Collaborative Interactive Evolution

Sean R. Szumlanski
University of Central Florida
School of Computer Science
Orlando, FL 32816-2362
seansz@cs.ucf.edu

Annie S. Wu
University of Central Florida
School of Computer Science
Orlando, FL 32816-2362
aswu@cs.ucf.edu

Charles E. Hughes
University of Central Florida
School of Computer Science
Orlando, FL 32816-2362
ceh@cs.ucf.edu

ABSTRACT
This paper examines the efficacy of genetic algorithms (GAs) in combining input from multiple users to control a single interactive system, such as an educational exhibit at a museum. Specifically, the idea of collaborative interactive evolution (that is, interactive evolution with input from multiple users) is introduced for this purpose. Two fitness functions are proposed to guide the collaborative interactive evolution, as well as two non-GA methods for combining user input. The usefulness and success of each of these methods is examined, and the GA is shown to be a viable means for combining user input for the control of a single interactive system.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *heuristic methods*.

General Terms
Human Factors

Keywords
Interactive Evolution, Collaborative Interactive Evolution, Genetic Algorithms, Real World Applications

1. INTRODUCTION
In the typical interactive evolutionary computation (IEC) paradigm, the GA produces candidate solutions at each generation, and a human user must evaluate each individual's fitness subjectively. This is what Takagi refers to as a "narrow definition" of IEC in [1], which also provides an extensive survey of IEC.

This work extends the idea of IEC into *collaborative* interactive evolution. Here, multiple humans provide input to this evolutionary system, as opposed to having just one user. The challenge then becomes that of determining how to combine input from multiple users, especially when those inputs conflict.

This form of collaborative interactive evolution essentially allows users to construct the fitness function for the GA interactively in conjunction with a fitness function template, before evolution begins. Users later engage in post-evolution evaluation of solutions, which gives rise to data that may be used to construct better fitness function templates.

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

2. APPLICATION
In our experimental system, users evolved fictional characters through a variety of parameter settings. In order to give users a motivation for creating the fictional characters, those characters were then brought to life using Haptek's People Putty software, and were used to tell an on-going interactive story whose direction was determined by the type of characters evolved by the users. The experiment unfolded in four stages:

2.1 Character Bids
Every day for six days, a consistent group of ten test users logged into a web-based system and designed fictional characters by choosing from a variety of parameters governing the physical and psychological traits of those characters. When a user designed a character, their input was encoded as a binary string and referred to as a "character bid." The premise behind character bids was that users could control the direction of a story through the careful selection of particular character traits to give cues to the storytellers as to how the story should unfold.

A character bid represents a chromosome in the GA system, in which there are 12 genes – 6 for physical characteristics and 6 for personality traits. Each gene is 3 bits, as there are 8 possible values for each gene. (For example, there were 8 face shapes to choose from, and a character's ego was rated on a sliding scale of 1 through 8, which made for easy binary encoding.)

2.2 Combining Character Bids
At the end of each day, all ten character bids were combined using four methods, two of which were GA-based:

In the first GA-based method, Fitness Function 1 awarded a fitness point to a candidate solution for every time that one of its traits matched a trait in one of the ten character bids. In this way, Fitness Function 1 worked towards finding a solution with the most frequently requested character traits.

The other GA-based method used Fitness Function 2, which awarded fitness points in ten passes – one pass for each character bid. In each pass, the fitness function counted the number of genes in the candidate chromosome that matched genes in the given character bid, and points were awarded as follows, based on the number of matches:

Table 1. Fitness bonuses awarded for the number of genes that a chromosome had in common with a user's character bid.

Number of Gene Matches

1	2	3	4	5	6	7	8	9	10	11	12
9	8	6	4	5	6	7	8	9	10	11	12

Fitness Bonus

The idea of Fitness Function 2 was to give a large fitness bonus for matching many genes in a character bid, but to also try to ensure that at least one gene is matched from each character bid – thus giving a large fitness bonus for matching *only* one gene.

The third method for character bid combination was averaging. Each gene of the character created by this method was the average value of all values requested for that gene among the ten character bids.

Finally, a character was created using frequency computation. Each gene of the character created by this method was simply the most frequently requested value for that gene among all character bids.

2.3 Tournament Pool

On the day following the combination of character bids, the top two individuals produced by each fitness function appeared in a "tournament pool" along with the individuals computed by the average and frequency methods. Users logged into the web-based system and examined the characters in pop-up windows that gave them a list of the characters' personality traits and allowed users to rotate and move 3D models of each of the six characters.

2.4 The Story

From the six characters in the tournament pool, users voted for the one character that they would most like to interact with. A team of creative storytellers then got together to examine the winning character's personality and appearance, and crafted a new scene in the week-long story that accompanied the experiment. The story team recorded voice clips, and on the day following each vote, users logged in and got to listen to the characters speak in real human voices while the animated avatars moved their lips to the words using Haptek's lip-synching animation technology.

3. RESULTS

Various statistics were tracked to determine whether the GA methods of combining user input produced results that could stand up to the methods of averaging and frequency computation.

3.1 Voting

With respect to votes cast in the tournament pool each day, characters produced by Fitness Function 1 were most popular, winning the tournament three out of the five days. A character produced by Fitness Function 2 won once, and a character produced by Frequency Computation won once, as well. To be truly fair, one should also examine exactly how many votes each character received (see Table 2).

Table 2. Number of votes cast for characters developed by each combination method throughout the experiment.

```
GA with Fitness Function 1: 18 votes (36%)
GA with Fitness Function 2: 17 votes (34%)
Avg. and Freq. Methods:     15 votes (30%)
```

3.2 Time Tracking

Another common method for determining how satisfied a user is with the output of an exhibit is to track how long they spend examining that output. As such, the time that users spent looking at each character was tracked throughout the week (see Table 3).

Table 3. Total number of seconds spent viewing Tournament Pool characters developed by each combination method.

```
GA with Fitness Function 1: 2800 seconds
GA with Fitness Function 2: 2318 seconds
Avg. and Freq. Methods:     2387 seconds
```

3.3 Reloading

Another popular method for determining how satisfied a user is with the output of a system is to track how many times the user returns to examine that output. In this system, it was easy to track how many times a user re-opened a pop-up window to examine a character again (see Table 4).

Table 4. Number of times Tournament Pool characters developed by each combination method were reloaded.

```
GA with Fitness Function 1: 94 reloads
GA with Fitness Function 2: 80 reloads
Avg. and Freq. Methods:     87 reloads
```

4. CONCLUSIONS

The most immediate and obvious conclusion is that users favored Fitness Function 1 over Fitness Function 2, and indeed over both the averaging and frequency computation methods combined.

To understand this result, it is important to note that there are certain restrictions that limited the abilities of averaging and frequency computation: averaging always resulted in middle-ground values for character traits, thereby providing little variety in the system. On the other hand, frequency computation had the undesired effect of creating characters that completely neglected most users' character bids whenever two users made similar character bids on a certain day. It might then be reasonable to assume that there is a tradeoff occurring here, where users appreciate the subtle variations provided by Fitness Function 1, which are a result of the random aspects of evolution, while still exhibiting a great degree of similarity with many of the character bids. Often times, characters produced by Fitness Function 2 seemed entirely random.

Overall, the success of Fitness Function 1 in this experiment indicates that GAs may be useful in combining the input of multiple users to control one exhibit. The GA method offers an element of randomness, mutation, and imperfection that may evolve sets of parameters that users find more engaging and interesting than the parameter sets that they themselves created.

5. REFERENCES

[1] Takagi, H. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. In *Proceedings of the IEEE, 89*. 2001, 1275-1296.