Evolution of a Fictional Dialogue

Carina M. Viljoen, Geoff S. Nitschke, Willem S. van Heerden
Computational Intelligence Research Group, Department of Computer Science, University of Pretoria
Pretoria, South Africa
carinamviljoen@gmail.com, *gnitschke@cs.up.ac.za*, *wheerden@cs.up.ac.za*

***Abstract*— This paper describes user-supervised *Evolutionary Algorithm* (EA) experiments that investigate the evolution of a sensible fictional dialogue. A user-supervised EA was used given the difficulty of defining a fitness function for evolving art tasks. Two EAs were tested for the task of evolving dialogue given an English word population. The EAs required user-assigned fitness values to be given as input with varying degrees of frequency during the evolutionary process. The success of the EAs were comparatively evaluated with respect to *two-point* recombination and a novel *complement gene scan* operator. Task performance was evaluated according to average fitness, word and genotype diversity, and the number of words used in the fittest evolved dialogue. Results indicated that for both EAs, complement gene scan was more effective for evolving complex, sensible and grammatically correct dialogue, comparative to sentences evolved by the EAs using two-point recombination.**

***Index Terms*— Evolutionary Algorithm, Aesthetic Selection.**

I. Introduction

This paper investigates the evolution of fictional dialogue[1] using evolutionary algorithms combined with human and computational aesthetic selection (user-supervised and automated fitness functions, respectively). Artificial language evolution using computer simulation [21], [16] is positioned within the larger field of linguistic theory [12]. The general research goal of such studies is to understand and to explain the ability of a speaker to form and understand new sentences, and to reject grammatically incorrect sentences.

There have been many research endeavors that apply bottom-up *Evolutionary Algorithm* (EA) based simulations to model artificial language evolution [2], and more generally to the task of evolving art. Such synthetic simulations allow the study of language and art as a complex, nonlinear, and analytically intractable system [3]. For example, evolutionary-based methods have been applied to evolve artistic forms such as music [23], [17] and images [7]. Bentley and Corne [1], and Romero and Machado [18] describe overviews of various artificial evolution methods used in the field of evolving art.

In evolutionary art simulations, the evolution of a fictional dialogue tantamount to that observed in theater and film, is an especially difficult task. That is, it is problematic to define unsupervised computational aesthetics fitness functions [11], [13] that evaluate subjective character dialogue, and direct the evolution of sensible and grammatically correct dialogue.

This research evolves fictional dialogue via combining aesthetic selection based EAs [9], and *Backus Naur Form* (BNF) rules [14]. Aesthetic selection refers to a user's judgement of how much sense and how grammatically correct a given evolved sentence was at given intervals during an EA process. That is, at given intervals of an EA adaptation process, a fitness value that reflected a user's evaluation was assigned in order to direct sentence evolution.

This research studied two EAs (EA1 and EA2). EA1 required user-assigned fitness at every generation of the EA process. EA2 accepted user-assigned fitness only at every tenth generation of the EA process. At other generations, EA2's fitness function used a grammar checker that automatically assigned a fitness, based on the grammatical evaluation of a sentence's correctness and how much sense a sentence made based on a given rule set. BNF rules were used to map genotypes to sentences at each generation of EA1 and EA2.

Aesthetic selection was chosen to guide EA fitness functions, given the difficulty of designing completely automated computational aesthetics [11], [13] (that is, fitness functions that automatically evaluate an evolved sentence, based on metrics of sense and grammatically correctness). In this study, EA2 was partially guided by an automated grammar checker. Aesthetic selection has been successfully implemented together with evolutionary methods in many evolutionary art experiments [20], [22], [5]. BNF rules were selected as the mechanism to map genotypes to dialogue since previous research has successfully applied BNF to map simple genotypes to relatively complex phenotypes such as programming language source code [19].

This study's goal was to ascertain the most appropriate aesthetic selection based EA for evolving sensible and grammatical correct dialogue in an evolutionary art simulation.

A. Research Goal and Hypotheses

- *Research Goal:* Conduct a study that comparatively evaluates two EAs (EA1, EA2). EA1 and EA2 accepted user-assigned fitness at every generation and every tenth generation, respectively. The goal was to ascertain which EA was more appropriate for evolving sensible and grammatically correct sentences, with respect to *two-point* [9] or the proposed *complement gene scan* recombination (extending *uniform gene scan* [6]).

- *Hypothesis 1:* EA2 will evolve sentences (for both recombination operators) with a statistically significant higher fitness, compared to EA1. This hypothesis was formulated based on related evolutionary art research that combines EAs with aesthetic selection [4]. Heijer

[1] *Dialogue* and *sentences* are used interchangeably throughout the paper.

and Eiben [4] demonstrated that it is often difficult for automated computational aesthetics methods to appropriately judge aesthetic features of evolved art.

- *Hypothesis 2:* The fittest sentences evolved by EA1 and EA2, using *complement gene scan* recombination, will contain a statistically significant higher genotype and word diversity, and more words, compared to EA1 and EA2 using *two-point* recombination. This hypothesis was formulated based on research on the related *uniform scanning* recombination operator [8], and is based on the notion that sentences with more diversity in words are more akin to natural speech. The research of Eiben *et al.* [8] indicated that uniform scanning recombination was effective for searching optimal or near optimal regions of many fitness landscapes. As an extension, complement gene scan recombination was hypothesized to be similarly effective for the task of evolving sensible and grammatical correct fictional dialogue.

B. Evaluation of Evolved Sentences

1) *Average Fitness:* For each EA, fitness was assigned by a user to an evolved sentence at a given generation. For EA2, an automated grammar checker also assigned fitness. Average fitness was calculated for each EA run (*n* generations), and over *N* simulation runs.

2) *Genotype and Word Diversity:* During an EA process, the diversity between genotypes and the words in sentences they represent, was measured. Genotype and word diversity was measured as an average *Euclidean distance* between real value and integer sets representing genotypes and words, respectively. Genotype diversity progression indicated how quickly an EA converged to a specific region of the search space. Word diversity indicated the complexity of evolved sentences. For example, low genotype diversity indicated that an EA was operating in a specific search space region, and was likely to evolve sentences of a limited complexity.

3) *Word Use in Fittest Sentences:* The average number of different words used in the fittest evolved sentence at each generation of an EA, provided a second indication of sentence complexity. During an EA process, word use in the fittest sentences invariably decreased as a result of an EA converging to one region of the search space.

II. Methods

This section describes the EA and BNF approaches that were combined to evolve sentences. In this study, only English grammar and words were used.

A. EA: Evolutionary Algorithm

Two versions of an EA (herein referred to *EA1* and *EA2*) were used in this study. EA1 and EA2 used the same experimental configuration, and differed only in the frequency with which user-assigned fitness evaluations were accepted. EA1 and EA2 accepted user-assigned fitness at every generation and every tenth generation, respectively.

1) Genotypes: All genotypes were of equal length, where each genotype was represented as set of six genes. Each gene was initialized to a random real value in the range $[0, 299]$. Each genotype in the population had a fitness in the range $[0, 10]$, and was assigned an initial fitness value equal to 10. In the case of EA1, genotype fitness was adjusted at every generation by user-assigned fitness (section III-B). In the case of EA2, genotype fitness was adjusted at every generation by a grammar checker, and at every tenth generation by a user-assigned fitness (section III-A.2).

2) Selection Operator: Experiments used *roulette* selection [9]. At each generation, three parent genotypes were selected with a degree of probability proportional to the genotype's fitness. The roulette operator was applied three times in order that three parents were selected. The probability of selection was 1.0 for a fitness of 10, and 0.0 for a fitness equal to 0. After genotypes were selected for recombination, three parent recombination was applied using either *two-point* [9], or *complement gene scan* recombination (section II-A.3).

3) Recombination Operators: The two-point and complement gene scan operators produced one and three child genotypes, respectively. An operator was applied until enough children were produced to completely replace the parent population. If a recombination operator was not applied, then recombination did not occur at the given generation, and the mutation operator was applied to all genotypes.

Two-Point: This operator randomly selected two points in three parent genotypes. Each of the three gene segments were swapped between two (randomly selected) of the three parent genotypes. Each gene segment was swapped between two selected parents with a 0.8 probability. For example, if three genotypes *a*, *b*, and *c* were selected for recombination, then two points would be randomly selected in *a*, *b*, and *c*, dividing each genotype into three gene segments A_0, B_1, C_2 for *a*, D_0, E_1, F_2 for *b*, and G_0, H_1, I_2 for *c*. Assume that *a* was selected to potentially have its gene segments swapped. If *a* was selected to have its first gene segment swapped with that of *b*, then gene segment A_0 would be swapped with D_0 with 0.8 probability. Second, if *c* was selected to have its second gene segment swapped with *a*, then B_1 would be swapped with H_1 with a 0.8 probability. Finally, if *b* was selected to have its third gene segment swapped with *a*, then C_2 would be swapped with F_2 with 0.8 probability.

Complement Gene Scan: This operator is a novel extension of *uniform scanning* recombination [8]. Each generation, the complement gene scan operator was applied with a 0.8 probability to produce three child genotypes from three parents. A marker of a value in the range $[1, 3]$ (randomly selected) was assigned to each gene in a child genotype with (initially) no gene values. Each marker value indicated which parent genotype (1, 2, or 3) would pass the value of its corresponding gene to a given child genotype. After markers had been assigned to each gene in a child genotype, the child's genes were given values from at least one of the parent genotypes.

To illustrate complement gene scan, consider the following example. Assuming three child genotypes (*A*, *B* and *C*) each comprising three genes, the operator initializes each genotype with a *null* value. Each gene in each child genotype is then randomly initialized with a marker that refers to the corresponding gene in a given parent genotype. Consider the following gene marker assignment to child genotype A (specifying parent gene values to be inherited):

$$A = \{[2], [1], [3]\} \tag{1}$$

That is, the values [2, 1, 3] are markers that refer to the corresponding gene values in parent genotypes 2, 1, and 3, respectively. The *complement* child genotypes are initialized with marker values via adding a value of one to the gene marker value in the previous complement child genotype. Child genotypes B and C are thus initialized as follows:

$$B = \{[3], [2], [4]\} \tag{2}$$

$$C = \{[4], [3], [5]\} \tag{3}$$

To encourage complementary inheritance of gene values from parent genotypes, a MOD operator is used for gene marker values greater than three. The three example child genotypes A, B, and C are now:

$$A = \{[2], [1], [3]\} \tag{4}$$

$$B = \{[3], [2], [1]\} \tag{5}$$

$$C = \{[1], [3], [2]\} \tag{6}$$

One of these three child genotypes (A, B, or C) is then randomly selected to be propagated into the next generation.

4) Mutation operator: After recombination, the mutation operator was applied to each gene in each genotype with a 0.05 degree of probability. Mutation was implemented by adding a random real value to a gene in the range $[0, 9]$.

B. Genotype to Sentence Mapping: Backus Naur Form (BNF)

The *Natural Language Processor* (NLP)[2] was used to load text files containing dialogue into a vocabulary used by the EAs. When the dialogue was read in by the NLP, a hash map was constructed. Each key in the hash map was a *Part Of Speech* (POS) descriptive tag that was assigned to sets of words. This hash map represented the vocabulary used by an EA. In constructing the vocabulary, a *Penn Treebank* POS tagger [15] was used to assign descriptive tags to words in the input dialogue. These tags indicated the word type based upon the part of speech that each word corresponded to, and the relationship of each word to adjacent words in a sentence. For terminal word sets in the dialogue, the POS tagger selected either *pre-modifier*, *determiner* or *head* as the most appropriate tag. For non-terminal word sets the POS tagger selected either *verb phrase* or *noun phrase* as the most appropriate tag. At each EA generation, when a genotype was mapped to a sentence, the following parse tree (in the form of a BNF rule set) was applied to the vocabulary. BNF is a notation that expresses the grammar of a language in terms of production rules [14]. BNF was found to be effective for performing genotype to sentence mappings in this study. The following parse tree was used in all experiments.

⟨sentence⟩→ ⟨clause⟩ ⟨punctuation⟩
⟨clause⟩→ ⟨noun phrase⟩ ⟨verb phrase⟩ |
⟨noun phrase⟩ ⟨verb phrase⟩ ⟨noun phrase⟩ |
⟨interjection⟩
⟨noun phrase⟩→ ⟨determiner⟩ ⟨premodifier⟩ ⟨head⟩
| ⟨determiner⟩ ⟨head⟩
⟨verb phrase⟩→ ⟨modal⟩ ⟨base verb⟩ | ⟨verb⟩
⟨participle⟩
⟨determiner⟩→ ⟨article⟩ | ⟨pronoun⟩
⟨premodifier⟩→ ⟨participle⟩ ⟨adjective⟩ ⟨noun⟩ |
⟨noun⟩|
⟨adjective⟩ ⟨noun⟩ | ⟨participle⟩ ⟨noun⟩
⟨head⟩→ ⟨noun⟩ | ⟨adjective⟩ ⟨noun⟩

A genotype to sentence mapping then worked as follows.

1) For each gene in a given genotype, the gene value *modulo* the number of mapping choices (defined by the BNF grammar) was calculated. The gene value was the dividend, and the number of choices was the divisor.

2) The result of applying the *modulo* operator (producing the remainder after division) indicated the phrase, word or punctuation mapped from the given gene value.

After a genotype was mapped to a word set, the word set was assembled, according to English grammar rules, into a sentence. Such sentences were then evaluated by the user (user fitness assignments only occurred at specific generation intervals), or by an automated grammar checker (section III).

BNF rules consisted of a set of terminals and non-terminals. The terminals were the *subject*, *verb*, *object*, *adverb* and *punctuation* in a sentence. Nonterminals were described by expressions, where an expression was comprised of sets of possible nonterminals and terminals. Each nonterminal was substituted with a word or form of punctuation (selected from the vocabulary). To demonstrate the genotype to sentence mapping process, consider the simplified example BNF rule *A*:

⟨sentence⟩→ ⟨subject⟩ ⟨predicate⟩ ⟨punctuation⟩
⟨predicate⟩→ ⟨verb⟩ | ⟨verb⟩ ⟨object⟩ | ⟨verb⟩
⟨adverb⟩
⟨subject⟩→ *Sally* | *I* | *Her*
⟨verb⟩→ *did* | *sings* | *dance* | *ate*
⟨object⟩→ *a box* | *the night*
⟨adverb⟩→ *well* | *quietly*
⟨punctuation⟩→ *!* | .

[2] The Natural Language Processor framework OpenNLP v1.4.3 is an open source project available at: http://sourceforge.net/projects/opennlp

In each genotype, gene values control genotype to sentence mapping. Consider the following example genotype *D*:

$$D = \{213, 7, 45, 11, 2\} \tag{7}$$

Given the BNF rule *A*, and the genotype *D*, then *D* would be mapped to its sentence as follows:

1) The first gene of *D* is 213. Since, in the BNF rule *A*, there are no alternate choices for the structure of the nonterminal *sentence*, the first choice is made for the nonterminal *subject*, which can be one of three possible terminals. The subject *Sally* is selected, since 213 *modulo* 3 is 0. In this example, 0 denotes that the first choice of *subject* be selected.

2) Next, a choice is made for the *predicate* nonterminal in the sentence. The *predicate* nonterminal is mapped to one of three nonterminal possibilities. The next gene is 7, and 7 *modulo* 3 is 1, so the predicate construct with a *verb* and *object* is selected.

3) The next gene in *D* is 45, and 45 *modulo* 4 is 1, meaning the verb *sings* is selected.

4) The next gene in *D* is 11, and 11 *modulo* 2 is 1, meaning the object *the night* is selected.

5) Finally, a choice is made for the punctuation nonterminal in the sentence. The next gene in *D* is 2, and 2 *modulo* 2 is 0, meaning the terminal punctuation *!* is selected.

Hence, in this example, the genotype *D* is mapped to the sentence *Sally sings the night!*.

III. Evaluation Metrics

This section describes the evaluation metrics used by the EAs. Namely, the *fitness function*, the *word* and *genotype diversity*, and a measure of *word use* in an evolved dialogue.

A. Fitness Functions

EA1 implemented a user-supervised fitness function, and EA2 combined user assigned fitness with fitness automatically assigned by a grammar checker.

1) User-Assigned Fitness: During either EA process, fitness was assigned by the user at a given generation interval, or by an automated grammar checker at every generation. The fitness value assigned was based on a user's evaluation of how grammatically correct and sensible an evolved sentence was at generation *i* of an EA process. User-assigned fitness values were always in the range $[0, 10]$. In general, an evolved sentence that contained many grammatical errors and was nonsense was assigned a fitness value of 0. Evolved sentences that contained no grammatical errors and made sense, were assigned a fitness value of 10. Since each genotype in an EA population had an initial fitness value of 10, user-assigned fitness *f* was calculated as a penalty $-(10 - f)$ applied to a given genotype's fitness value.

2) Grammar Checker: The grammar checker complemented user-assigned fitness for EA2. The grammar checker *JLanguageTool*[3], was integrated into the EA2 fitness function to automatically evaluated evolved sentences. The JLanguageTool used 475 English language grammar rules that identified formatting and punctuation errors, including common mistakes such as confusing the use of *you're* and *your*. At each EA2 generation, sentences with grammatical errors detected by JLanguageTool, applied a fitness penalty of $-x$, where *x* denoted the number of detected grammatical errors.

B. Average Word and Genotype Diversity

The diversity between words in sentences (mapped from genotypes), and between individual genotypes was measured over the course of the *n* generations of a given EA.

The diversity of words was calculated in terms of the average *Euclidean distance* between words that comprised the fittest evolved sentence (at generation *i*). In order to measure word diversity, words used in evolved sentences had their constituent characters converted to ASCII values. That is, characters in the range [a, z] were converted to their corresponding ASCII values in the range [97, 122] so as the Euclidean distance metric (equation 8) could be applied. An average Euclidean distance was calculated over all distances measured between all possible pairs of words in the fittest evolved sentence. A Euclidean distance metric was applied as a simple means of ascertaining the average difference between the characters that comprised words used in evolved sentences. Given an average Euclidean distance calculated between the words comprising two fittest evolved sentences, the relative average diversity of words used in each could be judged.

Average genotype diversity was similarly calculated via applying the Euclidean distance metric to all possible pairs of real value sets representing genotypes in an EA population, and then calculating an average Euclidean distance. The distance between two individuals (words or genotypes), x_1 and x_2, of length m was calculated as:

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{m} (x_{1i} - x_{2i})^2} \tag{8}$$

C. Word Use in Evolved Dialogue

Each EA used a vocabulary of 500 English words and forms of punctuation (table I). At each generation, genotypes were mapped to sets of words that were then assembled as sentences. During an EA process, as a user or grammar checker (or both) assigned fitness to genotypes, words in sentences mapped from the fittest genotypes became the most frequently used words. Similarly, words in the sentences mapped from less fit genotypes became less frequently used during an EA process. At each generation, *N* sentences were mapped from a population of *N* genotypes, and assigned a fitness. Word frequency for the words that appeared in the fittest sentence was then incremented.

[3] JLanguageTool v1.0.0 can be found at: www.languagetool.org/

TABLE I
EA1 / EA2 AND SIMULATION PARAMETERS.

EA1 / EA2 and Simulation Parameters	
Population size	15
Number of EA generations	50
Mutation operator	Add integer in range $[0, 9]$
Mutation rate per gene (σ)	0.05
Selection operator	Roulette
Gene value initialization	Random integer in range $[0, 299]$
Recombination operator	Two-point / Complement gene scan
Parents per recombination	3
Genotype	String: 18 Integers
Genotype length	40
Grammar checker	JLanguageTool v1.0.0
Fitness function	User / grammar checker assigned
Fitness range	$[0, 10]$
User-assigned fitness	Every 1 (EA1) / 10 (EA2) generations
Simulation runs per experiment	20
Words in initial vocabulary	500

IV. EXPERIMENTS

Each experiment applied EA1 or EA2 with a population of 15 genotypes, using either *two-point* or *complement gene scan* recombination. An experiment was executed for 50 generations, and 20 simulation runs. For each EA, average values for fitness, genotype and word diversity, and the number of words used in evolved dialogue (section III) were calculated at the end of the EA process, and over 20 simulation runs of a given experiment. When the average word diversity and number of words were calculated, only the fittest evolved sentence (at the final generation of a given EA's simulation run) was used. At each generation of EA1, the user was presented with 15 sentences, where one sentence was mapped from each genotype in the population. The user then assigned each sentence a fitness rating in the range [0, 10]. EA2 used the same procedure for fitness assignment, except that fitness was assigned once every 10 generations. Also, at every generation a grammar checker automatically assigned a fitness value. Table I presents the parameter values used in this study. These parameter values were derived in exploratory experiments that determined the parameter values were near optimal.

An EA's *task performance* was average fitness, genotype and word diversity, and the number of words in evolved sentences. Experiments compared EA1 and EA2 with respect to two-point and complement gene scan recombination. Seven users[4] participated in running experiments using EA1 or EA2.

Experimental Objective: The objective was to ascertain if either EA (with two-point or complement gene scan recombination) maximized any task performance measure.

Average Fitness: This task performance measure tested hypothesis 1 (section I-A). That is, that EA2 will evolve dialogue (for both recombination operators) that yields a statistically significant higher fitness, comparative to EA1.

Average Genotype and Word Diversity: Diversity in the fittest evolved sentences partially tested hypothesis 2 (section I-A). That is, the EAs using complement gene scan recombination, will evolve dialogue with a statistically significant higher average word and genotype diversity, compared to the EAs using two-point recombination. Hypothesis 2 is based on related research on *uniform scanning* recombination [8], and the notion that sentences with more diversity in words are more akin to natural speech.

Average Number of Words in Fittest Dialogue: This measure partially tested hypothesis 2 (section I-A). That is, the fittest dialogue evolved by the EAs using complement gene scan recombination will contain a statistically significant greater number of words, compared to two-point recombination.

V. RESULTS

This section presents task performance results for dialogue evolved by EA1 and EA2 using each recombination operator. To gauge comparative task performance results, statistical tests were applied to EA1 and EA2 result data, for each recombination operator. The Kolmogorov-Smirnov test [10] confirmed that EA1 and EA2 result data conformed to normal distributions. In order to determine if there was a statistically significant difference between the task performances of EA1 and EA2, an independent t test [10] was applied. A statistical significance of 0.05 was selected, and the null hypothesis stated as the data sets not differing significantly.

A. Average Fitness

The average fitness and standard deviation (in parentheses) of EA1 was 1.32 (0.65), and 5.16 (1.29), using two-point and complement gene scan recombination, respectively. The average fitness and standard deviation of EA2 was 8.10 (1.30), and 8.70 (1.05) using two-point and complement gene scan recombination, respectively.

Statistical t tests indicated that, for both recombination operators, EA2 yields a significantly higher average fitness, comparative to EA1. These results indicate that when the user-assigned fitness at every generation of an evolutionary process (as in the case of EA1), evolved sentences will have a lower fitness, on average. This was the case since users often judged evolved sentences as not making sense and containing many grammatical errors. However, when a user-assigned fitness at every tenth generation (as in the case of EA2), then user input to direct sentence evolution was comparatively infrequent. At other generations, grammatical errors and nonsensical constructs were evaluated by the grammar checker. Given that not all mistakes were detected by the grammar checker, this resulted in sentences evolved by EA2 having a higher average fitness, compared to EA1. This result supports hypothesis 1 (section I-A). That is, that EA2 will evolve sentences, for both recombination operators, that yield a statistically significant higher fitness, compared to EA1.

[4] Thanks are extended to the following people for their assistance in running experiments: Nikitah Bobhate, Gordon Wells, Waldo Delport and Ivan Sharpe.

B. Average Genotype Difference

Figure 1 (sub-figure *a* and *b*) presents the average Euclidean distance calculated between genotypes in EA populations, using *two-point* or *complement gene scan* recombination, respectively. The average genotype distance range presented in figure 1 has been normalized in the range [0.0, 1.0], where 0.0 indicates no difference between genotypes and 1.0 indicates the maximum difference between genotypes.

The average Euclidean distance (standard deviation given in parentheses) between genotypes in the EA1 and EA2 populations, using two-point recombination, was 0.47 (0.24) and 0.47 (0.27), respectively. The average Euclidean distance between genotypes in the EA1 and EA2 populations, using complement gene scan recombination, was 0.48 (0.23) and 0.44 (0.30), respectively. A statistical t test comparison indicated that there was no statistically significant difference between EA1 and EA2 using either two-point or complement gene scan recombination.

This result partially refutes hypothesis 2 (section I-A). That is, both EA1 and EA2, using complement gene scan recombination, will evolve sentences that contain more words and yield a statistically significant higher average word and genotype diversity, compared to two-point recombination.

C. Average Word Difference

Figure 2 (sub-figure *a* and *b*) presents the average Euclidean distance between words used in the fittest evolved sentences (at each generation), for EAs using *two-point* or *complement gene scan* recombination respectively. The average word distance was normalized in the range $[0.0, 1.0]$. A value of 0.0 indicated no difference between words in an evolved sentence. That is, where only one word was used. A value of 1.0 indicated the maximum possible difference between words used. That is, where very different types of words are used.

The average word difference (standard deviations are given in parentheses) in sentences evolved by EA1 and EA2, using complement gene scan recombination, was 0.45 (0.01) and 0.47 (0.01), respectively. The average word distance in sentences evolved by EA1 and EA2, using two-point recombination, was 0.37 (0.02) and 0.35 (0.04), respectively.

A statistical t test comparison of the average word distance in the fittest sentences evolved by EA1 and EA2, indicated that complement gene scan recombination resulted in a statistically significant higher word diversity, compared to the fittest sentences evolved using two-point recombination. This result partially supports hypothesis 2 (section I-A). That is, that either EA, using complement gene scan recombination will evolve sentences that contain more words and a higher word and genotype diversity, compared to sentences evolved by either EA using two-point recombination.

D. Average Number of Words Used in Evolved Sentences

Figure 3 (sub-figure *a* and *b*) presents the average number of words used in the fittest sentences evolved by EA1 or EA2 using *two-point* or *complement gene scan* recombination, respectively. A statistical comparison of the average number of words used in the fittest sentences (at the final generation of an EA process) evolved by EA1 and EA2, indicated that complement gene scan recombination resulted in sentences containing a statistically significant higher number of words, compared to sentences evolved using two-point recombination. The average number of words (standard deviation given in parentheses) in sentences evolved EA1 and EA2, using two-point recombination, was 64.06 (8.66) and 63.04 (8.75), respectively. The average number of words in sentences evolved by EA1 and EA2, using complement gene scan recombination, was 78.82 (5.12) and 80.50 (5.52), respectively. This result partially supports hypothesis 2 (section I-A). That is, both EA1 and EA2, using complement gene scan recombination, will evolve sentences that contain more words and yield a statistically significant higher average word and genotype diversity, compared to two-point recombination.

The results for average word diversity, and the number of words used in evolved sentences demonstrate that complement gene scan recombination is more effective, compared to two-point recombination. That is, complement gene scan recombination used with EA1 or EA2, facilitates the evolution of sentences that contain a greater number of words and greater diversity in words. However, there was no statistically significant difference in the average distance between genotypes in EA1 and EA2 using either recombination operator. Thus, even though both EA1 and EA2, using either recombination operator resulted in decreasing genotype diversity (figure 1), complement gene scan recombination facilitated the evolution of sentences containing more words and greater word diversity.

The supposition supporting this result is that the markers used by the complement gene scan operator allowed individual genes from multiple parents to be combined and propagated to child genotypes. This mechanism was appropriate for either EA1 or EA2 to evolve sentences that contained more words and greater word diversity, compared to sentences evolved using two-point recombination. However, this hypothesis is currently the subject of ongoing research.

The statistically higher average fitness of EA2, compared to EA1, is attributed to the stringent evaluation of user-assigned fitness at each generation of EA1. The automated grammar checker applied at every generation (of EA2) was not able to as effectively evaluate how much sense a sentence made, and was not as effective at detecting grammatical errors as a human evaluator. Furthermore, the fitness assigned by a user at every tenth generation of EA2 had little impact upon directing sentence evolution. That is, the fittest sentences evolved by EA2 were regarded by human users as conveying little sense, as well as containing grammatical errors.

E. Evolved Sentences

This section presents samples of the fittest sentences (dialogue) evolved for two characters: Luanne and Mark. Each sample was the fittest selected from one the 20 runs (50 generations each) of EA1 and EA2, that applied either *two-point* or *complement gene scan* recombination (section IV).

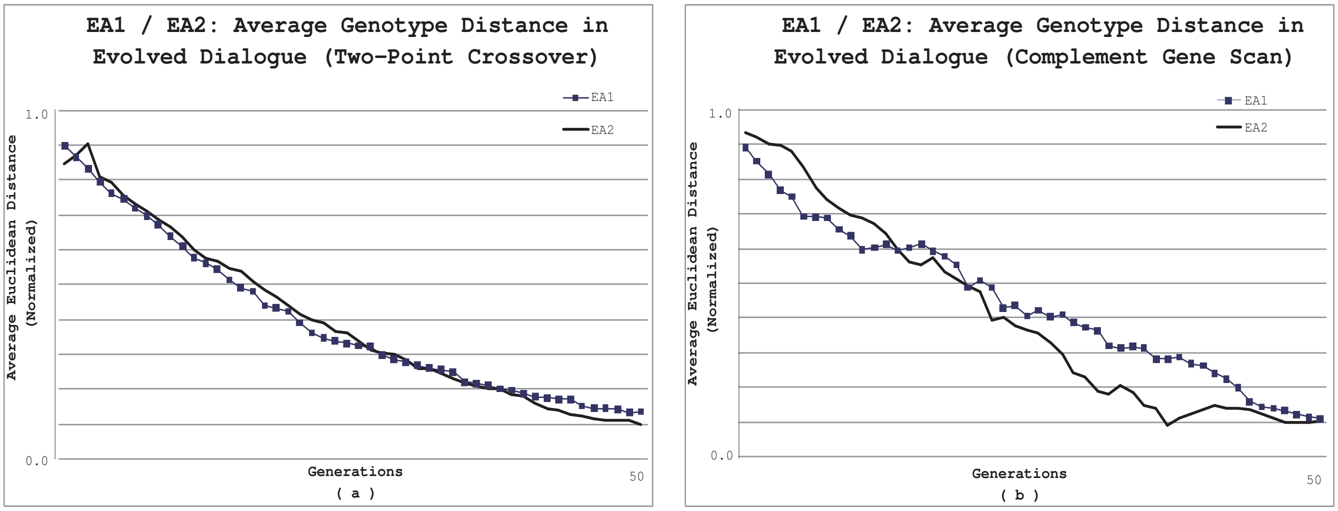

Fig. 1. Average normalized Euclidean distance between genotypes when EAs used *two-point* (sub-figure *a*) and *complement gene scan* (sub-figure *b*) recombination. Sub-figures *a* and *b* are presented on the left and right hand sides, respectively.

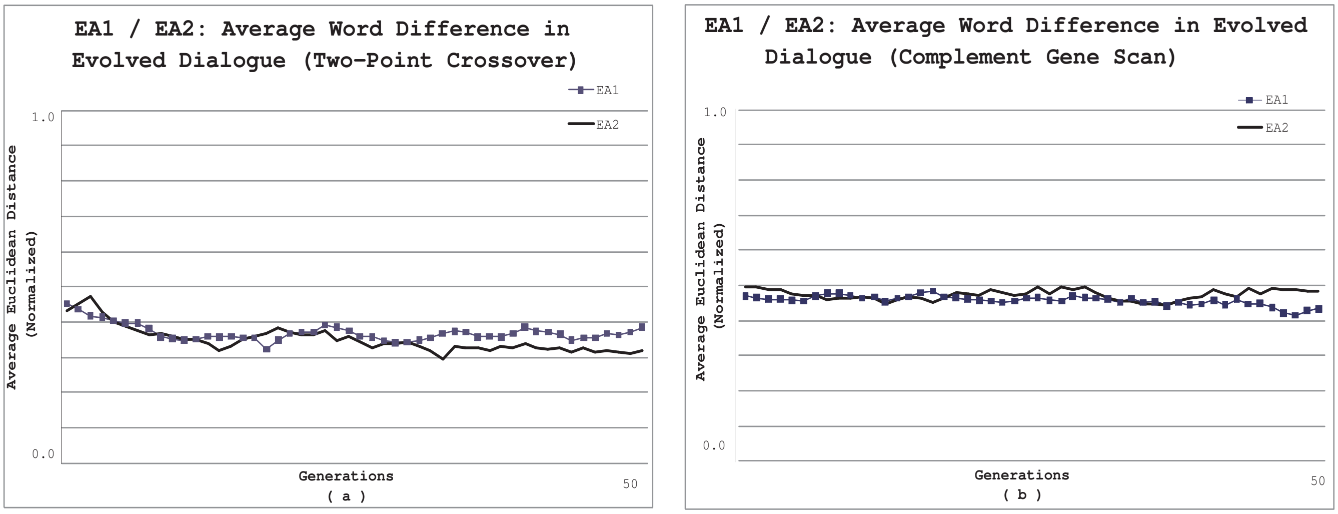

Fig. 2. Average normalized Euclidean word distance in fittest sentence (at each generation) with *two-point* (sub-figure *a*) and *complement gene scan* (sub-figure *b*) recombination. Sub-figures *a* and *b* are presented on the left and right hand sides, respectively.

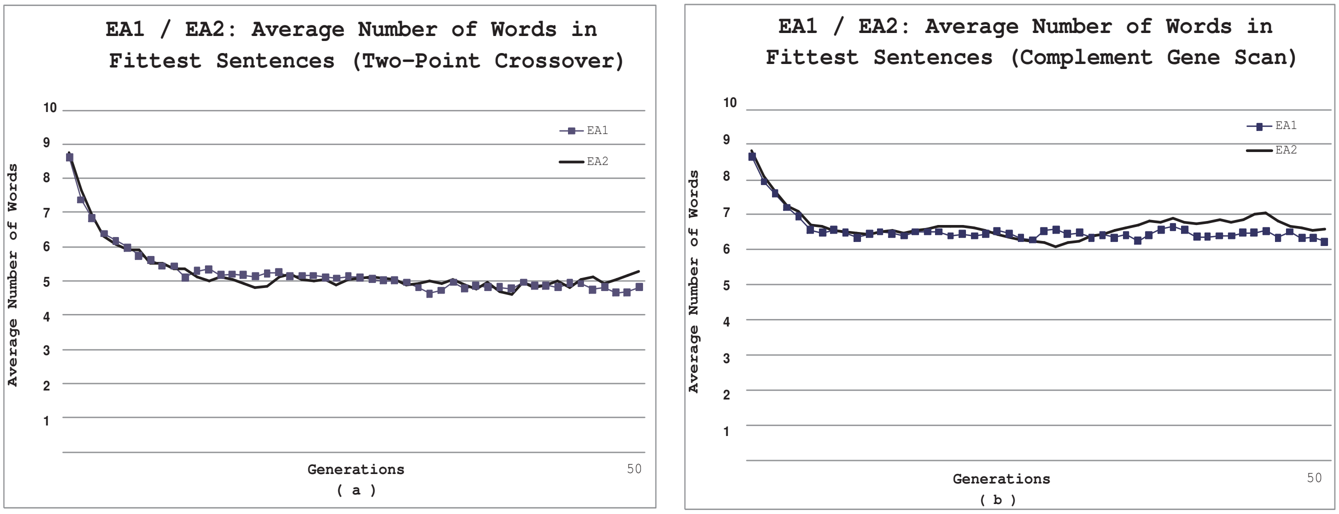

Fig. 3. Average number of words used in fittest evolved sentence (at each generation) with *two-point* (sub-figure *a*) and *complement gene scan* (sub-figure *b*) recombination. Sub-figures *a* and *b* are presented on the left and right hand sides, respectively.

1) EA1: Two-point recombination:

Luanne: Little Mark will look only!
Mark: Quiet Mark might look not.
Luanne: Heavier looking pie Luanne will eat again.
Mark: Your thinking slow Luanne would stay not?
Luanne: I little Mark will live alright!
Mark: My cousin can love back!

2) EA1: Complement gene scan recombination:

Luanne: Another great school Maye can go always!
Mark: My lot might go so...?
Luanne: My application won't go too!
Mark: All old school Mom got thinking.
Luanne: Another great looking pie Maye can eat lately?
Mark: A lot might stay back.

3) EA2: Two-point recombination:

Luanne: Another old picture little change was thinking?
Mark: I old Luanne will start later!
Luanne: Me thin cousin quiet day might happen so...!
Mark: His slow ideas did getting?
Luanne: My thin cousin whole look will find alright!
Mark: My old months made feeling.

4) EA2: Complement gene scan recombination:

Luanne: That heart might do really!
Mark: A quiet good night will happen always.
Luanne: That scholarship might do later!
Mark: Some quiet day was done?
Luanne: Your man was biding.
Mark: Some day old Ohio was coming.

VI. Conclusions

This research compared the efficacy of two *Evolutionary Algorithms* (EA1 and EA2) for the task of evolving grammatically correct and sensible fictional dialogue from an initial English vocabulary. Both EA1 and EA2 used fitness functions that worked with user-supervised fitness. EA1 and EA2 accepted user-assigned fitness at every and every tenth generation, respectively. For EA2, an automated grammar checker also modified fitness at every generation. Fitness was assigned based on how sensible and grammatically correct a user judged an evolved sentence to be. For both EA1 and EA2, experiments compared the effect of using two-point and complement gene scan recombination, upon the evolution of dialogue. For both EA1 and EA2, using complement gene scan recombination, the fittest evolved sentences contained more words, and a greater diversity in words, compared to sentences evolved using two-point recombination. However, there was no significant difference in the average genotype diversity in both EA1 or EA2 population when complement gene scan or two-point recombination was used. That is, decreasing diversity in an EA genotype population did not impact upon the capability of complement gene scan recombination to evolve more complex sentences, compared to sentences evolved by the EAs using two-point recombination.

Future work will investigate the mechanisms responsible for the evolution of greater sentence complexity by EAs using complement gene scan recombination. Furthermore, the complement gene scan operator will be tested in comparison to additional recombination operators using varying numbers of parent genotypes, for the task of evolving sensible and grammatically correct fictional dialogue.

References

[1] P. Bentley and D. Corne. *Creative Evolutionary Systems*. Morgan Kaufmann, San Mateo, USA, 2001.

[2] A. Cangelosi and D. Parisi. Computer simulation: A new scientific approach to the study of language evolution. In *Simulating the Evolution of Language*, pages 3–21. Springer, Berlin, Germany, 2002.

[3] E. de Jong. Analyzing the evolution of communication from a dynamical systems perspective. In *Proceedings of the 5th European Conference on Artificial Life*, pages 689–693, Lausanne, Switzerland, 1999. Springer.

[4] E. den Heijer and A. Eiben. Comparing aesthetic measures for evolutionary art. In *Applications of Evolutionary Computation Lecture Notes in Computer Science*, pages 311–320. Springer, Berlin, Germany, 2010.

[5] E. den Heijer and A. Eiben. Using aesthetic measures to evolve art. In *Proceedings of the IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, 2010. IEEE Press.

[6] A. Eiben. Multi-parent recombination in evolutionary computing. In *Advances in Evolutionary Computing, Natural Computing Series*, pages 175–192. Springer, Berlin, Germany, 2002.

[7] A. Eiben. Evolutionary reproduction of Dutch masters: The Mondriaan and Escher evolvers. In *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music, Natural Computing Series*, pages 211–224. Springer, Berlin, Germany, 2008.

[8] A. Eiben, P. Raue, and Z. Ruttkay. Genetic algorithms with multi-parent recombination. In *Proceedings of Parallel Problem Solving from Nature*, pages 78–87, Jerusalem, Israel, 1994. Springer.

[9] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, Germany, 2003.

[10] B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes*. Cambridge University Press, Cambridge, UK, 1986.

[11] G. Greenfield. On the origins of the term computational aesthetics. In *Proceedings of Computational Aesthetics 2005: Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 9–12, Girona, Spain, 2005. IEEE Press.

[12] M. Hauser, N. Chomsky, and W. Fitch. Three models for the description of language. *Transactions on Information Theory*, 1(1):113–124, 1956.

[13] F. Hoenig. Defining computational aesthetics. In *Proceedings of Computational Aesthetics 2005: Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 13–18, Girona, Spain, 2005. IEEE Press.

[14] D. Knuth. Backus Normal Form vs. Backus Naur Form. *Communications of the ACM*, 7(1):735–736, 1964.

[15] M. Marcus, M. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: the penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[16] M. Nowak, N. Komarova, and P. Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(1):611–617, 2002.

[17] T. Oliwa. Genetic algorithms and the ABC music notation language for rock music composition. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1603–1610, Atlanta, USA, 2008. ACM.

[18] J. Romero and P. Machado. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music. Natural Computing Series*. Springer, Berlin, Germany, 2007.

[19] C. Ryan, J. Collins, and M. O'Neil. Grammatical evolution: Evolving programs for an arbitrary language. In *Lecture Notes in Computer Science 1391, Proceedings of the First European Workshop on Genetic Programming*, pages 83–95, Paris, France, 1998. Springer-Verlag.

[20] K. Sims. Artificial evolution for computer graphics. In *Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, pages 319–328. ACM Press, 1991.

[21] L. Steels. The synthetic modeling of language origins. *Evolution of Communication*, 1(1):1–34, 1997.

[22] D. Thomas. Aesthetic selection of developmental art forms. In *Proceedings of Artificial Life VIII*, pages 157–163, Sydney, Australia, 2002. MIT Press.

[23] M. Unehara and T. Onisawa. Construction of music composition with an interactive genetic algorithm. In *Proceedings of the International Conference on Asian Design*, pages 84–89, Tsukuba, Japan, 2003. ACM.