilon = 0$ (the choppiness of the surface of the landscape) and how quickly it drops off to a *very low* value for $\varepsilon > 0$ (the depth – or shallowness – of the surface chop). A *medium* value at $\varepsilon = 0$ and *very low* values for $\varepsilon > 0$ indicate good search performance; a low value at $\varepsilon = 0$ indicates poor search performance; values in-between produce mixed search performance. This indicates that a fitness landscape that provides some information in the way of a "choppy" surface is an aid to a search algorithm, but that the level of "choppiness" or "ruggedness" is important and perhaps a little sensitive. Certainly it is evident from the results shown in Table 2 that insufficient choppiness inhibits search performance, and there is some indication that a landscape which is too rugged also affects search performance negatively.

VI. CONCLUSIONS

The aim of the work presented in this paper was to try to explain an unexpected result in the use of two different evolutionary representations for a non-photorealistic rendering problem. To do so we chose to use a number of well known fitness landscape measures to analyse the fitness landscapes underlying the evolutionary search for each of the representations. The measures did not give any clear understanding of the reasons for the difference in performance, but while we found no obvious, concrete correlation between performance and any of the fitness landscape measures, there is some suggestion of a correlation between performance and Vassilev's *Information Content* measure.

We utilised a definition of the fitness landscape as proposed in [21, 22], and which is described by the combination of the algorithm and the problem being investigated (the "black box" view).

Our results show that while we found no obvious, concrete correlation between performance and any of the fitness landscape measures, there is some suggestion of a correlation between performance and Vassilev's *Information Content* measure.

The optimal fitness landscape indicated by the results of the investigation – particularly the *information content* measure - is one which exhibits a "medium" amount of reasonably "shallow" choppiness or ruggedness. The results suggest that too much ruggedness on the fitness landscape results in the search algorithm spending too much time visiting local optima, and conversely a sparse fitness landscape results in the search algorithm spending too much time taking "hypersteps" over the landscape looking for optima of any kind.

More investigation of fitness landscape metrics and characteristics needs to be undertaken in order to gain more insight into the correlation between performance and the fitness landscape. Furthermore, while analysis of the fitness landscape described by a search algorithm can be useful in predicting and explaining the performance of the algorithm, some work to determine why the various evolutionary operators, or combinations of operators, produce the fitness landscapes they do would help design search algorithms that define fitness landscapes that are conducive to search.

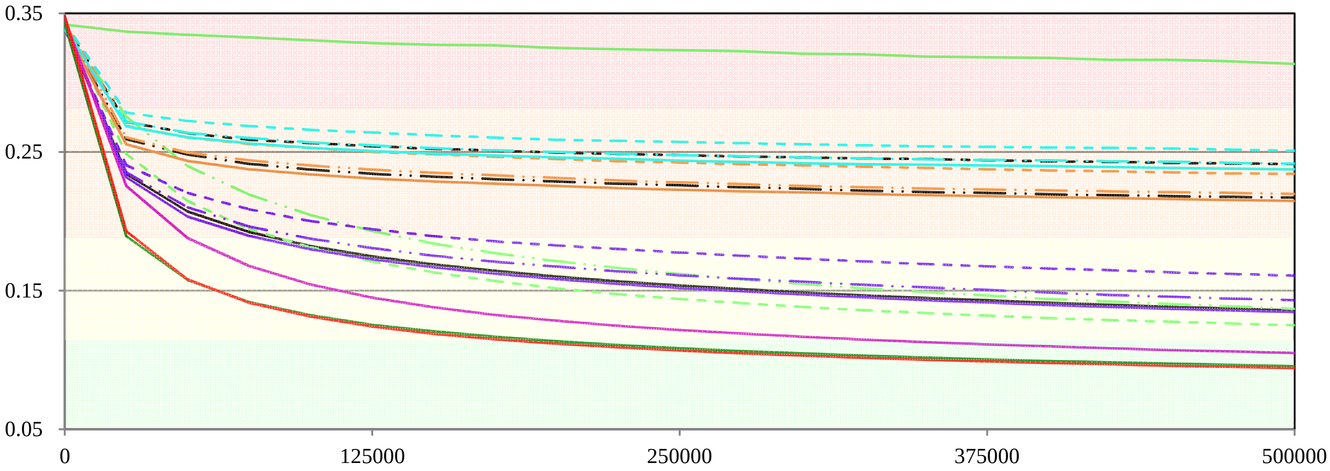


Fig. 5. Evolutionary runs (Fitness vs Evaluations ('000s)).

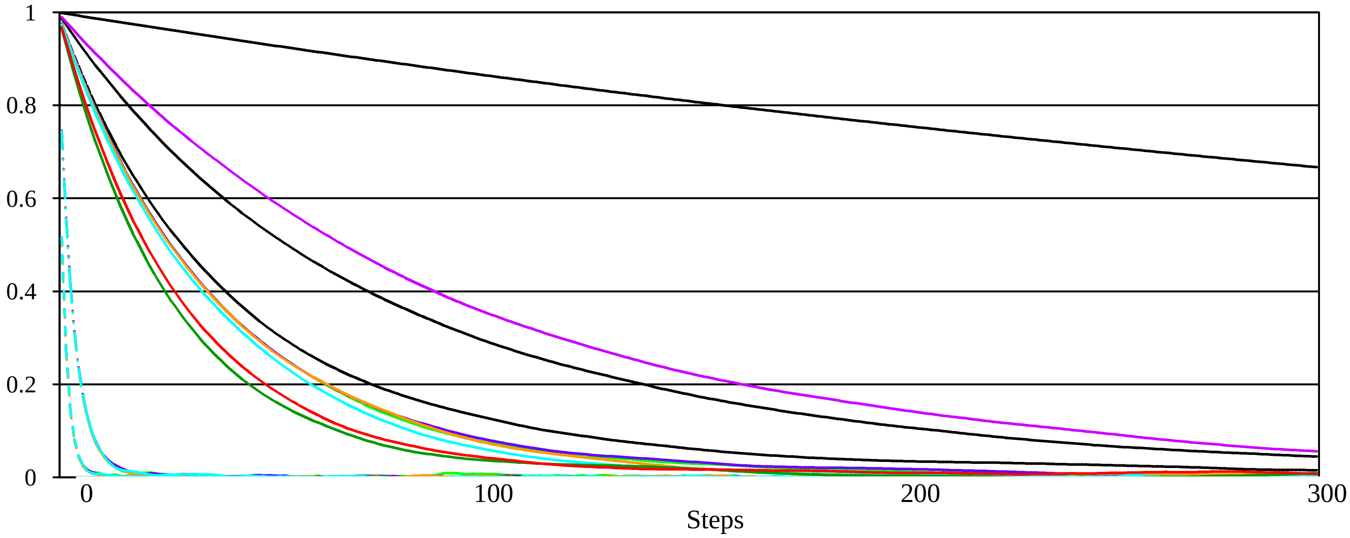


Fig. 6. Autocorrelation.

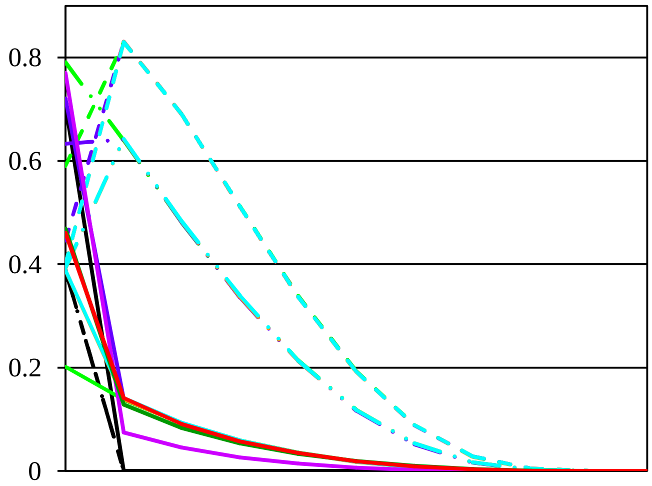


Fig.7. Information Content.

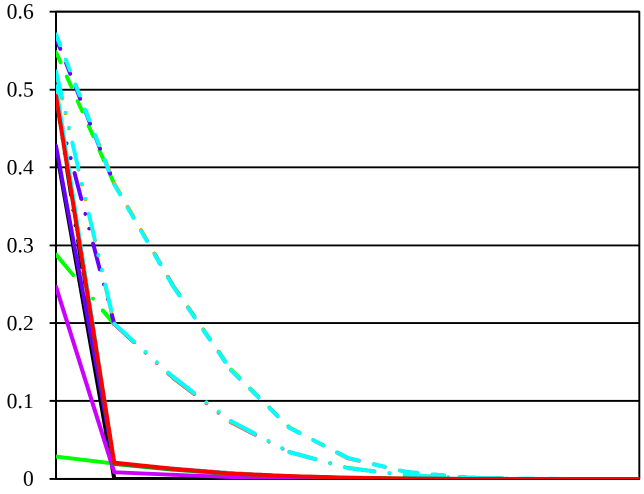


Fig. 8. Partial Information Content.

Table 2. Results.

Experiment	**Best Fitness**	**Autocorrelation at step 0**	**Correlation Length**	**Information Content**			**Partial Information Content**			ε^*
				$\varepsilon = 0.00$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	$\varepsilon = 0.00$	$\varepsilon = 0.01$	$\varepsilon = 0.02$	
treeGA p4e1c2m1	0.094154	0.967595	499	medium	very low	very low	medium	very low	very low	0.10
treeGA p4e1c0m3	0.095597	0.964622	180	medium	very low	very low	medium	very low	very low	0.10
treeGA p4e1c2m0	0.104986	0.989788	689	high	very low	very low	low	very low	very low	0.09
GA p10e1c0.95m0.00	0.125168	0.515882	10	medium	very high	high	medium	low	low	0.11
GA p10e1c0.05m0.05	0.134519	0.974415	252	high	very low	very low	medium	very low	very low	0.10
GA p10e1c0.00m0.05	0.135733	0.998568	1385	high	very low	very low	medium	very low	very low	0.01
GA p10e1c0.50m0.00	0.136945	0.745370	25	high	high	medium	low	very low	very low	0.11
GA p10e1c0.50m0.05	0.143093	0.746116	25	high	high	medium	medium	very low	very low	0.10
GA p10e1c0.95m0.05	0.160916	0.516395	10	medium	very high	high	medium	low	low	0.10
GA p10e1c0.05m0.50	0.214745	0.973874	187	low	very low	very low	medium	very low	very low	0.09
GA p10e1c0.00m0.50	0.217115	0.986515	421	low	very low	very low	medium	very low	very low	0.01
GA p10e1c0.50m0.50	0.219808	0.744541	19	low	high	medium	medium	very low	very low	0.10
GA p10e1c0.95m0.50	0.234276	0.514165	9	low	very high	high	medium	low	low	0.10
GA p10e1c0.05m0.95	0.237517	0.973043	205	low	very low	very low	medium	very low	very low	0.10
GA p10e1c0.00m0.95	0.241505	0.975454	452	low	very low	very low	medium	very low	very low	0.01
GA p10e1c0.50m0.95	0.241678	0.743528	27	low	high	medium	medium	very low	very low	0.10
GA p10e1c0.95m0.95	0.250849	0.516142	11	low	very high	high	medium	low	low	0.11
GA p10e1c0.05m0.00	0.313629	0.974470	241	low	very low	very low	very low	very low	very low	0.09

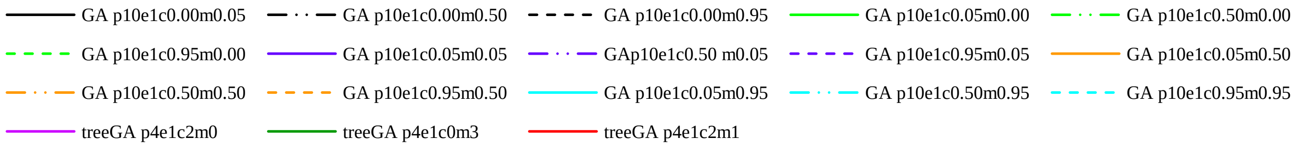


Fig. 9. Legend for charts.

REFERENCES

[1] Altenberg, L., The Schema Theorem and Price's Theorem, in Foundations of Genetic Algorithms 3, D. Whitley and M. Vose, Editors. 1995, Darrell Whitley and Michael Vose: San Francisco CA. p. 23-49.

[2] [Author], 2009.

[3] Barnes, J.W., Dokov, S., Dimova, G., and A. Solomon, A Theory of Elementary Landscapes, Applied Mathematics Letters, 2003. 16(3):p. 337-343.

[4] Borenstein, Y. and R. Poli. Information Landscapes, in Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, p. 1551-1522. 2005. Washington DC: ACM Press.

[5] Borenstein, Y. and R. Poli. Information Landscapes and Problem Hardness, in Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, p. 1425-1431. 2005. Washington DC: The ACM Press.

[6] Borenstein, Y. and R. Poli. Information Landscapes and the Analysis of Search Algorithms, in Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, p. 1287-1294. 2005. Washington DC: The ACM Press.

[7] Dimova, B., Barnes, J.W., Popova, E. and B. Colletti, Some Additional Properties of Elementary Landscapes. Applied Mathematics Letters, 2008. 22(2):p. 232-235.

[8] Frauenfelder, H., et al., eds. Landscape Concepts in Physics and Biology. Special Issue of Physica D. Vol. 107, p. 2-4. 1997, Elsevier Science: Amsterdam, Holland.

[9] Holland, J. Adaptation in Natural and Artificial Systems. Ann Arbor: The University of Michigan Press, 1975.

[10] Hordijk, W., A Measure of Landscapes. Evolutionary Computation, 1996. 4(4): p. 335-360.

[11] Jones, T., One Operator, One Landscape, in Technical Report 95-02-025. 1995, Santa Fe Institute: Santa Fe NM.

[12] Jones, T. and S. Forrest. Fitness Distance Correlation as a Measure of Problem Difficulty for Genetic Algorithms in L.J. Eshelman, ed., Proceedings of the Sixth International Conference on Genetic Algorithms, p. 184-192. 1995: Morgan Kaufmann.

[13] Kauffman, S.A., Adaptation on Rugged Fitness Landscapes, in Lectures in the Sciences of Complexity, D. Stein, Editor. 1989, Addison-Wesley. p. 527-618.

[14] Koza, J.R. Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press, Cambridge, 1992.

[15] Lipsitch, M. Adaptation on Rugged Landscapes Generated by Iterated Local Interactions of Neighboring Genes in Proceedings of the Fourth International Conference on Genetic Algorithms, p. 128-135. 1991. La Jolla CA: Morgan Kaufmann.

[16] Manderick, B., M. de Weger, and P. Spiessens. The Genetic Algorithm and the Structure of the Fitness Landscape in Proceedings of the Fourth International Conference on Genetic Algorithms, p. 143-150. 1991. La Jolla CA: Morgan Kaufmann.

[17] Merz, P. and B. Freisleben, Fitness Landscapes and Memetic Algorithm Design, in New Ideas in Optimization, D. Corne, M. Dorigo, and F. Glover, Editors. 1999, McGraw-Hill: London. p. 245-260.

[18] Merz, P. Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies. PhD thesis, Department of Electrical Engineering and Computer Science, University of Siegen, Germany, 2000.

[19] Raidl, R., and J. Gottlieb, Empirical Analysis of Locality, Heritability and Heuristic Bias in Evolutionary Algorithms: A Case Study for the Multidimensional Knapsack Problem. Evolutionary Computation, 2005. 13(4): p. 441-475.

[20] Reeves, C.R. Fitness Landscapes and Evolutionary Algorithms. In AE'99: Selected Papers from the 4th European Conference on Artificial Evolution, 1999. Springer-Verlag, p. 3-20.

[21] [Author], 2005.

[22] [Author], 2007.

[23] Smith, T., et al., Fitness Landscapes and Evolvability. Evolutionary Computation, 2002. 10(1): p. 1-34.

[24] Stadler, P., Landscapes and Their Correlation Functions. Journal of Mathematical Chemistry, 1996(20): p. 1-45.

[25] Tavares, J., Pereira, F.B., and E. Costa, Multidimensional Knapsack Problem: A Fitness Landscape Analysis. IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics. 2008. 38(3): p. 604-616.

[26] Vassilev, V., T. Fogarty, and J. Miller, Information Characteristics and the Structure of Landscapes. Evolutionary Computation, 2000. 8(1): p. 31-60.

[27] Vassilev, V.K. Information Analysis of Fitness Landscapes in P. Husbands and I.Harvey, eds, Proceedings of the Fourth European Conference on Artificial Life. 1997. Brighton, England: The MIT Press.

[28] Weinberger, E., Correlated and Uncorrelated Fitness Landscapes and How to Tell the Difference. Biological Cybernetics, 1990. 63: p. 325-336.

[29] Weinberger, E., Local Properties of Kauffman's N-K model: A Tunably Rugged Energy Landscape. Physical Review A, 1991. 44(10): p. 6399-6413.

[30] Whitley, D., Sutton, A. M., and A.E. Howe, A. E., Understanding Elementary Landscapes. In Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, July 2008. M. Keijzer, Ed. GECCO '08. ACM, New York, NY, 585-592.

[31] Wright, S. The Roles of Mutation, Inbreeding, Crossbreeding and Selection in Evolution in D.F. Jones, ed., Proceedings of the Sixth International Congress on Genetics, p. 356-366. 1932. Brooklyn NY: Brooklyn Botanic Garden.