Focus

Soft Computing 8 (2004) 318–324 © Springer-Verlag 2003
DOI 10.1007/s00500-003-0288-9

Routine automated synthesis of five patented analog circuits using genetic programming

J. R. Koza, M. A. Keane, M. J. Streeter

Abstract This article reports on a project in which we browsed patents issued after January 1, 2000 to commercial enterprises or university research institutions for analog electrical circuits. We then employed genetic programming to automatically design (synthesize) entities that duplicated the functionality of five post-2000 issued patents. The automated method works from a high-level statement of the circuit's intended function. The article addresses the question of what is actually delivered by the operation of the artificial problem-solving method in relation to the amount of intelligence that is supplied by the humans employing the method (something we refer to as the yield of an automated method). The article also addresses the question of the routineness of the artificial problem-solving method – that is, the amount of effort required to make the transition from problem to problem within a particular domain. The conclusion is that the artificial method routinely delivers high-yield, human-competitive (i.e., previously patented) results.

Keywords Genetic programming, Automatic circuit synthesis, Evolvable hardware, Automated design, Artificial intelligence

1 Introduction

The common goal of the fields of artificial intelligence, machine learning, automated logic, and automated design is to get a machine to produce a human-like solution from a high-level statement of a problem.

One question that arises in attempting to evaluate artificial problem-solving methods is whether there is any real substance to the demonstrative problems. Published demonstrative problems are often contrived problems that circulate exclusively inside academic groups that study a particular artificial method, but have no importance or relevancy to any actual scientific or engineering research or development effort.

A second question concerns the yield of the artificial system. What is actually delivered by the operation of the artificial method in relation to the amount of analysis and intelligence that is pre-supplied by the humans employing the method? The Deep Blue system has, for example, produced noteworthy human-competitive results in the domain of chess. However, a team of very intelligent humans spent years analyzing this particular game. Then, they spent considerable effort developing specific computer software and hardware to efficiently evaluate large numbers of alternative moves and to retrieve specific sequences of plays that are known to be useful in particular situations. Given the massive infusion of human analysis and intelligence, the "value added" during the actual operation of the artificial method during an actual game is relatively minor. Thus, Deep Blue has a very low A-to-I ratio (artificial-to-intelligence ratio). That is, this particular artificial problem-solving method has a very low yield.

A third question concerns the routineness of the artificial system. The word "routine" means more than that the problem-solving method is general. Routineness addresses the difficulty of the transition required to grapple with new problems. When we use the term "routine" to describe a problem-solving method, we mean that relatively little human effort is required to get the method to successfully handle additional problems within a particular domain and to successfully handle additional problems from another domain. A problem-solving method is not routine if each new problem requires a massive readjustment of the method or a massive new infusion of human intelligence.

This article reports on a project in which we browsed patents issued after January 1, 2000 to commercial enterprises or university research institutions for analog electrical circuits.

We then employed an artificial method (genetic programming) to automatically design entities that duplicated the functionality on the issued patents. Specifically, we automatically synthesized the entities contained in five patents issued after January 1, 2000.

Section 2 describes the five patented inventions that are discussed in this article. Section 3 discusses the artificial problem-solving method. Section 4 presents the five

Published online: 4 September 2003

J. R. Koza (✉)
Department of Medicine,
Department of Electrical Engineering,
Stanford University, PO Box K, Los Altos, CA 94023
e-mail: koza@stanford.edu

M. A. Keane
Econometrics Inc.,
1960 N. Lincoln Park West #1103, Chicago, IL 60614

M. J. Streeter
Genetic Programming Inc., PO Box K,
Los Altos, CA 94023

human-competitive results, including an analysis of the yield and routineness of these results.

2 The five patents

The five patented inventions are from the domain of designing (synthesizing) analog and mixed analog-digital electrical circuits. The goal of each of the five problems is to automatically synthesize the patented inventions from a high-level statement of circuit's desired performance and characteristics. Since all five patents were issued after January 1, 2000, these five circuits represent actual current research and development efforts by engineers and scientists in the field.

2.1 Balun circuit

The purpose of a "balun" (balance/unbalance) circuit is to divide an input signal into two half-amplitude signals which are 180 degrees out of phase from each other. The circuit described in U. S. patent 6,265,908 [9] is noteworthy in that it uses a power supply of only 1 Volt.

2.2 Voltage-current conversion circuit

The purpose of the voltage-current conversion circuit in U. S. patent 6,166,529 [5] is to take two voltages as input and to produce as output a stable current whose magnitude is proportional to the difference of the voltages.

2.3 Cubic signal generator

U. S. patent 6,160,427 [4] covers a "Compact cubic function generator". This is a computational circuit designed to produce as output the cube of an input signal. The patented circuit is "compact" in the sense that it requires a voltage drop across no more than two transistors at any point in the circuit.

2.4 Register-controlled variable capacitor

U. S. patent 6,013,958 [1] covers a circuit that is equivalent to a capacitor whose capacitance is controlled by the value stored in a digital register.

2.5 High-current load circuit

U. S. patent 6,211,726 [3] covers a circuit designed to sink a time-varying amount of current in response to a control signal. Toward this end, Daun-Lindberg and Miller of IBM employed a number of FET transistors arranged in a parallel structure, each of which sinks a small amount of the desired current.

3 Methods

We used genetic programming to breed a population of computer programs representing circuits over a series of generations. Genetic programming starts with thousands of randomly created computer programs and uses the Darwinian principle of natural selection, recombination (crossover), mutation, gene duplication, gene deletion, and certain mechanisms of developmental biology to breed an improved population over a series of many generations.

The five major preparatory steps for genetic programming entail determining

(1) the set of primitive functions for the to-be-evolved program,
(2) the set of terminals for the to-be-evolved program,
(3) the fitness measure,
(4) parameters for controlling the run, and
(5) a termination criterion.

The function and terminal sets for all five problems involving the post-2000 patented analog circuits were the same. They permit the construction of any circuit that is composed of transistors, resistors, and capacitors and that is connected in any topological arrangement. That is, the terminal and function sets incorporate *de minimus* and platitudinous information about circuits.

The control parameters and termination criterion were the same, except that we used different population sizes for some of the runs. The population size was chosen on a practical (not analytic) basis, namely to ensure the reasonableness of each run's elapsed time.

The main difference between the runs was that a different fitness measure was used for each problem.

The fitness measure specifies what performance in the time-domain or frequency-domain is desired, given various inputs.

3.1 Low-voltage balun circuit

The fitness measure for this problem consisted of a (1) frequency sweep analysis designed to ensure the correct magnitude and phase at the two outputs of the circuit and (2) a Fourier analysis designed to penalize harmonic distortion.

3.2 Voltage-current conversion circuit

The fitness measure for this problem employed four time-domain input signals (fitness cases). For each fitness case, the goal was for the circuit to output a stable current proportional to the difference between the two voltage signals. Fitness is the sum, over the fitness cases, of the absolute difference between the desired output current and the circuit's actual output current. We included a time-varying voltage source beneath the output probe point to ensure that the output current produced by the circuit was stable with respect to any subsequent circuitry to which the output of the circuit might be attached. The relative weight of each fitness case was defined as the reciprocal of the patented circuit's error for that fitness case, so that the patent circuit was defined to have a fitness of 1.0.

3.3 Cubic signal generator

Fitness is measured using four time-domain fitness cases reflecting various input signals and time scales. Fitness is the sum, over the fitness cases and time steps, of the absolute value of the difference between the cube of the input voltage and circuit's output. The patented circuit is "compact" in the sense that it requires a voltage drop across no more than two transistors at any point in the circuit. The compactness constraint is enforced by allowing the evolutionary process access to only a 2-Volt power supply.

3.4
Register-controlled variable capacitor

For this problem, we used 16 time-domain fitness cases. The 16 fitness cases ranged two different input signals and all eight possible values of a three-bit digital register. Fitness is the sum, over 16 fitness cases, of the absolute value of the difference between output voltage associated with a capacitor having the ideal value for a particular setting of the three-bit register and the circuit's actual output voltage.

3.5
High-current load circuit

The fitness measure for this problem consisted of two time-domain fitness cases, each representing a different control signal. Fitness is the sum, over the fitness cases, the absolute value of the difference between the amount of current sunk by the circuit and the desired amount of current given by the time-varying control signal. Each fitness case was weighted by the reciprocal of the patented circuit's error on that fitness case, so that the patent circuit was defined to have a fitness of 1.0.

3.6
Additional details

We generally followed the methods in [6, 7].

To ascertain the performance of candidate circuits during the run, circuits were simulated (analyzed) using the SPICE simulator [10]. SPICE analyzes and simulates circuits that it is given. It has no ability to synthesize circuits. The (considerable) knowledge embodied in the SPICE simulator is not knowledge about circuits, but knowledge about how to efficiently converge on a solution to a system of non-linear differential-integral equations. Whatever knowledge may be embodied in the simulator, the artificial method (genetic programming) neither acquires nor uses any knowledge from the simulator about how to synthesize a circuit. Indeed, this work could have avoided the use of a simulator altogether by building breadboards for the candidate circuits on each generation of the run. We could then have physically measured the performance of each physical entity and then inserted the observed performance data into the run. We make this observation to emphasize that when an artificial method uses a simulator to ascertain the performance of a candidate individual, the method is not thereby using or acquiring any knowledge concerning circuits. The differences between using a simulator and building thousands of bread boards on each generation of the run is a matter of cost (in time and money) and accuracy of the data.

4
Results for the five post-2000 patented inventions

4.1
Balun circuit

The best-of-run evolved circuit (Fig. 1) was produced in generation 84 and has a fitness of 0.429. The patent circuit had a total fitness of 1.72. That is, the evolved circuit is roughly a fourfold improvement over the patented circuit in terms of our fitness measure. Figure 2 shows the time-domain behavior of the best-of-run balun circuit from generation 84 for both the in-phase and out-of-phase output ports. The evolved circuit is superior to the patented circuit both in terms of its frequency response and its harmonic distortion. Figure 3 shows the balun circuit of

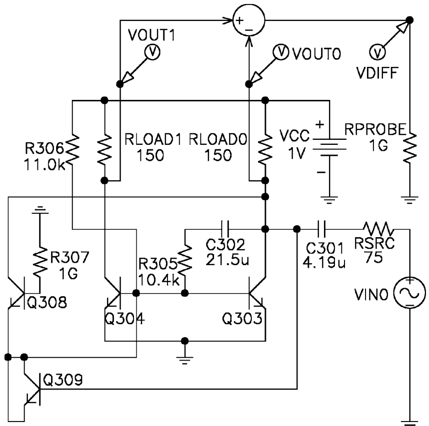

Fig. 1. Best-of-run balun circuit from generation 84

Table 1. Five patented inventions

Invention	Year	Inventor	Place
Mixed analog-digital integrated circuit for producing variable capacitance	2000	Turgut Sefket Aytur [1]	Lucent Technologies Inc.
Voltage-current converter	2000	Akira Ikeuchi and Naoshi Tokuda [5]	Mitsumi Electric Co., Ltd.
Cubic function generator	2000	Stefano Cipriani and Anthony A. Takeshian [4]	Conexant Systems, Inc.
Low-voltage high-current transistor circuit for testing a voltage source	2001	Timothy Daun-Lindberg and Michael Miller [3]	International Business Machines Corporation
Low-voltage balun circuit	2001	Sang Gug Lee [9]	Information and Communications University

U. S. patent 6,265,908. The inventor identifies one key capacitor as the difference that distinguishes his patented circuit from the prior art (C2 in Fig. 3 and C302 in Fig. 1). The evolved circuit contains that capacitor.

4.2
Voltage-current conversion circuit

A circuit (Fig. 4) emerged on generation 109 of our run of this problem with a fitness of 0.619 (over the four fitness cases used for purposes of evolution). That is, the evolved circuit has roughly 62% of the average (weighted) error of the patented circuit. Figure 5 compares the voltage produced by the best-of-run voltage-current-conversion circuit from generation 109 with the ideal output for fitness case 2. As can be seen, the two curves are almost indistinguishable. The comparisons for the other three fitness cases are similarly close. The evolved circuit was subsequently tested on unseen fitness cases which were not part of the fitness measure and outperformed the patented circuit on these new fitness cases.

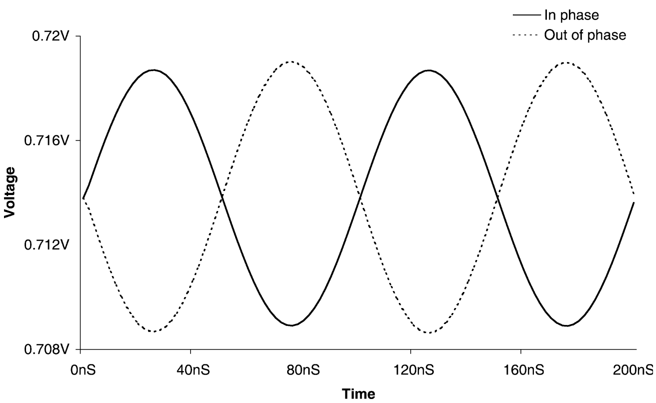

Fig. 2. Time-domain behavior of best-of-run balun circuit from generation 84

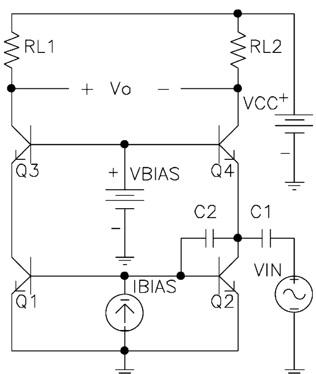

Fig. 3. circuit of U. S. patent 6,265,908

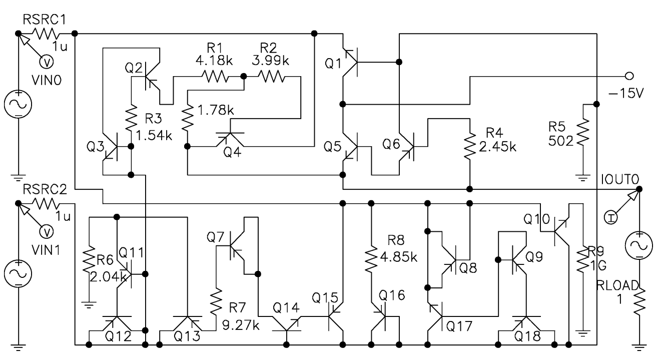

Fig. 4. Best-of-run voltage-current-conversion circuit from generation 109

4.3
Cubic signal generator

The best-of-run evolved circuit (Fig. 6) was produced in generation 182 and has an average error of 4.02 mV. The patented circuit had an average error of 6.76 mV. That is, the evolved circuit has approximately 59% of the error of the patented circuit over our four fitness cases. Figure 7 shows the output produced by the best-of-run cubic signal generation circuit from generation 182. As can be seen, the two curves are almost indistinguishable. Figure 8 compares the error of the best-of-run cubic signal generation circuit from generation 182 and the error of the circuit of U. S. patent 6,160,427. As can be seen from the figure, the error produced by the genetically evolved circuit is

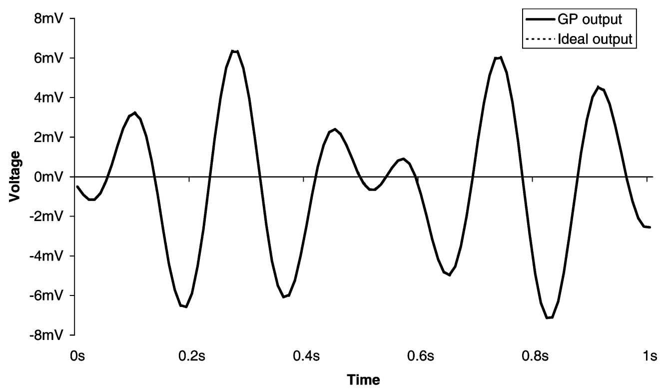

Fig. 5. Comparison of the voltage produced by the best-of-run voltage-current-conversion circuit from generation 109 with the ideal output for fitness case 2

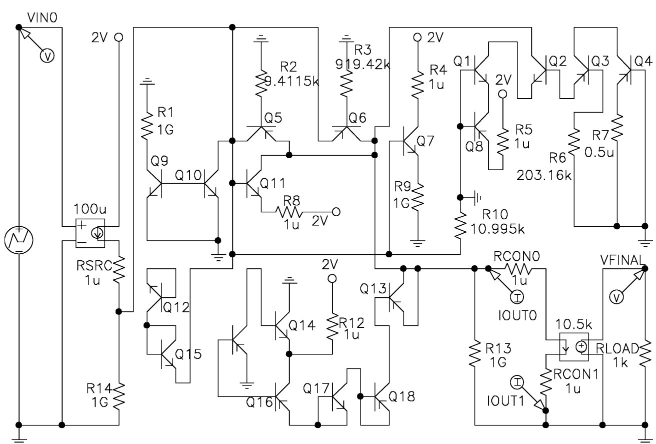

Fig. 6. Best-of-run cubic signal generation circuit from generation 182

generally smaller than that produced by the patented circuit. The evolved circuit duplicates the functionality of the patented circuit but does not infringe on the patented circuit. That is, the evolved circuit is a different way of solving the same problem.

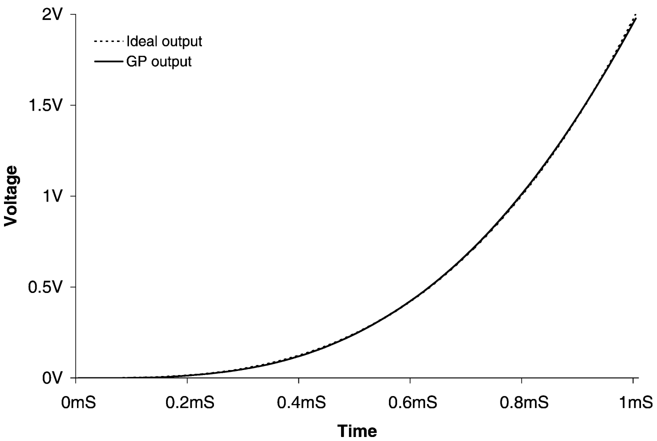

Fig. 7. Output produced by the best-of-run cubic signal generation circuit from generation 182

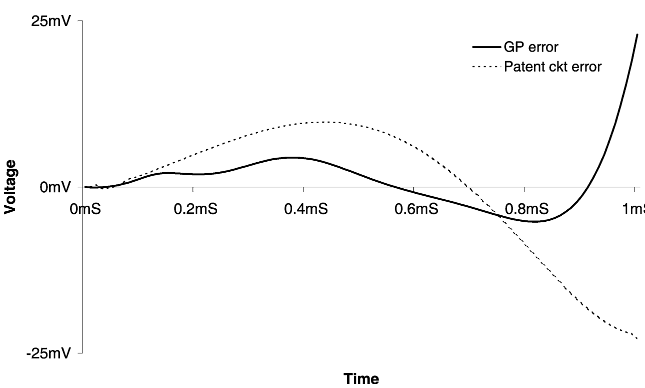

Fig. 8. Comparison of error of the best-of-run cubic signal generation circuit from generation 182 and the error of the circuit of U. S. patent 6,160,427

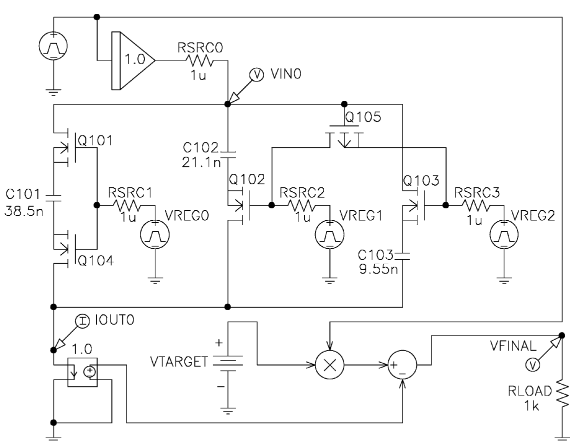

Fig. 9. Smallest compliant register-controlled capacitor circuit from generation 98

4.4 Register-controlled variable capacitor

Over our 16 fitness cases, the patented circuit had an average error of 0.803 mV. In generation 95, a circuit emerged (Fig. 9) with average error of 0.808 mV, or approximately 100.6% of the average error of the patented circuit. During the course of this run, we harvested the smallest individuals produced on each processing node (deme) which were compliant with a certain maximum level of error. Examination of these harvested individuals revealed a circuit created in generation 98 which matched the topology of the patented circuit. Additionally, the evolved circuit employs capacitors of 9.55, 21.1, and 38.5 nF associated with the low, middle, and high-order bits, respectively, of the three-bit register. That is, the evolved circuit employs an approximately exponential weighting of capacitors. This exponential weighting is one of the key feature of the patented circuit. Figure 10 shows the register-controlled variable capacitor circuit of U. S. patent 6,013,958.

4.5 High-current load circuit

Our run for this problem eventually reached a plateau and produced a circuit that sunk the desired current into one of the negative power supplies (rather than to ground). This "cheating" circuit was not in the spirit of the patented invention. However, on generation 114 of this run (before the cheating solution appeared), a circuit emerged that duplicated Daun-Lindberg and Miller's parallel FET transistor structure. This circuit had a fitness (weighted error) of 1.82, or 182% of the weighted error for the patented circuit. This circuit is presented in Fig. 11. Figure 12 compares the output of the best-of-run high current load circuit from generation 114 for fitness case 1 and the ideal output. As can be seen, the two curves are almost indistinguishable. Figure 13 shows the high current load circuit of U. S. patent 6,211,726.

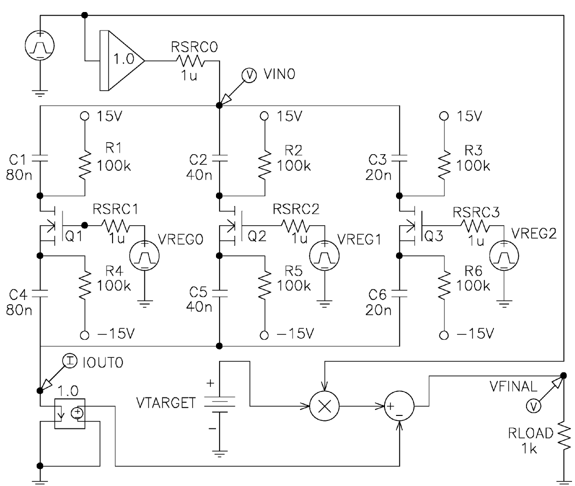

Fig. 10. Figure Register-controlled variable capacitor circuit of U. S. patent 6,013,958

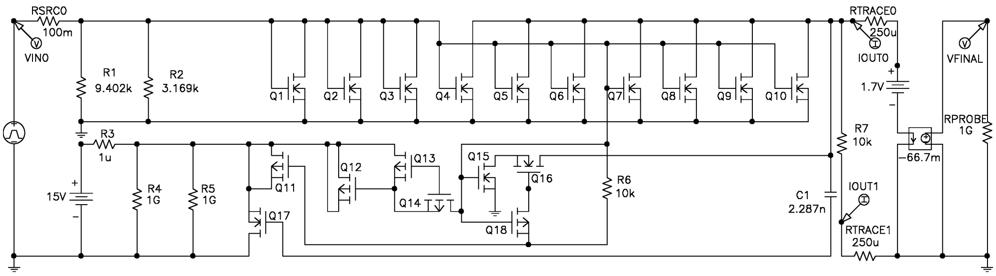

Fig. 11. Best-of-run high current load circuit from generation 114

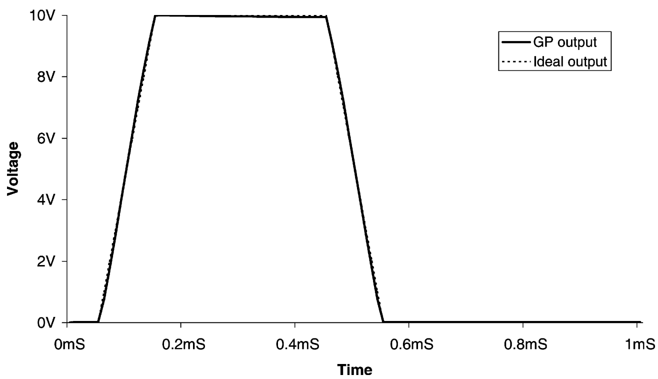

Fig. 12. Comparison of output of the best-of-run high current load circuit from generation 114 for fitness case 1 and ideal output

5
Yield

We define the *AI ratio* (artificial-to-intelligence ratio) of a problem-solving method as the ratio of that which is delivered by the automated operation of the *artificial* system to the amount of *intelligence* that is supplied by the human applying the problem-solving method to a particular problem.

The results produced by genetic programming on all five problems are human-competitive because the results was patented as an invention. Thus, the operation of the artificial system delivers a large amount of "A" and places a large number in the numerator of the AI ratio.

Determining how much intelligence was supplied by the human user requires the discipline of drawing a bright line between that which is delivered by the artificial system and that which was supplied by the intelligent human user prior to the launch of the run. We imposed that discipline when we identified the preparatory steps for each problem. We used *de minimus* knowledge and platitudinous information about circuits in carrying out the preparatory steps (thereby placing a small number in the denominator of the AI ratio).

Thus, the yield (the AI ratio) of the artificial problem-solving method here is high.

6
Routineness

The routineness of an artificial problem-solving method transition reflects the amount of effort required to get the method to successfully handle additional problems within a particular domain and additional problems from a different domain.

The results for the five problems involving circuits reflected the five different high-level statements of "what is to be done" for the five problems – that is, the five different fitness measures.

All five patented inventions are from the domain of analog electrical circuits. Referring to the preparatory steps listed above, the transition between the domain of circuits and another domain would require changing the function and terminal sets. For example, genetic programming is capable of automatically creating both the topology and sizing (tuning) for controllers [8] and automatically creating the topology and sizing of antennas [2]. Making the transition to the domain of controllers and antennas requires changing the function and terminal sets. The new function and terminal sets contain only platitudinous information about controllers and antennas. Of course, different fitness measures are used for problems from the field of controllers and antennas (just as different fitness measures are used for each of the five problems herein).

Thus, the routineness of the artificial problem-solving method here is high.

In summary, genetic programming routinely delivered high-yield, human-competitive results for the five problems discussed herein.

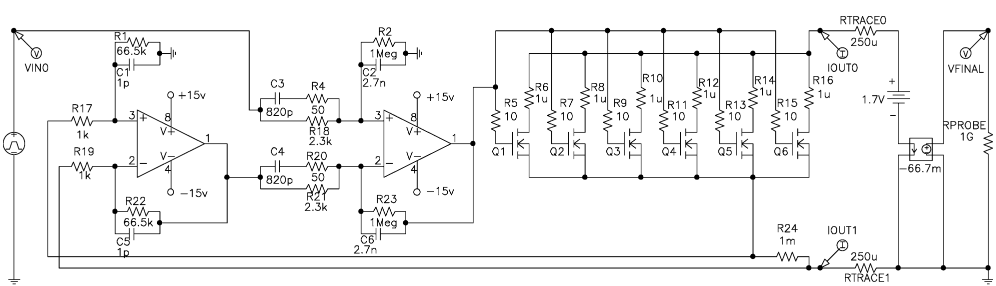

Fig. 13. High current load circuit of U. S. patent 6,211,726

7
Machine intelligence

In 1950, Turing proposed a three-person "imitation game" that might be used to determine whether machine intelligence had been achieved [11]. In the "imitation game," a judge tries to decide whether typewritten replies to questions came from a man or a woman. Turing's original test has been paraphrased in various ways over the years. One popular restatement of Turing's original test for machine intelligence is a two-person game in which a judge receives messages "over a wall" and tries to decide whether the messages came from a human or a machine.

The Turing test involves natural language processing and testing human competitiveness in many subtle ways. No current automated method satisfies the Turing test. Nonetheless, one aspect is relevant here. Patent Offices in various countries have been in the business of performing a similar kind of "over the wall" test for hundreds of years. For example, the U.S. Patent Office receives written descriptions of human-designed inventions and judges whether they satisfy the statutory requirement (35 United States Code 103a) of being "[un]obvious ... to a person having ordinary skill in the art to which said subject matter pertains."

The Patent Office operates at arms-length and does not know who (or what) actually conceived the proposed invention when it passes judgment on the merits of a patent application. The inventor could be an exceptionally creative human or it could be something else (e.g., an automated process). If an automated method were able to duplicate a previously patented human-created invention, the fact that the original human-designed version satisfied the Patent Office's criteria of patent-worthiness means that the automatically created duplicate would also have satisfied the Patent Office's criteria. Thus, whenever an automated method duplicates a previously patented human-designed invention, it can be viewed as satisfying a Patent-Office-based variation of the Turing test.

When an institution or individual allocates time and money to invent something and then also embarks on the time-consuming and expensive process of obtaining a patent, it has made a judgment that the work is of some practical or scientific importance. Moreover, the Patent Office also applies a statutory test of utility as a precondition to issuing a patent. Thus, the above Patent-Office-based variation of the Turing test differs from the original Turing test in that patented inventions represent non-trivial work by exceptionally creative humans.

8
Conclusions

We used an automated method to create entities that duplicate the functionality of five post-2000 patented inventions involving circuits. The automated method used *de minimus* human-supplied information. The artificial method routinely delivered high-yield, human-competitive results.

References

1. **Aytur, Turgut Sefket** (2000) Integrated Circuit with Variable Capacitor. U. S. patent 6,013,958. Filed July 23, 1998. Issued January 11, 2000
2. **Comisky William, Yu Jessen, Koza John** (2000) Automatic synthesis of a wire antenna using genetic programming. Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference, Las Vegas, Nevada. pp. 179–186
3. **Daun-Lindberg, Timothy Charles, Miller Michael Lee** (2001) Low Voltage High-Current Electronic Load. U. S. patent 6,211,726. Filed June 28, 1999. Issued April 3, 2001
4. **Cipriani Stefano, Takeshian Anthony A** (2000) Compact cubic function generator. U. S. patent 6,160,427. Filed September 4, 1998. Issued December 12, 2000
5. **Ikeuchi Akira, Tokuda Naoshi** (2000) Voltage-Current Conversion Circuit. U. S. patent 6,166,529. Filed February 24, 2000 in U. S.. Issued December 26, 2000 in U. S.. Filed March 10, 1999 in Japan
6. **Koza John R, Bennett III Forrest H, Andre David, Keane Martin A** (1999) Genetic Programming III: Darwinian Invention and Problem Solving. San Francisco, CA: Morgan Kaufmann
7. **Koza John R, Bennett III Forrest H, Andre David, Keane Martin A, Brave Scott** (1999) Genetic Programming III Videotape: Human-Competitive Machine Intelligence. San Francisco, CA: Morgan Kaufmann
8. **Koza John R, Keane Martin A, Yu Jessen Bennett, Forrest H III, Mydlowec William** (2000) Automatic creation of human-competitive programs and controllers by means of genetic programming. Genetic Programming and Evolvable Machines 1: 121–164
9. **Lee Sang Gug** (2001) Low Voltage Balun Circuit. U. S. patent 6,265,908. Filed December 15, 1999. Issued July 24, 2001
10. **Quarles Thomas, Newton AR, Pederson DO, Sangiovanni-Vincentelli A** (1994) SPICE 3 Version 3F5 User's Manual. Department of Electrical Engineering and Computer Science, University of California. Berkeley. March 1994
11. **Turing Alan M** (1950) Computing machinery and intelligence. Mind. 59(236): 433 – 460. Reprinted in Ince, DC (Ed.) 1992. Mechanical Intelligence: Collected Works of A. M. Turing. Amsterdam: North Holland. pp. 133–160