Evolutionary Visual Exploration: Experimental Analysis of Algorithm Behaviour

Waldo Cancino,
INRIA Saclay - Île-de-France
Bat 660, Universite Paris-Sud
91405 Orsay Cedex, France
Waldo.Cancino@inria.fr

Nadia Boukhelifa
INRIA Saclay - Île-de-France
Bat 660, Universite Paris-Sud
91405 Orsay Cedex, France
Nadia.Boukhelifa@inria.fr

Anastasia Bezerianos
L. Recherche en Informatique
Bat 650, Universite Paris-Sud
91405 Orsay Cedex, France
Anastasia.bezerianos@lri.fr

Evelyne Lutton
INRA-AgroParisTech
78850 Grignon, France
Evelyne.Lutton@grignon.inria.fr

ABSTRACT

Recent publications in the domains of interactive evolutionary computation and data visualisation consider an emerging topic coined *Evolutionary Visual Exploration (EVE)*. EVE systems combine visual analytics with stochastic optimisation to aid the exploration of complex, multidimensional datasets. In this work we present an experimental analysis of the behaviour of an EVE system that is dedicated to the visualisation of multidimensional datasets, which are generally characterised by a large number of possible views or projections. EvoGraphDice is an interactive evolutionary system that progressively evolves a small set of new dimensions, to provide new viewpoints on the dataset, in the form of linear and non-linear combinations of the original dimensions. The criteria for evolving new dimensions are not known a priori and are partially specified by the user via an interactive interface: (i) The user selects views with meaningful or interesting visual patterns and provides a satisfaction score. (ii) The system calibrates a fitness function to take into account the user input, and then calculates new views, with the help of an evolutionary engine. In previous work (an observational study), we showed that EvoGraphDice was able to facilitate "exploration" tasks, helping users to discover new interesting views and relationships in their data. Here, we focus on the system's "convergence" behaviour, conducting an experiment with users who have a precise task to perform. The experimental task is set up as a geometrical game, and collected data show that EvoGraphDice is able to "learn" user preferences in a way that helps users fulfill their task (i.e. converge to desired solutions).

GECCO'13 Companion, July 6–10, 2013, Amsterdam, The Netherlands.
Copyright 2013 ACM 978-1-4503-1964-5/13/07 ...$15.00.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; H.5.m [**Information Interfaces and Presentation**]: Miscellaneous

Keywords

Interactive evolutionary computation; visual analytics; genetic programming

1. INTRODUCTION

Recent work combining visualisation and optimisation [7] shows a growing interest of the Visual Analytics community toward optimisation and interactive evolutionary algorithms, and of the evolutionary computation community toward visualisation [6, 8, 10].

The purpose of visual exploration is to find meaningful patterns in the data that can lead to insight [15]. In high-dimensional datasets, this task becomes challenging as viewers may be faced with a large space of alternative views on their data. The quest for finding an interesting subset of elements, such as data dimensions or projections, is shared by other research fields. In machine learning and data mining this issue is often treated as a classification problem where dimensions refer to features [12]. Formalised as an optimization problem (e.g. Evolutionary Computation), interesting features are selected based on a fitness function [4].

For scientific and engineering applications, Interactive Evolutionary Computation (IEC) is interesting when the exact form of a more generalised fitness function is not known or is difficult to compute, say for producing a visual pattern that would interest a particular user. Here, the human visual system, together with the emotional and psychological responses of the user in question can outperform a pattern detection or learning algorithm. Similarly, Evolutionary Visual Exploration[1] (EVE) relies on the premise that IEC can help guide users in exploring complex datasets.

EvoGraphDice [3, 4] is an EVE prototype for the exploration of multidimensional datasets: the system proposes interesting views based on both objective measures, such as visual patterns in the two-dimensional projections of the

[1] http://www.aviz.fr/Research/EVE

data, and subjective measures corresponding to user satisfaction with the presented view. *EvoGraphDice* was built as an extension of an existing scatterplot matrix inspection tool. It uses low dimension projection to handle data multidimensionality, and linear and non-linear combinations of dimensions for an axis of the projection plane to propose alternative views. User exploration is guided by an Interactive Evolutionary Algorithm (IEA) which can both generate new views and adapt to user interest (see figure 1).

The paper is organised as follows. Section 2 presents the main mechanisms of *EvoGraphDice*. The experimental setup and analysis are described in section 3. We discuss our main findings in section 4, before giving a conclusion and future research in section 5.

2. EVOLUTIONARY VISUAL EXPLORATION

In this section we first describe the visual interface of *EvoGraphDice* and the main components of the genetic engine we implemented behind the visualization tool. Combining visual analytics with stochastic optimisation using our framework (EVE) raises interesting issues related to algorithm convergence, described next, that motivated our experimental study.

2.1 EvoGraphDice Visual Interface

EvoGraphDice has been built over an existing visualization tool (GraphDice [2, 5]) to manage the various projections of the data. Views are organised in a scatterplot matrix (SPLOM) of 2D projections, figure 1-a. Users can do brushing and linking using a lasso tool. *EvoGraphDice* displays the dimensions proposed by the IEA as additional rows (and columns) in the SPLOM. The system initially displays dimensions returned by a PCA, after which the user can evolve new dimensions by pressing the "evolve" button, figure 1-d. The proposed views are displayed in yellow background; the darker the color the more interesting the view. The system provides an initial score (1 to 5) for each new view but the user can adapt this score using the slider in figure 1-d. User evaluated cells are flagged (using a small black square) to distinguish them from system evaluated cells. *EvoGraphDice* can be initialised at any time using the "restart" button which resets parameters of the IEA. Users can save views (figure 1-f) and bring them back into the SPLOM if they have been replaced during the exploration.

The current population of evolved dimensions is also displayed as a table (figure 1-h) where each row corresponds to a combined dimension described by a mathematical expression and various components of the fitness function such as the scagnostics measures defined in section 2.2. The user can edit an individual (i.e. a combined dimension) using the "dimension editor" in figure 1-j, or limit the dimension search space (figure 1-i), which results in a system reset similar to pressing the "restart" button. Note that many EA parameters can be tuned, such as the fitness threshold and crossover/mutation/replacement rates (see [4]).

A first version of *EvoGraphDice* [4] was based on an IEA that only manipulated linear combinations of dimensions. Recent extensions, described in [3], are (i) a Genetic Programming (GP) algorithm allowing the manipulation of nonlinear combinations of dimensions as variable size mathematical formulae, (ii) user assessment of proposed views is explicitly captured via a slider, (iii) a surrogate function based on some specific geometric measurements (*scagnostics*) is used to predict and simplify the interactions of the user with the IEA, (iv) color highlighting of cells is used to draw user attention to the most interesting views.

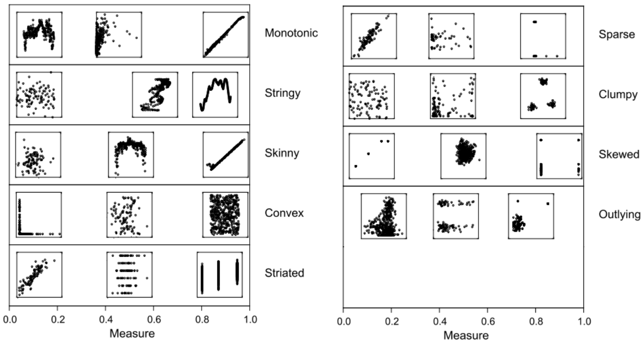


Figure 2: Nine scagnostics measures from [16].

In [3], we validated our general approach of combining visual analytics with stochastic optimisation to aid data exploration, by conducting an observational study with expert users from various domains. Our results showed that *EvoGraphDice* can help users quantify qualitative hypotheses and try out different scenarios to dynamically transform their data. Importantly, it allowed our experts to think laterally, better formulate their research questions and build new hypotheses for further investigation. Our work demonstrated that tightly combined visualization and optimisation techniques yields exciting results in data analysis, and opens new venues for research. It also highlighted challenges such as monitoring algorithm convergence which we try to address in this paper in more detail.

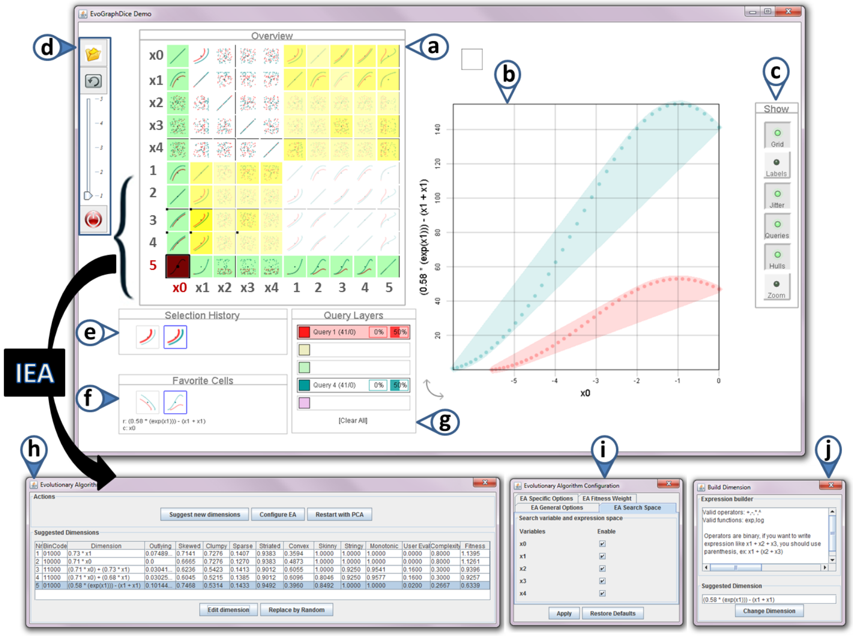
2.2 User fatigue

Despite efforts to design good user interfaces for IEA, human interaction raises "user bottleneck" problems [1,11,14]. Among the various possible strategies to deal with this issue, we choose to use a small population size of suggested dimensions, and to deploy an approximated user model: a *surrogate function* based on a series of geometric measures that have proven statistical properties and are computable for moderately large data sets [16]. *Scagnostics*[2] are based on geometric graphs which are calculated from areas, perimeters and lengths of these graphs. They include nine measures to characterise scatterplots (figure 2) and are useful for quickly discovering regularities and anomalies in scatterplot matrices. The underlying algorithm detects different types of point distributions including multivariate normal, log normal, multinomial, sparse, dense, convex and clusters. It does so by binning, detecting outliers and computing measures based on the following three statistical properties: *shape* for convex, skinny and stringy distributions; *trend* for monotonic distributions; and *density* for skewed, clumpy, outlying, sparse and striated.

2.3 Search Space and Genetic Engine

The space searched by the evolutionary process is the set of all dimensions that can be built by combining the initial dimensions with operators and constants, encoded as trees

[2] Available as a free downloadable package in R from `http://www.rforge.net/scagnostics/`



r: (0.58 * (exp(x1))) - (x1 + x1)
c: x0

Nr	BinCode	Dimension	Outlying	Skewed	Clumpy	Sparse	Striated	Convex	Skinny	Stringy	Monotonic	User Eval	Complexity	Fitness
1	01000	0.73 * x1	0.07489...	0.7141	0.7276	0.1407	0.9383	0.3594	1.0000	1.0000	1.0000	0.0000	0.8000	1.1395
2	10000	0.71 * x0	0.0	0.6665	0.7276	0.1270	0.9383	0.4873	1.0000	1.0000	1.0000	0.0000	0.8000	1.1261
3	11000	(0.71 * x0) + (0.73 * x1)	0.03041...	0.6236	0.5423	0.1413	0.9012	0.6055	1.0000	0.9250	0.9541	0.1600	0.3000	0.9396
4	11000	(0.71 * x0) + (0.68 * x1)	0.03025...	0.6045	0.5215	0.1385	0.9012	0.6096	0.8046	0.9250	0.9577	0.1600	0.3000	0.9257
5	01000	(0.58 * (exp(x1))) - (x1 + x1)	0.10144...	0.7468	0.5314	0.1433	0.9492	0.3960	0.8492	1.0000	1.0000	0.0200	0.2667	0.6339

Variables	Enable
x0	☑
x1	☑
x2	☑
x3	☑
x4	☑

Valid operators: +,-,*,^
Valid functions: exp,log

Operators are binary, if you want to write expression like x1 + x2 + x3, you should use parenthesis, ex: x1 + (x2 + x3)

(0.58 * (exp(x1))) - (x1 + x1)

Figure 1: EvoGraphDice prototype showing an exploration session of a synthetic dataset (the spiral game). Widgets: (a) an overview scatterplot matrix showing the original data set of 5 dimensions (x0..x4) and the new dimensions (1..5) as suggested by the evolutionary algorithm, (b) main plot view, (c) tool bar for main plot view, (d) a tool bar with (top to bottom) "favourite" toggle button, "evolve" button, a slider to evaluate cells and a restart (PCA) button, (e) the selection history tool, (f) the favourite cells window, (g) the selection query window, (h) IEA main control window, (i) window to limit the search space, and (j) dimension editor.

according to the Genetic Programming (GP) framework [9]. Evolved expressions can be any combination using $+$, $-$, $*$, $/$, $(.)^{(.)}$, *exp* and *log* operators.

A small set of combined dimensions is evolved: if n is the number of initial dimensions, a population of another n combined dimensions is evolved. Each time the user clicks on the "evolve" button, a new generation is computed, and the user can explicitly evaluate a cell in the SPLOM with a slider attached to each new view.

Users evaluate cells, while the evolutionary engine manipulates individuals, i.e. new dimensions. A user evaluation of a dimension is computed by averaging the user evaluations of all cells that involve the new dimension.

A set of *a priori* interesting dimensions has been chosen as a starting point: a PCA analysis is performed [13] on the original data and the corresponding n linear combinations form the initial population.

The fitness function, that is optimised by the genetic engine, is a sum of three terms:

1. **The surrogate function** f_{sc}, that plays the role of a predictor. It is based on scagnostics measurements computed for every scatterplot cell y_i, x_j of each dimension y_i. The corresponding fitness term is a linear combination of the highest values of the scagnostics $(SC_k(y_i, x_j))$ of each scatterplot cell (y_i, x_j):
$$f_{sc}(y_i) = \sum_{k=1..9} w_k(\max_j SC_k(y_i, x_j)). \qquad (1)$$
The weights w_k that govern the relative importance of each scagnostic measurement are initialised to a uniform weight (1/9). Then, when there are more than $m > n$ user evaluations recorded (n, the number of variables), the w_k are updated via a simple multilinear regression on the m past interactions ($m \geq n$ corresponds to the length of a "memory" of the system).

2. **A complexity term** that favours dimensions made of a small number of variables and simple mathematical expressions :
$$f_c(y_i) = \left(1 - \frac{nvars(y_i)}{n}\right) \times \frac{1}{depth(y_i)}, \qquad (2)$$
$nvars(y_i)$ is the number of original variables involved in the mathematical expression of y_i, and $depth(y_i)$ is the depth of the GP tree representing y_i.

3. **A user evaluation term,** $f_u(y_i)$, that is an average of the user evaluation for each cell corresponding to y_i (range of 1 to 5 from "bad" to "excellent").

Diversity management: for small populations sizes, there is a major risk of premature convergence if no diversity preservation mechanism exists. To deal with this issue in *EvoGraphDice* each time a new dimension y'_i is generated, its Euclidean distance to the current population is computed. If y'_i is too close to one of the individuals of the current population, it is replaced by a random individual.

2.4 Issues of Algorithm Convergence

As we are dealing with a small population of dimensions evolved during only a few generations (see experimental section next), the algorithm cannot be considered as having converged in the classical sense. Theoretical analysis considers two main mechanisms that govern the behaviour of EAs:

focus (convergence or exploitation) and diversity (random search or exploration). In their most classical uses, i.e. computationally expensive optimisation, the exploitation mechanism is privileged and the exploration component is used only to ensure the robustness of the results. In the interactive framework where creativity or new feature discovery are sought, the same mechanisms operate but with a different balance: exploration capability seems to have a bigger impact.

Additionally, talking about convergence for EVE systems is even more difficult as usually the users themselves do not clearly know what they are searching for. In previous experimental analysis [3], it has been noticed that the "guided" exploration ability of *EvoGraphDice* is exploited in different ways by experts: some explored, thus it can be said they focused on the random search ability of the algorithm, while others exploited with longer runs (>10 generations) thus focused on guided search/convergence. In both cases, the IEA provides a unified framework for users that sometimes are interested in focussed search, e.g. if they know what they want, or in explorative suggestions if they know less what they may find.

The experiments described in the next sections have been designed to evaluate the focused search/exploitation mechanisms of *EvoGraphDice* where the user has a precise and well defined task to perform.

3. ALGORITHM BEHAVIOUR: EXPERIMENTAL ANALYSIS

We conducted an experiment to collect data about user interactions and the fitness function. In particular, we wanted to monitor the IEA's convergence focusing on (i) the learning behaviour of the algorithm; and (ii) its ability to adapt to user focus.

3.1 Experiment Setup and Procedure

We run our experiment with 12 participants (5 female), ages $23 - 43$ (mean 28.5). Participants were researchers from two different institutions who had limited experience with SPLOM-based visualizations (only 3 previously used a SPLOM-based visualization tool). We ran *EvoGraphDice* on an HP Z800 workstation PC with a 1900 dual monitor (1280 x 1024 screen resolution), and on a MacBookAir connected on a 30″ monitor (same resolution). Each session with a user lasted $1 - 2$ hours.

A 5D dataset was synthesised with two enclosed curvilinear dependencies between two variables (x0 and x1) and random data for the rest of the dimensions. Participants were asked, after a brief introduction to the tool, to evolve a scatterplot where it is possible to separate the two curves in figure 3 (left) with a straight line, and were not given a time limit to complete the task (this task is equivalent to separating the two convex hulls in figure 3). Two levels of difficulty for the game were generated, where difficulty relates to the amount of enclosure between the curves–the bigger the overlap area between the curves the more difficult it is to find the solution. Participants started with level 0 and depending on their performance, they also did level 1. For struggling participants we allowed them to restrict the search space to dimensions x0 and x1 in the hope of helping them find a solution more quickly. Participants stopped the game when they found a solution or when they felt the tool is no longer proposing interesting views (for struggling users, a minimum exploration time of 20 minutes was always respected). With the exception of one user, all participants successfully evolved a view separating the two convex hulls for level 0 (average time to find a solution was 17 minutes), but only 6 out of the 11 participants who tried level 1 managed to find a solution (average time 11.5 minutes). We note that some users needed several tries (i.e. game restarts) in order to solve the game.

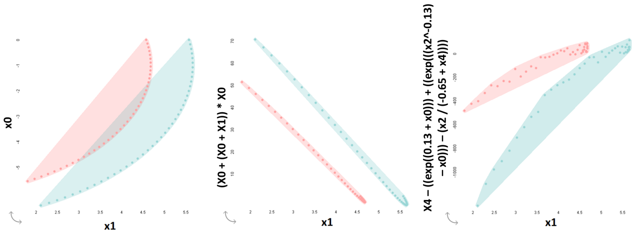


Figure 3: Two different solutions (screenshots of plots) for the training game problem-level0 (left) that involve a simple dimension combination (middle) and a complex formula (right).

Table 1: Log data capturing information on the GP.

For each generation	9 scagnostics weights, w_k.
For each individual	generation number, genome (math. formula), surrogate function term, f_{sc}, complexity term, f_c, average user evaluation, f_u, resulting fitness.
For each evaluated cell	generation number, 9 scagnostics, predicted evaluation $(= \sum w_k$ x Cell Scagnostics), user evaluation.

3.2 Data Collection

Log data was gathered for further analysis and includes two types of information related to: (a) user interactions with the tool such as cell selections and evaluations via the slider; and (b) genetic engine status at each generation (see Table 1), such as details about the individuals in each generation (including their fitness components and scagnostics scores), the cells these individuals participate in (i.e. the yellow cells in figure 1-a); and (c) the overall learned scagnostics weights. A total of 27 log files were collected and further analysed.

3.3 Data Analysis

3.3.1 System Evaluation vs. User Evaluation

User scores of evaluated cells (i.e. when the user moves the slider) have been compared to their "predicted" values, based on the current scagnostic weights learned at each generation and the scagnostics values of the corresponding cell (see equation 1): $\sum_{k=1..9} w_k(SC_k(y_i, x_j))$. These values are averaged per generation and are plotted in figure 4 (for successful players) and figure 5 (for unsuccessful players) for

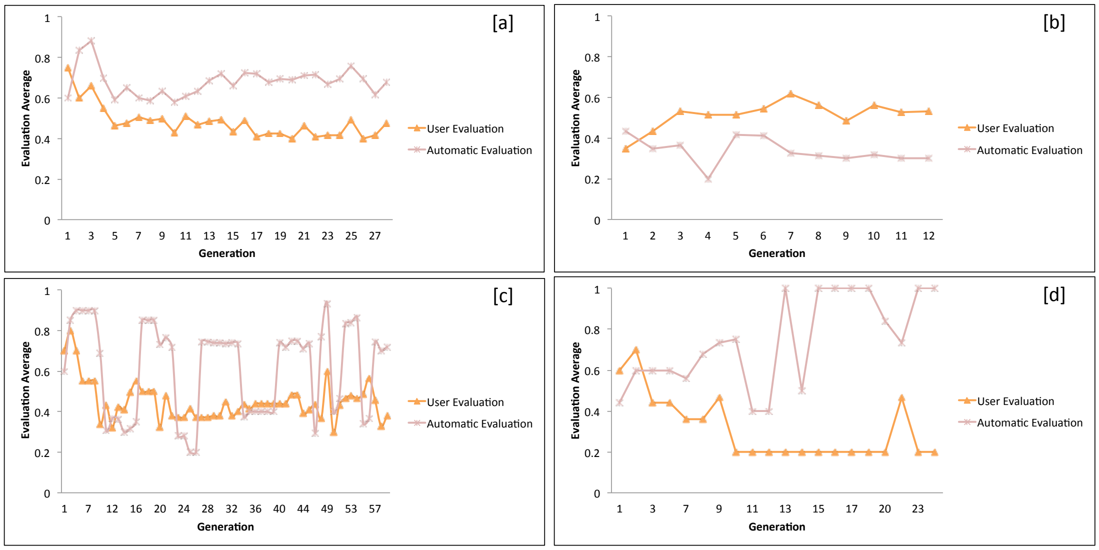


Figure 4: Average user evaluation of cells versus scagnostics-based prediction over generations for four successful participants [a–d] for level 0 of the game. The orange curves correspond to user evaluations while the pink ones represent the predicted values.

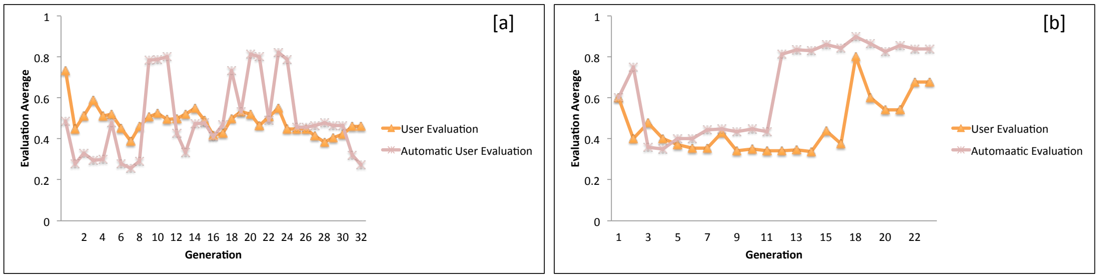


Figure 5: Average user evaluation of cells versus scagnostics-based prediction over generations for two unsuccessful participants [a–b] for level 0 of the game. The orange curves correspond to user evaluations while the pink ones represent the predicted values.

level 0 of the game. The predicted values seem to roughly follow the user evaluations in the majority of cases, independently from the output of the game (success or failure). An exception to this is figure 4-d (generations $17 - 23$) where a user decided to rate the cells according to a different criterion (aesthetics) after having obtained the result.

3.3.2 *Learning User Strategy Through Scagnostics*

A second type of analysis was performed on the scagnostics weights distribution along generations. For comparison purposes (and since the number of generations per exploration session differ between users), generation values have been averaged in three bins, corresponding to the start, middle and end of the game session. We tried to get the same bin size for all groups when possible, and at the very least ensure start and end bins always have the exact same size when integer division by 3 was not possible. Figures 6 to 9 display the 9 histograms of various game sessions, for levels 0 and 1, with limited and unlimited search spaces.

For level 0-no restricted space (figure 6), successful explorations are concentrated around either sparse, and/or a combination of skinny and stringy distributions (if we focus on the end generation bins). For the unsuccessful session, there was no strong pattern, as expected. Also, for sessions where sparse was a dominant exploration pattern, we can see a clear increase from start to end generations, e.g. figure 6-c, e and f.

When restricting the search space for level 0, the solution to the game was found much quickly than when the search space was not limited (5 generations as opposed to on average 23 generations as seen in figure 7-a). The successful exploration strategy was a combined approach between sparse and stringy. For the unsuccessful exploration session (figure 7-b), it might show that just ‘skinny’ as the dominant scagnostic might not be enough to reach a solution, or that the user did not explore enough generations (given the chosen search strategy, a solution might be just around the corner).

Again, and for level 1 this time, in figure 8 the unsuccessful candidate had a random exploration pattern, while focusing on sparse distributions still leads to a successful exploration strategy. When limiting the search space (figure 9), the average number of dimensions ran to get a successful solution was 24 (slightly higher than for level 0), but in general the same observations regarding successful pattern searches mentioned above also hold for this game level.

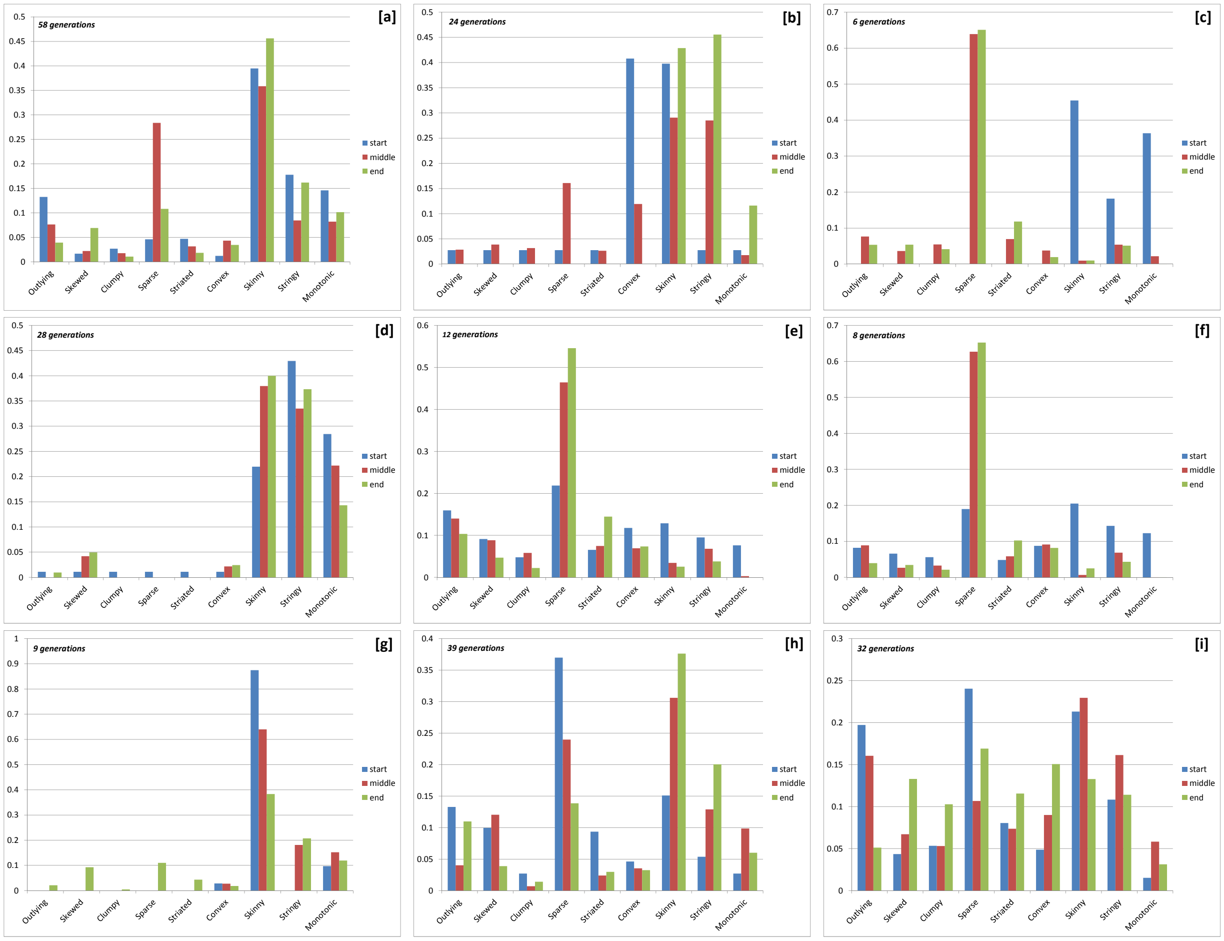


Figure 6: Scagnostics weights over time for successful participants [a–h] and unsuccessful participants [i] for level 0 of the game without limiting the search space.

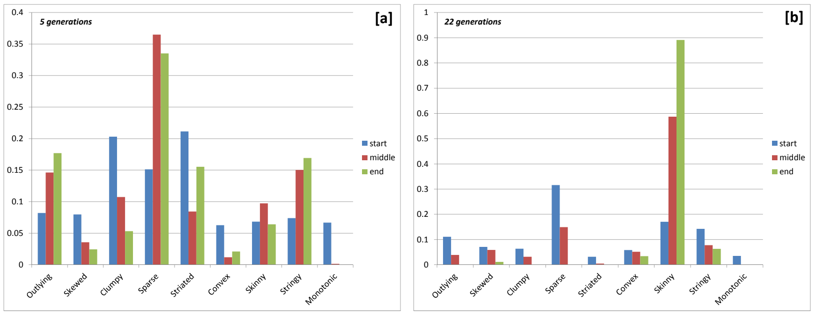


Figure 7: Scagnostics weights over time for successful participants [a] and unsuccessful participants [b] for level 0 of the game when restricting the search space to dimensions x0 and x1.

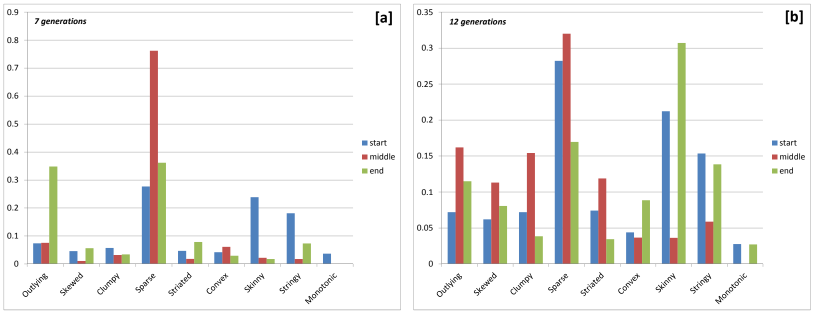


Figure 8: Scagnostics weights over time for successful participants [a] and unsuccessful participants [b] for level 1 of the game without restricting the search space.

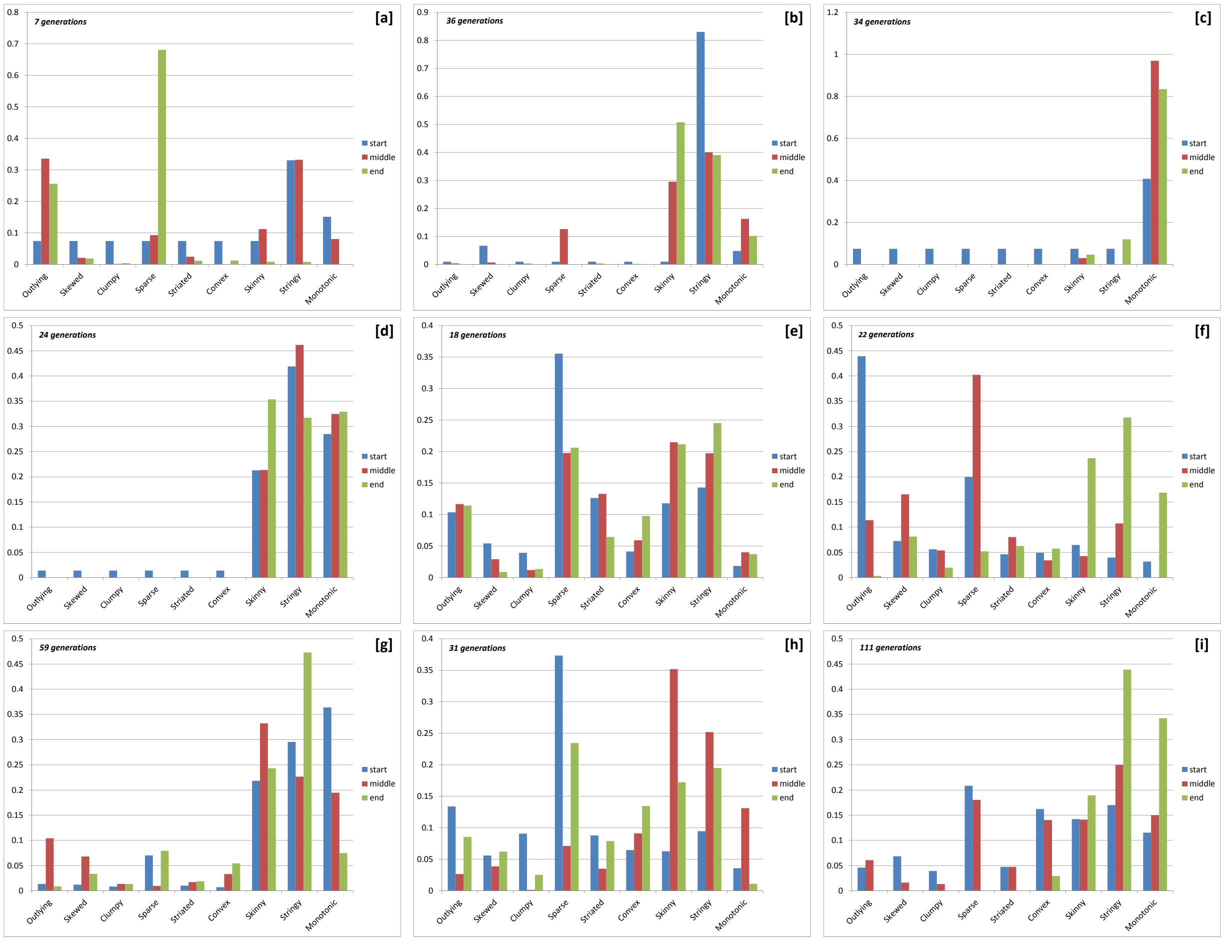


Figure 9: Scagnostics weights over time for successful participants [a–e] and unsuccessful participants [f–i] for level 1 of the game when limiting the search space to dimensions x0 and x1.

4. DISCUSSION

EvoGraphDice seems to exhibit a learning behaviour controlled by the diversity component which aims at maintaining an exploration capability (section 3.3.1). This effect is noticeable even in the case of unsuccessful game sessions (e.g. figure 5-b) and corresponds to an exploitation component (focus) of the genetic engine, regardless of user aim. Analysis of section 3.3.2 shows different exploration patterns for successful and unsuccessful sessions. The analysis of the content of the surrogate function, via the observation of the variation of the learned weights of the scagnostics measurement, highlights a difference in users' focus of attention (i.e. sought after pattern). For successful game sessions, there are clearly two strategies, one tending to "unfold" the curved shapes by favouring linear scagnostic measurements (e.g. middle solution in figure 3), the other trying to spread the figures laterally by favouring sparse scagnostics (e.g. right solution in figure 3).

The surrogate function that "approximates" the user evaluation, is clearly not able to embed the explicit aim of the game (that is separating the convex hulls of two geometrical subsets) as it only performs calculations on the whole set of points of the scatterplot. However, what is clear from the analysis is that the surrogate function seems to be discriminative enough to allow various search strategies.

5. CONCLUSION AND FUTURE WORK

This work is focussed on the evaluation of the convergence behaviour of the *EvoGraphDice* system through a user study having a well specified task, while previous studies [3] assessed the tool's exploration and lateral thinking capabilities. The experiments and analysis presented above exhibit an obvious learning behaviour that seems to facilitate the algorithm's convergence toward a satisfying solution, while still allowing users to adopt various search strategies. The data we collected for this study have not been fully explored and future work will involve more in-depth analysis, for example examining trends for different GP runs per user exploration session (instead of concatenating all generation runs). In addition, we would like to re-examine our notion of start-middle-end of an exploration session to find more meaningful ways to characterise various stages of exploration (e.g. change of user focus).

Our current formal and informal experimental analysis will guide our future developments. We will for instance improve the surrogate function calculation in order to detect more complex visual patterns (beyond those currently detected by Scagnostics, e.g. kurtosis and entropy). A bridging of *EvoGraphDice* with existing statistical packages will also be considered, in order to combine powerful statistical analysis with flexible and intuitive visual exploration.

Acknowledgements:
We thank all the users who participated in our experiment.

6. REFERENCES

[1] W. Banzhaf. *Handbook of Evolutionary Computation*, chapter Interactive Evolution. Oxford University Press, 1997.

[2] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J.-D. Fekete. GraphDice: A System for Exploring Multivariate Social Networks. *Computer Graphics Forum (EuroVis 2010)*, 29(3):863–872, 2010.

[3] N. Boukhelifa, W. G. Cancino Ticona, A. Bezerianos, and E. Lutton. Evolutionary Visual Exploration: Evaluation With Expert Users. *Computer Graphics Forum (EuroVis 2013, June 17–21, 2013, Leipzig, Germany)*, 32(3), June 2013.

[4] W. Cancino, N. Boukhelifa, and E. Lutton. Evographdice: Interactive evolution for visual analytics. In *IEEE Congress on Evolutionary Computation, June 10-15*, 2012. June 10-15, Brisbane, Australia.

[5] N. Elmqvist, P. Dragicevic, and J.-D. Fekete. Rolling the Dice: Multidimensional Visual Exploration using Scatterplot Matrix Navigation. *IEEE Transactions on Visualization and Computer Graphics (Proc. InfoVis 2008)*, 14(6):1141–1148, 2008.

[6] A. Guettala, F. Bouali, C. Guinot, and G. Venturini. A user assistant for the selection and parameterization of the visualizations in visual data mining. In *Information Visualisation (IV), 2012 16th International Conference on*, pages 252–257, 2012.

[7] H. Hauser. The iterative process of interactive visual analysis. Keynote talk at the EuroVA 2012 workshop, June 2012.

[8] N. Hayashida and H. Takagi. Visualized IEC: interactive evolutionary computation with multidimensional data visualization. In *IECON 2000. 26th Annual Conference of the IEEE*, volume 4, pages 2738–2743, 2000.

[9] J. R. Koza. *Genetic Programming* . MIT Press, 1992.

[10] X. Llorà, K. Sastry, F. Alías, D. E. Goldberg, and M. Welge. Analyzing active interactive genetic algorithms using visual analytics. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 1417–1418, New York, NY, USA, 2006. ACM.

[11] R. Poli and S. Cagnoni. Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement. In *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 269–277. Morgan Kaufmann, 1997.

[12] J. Seo and B. Shneiderman. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *in Proceedings of IEEE Symposium on Information Visualization*, pages 65–72, 2004.

[13] I. Smith. A tutorial on Principal Component Analysis, 2002.

[14] H. Takagi. Interactive Evolutionary Computation: System Optimization Based on Human Subjective Evaluation. *INES'98*, 1998.

[15] J. J. Thomas and K. A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* National Visualization and Analytics Ctr, 2005.

[16] L. Wilkinson and G. Wills. Scagnostics Distributions. *Journal of Computational and Graphical Statistics*, 17(2):473–491, 2008.