Implementation Issues for an Interactive Evolutionary Computation System

M. R. N. Shackelford
Advanced Computational Technologies
2 The Priory, Winnersh
Berks, RG41 5DE
mshackelford@ad-comtech.co.uk

ABSTRACT
The design and development of an Interactive Evolutionary Computation (IEC) system needs to take into account the implementation issues found when delivering the system to "Real World" users. This paper reports and reflects on the implementation issues found when rolling-out an IEC-based Programme Management system [1]. Although the system under discussion was for a specific application, it is felt that the issues and their resolution are generally applicable to a wide range of IEC systems.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Ergonomics, Evaluation/ methodology, Graphical user interfaces (GUI), Screen design, User-centered design.*

General Terms
Design, Reliability, Human Factors.

Keywords
Interactive, Evolutionary, Computation, GUI.

1. INTRODUCTION
An Interactive Evolutionary Computation system includes within the process a human being, used to set, or assist the application in setting, the Fitness Function values. This is often necessary when the algorithm that defines the Fitness Function is either too complex to specify, or in many cases, impossible to quantify and code.

The issues found and resolved when implementing an Interactive User Interface to the Evolutionary Algorithm for the system (described in [1]) are outlined in the following sections:

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

- **Encouraging the User** – overcoming initial resistance, simplifying the user interface, and providing suitable feedback.
- **Preventing Fatigue** – avoiding user fatigue and unnecessary user interaction.
- **Handing Errors** – managing the inevitable input errors made by the user when setting fitness values.
- **Areas of Concern** – further issues that need to be considered in the IEC design.

2. ENCOURAGING THE USER
When developing an Evolutionary Algorithm [EA] based program to deliver a new method for solving existing problems it is necessary to be able to define a Fitness function. Often this is fairly straight-forward (for example when designing a bridge the calculations for strength and stiffness can be used to determine the fitness based on the geometry), but in some cases (such as the Programme Management system discussed in this paper [1]) the fitness is derived from a wide range of parameters based on the project manager's knowledge and experience.

This information is hard, if not impossible, to formalise and it is at this point that the design of the EA requires the addition of an Interactive component. This issue is discussed in [2], [3], [4], [5], [6], [7], [16] as well as [1].

When a computer user starts to use a new, unfamiliar system there is often resistance to using it or else a great deal of training required. The Interactive component of the EA must not only be seen to be familiar (and even friendly), but also enable the user to easily use his tacit, implicit knowledge in setting the fitness of the various individuals in each generation.

To this end the interface must meet the following criteria:

- **Explicable**: Display the information about each generation in a format that the user can easily understand and recognise.
- **Familiar**: Display the information using a format that the user is familiar with, rather than presenting an alien view.
- **Manageable**: Show appropriate amounts of information. Too little and the user is unable to make sensible decisions, too much and the user is overwhelmed by the amount of data.
- **Intuitive**: Enable direct (and obvious) input and manipulation of information.

- **Visible**: The interface must enable the user to have a level of understanding of the underlying process.
- **Feedback**: The effect of settings entered by the user should be apparent.

In the IEC for Programme Management, the above principles meant that the information display took the form of a GANNT chart (standard project management diagram of Bars against a timescale) along with Resource profiles (cumulative resource usage over time).

16 charts were found to be the most appropriate number to be displayed at each iteration, both for the population size and for the user to cope with the information.

Fitness was set by the user with a single Mouse-click for each individual, and because the display was based on GANNT charts, each iteration effectively showed the results of the selections from the previous generation.

3. PREVENTING FATIGUE

As part of the development of the IEC ([1]) the system was presented to a number of focus groups, and there were a number of negative observations, amongst which the most popular seemed to be:

"*Too boring to use*"

"*Too much interaction required*"

"*Too much time required*"

The issues involved with fatigue include:

- **Time-consuming** – setting the fitness for many individuals in many generations
- **Boring** – repetitive tasks soon lose the user's interest
- **Unreliable** – the user's concentration falls off, allowing entry of conflicting or contradictory settings.

Takagi ([8], [9], [10]) explores human fatigue, and suggests a number of techniques to overcome this:

- Reduce number of Fitness values selected.
- Display solutions in a pre-sorted order.
- Reduce the number of solutions.
- Reduce the number of iterations.
- Embed user knowledge in the application.

Using these ideas and further suggestions from the users, the IEC system was enhanced to reduce the amount of interaction, but still required the user to confront a large number of iterations.

The next issue raised was "*Why can't the system learn what I want it to do?*"

To this end, the IEC was extended to learn the user's preferences, reported in [12], based on a case-based template system. Other ideas are reported in [13], [14], [15].

The user ran the system for the first few iterations, while the system stored templates of the preferred selections, and the IEC was then able to use these templates to predict the users preference in subsequent iterations, without requiring user interaction.

The IEC was able to learn from the user, and was able to use the templates to extract the user's implicit knowledge and experience without the user needing to formally define his preferences.

There is potential for this technique to be used in more general interactive optimisation and decision-support systems. Any such development would need to have a simple and direct graphical representation of the solutions, and the ability for the user to be able to easily enter target solutions. These target solutions may well be extrapolations of the actual solution (perhaps they could be called meta-solutions?), as in this case where the target solution is represented by a Resource Profile which has been calculated from the actual Programme Schedule based on dates and times.

4. HANDLING ERRORS

It is quite likely that, in the "rough and tumble" that is the normal life of a Project Manager (or any other IEC user), who would be trying to use the IEC system, whilst talking on the phone, swinging back on his chair and trying to avoid pouring coffee into his keyboard, that he will specify the wrong fitness score value for one or more individuals (GANTT charts). This would give the wrong fitness value to the selected chromosome and change the selection criteria for the next generation.

The occasional error is easily dealt with, given that the GA is using a population, many of whose members may share the same genes as the incorrectly chosen chromosome, it is likely that similar chromosomes can be regenerated (assuming that the error made a fit individual look worse), or (in the case of a poor individual being scored fitter) the GA will quickly abandon the upgraded individual as the laws of 'Survival of the Fittest' take action over the next few generations.

When testing the effects of user error (reported in [11]) it was shown that overall the levels of error do not have much effect until the error level reaches 7% (about 1 in 14). At this level, for a population of 16 individuals, we get over 1 error in each generation, and these errors can therefore be expected to affect the best individual once every 16 iterations.

An interesting result from these experiments was the improvement in solutions with low error levels (1% to 5%) compared to the benchmark 'perfect' (0% error rate) user. It is suggested that the IEC is using the occasional user error (when an individual is wrongly scoring) in a similar manner to the Mutation operator to allow the population diversity to be maintained and hence help broaden the search just enough to avoid poor results.

Another problem with high error rates occurs when the current highest fitness individual is wrongly scored, and therefore lost. The IEC system runs with an Elite factor set to 1, passing on the best member of the current population to the next generation without change. When the User mis-scores the best individual, and thereby lowers the individual's fitness value, the Elite function chooses the next best individual. A potential solution is to set a 'lock' on the best individual and prevent the user from changing its fitness.

5. AREAS OF CONCERN AND SUMMARY

Discussions with the focus group regarding the techniques used to resolve the issues outlined in the previous sections brought to light some further issues that should be borne in mind when developing an IEC system:

- How to determine at what point the user's concentration starts to fall off.
- What is the effect of reintroducing individuals that have already been scored as unwanted.
- The use of a GA restricts innovation by the limits of the user's knowledge.
- What is the effect of a user changing his selection criteria over time, as his preferences change as new solutions are displayed.

Many of the issues raised by the implementation of an IEC can be resolved or avoided by careful design and taking into account the existing experience and environment of the potential users. Good visualisation is key to the success of the IEC, and so significant effort should be put into the user interface design.

Using a GA within the IEC has positive effects, especially for coping with user error, and it also allows the user to combine partial solutions in the search for more acceptable results.

Fatigue is a major problem, but can be circumvented (or mitigated to a certain extent) by the use of learning techniques to develop profiles or templates of the user's preferences. This technique also enables the system to absorb the user's implicit knowledge without requiring the user to formalise this information.

6. REFERENCES

[1] Shackelford M R N, Corne D W (2001), Collaborative Evolutionary Multi-Project Resource Scheduling, in Proceedings of the 2001 Congress on Evolutionary Computation, IEEE Press

[2] Bullock S, Cartlidge J and Thompson M (2002), Prospects for Computational Steering of Evolutionary Computation, in Proc Alife VIII Workshop, MIT

[3] Esquirol P and Lopez P (1997), Constraint-Oriented Cooperative Scheduling for Aircraft Manufacturing, IEEE Expert, Jan-Feb 97

[4] Dorn J (1995), Interaction with a Scheduling System by Soft Constraints, AAAI SIGMAN Workshop, New Orleans, November 95

[5] Sims K (1991), Artificial Evolution for Computer Graphics, Computer Graphics 25(4)

[6] Baker E and Seltzer M (1994), Evolving Line Drawings, Graphics Interface '94 Proceedings

[7] Klau G W et al. (2002), The HuGS Platform: A Toolkit for Interactive Optimization, TR2002-08, in Advanced Visual Interfaces, Trento, Italy

[8] Takagi H (1998), Interactive Evolutionary Computation: System Optimization Based on Human Subjective Evaluation, IEEE Int'l Conf. On Intelligent Eng. Sys (INES '98)

[9] Takagi H (1999), Active User Intervention in an EC Search, Kyushu Institute, Fukuoka, Japan

[10] Takagi H (2001), Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation, Proc. of IEEE

[11] Shackelford M R N, Corne D W (2004), A Technique for the Evaluation of Interactive Evolutionary Systems, in Proceedings of the Sixth International Conference on Adaptive Computing in Design and Manufacture (ACDM 2004).

[12] Shackelford M R N, Corne D W (2006), Reducing Interaction Fatigue in an Interactive Evolutionary Systems, in Proceedings of the Seventh International Conference on Adaptive Computing in Design and Manufacture (ACDM 2006).

[13] Ziegler J, Banzhaf W. (2003), Decreasing the number of Evaluations in Evolutionary Algorithms by using a meta-model of the Fitness Function, EuroGP 2003, LNCS 2610.

[14] Johanson B and Poli R (1998), GP-Music: An Interactive Genetic Programming System for Music Generation with Automated Fitness Raters. In Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22-25 98

[15] Lee K-M and Street W N (2000), Learning Shapes for Automatic Image Selection, INFORMS & KORMS 1468 Seoul

[16] Machwe A.T., Parmee I. C. & Miles J. C. (2005), Integrating Aesthetic Criteria with a User-Centric Evolutionary System via a Component-based design representation, International Conference on Engineering Design ICED 05, Melbourne, August 15-18 2000