A Computational Model of Poetic Creativity with Neural Network as Measure of Adaptive Fitness

Robert P. Levy

Oswego State University
Oswego, New York 13126
levy@oswego.edu

Abstract

The prevailing metaphor used in the study of creative cognition is the metaphor of evolution (Gruber and Davis 1988). Naturally, AI experimenters have implemented algorithms based loosely or precisely on Darwinian models of information transfer and change over time in attempts to make machines more creative (Fogel and Owens 1966). Simulating an evolving systems model of creative expression raises the important technical issue of how to judge the fitness of the forms the system produces. The use of a neural network is an appropriate and fairly novel (see also Biles 1996 and Burton and Vladimirova, 1997) answer to the question of how a fuzzy and subjective measure can be captured and put to use. This technical paper proposes a plan for the design of a creative computational system of this kind, and documents a working prototype called Poevolve that embodies some of these features.

Introduction

Creativity is an important and ubiquitous fact of human nature. It is of fundamental interest to cognitive science and artificial intelligence. Better models of creativity are needed both to make machines more intelligent and to better understand our own highly fluid cognitive processes.

Data on creativity come in from a number of sources. Anthropologists and ecologists look at the interplay between genetically rooted processes, imitation, and creativity as sources of existing and changing ethnopsychological processes. Social psychologists study the relationship between social interactions and creativity. Computer scientists and architects of computational proto-minds construct models of scientific problem-solving and artistic generativity to test models (as in computational cognitive psychology) and to make practical applications (as in artificial intelligence). Cognitive psychologists and linguists study the specific details of how simple processes coalesce into more surprising and unpredictable ways of thinking. They study the simple machines of creativity extensively: metaphor, blending, and combination of concepts are examples of some significantly useful devices (Finke, Ward, and Smith 1992). Although each of these backgrounds to the study of creativity has its characteristic methods and tools, the fact that they must rely on cross-disciplinary findings inextricably binds them up together as a cognitive science of creativity.

Controversies over the meaning of creativity fortunately do now meet on common terms, making meaningful dialog possible. The concensus seems to be that a creative process is any process that results in something new and valued (Csikszentmihalyi 1999). Creativity seems to often be a process of finding and optimal solution, something that mainly occurs in the cognitive unconscious in a flurry of massively parallel activity. Often when a person experiences a desire to produce creative output, it is immediately followed by a thought or action expressing the results of a creative process. The intuitive nature of creativity gives it a mysterious allure. The tendency of many to use the term "creativity" in an undisciplined and often superstitious way may cause some to adopt the term "generativity" in lieu of "creativity". The problem with this term is that it only denotes one part of the process, the production of something. A calculator generates the products of numbers but is not creative because it does not reflect on the interesting properties of what it is doing. Consequently it is not capable of creating original and distinctive artifacts (discovery of new mathematical axioms for example).

If it is an accident that the accepted definition of creativity is highly compatible with the Darwinian algorithm of evolution, it is truly a serendipitous accident. When we view the creative process as centering around the goal of creating something "good" (preferably good in many different ways) and also "new", it is inevitable to adopt something like evolution. In a given ecology, the animals that may safely go about reproducing with one another are the most prolific, and consequently their strands of DNA are the more prevalent variety in circulation. Just as the next population of animals, despite all of their individual differences, will reflect adaptive saving traits, in a space of concepts and objects, the future inhabitants of that space are bound to be those who can make it past the highly picky checks for qualities defined within a context the creator defines and works within. In much the same way that a planetary system metaphor gave way to a precisely accurate conceptualization of the atom, the evolution model is not the final word on creativity. To contribute to the process of developing a better fit model we first take creativity out of the box of abstract processing and discover it where it occurs naturally: in the embodied mind/brain during specific kinds of behavior. The behavior we explore here is the writing of poetry. On this walk through what a poet does, we take ideas from the evolution model of creativity and bring them to bear on the details of the situation. The section entitled Building a Computer Poet introduces a workable plan for a program that incorporates many of the elements of actual poets' creative processes. Next, the section called Poevolve User's Guide documents the latest prototype of the creative system introduced here.

Creativity and the Process of Writing a Poem

Imagine yourself sitting down at a table or keyboard to work on a poem. You could start in a number of ways. There might be a concept you have in mind you would like to express in poetry, or perhaps you have a line of poetry already in mind, fully written. Maybe you are specifically interested in writing something funny. There could be a combination of words you have been playing with for quite some time. Now if it is possible for you to imagine that you have suddenly been afflicted with "writer's block", then do so at this time (this means you have nothing to work with, nothing at all poetic is happening). Where do we start the project of writing this poem? The obvious answer is to start with the most methodical and formulaic side of poetry, the application of a poetic form. A poetic form defines the shape of the poem in terms of sonic characteristics and can define semantic constraints as well. You can create your own poetic form out basic prosodic, rhyming ideas, and so on, as the free verse poets do, or you can choose from any number of poetic forms (which is probably what you'll need to do since you have "writer's block" right now). Choosing the form is not necessarily the first thing you would choose to do, but you don't seem to have much choice given your current situation. Books on poetics list hundreds of forms, so you now look up a form and go with it.

For no special reason you are aware of, you opt to use the limerick form. The limerick is a form in which a humorous story with a surprise ending is bottled up into five lines. The third and the fourth lines rhyme and share a fixed rhythm ([./../] as in "he felt a great drop"). The first, second, and fifth share a rhyme and different fixed rhythm ([./../../] as in "there once was a cat from Siam"). The last line is the suggested punch line, because the sonic configuration of two short lines before a long line affords the effect of surprise and release (followed by laughter on some occasions). With all these new constraints, the job is much easier and you are now closer to getting past your lack of ideas. A line might come to mind like "There once was a parrot named Sam," which seems like a good first line since you are telling a story. What just happened? Somehow without conscious awareness of the process, you produced a viable line of poetry. A number of things probably occurred: 1. a shuffling of words, 2. experimentation with combinations of words, 3. experimentation with drafts of lines, 4. experimentation with drafts of concepts for the line, 5. selection of lines or word combinations on the basis of matching the rhythmic template, 6. selection on the basis of concepts, 7. on the basis of the line's ability to initiate a story, 8. on the basis of syntactic clarity. These are at least a few of the things that are occurring under the scenes.

It is important that we answer the question of what might be occurring at the neuronal level to make this possible. The elegant explanation is that the same patterns of activation that have been involved in direct and conscious involvement with such pieces of information in the past are operating as they would, but without data from the senses and without impulses being directed to the muscles (Campbell 1960). The ability to substitute such simulations of real sensory/ motor activity for actual test trials in the world (e.g. using up reams of paper in order to devise the first line of the poem) is a fact that is probably directly linked to its adaptive value in prehistoric ecologies. Possessing such an ability improves performance on production of survival-critical decisions. Neuronal patterns that do what a biological agent could do, but without the biological agent doing anything, are the neurophysiological basis for agent-based models of cognition. When these normally serial behavioral processes are allows to progress on several intersecting tracks, unexpected and convergent discoveries become nearly commonplace.

Continuing to write your poem, you might come up with several new lines, some of which penetrate the threshold of your direct attention but are still discarded, and others that you send back for additional work. Since you have literally written down the first line and plan to keep it, there is a constraint of rhyming now in addition to the constraints that were in place when you set out to write the poem: the two other long lines must rhyme with "Sam". An idea to use a cat character from Siam comes to mind, but this idea needs to be refined and expressed within the poem. A number of lines enter awareness that seem to fit some important criteria but lack in others, namely grammaticality. You think of "who's eye on a cat from Siam", "got caught by a cat from Siam", and other ideas. Although you had thought the first line was done you now realize that if you change the first line you can say "A wonderful parrot named Sam/ got caught by a cat from Siam".

This example presents a new set of features. First, there is the element that what has already been established will have an effect on the direction of the poem's development. Second, there is element of solidity and fallibility of ideas and poem fragments. A poet might seem to have generated something highly valuable and consider it a "keeper", but then realize that by modifying or sacrificing it, something better can take its place. Third there is the fact that the creative process involves a dialog between actively being engaged in creating the work and being receptive to ideas and pieces of poetry produced underneath the threshold of awareness. Being actively engaged, we experience the process as a meta-level executive judge who selects constraints for evaluation. These ground the multiple criteria for evaluation that are applied covertly. Not only do we experience and guide the production of a high-level structure for evaluation, but we experience the top level of pattern manipulation also. As we move items around we might have a need for some new ideas, to which the unconscious creative process then responds producing something.

If you allow this process to go on you might have a very good limerick. This could take some time. The process of writing a poem specifies no obvious length of time to take writing it. Perhaps your mediocre draft will surface later this year on a crumpled receipt you discover in a cupboard drawer. Doing some more work on it, you now realize what was missing in it and you create an clever poem with a surprising and funny finale! This can be due to a number of complexities of your personal life and state of mind. Perhaps your internal ecology of evaluations has developed in such a way that you now find it easier subject poems to scrutiny and modify them to meet these expectations. This unrefined explanation is not a symptom of the author's laziness but rather is due to a void in specific knowledge of technical details; understanding the way in which a creative mind changes over time is one of the most important areas of study that contemporary cognitive science can offer to our understanding of what we are.

The need for objective models that can be ethically studied as they change and grow is one reason to create a computer simulation like the one we will be exploring now. Among many other reasons for having them, computerized models of creative activity might help us to create interesting and new kinds of work, and even to boost our potential beyond our present expectations. These possibilities will be discussed in greater detail below.

Building a Computer Poet

The initial design of software that models how human beings write poetry seems unlikely to capture the essence of how it realistically happens, but what follows is a plan for an accurate model that can be made more specific in time as knowledge from trial runs and from other sources pours in. An ideal computer model of writing poetry should be capable of acting independently of human helpers. Like a human poet, the computer poet should have artistic sensibilities that are in some ways unique and in many ways contiguous with the senses of other (in this case, human) poets. It is important that the poet have the ability to develop constraints on evaluation at a high level and that there be a dialog between the top level of evaluation and lower levels. A poet must possess a lexicon, some knowledge of poetics, the ability to form concepts, the ability to form grammatical expressions, and a certain amount of cultural knowledge. The poet should also be capable of evaluating and making changes to its work at several levels.

Although this may seem to be a daunting task, translating these requirements into a computational description is not difficult. It is the development of a working model that actually measures up to all of these expectations that should prove challenging (the next section documents a working model which measures up to some of these expectations).

The architecture of the system is as follows. There are several parts: the generator module, the evaluator module, the work space, the lexicon, the conceptual knowledge base (conceptual dependency), and the syntactical knowledge base (explicit grammar rules). The generator module and the evaluator module run as parallel processes. The generators actively fill the work space with poetic objects as the passive evaluators are aroused by various features which cause them to apply numerical ratings of their interest. The evaluators are set up within a two-tiered structure, the upper tier controlling their organization and the lower one being the actual evaluators. The upper tier also acts to represent "consciousness" in that it controls the focus of attention (amount of processing devoted) to areas and aspects of work on the poem.

The action begins with the production of limerick container (the limerick form has no special significance to this work; any form is usable). Several copies of the limerick container with initially random content are released into circulation. Limerick containers are the computational objects that correspond to "ideas for whole poems", "ideas for lines", or "ideas for parts of lines". Both manipulation and evaluation modules can target any of these. Manipulation of these objects means either modifying them or combining them together. Modification and combination can be literal, or can be flavored with the bias of syntax, word choice, and conceptual organization. Evaluation means applying a value based on interest in working with what is in the limerick container. This value is used by the generator module in picking which forms to work with. Forms ignored for too long disappear.

Limerick containers and their subcompartments have the potential to increase in "solidity". Solidity is the poet's assessment of the probability of holding on to something in favor of replacing it with something else or changing it. Solidity is built up as a result of positive numerical comments from the evaluators module.

The two-tiered evaluators module is based on the idea that a poet can actively modify the organization of qualities that are operative in the creative work. The lower tier of the module contains trained neural networks. These networks are trained prior to the work of doing poetry because most people do not actively modify their fundamental poetic sensibilities in the course of writing a poem. These fundamental ideas take the form of trained neural networks. The networks are trained on human judgments of specific qualities of word arrangements, lines, and poems. Many of the qualities should be highly subjective and simple values like cute, dark, pretty, melodic, nice, abrasive, light, and so on. Once a neural network has picked up on patterns in words and in poetic forms, it may generalize to forms it has never seen in the training set. It is important to make sure that the networks generalize properly before introducing them to the system. In addition to subjective qualities, raters for other things including syntax can be included either as neural networks or as explicit rules.

The upper tier of the evaluator module performs two functions. One is to control the organization of the evaluators, and the other is to control the focus of attention on problem areas of a poem. Controlling the organization can mean determining how much weight should be given to specific qualities, how much weight should be given to combinations of qualities, or what combinations of qualities are especially interesting. It can be designed in such a way that it will give extra value to a quality it detects a second time after having discovered it once. It can also include a means weigh the importance of having isolate bits of highly interesting content with the need to balance levels of interesting content throughout the course of the poem.
This function of the upper tier is only vaguely defined at this time but is important nonetheless. The other function, the "consciousness" of the system, is more clear cut. As soon as the system reaches a point where one limerick container has become significantly more solid than other ones, the consciousness of the system focuses in on this container and looks for problems with it. Keep in mind that a limerick container does not necessarily contain a whole poem, but can be a single line or even a piece of a line. Consciousness, which of course is not actually consciousness, devotes more focus of work onto refining the already interesting piece of work into something more complete.

We have described the computational system. Now let us walk through an ideal poetic simulation. First the system is initiated: the generator module immediately begins shooting out random limerick container objects into the work space, and the evaluator module listens in, placing ratings on various parts and pieces. The generator module manipulates forms that have been rated well, combining them and modifying them. The evaluator module continues to apply ratings as the generator works with manipulations on the more well liked forms. Meanwhile less interesting containers disappear making room for new ones (mainly combinations or modifications of existing containers). This process continues for an indefinite stretch of time until the evaluator module (in its role of conscious director) registers that a container that has surpassed the numerical rating value necessary in order to be noticed. A large amount of lesser work disappears prematurely to make space for work on this one container. Based on problems detected with various aspects of the container, work is done on it as it grows in value. Eventually the system stops because the poem is finished and is interesting in terms of what the system considers to be good work.

The computational model we have just defined is intended to be developed into a working program with the intention that by running simulations and altering parameters in many different controlled ways, a more detailed idea of how poetic creativity works will emerge. The first steps toward developing this model have been taken already and the first working executable version is described in the section that follows.

Poevolve User's Guide

Although the system just described has more merit than the one I am about to describe, this one has the one major advantage of already having been implemented and run a number of times. Though the system just described would be capable of most of what is involved in the creative composition of poetry, the Poevolve (poe[try] + evolve) software we now turn our attention to is not incapable of all of these important abilities. Poevolve is a working prototype of the more advanced creative poetry program that has not yet been constructed. It has the important features of being evolution based and having a trained neural network as its fitness measure. The program does lack a number of essential features that would improve its chances of being truly creative: it lacks the ability to focus in on aspects of poems, the ability to work piecemeal with parts of poem ideas, a dynamic evaluator, concepts, syntax, and the model of the dialectic interaction between top-down and bottom-up processes that are included in the above description. Even though it lacks these things it is interesting to see the tentative results of what has been done so far.

Poevolve, unlike the system its maker aspires to have it be, is very straightforward and minimal. The user of the program sets the initial population size (which does not change) to some number, and also sets the number of generations for which the members of populations will be evaluated and be subject to combination (crossover) and modification (mutation). The overarching architecture of Poevolve is the standard Darwinian algorithm expressed the same way it is typically expressed in computer code. A population evolves from time A to time B, and is then tested to see if any improvement has occurred. Note that

Poevolve plays the important role as a testing space for parts of the evolution program before they are introduced into the new model of poetic creativity. This highlights another (however obvious) difference between the current and future software: Poevolve is an instance of evolutionary computation, where as its future incarnation will be a model of a poet writing poetry, invoking evolutionary (creative) processes in order to come up with ideas. The program we are describing now is a step in the direction of simulating the more realistic model.

After the user defines the size of the population and for how long it may evolve, a population is instantiated, comprised of the specified number of individual limerick poems. This initial population is referred to as population zero (0). Population 0 is randomly generated. Randomly generated poems are produced using a single method that calls on other methods to select some words from the lexicon, to search for rhyming words, and to find some words that fit the correct stress patterns necessary to produce a limerick structure. A word on the lexicon: the lexicon contains 1107 entries, where each entry contains a word and some information on the word's stress, phonetic structure, and syllables. 1107 seems like an arbitrary number of words to include, and in fact this is simply the number of words that remain after having mixed together the words from a collections of limericks (some of which appeared on the questionnaire-- more on this below) and removing repetitious instances of words from the list. This method of getting the words for the lexicon was devised because it is natural to use words that have already been used in limericks to produced more limericks. This ensures that a usable ratio of words of long and short lengths exists, and that enough rhyming combinations exist. In short, the lexicon is considered to be "limerable" (good for composing limericks).

Population 0 is first completely filled with randomly generated poems, and then the fitness method rates them. Using an artificial neural network is an essential part of the system because it gives the system autonomy. Instead of having to keep a person near the system at all times to offer their personal neural networks as fitness evaluators, the system should be able to take care of its own self-evaluations. Not only are there very few people who are willing to sit day after day grading a computer's output, but even the ones who are willing to do this for some length of time are too slow to be of any use. So these useless people have been replaced by a high speed neural network that has been trained on the ratings done by 160 highly useful undergraduate students of psychology, computer science, and cognitive science.

The decision to get ratings from a large group of people in order to get the central tendency of their impressions of the poems makes it so that the ratings the neural net has been trained on do not just reflect a single individual's impressions. On the other hand a good argument for training the network on the ratings made by a single individual is that reading poetry is a highly personal activity and that it is possible to lose the nuances of the distinctions made in the process of taking the mean or mode of a group.

These ratings that are used to train the neural network are the modal ratings of research participants who have been asked in questionnaire form how "creative" each poem is on a scale where 1 is not at all creative and 6 is extremely creative. The choice of asking how creative the poems are is based on the fact the poems (for reasons of simplicity) will be judged based on a single measure. For this reason the generic multipurpose measure to be used could be either "likability", "creativity", or something equally as general in scope. The questionnaires used contained a random selection of 25 poems drafted by humans and 25 by a random limerick generator that will be discussed in greater detail below. Four different versions of the questionnaire that have been distributed to the research participants differ only in that the order is randomly arranged differently in order to iron out any carryover effects caused by the particular order of the succession of poems being rated.

The neural network is a recurrent network, meaning that since some of the processing units propagate their signals in a loop back to a prior layer, a poem may be fed into the network one syllable at a time as it builds up a rating for the whole poem. The reasoning behind using this kind of network is that the network should be able to detect facts about syllables, words, and lines in the context of the limerick as a whole. This particular network has a layer of 77 input units, a layer of 40 hidden units, a layer of 40 context units, and a layer of 6 output units. The output layer displays a number from 1 to 6 indicating the rating given to the limerick entered into the input layer. The hidden layer is necessary in order for the network to make anything more than simple kinds of associations. The layer of context units refers back to the hidden units in a loop, allowing the network to form recurrent connections, building up a pattern over time.

Input units encode information on a single syllable from a word in the lexicon. Of the 77 units in the input layer, each group of units encodes some information on this lexical entry. 11 of these units make up a distributed coding of a number between 1 and 1107 corresponding to the index of the entry in the lexicon the syllable belongs to. This allows for the possibility of the neural network forming connections based on the fact that a particular word has been used or that a group of words has occurred in a certain order. This makes some semantic connections possible. Phonology is then encoded as a 66 unit CCVSCC cluster where "C" is a set of 13 units encoding a consonant, "V" is a set of 13 units encoding a vowel, and "S" is single unit indicating that there is stress or that there is not stress on the syllable. Each of the five 13-unit phonemes contains information on whether it is a consonant or vowel, whether it is voiced or not, and the manner and place of articulation. Using this encoding, the neural net can pick up distinctions and relations among specific sounds, thus increasing the comprehensiveness of the mapping it may make between poem and numeric rating.

The neural network has been trained on an incomplete set of data from the questionnaire and has demonstrated the ability to generalize to the rest of the data from the questionnaire (which it was not previously exposed to). The degree of detail to which it may generalize is not that great, however. The network makes distinctions between two values: one covers the high ratings given by people to good limericks written by people, and the other covers the low ratings granted by people to the random limericks composed by machine. This binary rating network is imperfect but it is perfectly viable for the purposes of the poetry generating program.

It is somewhat significant to note that the idea of using a neural network as a method of gauging fitness in an evolutionary scenario is relatively original. The idea should be the focus of more work in the future. The author of this paper is aware of only four other instances of the idea. Of these four, one is an industrial application (Dagli 1993), another is an analytic and synthetic tool for biologists (Schneider 1995), and two are musicological. One, the GenJam model (Biles 1996) has had limited success at its goal of modelling jazzy improvisation guided by a skilled mentor. The other known musicological use of the idea (Burton and Vladimirova 1997) is an incomplete work in progress, including a proposal for the invention of an interactive drum machine that would act with some degree of freedom in creating rhythms while also learning from ongoing external stimulation provided by a drummer. Burton and Vladimirova's model is in theory a genetic algorithm in which the neural net (acting as judge of the fitness of rhythmic patterns) belongs to a class of unsupervised neural networks called Adaptive Resonance Theory (ART) networks. Like the network in the GenJam model, this network would continue to learn throughout the course of the system's evolution. This fact makes these models different from the Poevolve model in that the network here acts as a pre-established sense of a type of subjective judgement rather than a changing, learning network. The changing and learning occur here as the generation of novelty; the means of rating these novel forms remains stable. The conceptualization of a model of creativity expressed in the previous section outlines a way of making evaluation dynamic without compromising the idea that senses represented by networks should not change during the creative process.

A fitness function is the key component of an evolution program, but without some means of generating novelty there can be nothing for it to judge. This is where the mutation and crossover methods come into play. The mutation and crossover methods are somewhat simple but they are laden with combinatorial potential. The mutation operator either does something or nothing to the set of words this limerick carries around (its minilex), it then either does something or nothing to the rhyme, and then either does something or nothing to the lines of the poem. Modifying a poem's minilex means either replacing one quarter of its words with new words or replacing one half of its words with new words. Altering the rhyme means getting a new A rhyme scheme, a new B rhyme scheme, or both. Changing lines can mean that one or more line of the poem will change. Changing a line means modifying 1 to 3 words within the line or modifying every word (with the exception of those in the rhyme (which are treated as separate).

The crossover method does some work on two existing poems and returns a third poem. When the crossover method is called on some poems, first it may merge the former poem's minilex with the latter's or ignore the latter minilex, next after that it may take into consideration the other poem's rhyming or ignore the other poem's rhyming altogether, and finally after that it will either work with the former and the latter's lines or ignore the latter's lines. At every stage if the latter poem's features are ignored there is no interaction and the original poem's features are left unchanged. Just as in the the case of mutation there is a possibility that no changes will be made at all. For crossover, working with two minlexes means either borrowing one quarter of the other poem's minilex or borrowing one half of the other poem's minilex. Crossing the rhymes means either borrowing the A rhyme scheme from the other poem or borrowing the B rhyme scheme from the other poem. Crossing lines can mean that lines will either be shared or merged. Sharing lines means that the A lines or the B lines will be borrowed exclusively. Merging lines means that either all the lines or just the A or B lines will be merged together. Merging together lines consists of using some of the words from each line and patching them up with additional words from the minilex.

A small selected few of the limericks from population 0 (the poems given the highest ratings) are the poems on which mutation and crossover operations are performed. Mutation, crossover, and direct copy produce the next generation of limericks. 70 percent of the new generation are products of crossover, 20 percent result from mutation, and 10 percent come from directly copying poems from the highly rated set. Returning again to the neural network fitness evaluator, the process repeats until the predetermined number of generations have passed.

The results of Poevolve are interesting but are inconclusive and additional work needs to be done. The fact that the neural network fitness measure works well indicates that it is the generative side that needs work at this time. The current system produces limericks that in many ways seem random, but increase in value over the course of of program runs according to the results so far. Much of this seems to be due to lack of explicit syntax rules. In addition to the need for conceptually based and syntactically interesting operations on poems, performance issues need to be dealt with so that larger populations can be worked on for longer periods of time. If this is the primary problem, it follows that the full-fledged version of the model of poetry composition will require an even more massive amount of processing power to implement.

Concluding Remarks

The computational model of creativity introduced in this paper aims to model accurately the creative process involved in working on writing a poem. As more and progressively better results from simulations come about the model may become refined in such a way that it could yield important and unexpected findings and predictions that may generalize beyond poetry to other domains in which creativity is prominent.

This model and other related creative software may also play a role in advancing poetry and poetics. By understanding creativity and automating some processes that can be called creative, many writers will perhaps be challenged to go beyond normal means of expression developing a repertoire of techniques for reshaping the creative process itself, in contrast to only reshaping its articulations and styles.

These goals are a far cry from the results of recent program runs, but more developed versions aim to move progressively in the direction of closely mirroring human creativity and producing high quality output.

References

Burton, A. R., and Vladimirova, T. 1997. Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition, Technical Report, University of Surrey.

Biles, J. A., Anderson, P. G., and Loggi, L. W. 1996. Neural Network Fitness Functions for a Musical IGA, Technical Report, Information Technology and Computer Science Dept., Rochester Institute of Technology.

Csikszentmihalyi, M. 1999. Creativity. *MIT Encyclopedia of the Cognitive Sciences*. Cambridge, Mass: MIT Press.

Campbell, D.T. 1960. Blind Variation and Selective Retention in Creative Thought as in Other Knowledge Processes. *Psychological Review* 67:380-400.

Dagli, C. H., and Sittisathanchai, S. 1993. Genetic Neuro-scheduler for Job Shop Scheduling. *Computers and Industrial Engineering* 25(1-4):267-270.

Finke, R.A., Ward, T.B., and Smith, S.M. 1992. *Creative Cognition: Theory, Research, and Applications.* Cambridge, Mass: MIT Press.

Fogel, L.J., Owens, A.J., and Walsh, M.J. 1966. *Artificial Intelligence Through Simulated Evolution*. New York, New York: John Wiley.

Gruber, H.E., and Davis, S.N. 1988. Inching Our Way Up Mount Olympus: The Evolving-systems Approach to Creative Thinking. *The Nature of Creativity* ed. Sternberg, R. Cambridge, Mass: Cambridge University Press.

Schneider, G., Schuchhardt, J., and Wrede, P. 1995. Amino Acid Sequence Analysis and Design by Artificial Neural Network and Simulated Molecular Evolution-- An Evaluation. *Endocytobiosis and Cell Research* 11(1):1-18.

Turco, L. 1986. *The New Book of Forms: A Handbook of Poetics.* Hanover, NH: University Press of New England.