Automatic Design of Analogue Circuits

by
Yerbol Sapargaliyev

A thesis submitted for the Degree of Doctor of Philosophy

School of Engineering and Design
Brunel University, UK

October 2011

Abstract

Evolvable Hardware (EHW) is a promising area in electronics today. Evolutionary Algorithms (EA), together with a circuit simulation tool or real hardware, automatically designs a circuit for a given problem. The circuits evolved may have unconventional designs and be less dependent on the personal knowledge of a designer. Nowadays, EA are represented by Genetic Algorithms (GA), Genetic Programming (GP) and Evolutionary Strategy (ES). While GA is definitely the most popular tool, GP has rapidly developed in recent years and is notable by its outstanding results. However, to date the use of ES for analogue circuit synthesis has been limited to a few applications.

This work is devoted to exploring the potential of ES to create novel analogue designs. The narrative of the thesis starts with a framework of an ES-based system generating simple circuits, such as low pass filters. Then it continues with a step-by-step progression to increasingly sophisticated designs that require additional strength from the system. Finally, it describes the modernization of the system using novel techniques that enable the synthesis of complex multi-pin circuits that are newly evolved.

It has been discovered that ES has strong power to synthesize analogue circuits. The circuits evolved in the first part of the thesis exceed similar results made previously using other techniques in a component economy, in the better functioning of the evolved circuits and in the computing power spent to reach the results. The target circuits for evolution in the second half are chosen by the author to challenge the capability of the developed system. By functioning, they do not belong to the conventional analogue domain but to applications that are usually adopted by digital circuits. To solve the design tasks, the system has been gradually developed to support the ability of evolving increasingly complex circuits.

As a final result, a state-of-the-art ES-based system has been developed that possesses a novel mutation paradigm, with an ability to create, store and reuse substructures, to adapt the mutation, selection parameters and population size, utilize

automatic incremental evolution and use the power of parallel computing. It has been discovered that with the ability to synthesis the most up-to-date multi-pin complex analogue circuits that have ever been automatically synthesized before, the system is capable of synthesizing circuits that are problematic for conventional design with application domains that lay beyond the conventional application domain for analogue circuits.

To my family

for their pride of every progress I made during this research

Statement of Originality and Previously Published Work

Much of the work presented in this thesis has been previously published as listed below. Although some of these papers have co-author, the work appearing in this thesis is entirely my own, with the exception of parts acknowledge in the text. I hereby declare that this thesis has not been submitted, either in the same or different form, to this or any other universities.

List of Previous Publications

[1] Yerbol Sapargaliyev, Tatiana Kalganova, "EHW from Consumer Point of View: Consumer-Triggered Evolution", International Enformatika Conference, IEC'05, August 26-28, 2005, Prague, Czech Republic, ISBN 975-98458-6-5.

[2] Yerbol Sapargaliyev, Tatiana Kalganova, "Absolutely free extrinsic evolution of passive low-pass filter", IEEE Canadian Conference on Electrical and Computer Engineering, pp.1210 – 1213, Print ISBN: 1-4244-0038-4, May 7-10, 2006, Ottawa, Canada

[3] Yerbol Sapargaliyev, Tatiana Kalganova, "On Comparison of Constrained and Unconstrained Evolutions in Analogue Electronics on the Example of "LC" Low-Pass Filters", IEICE transactions on Electronics Vol.E89-C No.12 pp.1920-1927; (IEICE: The Institute of Electronics Information and Communication Engineers), 2006 2006/12/01 Online ISSN: 1745-1353, Print ISSN: 0916-8516

[4] Yerbol Sapargaliyev, Tatiana Kalganova, "Constrained and Unconstrained evolution of "LCR" low-pass filters with oscillating length representation", 2006 IEEE Congress on Evolutionary Computation, pp. 1529-1536, July 16-21, 2006, Vancouver, BC, Canada

[5] Yerbol Sapargaliyev, Tatiana Kalganova, "Unconstrained Evolution of Close-to-ideal "LCR" Low-pass Filter", INES 2006, 10th International Conference on Intelligent

Engineering Systems, pp.145 – 150, June 26-28, 2006, London Metropolitan University, London, United Kingdom

[6] Yerbol Sapargaliyev, Tatiana Kalganova, "Unconstrained evolution of analogue computational "QR" circuit with oscillating length representation", pp. 1-10, ICES 2008, September 21-24, Prague, Czech Republic. (ICES: The 8-th International Conference on Evolvable Systems: from Biology to Hardware)

[7] Sapargaliyev Y., T. Kalganova, "Automated Synthesis of 8-Output Voltage Distributor using Incremental Evolution", In Proc. of 2010 NASA/ESA Conference on Adaptive Hardware and Systems, IEEE, 2010, p. 186-193 (15-18 June).

[8] Yerbol Sapargaliyev, Tatiana Kalganova, "Challenging the Evolutionary Strategy to Synthesis Analogue Computational Circuits", *Journal of Software Engineering and Applications*, Vol. 3, No. 11, 2010, ISSN Print: 1945-3116, pp.1032-1039, Scientific Research Publishing, USA, November 2010.

[9] Yerbol Sapargaliyev, Tatiana Kalganova, "Open-Ended Evolution to Discover Analog Circuits for Beyond Conventional Applications", Accepted for the *Springer journal of Genetic Programming and Evolvable Machines*, November 22, 2011.

[10] Yerbol Sapargaliyev, Tatiana Kalganova, "Synthesis of Time-to-Amplitude Converter by Mean Coevolution with Adaptive Parameters", *Journal of Software Engineering and Applications*, Vol. 4, No. 8, 2011, pp. 447-464 Scientific Research Publishing, USA, August, 2011.

Acknowledgements

First of all, I was extremely fortunate to get under a supervision of Dr. Tatiana Kalganova, who gave me important feedback, guidance and support during the development of this research. She was far more helpful than what could reasonably be expected. She has always been welcoming for discussions and willing to help, for which I am deeply grateful.

People in Bio-inspired intelligent system research group have helped me in many ways: I am grateful to Emanuel, Mujtaba, Buddika and Gishantha for their support, cooperation and sincere friendship during my stay at Brunel University.

I am also grateful to Mr. Liu Maoxin for providing the financial support at the early stages of my research.

I am deeply indebted to my younger brother Kenen Sapargaliyev and my business partners Katia Guo and Mr. Xu for helping me to earn tuition and carrying extra duties at our common business.

I also like to thank Dr. Stoica from NASA Jet Propulsion Laboratory for his useful conversations, support and warmth.

Finally, I would like to thank my wife Janna Aitjanova for so much: encouraging me, keeping me optimistic and carrying a lot of additional household duties. I love you!

Yerbol Sapargaliyev

Contents

List of Figures

List of Tables

Acronyms

ADC – analogue-to-digital converter

AE – Artificial Embryology

AI – Artificial Intelligence

AL – Artificial Life

ANEM – add new element mutation

ASR – aggregated selection rate

CC – computational circuit

CPLD – complex programmable logic device

CSM – circuit structure mutation

CAD – computer-aided design

DAC – digital-to-analogue converter

DC – direct current

DEM – delete element mutation

DL – digital logic

DM – differentiated mutation

EA – evolutionary algorithms

EE – Evolutionary Electronics

EHW – Evolvable Hardware

ER –Evolutionary Robotics

ES – Evolutionary Strategy

FF – fitness function

FPAA – field programmable analog array

FPGA – field programmable gate array

FPTA – field programmable transistor array

GA – Genetic Algorithm

GP – Genetic Programming

HW – hardware

IC – integrated circuit

ILG – increasing length genotype

JPL – Jet Propulsion Laboratory

LC – circuits consisted of 2 components: L – inductor and C – capacitor

LCR – circuits consisted of 3 components: L – inductor, C – capacitor and R – resistor

LCRQQ – circuits consisted of 5 components: L – inductor, C – capacitor, R – resistor, Q – n-p-n bipolar transistor and Q – p-n-p bipolar transistor

MR – mutation rate

NASA - National Aeronautics and Space Administration

OLG – oscillating length genotype

PC – personal computer

REMP - rule of equal mutation probabilities

RNG – random number generator

PC – personal computer

PLC – programmable logic array

PS – population size

PSPICE - Personal Simulation Program with Integrated Circuit Emphasis

SOC - systems-on-a-chip

SPICE - Simulation Program with Integrated Circuit Emphasis

SR – selection rate

SRM – substructure reuse mutation

SSDN – single-source divergent neuron

System – the system that exploit the EHW toward the task set, composed with all necessary in-build hardware and software.

SW – software

TAC - time to amplitude converter

TIMC – time interval meter circuit

UDIP – uniformly distributed initial population

VDC – voltage distributor circuit

VNFE – very narrow focused evolution

WWW – web world wide

WDWC – winner dominated winner cooperates.

Chapter 1. Introduction

This chapter aims to state the goal of the whole thesis and to describe the thesis' overall structure, emphasizing its objectives and its contributions to the EHW field.

1.1 Background

The importance of analogue circuit design cannot be overestimated. Analogue components are the building blocks of all digital circuits. With analogue designs becoming more complex, there is increasing need for analogue circuit design automation. In a digital circuit design, the methodologies have been changing from gate-level design through hardware description language to a system-level design. By contrast, the methodologies of analogue circuit design have not been automated to a great extent so far, and have not changed from the early days of integrated circuit technology [1].

The main reason for the above difference comes from the fact that digital circuit design is ruled by discrete variables of *Boolean logic*, while analogue design is based on continuous variables of conductor/semiconductor physics. Therefore, with analogue circuit design it has been thought that it requires experience and the inspiration of human designers [1]. There are also several other differences in digital and analogue design methodologies [2].

The automated analogue circuit synthesis methodology can be broadly classified according three approaches. The first approach is that of knowledge-based synthesis ([4]-[7]), with broad utility of known substructures. The second approach is the sizing-based approach ([8]-[11] and [16]) and the third one is the sized topology generation approach ([2], [12]-[15], [17] and [18]). The works on automated analogue circuit synthesis - including both topological and numerical optimization with a minimum knowledge about circuit topologies - started to appear more than a decade ago. It has

been called Evolvable Hardware (EHW) when the synthesis strategies started to be combined with Evolutionary Algorithms (EA) and the idea to represent the analogue circuit in the form of code. EHW is the name of such systems, which use artificial evolution as a synthesis method of electronic circuits. Recent remarkable progress in processor speed enabled the heuristic approach to be taken for synthesize analogue circuits automatically [22].

EA refers to the heuristic techniques based on the principles of natural evolution. It was in the 1960s when American and European researchers independently developed stochastic search methods inspired by Darwinian evolution theory. The main techniques that represent EA are Genetic Algorithms (GA) [19], Genetic Programming (GP) [21] and Evolution Strategies (ES) [22], which nowadays are well-known optimization methods in realizations of EHW. The operations involved and the structure of the individuals in the population made the techniques different:

- Genetic algorithms – the solution of a problem is in the form of strings of numbers (traditionally binary), virtually always applying recombination operators in addition to selection and mutation [19];

- Genetic programming - the solutions are of computer programs, structured in the form of trees [21], [66], [46];

- Evolutionary strategy - works with vectors of real numbers as representations of solutions and does not have a crossover (recombination) procedure [22], [64].

All of them are similar in spirit, but differ in the details of their implementation and the nature of the particular problem to which they have been applied. The basic idea of an EHW system, adapted for the evolution of circuits, is illustrated in Figure 1-1. The synthesis strategy of EHW is based on a combination of EA and the concept process into an instance, called the *phenotype* [72]. According to [20], the *phenotype* is "any observable characteristic or trait of an organism: such as its morphology, development, biochemical or physiological properties, behaviour and products of behaviour" and by *coding* the *phenotype* can be transformed into a *genotype* [72]. The last one represents

the targeted circuit in the form of code, which makes it possible to process candidate solutions by mean of computers. Being packed into the population, the multitude of candidate solutions - named *chromosomes* - are managed by EA. A *fitness value* for each individual is assigned by a *fitness function* (FF) during an *evaluation* operation. During evaluation, the genotype is decoded into a *phonotype* and tested by means of simulation software or real hardware. The fitness value characterizes how the particular chromosome fits the target by its current intrinsic and extrinsic features. The best-ranked individuals are selected to the next *generation* of the iterative process. The last one starts from reproduction (or cloning) accompanied by the action of *genetic operators* that are intended to mimic the process of mutation. These operations are between the evolutionary part and the reconfigurable HW/SW (i.e. providing circuit configurations and circuit response, the last iteratively so) that over time the "quality" of the individuals persisting in the population tends to increase until the HW/SW is able to provide the circuits that satisfy the requirements.

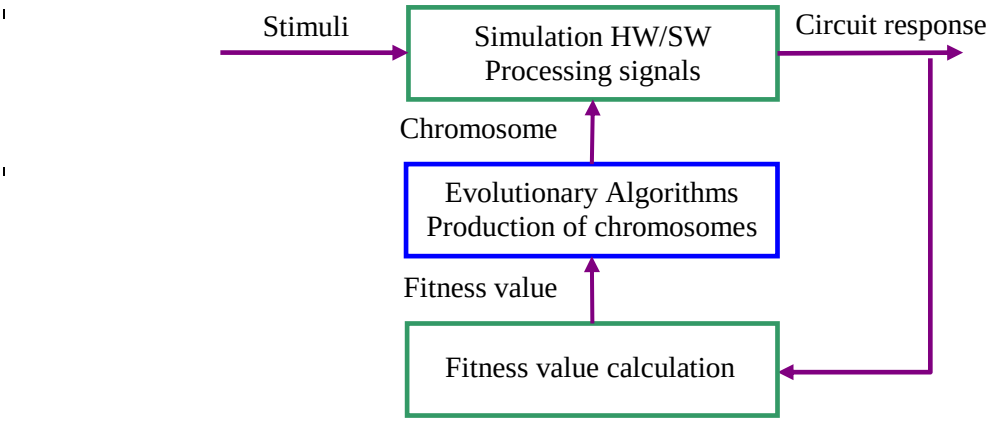


Figure 1-1. The conceptual scheme of functioning of EHW.

1.2 Motivations, Aims, Objectives and the Scope of the Work

The following four events before the start of this research mostly contributed to inspire, stimulate and drive the work presented in this thesis:

1. In 1996, Adrian Thompson evolved an analogue circuit that utilized ten times fewer components than the conventionally designed circuits required to solve the same task (voice discriminator) [38].

2. In 1999, Hugo de Garis claimed that the “killer application” of EHW is urgently needed and would soon appear [3].

3. In June 2001, Dr. Steve Zornetzer, Director of NASA Ames Information Systems and Technology, made a strong statement in a speech at the Second NASA/DoD Workshop on EHW: "Maybe NASA's future will depend on Evolvable Hardware" [73]. At the same workshop, Nikzad Toomarian from JPL stated that “EHW is needed for deep space exploration in extreme environments” [73]. For example, Pluto express - with a flight time of 8-9 years - and other interstellar missions need to emphasize long-term survivability. Moreover, JPL needs future space systems based on EHW that can adapt in seconds and survive for over 100 years with low power and high intelligence [73].

4. In the 2003 issue of the IEEE Intelligent Systems journal, EA were regarded as one of the major achievements of AI in the 21st century [88].

The first event may easily inspire someone to devote himself/herself to EHW, because it seeds the hope that you have a chance to take part in the next technological revolution. It comes not just from the fact of the evolution of analogue circuits based on digital-purposed FPGA, but due to the fact that the evolved frequency discriminator is the first analogue circuit from the application domain that is “problematic for conventional design” [38]. It also indicates the direction of the research which should be done:

- Evolution of analogue circuits,
- Unconstrained open-ended evolution,
- Fine-grained or component level evolution.

As a result, you should get super-compact untrustworthy analogue circuits with unconventional designs.

The second event inspires in two ways. On one hand, the idea of EHW is met by high demand from the commercial market. On the other hand, EHW researchers have the appropriate tools and know-how that enables the bringing of the technology of EHW to a wide application very soon.

The third event raises similar thoughts as the previous events, together with additional details on the direction of the research. It confirms that analogue open-ended unconstrained EHW has significant potential to result in the real world devices anticipated by NASA.

Finally, the fourth event insures that EA-based techniques - including EHW - are highly likely to become widely needed by humans.

Thus, in brief, the main motivation for this work is *the open-ended unconstrained evolution of commercial application oriented unconventional analogue circuits.*

From the literature review, EHW is alluring because it is able to create designs that may outperform man-made circuits [92], and this fact definitely is an additional motive for exploring EHW. EHW has three competitive advantages over the conventional design approach:

- First of all, evolution is able to search a larger solution space for designs that better utilize the available resources than could be found by a human designer. EHW is able to synthesise circuits that are not constrained by human design

methods. For example, in [92] the range of the post-2000 patented circuits was reinvented with the help of GP and the evolved circuits significantly outperformed the conventionally designed ones.

- EHW is able to produce circuits that cannot be fully specified in advance, but where the required behaviour is known. Therefore, EHW suggests an opportunity of synthesizing circuits that are able to adapt to circumstances that cannot be foreseen. This feature is widely exploited in evolutionary robotics [60].

- Since parallelism and asynchronism are inherent features of analogue unconstrained EHW, it enables the full exploitation of the physical medium, ignoring constraints introduced by the human designer that prevent the physical features of the hardware from being exploited. For instance, in [38] evolution exploited the analogue properties of a digital device (FPGA).

Inspired and motivated by the above, intuition suggests that there is need for work that will explore the potential capabilities of unconstrained analogue EHW that will answer the question: *how far could the system reach in the evolution of complex unconventional circuits?* Now the overall aim of the thesis can be stated as:

To develop an evolutionary system for the design of analogue circuits and to challenge it with complex tasks that are problematic for conventional design.

The meaning of "problematic for conventional design" is the same as in [38] claimed by A. Thompson. The term was not defined clearly, but the circuit related to this notion was described as follows:

"The task was intended as a first step into the domains of pattern recognition and signal processing… Such a circuit could be used to demodulate frequency-modulated binary data received over a telephone line… Conventional design would require 1-2 orders of magnitude more silicon area to achieve the same performance, and even then it would be difficult."

From this quotation it can be clearly understood that the problematic area for human designed analogue circuits is that:

- regarded as difficult to design, i.e. the design method is not foreseen in advance by an expert with substantial experience and who claims that the design of the circuit with purely analogue parts is meaningless due to the enormous efforts and time required;

- but, at the same time, the purposed circuit function could be designed with the help of digital logic.

The last term uncovers the reason why such analogue circuits are needed: it may replace and make redundant the number of digital circuits, such as those that are too bulky (clocks, oscillators), or too power consuming (ADC/DAC) or too expensive (microprocessors, microcontrollers), etc., in up-to-date commercial devices. This concept is in common with System-on-a-chip (SOC) in digital electronics in that like digital SOC it integrates all the components, necessary circuits and parts for a proposed electronic system into a single integrated circuit (IC) [59], but processes purely analogue signals.

Being problematic for design, the application domain of such unconventional circuits is also unconventional and - thus - such circuits can be called *analogue circuits for unconventional applications*.

In targeting the overall aim, the following intermediate objectives have been set up:

1. The evolved LCR circuits by the proposed system should exceed the previously evolved LCR circuits by component economy and functionality with fewer computer resources spent,

2. The evolved LCRQQ[1] circuits of the proposed system should exceed the previously evolved LCR circuits by component economy and functionality with fewer computer resources spent,

3. The proposed system should be able to successfully evolve complex *analogue circuits for unconventional applications*.

Thus, the scope of the thesis can now be clearly stated: the work should be focused on the system that is capable of reaching the three objectives listed above. That is, the (framework) system should be created from scratch (Chapter 3), where motivating features - like unconstrained evolution, analogue fined-grained evolution, etc. (Section 2.3) - should be established, along with the basic system parts, such as representation, mutation, evaluation, etc. (Section 2.2).

It was encouraging and exciting on the one hand (but naive on the other), to think that the framework system - even unconstrained to the maximum - would be able to tackle all the objectives simultaneously. However, and being designed, the LCR-focused and then the LCRQQ-focused system (described in Chapter 3) were able to reach only the first objective while only approaching a successful solution to the second one (Experiment 8). Therefore, it was necessary at the second stage of the research (Chapter 4) to explore some known enhancing techniques (Section 2.4) and novel methods that became feasible during the first stage's development (Sections 3.5 and 3.9). However, the author wonders if these techniques, even with novel methods, have failed to reach the third objective (Section 5.2.4). Therefore, in Chapter 5 the level of the novelty of the additional techniques has been significantly increased. As a result, the parallel incremental technique was developed and utilized. After that, all of the objectives - as well as the overall aim - were reached.

Within each version of the system, some local system developments were described, where each sub-version was tested again, and if the experiment results were not

[1] LCRQQ is a circuit composed of inductors, capacitors, resistors and two types of transistors.

satisfactory or the current version was unable to tackle the problem, new local developments took place.

The further analysis of works of Thompson et al. [38], [51], [55], [82], [29], [36], [93] and the JPL research group [28], [39], [40], [56], [59], [71], [77], [94], [73], [149], [165] gives the following milestones for the proposed research:

- The evolved unconventional and untrustworthy circuits could be robust [55], [28], [59];

- The untrustworthiness of the evolved circuits is not a problem for their exploitation;

- Evolution may prefer another environment than that of FPGA [165], because another substrate could suggest more freedom of interconnections, and thus more space of unconventional solutions.

The assessment criteria for the success of the research undertaken demand that two main values are established that estimate the performance of the evolutionary system and the resulting design:

1. The relative comparison of computational efforts spent (number of individuals evaluated) in the current approach and the ones undertaken before;

2. The relative comparison of the number of components in the evolved solution and the solutions evolved or designed by some way previously.

1.3 Thesis Achievements & Contributions

The main contribution of this dissertation is *the discovery of the fact that with the help of evolution it is possible to design analogue circuits with applications that lie beyond the conventional application domain of analogue circuits.*

The following features of the developed system are the accompanying achievements and contributions to the area of research:

1. The ability of the developed evolutionary system to simultaneous synthesise the topology and parameters of the large analogue circuits. In this work, the largest circuits developed reach 87 and 138 components after pruning; to the author's knowledge these circuits have the greatest number of functional components in the area of EHW.

2. The ability to develop an evolutionary system to synthesise circuits with a multitude of inputs/outputs. Circuits with 9 and 5 pins are synthesized.

3. Contrary to doubts about Evolutionary Strategy (ES) as a method for topologically open-ended analogue circuit design [23], this work has shown that ES, along with GA and GP, could be regarded as an effective tool in the domain of analogue circuit synthesis. This fact has been proven with every example described in the work. The evolvability of different functions has been investigated and several techniques were proposed that significantly improved the scalability of the existing ES-based evolutionary system. The comparison of the results with ones produced by GP and GA shows the fast speed of evolution, the economy in the components of the resulting solutions and the highest precision of the functions performed by the evolved circuits.

4. The novel automatic incremental evolution technique for analogue circuits has been proposed for the synthesis of analogue multi- input/output large scale circuits.

5. Based on the state-of-the-art concept *virtual mutation*, the novel adaptive individual-level mutation technique called the *differentiated mutation technique* has been proposed, the application of which significantly improves the speed and convergence of the designs targeted.

6. The new ES-based parallel island-model strategy called Winner-Dominates-Winner-Cooperates (WDWC) has been proposed. Together with the *differentiated mutation technique*, the new approach composes a novel Very Narrow Focused Evolution (VNFE) with extremely small selection rates. The VNFE is able to synthesize large scale circuits, demonstrating the sustainability, scalability and reliability of the system for a sustainable evolutionary search.

7. Due to the power of the system developed in this work, the scope of potential solutions to be discovered by analogue EHW is considerably widened. This enables the synthesis of circuits which have applications that may lie outside of the conventional application domain of analogue circuits. For example, as has been shown, the system enables the design of analogue circuits with functions that are performed by digital circuits in commercial applications.

1.4 Organization of the Dissertation

The rest of the dissertation is organized into six chapters.

Chapter 2 describes the background of the field of EHW and EA, emphasizing the current issues and techniques available to tackle these issues. The crossroads in the field of evolutionary analogue circuit synthesis are shown with the analysis of choices preferred. The chapter suggests discussions on different options to choose from, if one builds an evolutionary system, such as: *untrustworthiness* vs. *untrustworthy*, *digital* vs. *analogue*, etc.

Chapter 3 is devoted to the specification of all the details of the actual implementation of the framework system with the details of each mutation procedure

and other genetic operators. The chapter consists of 13 sections that follow the historical development of the system from the basic ideas behind EHW to the techniques that sharpened the system to the level at which the synthesis of the *initial level of complexity* of circuits with excellent functionality and economic resources becomes feasible. This chapter experimentally presents the answers to some basic questions, such as what is the potential of *constrained* or *unconstrained* evolution, and what kind of varying strategy is better: *incremental length genotypes* (ILG) or *oscillating length genotypes* (OLG)? Each version of the system has been challenged by the design problem. A comparison of each solution evolved with those previously made by others is given in order to define which future direction to follow. The chapter ends with an encounter with a problem during an experiment with the evolution of the *cube root circuit*. The narration steps forward to the next chapter, continuing to follow the further development of the proposed system.

The target of Chapter 4 is to describe the essence of the *differentiated mutation* approach in evolutionary analogue circuit synthesis. At first it introduces the Substructure Reuse Mutation (SRM) technique with testing experiments. The mutation procedure in this chapter has been united and generalized towards the novel mutation approach in analogue circuit synthesis. The chapter finishes with the evolution of circuits that significantly exceed the ones evolved before by functionality, component economy and computing efforts spent. In the last experiment of the chapter, the developed system could evolve the circuit with five input/output pins that has never been done before in the area of the automatic synthesis of analogue circuits. This circuit is problematic for conventional design and belongs to the unconventional application domain.

Moving towards increasing the complexity of the evolutionary targets, further upgrades of the evolutionary system are required. In Chapter 5, the incremental technique for analogue circuits is proposed. A new evolutionary tool is used to approach the challenging task of evolving a circuit with a total of nine pins and which belongs to the unconventional application domain. However, even with the incremental technique the developed system was unable to successfully finish the task. Therefore, in the second part of the chapter, the parallel evolution technique is applied to the second phase of the

9-pin circuit and the evolution succeeded with the circuit where the number of components reached 158 and the number of pins reached nine. In the last theoretical part of the chapter, the parallel evolution methodology is renewed with a novel strategy called Winner-Dominates-Winner-Cooperates (WDWC). Being modernized, the new system becomes very narrowly focused during evolution due to extremely small selection rates, which has been called Very Narrow Focused Evolution (VNFE). Next, the new system is tested on a new target, the analogue circuit that combines a number of digital circuits in its functionality inside the up-to-date device.

Chapter 6 summarizes and evaluates the work presented and the limitations of the approach, going on to discuss the issues and insights that have been discovered and whether or not the aims and objectives have been achieved. It briefly considers some further questions that are relevant to the proposed approach but which could not be addressed in the context of this thesis and should instead be addressed to future work. The perspectives that this research opens up and its applications are also discussed.

The diagram on the remaining page of this chapter gives a schematic overview of the thesis along with the experiments made. The diagram is meant to be used as a reference during the reading, assisting in the comprehension of the structure of the thesis, in the placing of each subject in the global context, and in the understanding of the relations between the different parts of the thesis. The rectangles with a dark background refer to the milestone versions of the developing system, whereas the white rectangles detail the particular modifications made towards the system, with the horizontal links providing either a further level of detail about particular subtopics or the corresponding experiment.

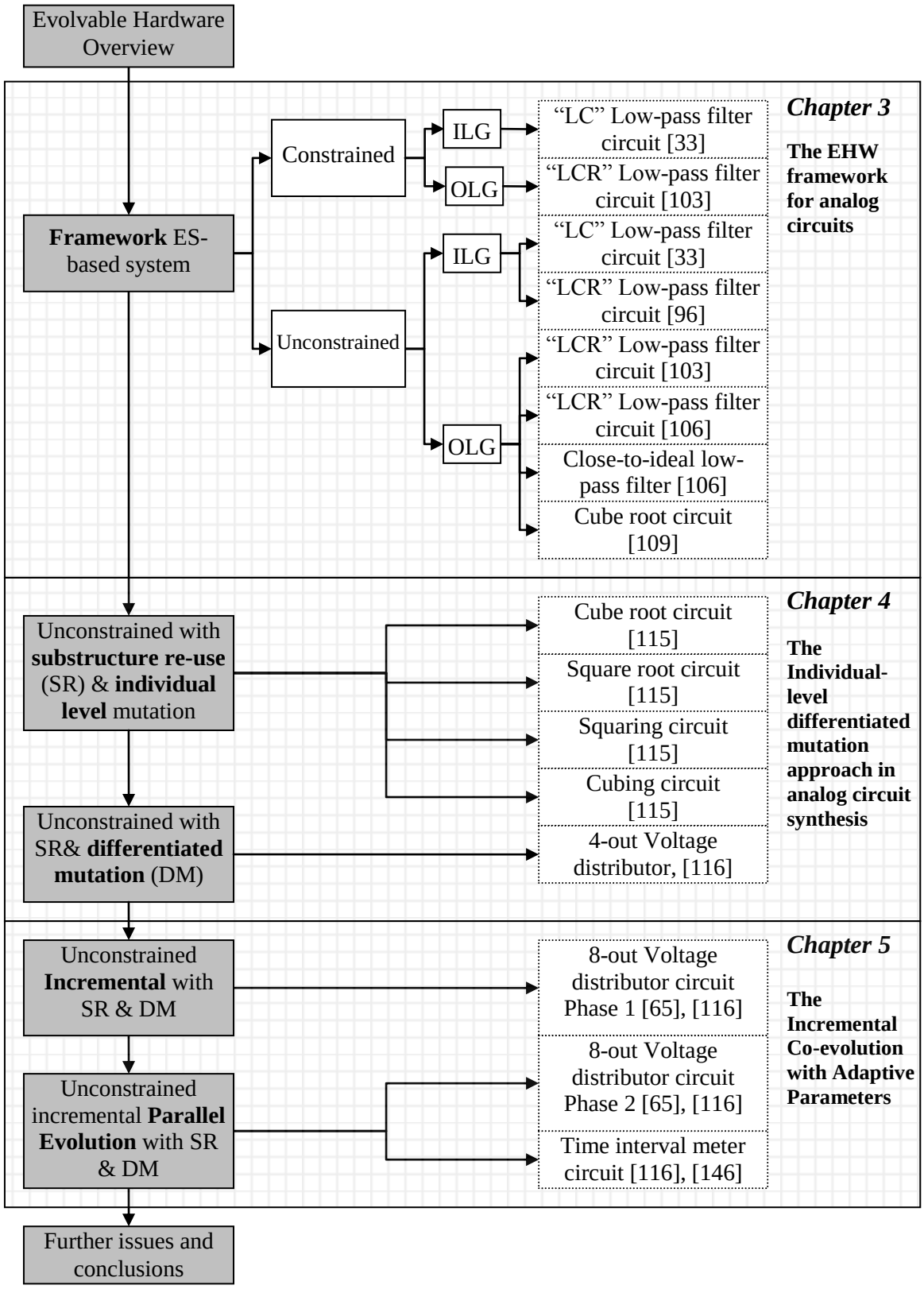


Figure 1-2. The structure of the thesis along with the circuits evolved. The techniques in bold inside the rectangles with a dark background refer to the main system modifications.

Chapter 2. Evolvable Hardware

This chapter describes evolvable hardware from its beginnings to some techniques that are necessary for the evolution of analogue electronic circuits. The chapter contains a comprehensive number of references to works particularly selected as close to analogue circuits as possible, emphasizing the techniques that are utilized in this thesis. It also outlines the milestones and most promising techniques that other researchers have developed to tackle the basic problems in evolutionary analogue circuit design.

Further, the following essential features of EA will be discussed: Representation, Mutation, Evaluation, Ranking and Selection. Then, the narration will continue with other techniques that are able to enhance the power of the core system: Parallel EA, Adaptation and Incremental Evolution. The target of the chapter is to describe and discuss the advantages and disadvantages of these techniques.

2.1 Analogue Circuit Overview

2.1.1 Analogue circuits for unconventional applications

The human ability to design analogue circuits has some limits. This is supported by Aaserud's notion from [140]:

"Analogue designers are few and far between. In contrast to digital design, most of the analogue circuits are still handcrafted by the experts or so called 'zahs' of analogue design."

Moreover:

"Analogue circuit design is known to be a knowledge-intensive, multiphase, iterative task, which usually stretches over a significant period of time and is performed by

designers with a large portfolio of skills. It is therefore considered by many to be a form of art rather than a science."

This is due to the physical nature of interactions among analogue components. For example, the patent in [108] presents the human-designed cubing CC. Intuitively, one may guess that the overall cubic curve is in some way contributed to from the inherent electrical characteristics of the transistor; more precisely, it is composed of a multitude of function fragments of circuit transistors. However, there is no evidence as to how the transistors should be organized, what kinds of network laws should be applied and how many times. Being designed conventionally, this CC has an average error of 7.1mV against 0.99mV for unconventional circuits evolved and patented in [99] and against 0.29mV evolved in this work. This example suggests that complex circuits are amenable to handcrafting, but that there is a trade-off with the precision of their performance.

Another kind of example is that of the tone discriminator evolved in [38] by Thompson. In Thompson's opinion, this circuit is a challenge for an analogue circuit designer, but not for a digital one. Furthermore, the circuits exemplified in this work - where the output signal is a complex discrete or singular function of a continuous input signal - are even less amenable to human design.

On the other hand, relatively less complex circuits - such as the low-pass filter - being conventionally designed [102], are unrivalled in terms of economy and attenuation in the stop band against every evolved low-pass filter in [12], [13], [17], [78], [33], [16], [103], [106] (Table 3-9).

In this sense, the analogue circuits that are problematic for conventional design or else being conventionally designed, but yield to those designed by any other heuristic way, including circuits designed with the help of evolution, are limited in their applications and belong to the domain of *circuits with unconventional applications*. With new terminology, this thesis now aims, overall, to: *develop an extrinsic evolutionary system for the design of analogue circuits and to challenge it with analogue circuits for unconventional applications.*

2.1.2 Area overview of complex analogue circuits

If we look at the past when analogue EHW had just started up and follow the development of analogue EHW, it is possible to distinguish four different stages. Firstly, the EA technique has been applied for sizing analogue circuits [6], [7], [175], [176], and [178]. Since the middle of the 1990s, the second stage started with the appearance of successful attempts at designing both their topology and parameters [117], [87]. For several years, an increasing number of researchers presented results where circuits had been evolved extrinsically [24], [17], [79] and intrinsically [38], [93], [121]. At that time, the questions that these works tried to answer could be generalized thus: is it possible to design analogue circuits with the help of EA and could EA create *more complex* circuits? Chronologically following the publications of research groups during this period, it is evident that the size of the reported circuits designed by each group grew each time. For example, Koza et al. in [37] (2006) evolved filters and computational circuits in [24] (2007), Zebulum et al. evolved filters and amplifiers in [28] (1998) and [149] (2000), and Lohn et al. evolved the same circuits in [13] (1999) and in [73] (2001).

The beginning of the third stage began with the question: is it possible to design industry-feasible analogue circuits with the help of EA? There appeared papers about robust and industry-feasible analogue circuits [93], [55]. Since then, many researchers in the area of analogue circuit evolution turned from the first stage to the second [57], [53], [58], [55], [54], [45].

In this sense, the current research, being inspired by works from the second stage, continues to keep answering the question: can EA create *larger* circuits? It is a theoretical exploration of the capabilities of evolution that could create complex useful structures composed of numerous analogue components.

The interest in larger circuits is based on the intuitive assumption that a more complex function requires more components and, vice versa, that more components in a circuit may determine a more complex function. The proof for the assumption requires a definition of the level of the complexity of the function performed by a circuit.

It is evident that larger handcrafted analogue circuits require more effort in design, involving the application of network laws (Ohm's, Kirchhoff's, Norton's, etc.) and calculations [1]. For conventional circuits - such as filters, amplifiers, switches, etc. - where every component's contribution is definite, the level of their complexity could be (without doubt) defined as proportional to the effort spent in design and thus to the number of components. For unconventional circuits, this may become true if every component effectively contributes to the overall circuit functioning as it happens in a conventional circuit, where the design is made component by component. There are a lot of examples where the evolutionary designed circuits exceed the conventional one in terms of component economy and functionality (e.g. [38], [92]) proving the high level of involvement per component. Thus, it is appropriate to claim that the above assumption - that more components in a circuit determine a more complex function - also suits evolved circuits.

The relation between complexity and involvement had been mentioned by John von Neumann [181]. He did not give a formal definition of *complexity,* but rather tried intuitively to explain this concept:

"This concept clearly belongs to the subject of information, and quasi-thermodynamical considerations are relevant to it. I know no adequate name for it, but it is best described by calling it "complication." It is affectivity in complication, or the potential to do things. I am not thinking about how involved the object is, but how involved its purposive operations are. In this sense, an object is of the highest degree of complexity if it can do very difficult and involved things."

Therefore, to allege that the function, performed by N components in a circuit, is *significantly* less complex than is claimed will be possible only if one provides another circuit composed of M components and performing the same function with the same precision, but where M is *significantly* less than N. Thus, when referring the successfully evolved circuits presented in this thesis (after pruning) as well as in other works, we make the *a priori* assumption that the involvement of every component is purposive and that the "circuit complexity" or "circuit size" is directly proportional to the "function complexity" that this circuit performs, while it is not refuted by direct comparison.

Based on this idea, the selection of the foremost related literature is made purely upon the size of the circuits evolved. Table 2-1 represents the research groups against the maximum sizes of analogue circuits evolved, in descending order.

On the other hand, based on the overall aim of the thesis, the interest covers those circuits that are of *unconventional applications*. The search does not concern those purely analogue circuits such as oscillators, amplifiers, dc-converters, etc., as well as those analogue circuits that have the same definition in a digital domain, like digital/analogue filters, digital/analogue controllers, etc. This limits the search to the most relevant works of Thompson [38] and Koza [24] (the last column of Table 2-1). Other works have neglected to evolve this category of circuits.

Table 2-1. Developers in Evolution of Analogue Circuits: Unconventional Applications

Researcher	**Pruning**	**Type of evolution**	**Circuit size**	**Circuit name**	**Attempts to evolve circuits for unconventional applications**
Thompson [38]	After pruning	Intrinsic	20 cells	Tone discriminator	1: tone discriminator
Koza et al. [24]	After pruning	Extrinsic	64	Square root	2: NAND, two-instruction arithmetic unit
Mattiussi et al. [43]	After pruning	Extrinsic	55	Temperat. sensing	No
McConaghy et al. [44]	After pruning	Extrinsic	48	Amplifier	No
Sripramong et al. [83]	After pruning	Extrinsic	41	Amplifier	No
Shibata et al. [84]	Before pruning	Intrinsic	36	Absolute function	No
Trefzer [148]	After pruning	Intrinsic	34	Amplifier	No
Layzel [93]	After pruning	Intrinsic	33	Oscillator	No
He et al. [170]	After pruning	Extrinsic	28	Amplifier	No
Hu et al. [45]	After pruning	Extrinsic	26	Low-pass filter	No
Lohn et al. [59]	After pruning	Extrinsic	23	Low-pass filter	No
Kruiscamp et al. [87]	After pruning	Extrinsic	22	Amplifier	No
Ando et al. [78]	Before pruning	Extrinsic	22	Low-pass filter	No
Zebulum et al. [77]	After pruning	Extrinsic	19	Amplifier	No
Xia et al. [85]	After pruning	Extrinsic	18	Low-pass filter	No
Dastidar, et al. [86]	After pruning	Extrinsic	18	Amplifier	No
Chang et al. [81]	After pruning	Extrinsic	17	Low-pass filter	No
Langeheine et al. [121]	After pruning	Intrinsic	15	Amplifier	No
Das et al. [89]	After pruning	Extrinsic	15	Low-pass filter	No
Ohe et al. [171]	After pruning	Extrinsic	15	Amplifier	No

Kim et al. [54]	After pruning	Extrinsic	14	Low-pass filter	No
Conca et al. [177]	After pruning	Extrinsic	14	Low-pass filter	No
Yuan et al. [173]	After pruning	Extrinsic	14	Amplifier	No
Wang et al. [47]	After pruning	Extrinsic	13	Amplifier	No
Goh et al. [17]	After pruning	Extrinsic	12	Low-pass filter	No
Sabat et al. [172]	After pruning	Extrinsic	10	Amplifier	No
Grimbleby [79]	After pruning	Extrinsic	10	Low-pass filter	No

It is supposed that the number of pruned components is at least 5% of the total number of components.
This table does not include some intrinsic approaches (i.e. Stoica [40]), where the count of the components of the resulting circuits has not been made. It also does not present circuits that contain less than 10 components.

The significance of Thompson's work [38] in relation to this thesis has already been mentioned. He was the first who, with the help of *intrinsic* evolution, evolved an unconstrained analogue circuit that was problematic for conventional analogue design, instead belonging to the domain of digital circuit design. The frequency discriminator - that distinguished between 1kHz and 100kHz frequencies by supplying a steady voltage to an output - was evolved by involving 20 FPGA cells. FPGA worked in analogue mode and thus the analogue behaviours of analogue components composed of the cells have been exploited in circuit functioning. Attempts to analyse the work of the circuit have failed, but Thompson declared that the trustworthiness of a circuit is not obligatory for the exploitation of it, at least in principle [38]. The work has contributed to the current research and the important idea that unconstrained evolution which designs unconventional circuits could be applied towards targets that belong not only to the domain of analogue application domain but also to the domain of digital application. The circuit was not robust to component variations and temperatures. In 2000, he published work concerning the evolution of a robust temperature circuit [158]. In this paper he answered the question of the robustness of unconstrained circuits, where the detailed model of how the behaviour of each component affects the circuit function does not exist. The initially designed unrobust circuit from [38] - the frequency discriminator - became robust after evolution. Unfortunately, Thompson did not continue his work in the same direction and in 2002 he switched to adaptive computing [180].

As can be seen from Table 2-1, the largest circuit evolved after pruning was made by Koza et al. [24] in 1997. One of the reasons why the 14-year old work is still on top is

the powerful computing support, multi-cluster system that enabled them to operate the largest population (640,000) and the highest number of individual evaluations (6,700,000,000) in the EHW domain [98].

The works of Koza [24], [12], [21], [46], [58], [63] are the most close in spirit to the work of this thesis, and many of the problems and results described will be used as benchmarks and references for the results obtained. After the discovery of Genetic Programming [21] in 1992, he began the evolution of analogue circuits from low pass filters in 1995 [12]. In 1997, he achieved the largest circuits in EHW [24]. His circuits have been the most distinguished in terms of their size and functions ever since. The milestone work appeared in 2000, where he evolved the analogue circuits that perform digital functions: the NAND circuit and a *two-instruction arithmetic logic unit circuit* [52]. This was the first attempt with the help of *extrinsic* evolution to enter the digital circuit domain. However, the evolved circuits were modest in size (6 and 26 components) and took considerable computing effort (2.2mln and 43.6mln evaluations respectively). Unfortunately, since then Koza *et al.* did not make any further attempts in the evolution of circuits for unconventional applications. In 2004, Koza *et al.* published the first work on industrial-strength circuits utilizing a multi-objective approach, where they evolved an operational amplifier with 31 components and with considerable computer effort (7.7E+7) and which almost fitted the multi-objective function [58].

2.2 Evolutionary Algorithms

According to [19], the art of EA consists in determining how an evolutionary algorithm can be tuned to a problem while solving it. It is suggested that the following ways are essential for significantly increasing the efficiency of the algorithm:

- Adapting and tuning the algorithm to the specific problem, i.e. “the problem-specific knowledge incorporated into the system enhances an algorithm's performance and narrows its applicability;”

- The parallel implementation of evolutionary algorithms.

These points are referenced since each of them plays a significant role in the construction of the evolutionary system, and the point made in the quotation above has directed the shifting of most of the details on the system from Chapter 2 to Chapter 3.

In the following sections, an introduction to the main operators in Evolutionary Strategy is given with a presentation of some different methods. The references to the literature can be found to help in the understanding of each concept. While the details of an approach are given in Chapter 3, only short comments on what kind of operator has been utilized in the thesis and why it was utilized are presented.

2.2.1 Representation

In general, there are two approaches for *representations*. The first one is to choose any of the standard algorithms and to design a decoding function according to the requirements of this algorithm [26]. The second approach is to design the representation as close as possible to the characteristics of the *phenotype*. While the *direct encoding* scheme - where all connection parameters are individually encoded - saves computational time, the indirect encoding scheme - where the details of the architecture, such as the connections - are left to a training scheme or to *developmental rules*, allowing for a more compact representation of the circuit [123].

In his 1992 PhD thesis, De Garis [25] predicted that the combination of evolution and *development* would be applied to electronic circuits, and he called this field *Artificial Embryology* (AE). A lot of publication appeared over last decade on the application of *development* towards circuit synthesis [123], [43], and [124]. This approach - and foremost among its various properties - has an ability to achieve greater scalability [123]. However, this thesis does not focus on *development* technique, concentrating more on other kinds of tools. Thus, *direct coding* has been utilized in this work.

The second aspect that has to be addressed in the frame of the *representation* is what kind of behaviour the chromosome length should have during evolution: *dynamic* or *static*? The *static* representation requires the knowledge of a component number in the

targeted circuit [83], [15], [87], which is impossible in most of the application cases. Therefore, a *dynamic* or *variable* genotype strategy is set as basic for the proposed system.

In [143] a novel approach with a special type of direct encoding for digital circuits was developed, called *Cartesian Genetic Programming* (CGP). CGP represents circuits as a two dimensional grid of program primitives. Each node is encoded by a number of genes representing a particular function and its inputs. Each node takes its inputs in a feed forward manner from either the output of a previous node or from one of the circuit inputs. The limit of CGP is in the difference between CGP and linear GP, i.e. the restrictions that CGP imposes on some interconnectivity, namely, the feed-forward connectivity of their directed acyclic graphs.

In this work, direct representation has been utilized (Chapter 3), similar to that used by Zebulum et al. in [149], [28], [77]. In this approach, the minimal genetic code is a locus which is coded for a component's parameter, pin number and component name. Four loci compose a gene that represents a component. The circuit is composed of components and is represented by a list of genes inside a chromosome. The main reason for applying this kind of coding is the simplicity of the approach.

2.2.2 Bloat

Evolutionary Algorithms that use variable length representations suffer from bloat. Bloat occurs when the average genome size tends to grow as the evolution progresses. The main side effect is that progress toward a solution slows dawn and the probability of evolution failure increases.

Koza [21] used the following techniques to fight bloat: a) the individuals that are created for the initial populations are restricted to depths between 2 and 6; b) sub-trees that are generated for sub-tree mutation are limited to a depth of 4 and non-terminal nodes are selected as mutation points with a probability of 90%; c) the creation of new individuals is limited to a depth of 17.

Another often used technique to slow bloat is the use of parsimony pressure. In this case, the fitness function of an individual is a combination of its performance and its size, where a smaller size increases the fitness value [126]. In [28] Zebulum et al. explored *different genotype length varying strategies.* As a result, they declared that a successful strategy is that which suggests evolution - besides genotype increase - of the pathways from large to smaller genotypes. Namely, the best strategy was that of *Oscillating Length Genotype* (OLG), according to which the genotypes are enabled to decrease their lengths during evolution.

Code editing is another way of reducing the size of individuals. In this approach, parts of individuals that have no effect on the result of the individual - called introns - are removed from the individual [126]. However, there are strong opinions that suggest that introns are not responsible for bloat [128], [127].

Finally, it is possible to introduce a "Delete Element Mutation" (DEM). Using this approach, individuals during the mutation stage may get their genotype reduced as in [125], [43], [80]. The DEM procedure is one such approach, where the randomly chosen component is deleted from a circuit and the abandoned connections are randomly connected to each other or the circuit nodes. This technique is described in detail in Chapter 3, with comparative testing of the example of low-pass filters. It will become an essential part of the OLG varying strategy. Table 2-2 displays some previous approaches in analogue circuit synthesis, where the third column shows whether or not the DEM is used.

Table 2-2. Statistics of a type of EA and the "Delete Component Mutation" used by others

Researcher	**EA type**	**DEM**
Koza et al. [12]	GP	No
Lohn, et al. [13]	GA	No
Goh, et al. [17]	GA	No
Zebulum, et al. [28]	GP,GA	Yes
Grimbleby [18]	GA	No
Dastidar, et al. [86]	GA	Yes
Sripramong et al. [15]	GP	No

Walker et al. [49]	GA	No
Chang, et al. [81]	GP	No
Mattiussi, et al. [43]	GA	Yes
Trefzer [148]	GA	Yes
Gan, Yang et al. [80]	Gene Expression Programming	Yes
McConaghy et al. [53]	GP	No
Kim et al. [54]	ES	No
Das, et al. [89]	GA	No

2.2.3 Mutation (Modification)

As is already understood, ES is a type of EA that does not utilize recombination or crossover operation during modification. Thus, the term 'mutation' is further used to describe any modification of some number of gene loci according to the mutation rate that involves the genotype of a single chromosome.

The mutation parameter has been suggested as being the most sensitive parameter in the theory of EA [34], as well as when it concerns a digital electronic circuit synthesis [35]. Harvey, in [36], argued with the example of SAGA that mutation is a driving force of the natural evolution of species. It has been noticed that different values of the mutation rate are desired at different stages of the evolutionary process in order to achieve balance between global and local searches. For example, in [19] it is proposed that the art of EA consists of how the algorithm is adapted to the problem, especially the mutation operator. Thus, there is no conventional paradigm as to what the optimum mutation parameter or mutation operator should be, and a number of publications have addressed this problem [129], [130], [131].

In Chapter 3, the static mutation is used for the framework system. This helped in the estimation of how powerful the unconstrained evolution is in comparison with other approaches in the design of low-pass filters. Next, in Chapter 4, the mutation operator is developed according to a novel mutation strategy so as to tackle the stalling effect problem.

2.2.4 Evaluation-Ranking-Selection

A common thread in EA is the *evaluation-ranking-selection* mechanism. This process determines which individuals of the population are selected to be members of the next generation. It is done according to how good or bad a solution is. The better it is, the higher the probability of its survival, and so it has a higher probability of being selected for the next generation [132].

Primarily, individuals have to get through the *evaluation* procedure, where the main target is the fitness assigning. The *fitness function* (FF) contains some of objectives with *weights* that are applied toward the individuals. The FF calculates how close an individual is in terms of its features to the objectives set [133].

As is usual, with the fitness values assigned each chromosome is ranked by the *ranking* operation. The last one can also operate selectively, based on the additional number of objectives [132].

In [150], Bentley et al. proposed the classification of ranking methods for multi-objective problems based on the criteria of whether the method is range-dependent or not, where the “range” is a practical interval between the maximum and minimum fitness values. Being independent from different fitness ranges that refer to different objectives, no objective is directly compared to another. Furthermore, the range-independent ranking does not require specific knowledge of the nature of the problem.

The final operation of *selection* chooses the individuals from the *ranking list* to the next generation. That is, the fittest chromosomes are not guaranteed to be top-ranked and the top-ranked members are not guaranteed to be selected for further evolution. For example, there are many methods for selecting individuals for survival [132]. These methods include: (1) *proportional selection*, or a “roulette-wheel” selection scheme, where the probability of selection is proportional to the individual's fitness; (2) *ranking-based* methods, where all the individuals in a population are sorted from the best to the worst and the probabilities of their selection are fixed for the whole evolution process; and (3) *tournament selection*, which involves running several "tournaments" among a

few individuals chosen at random from the population, where selection pressure is easily adjusted by changing the tournament size - if the tournament size is larger, weak individuals have a smaller chance of being selected [133].

The *evaluation-ranking-selection* procedure is a suitable stage within EA where one could apply multi-objective pressure towards evolution [32]. In the case of a single objective, it is obvious: the better the fitness value that the individual has, the better the solution is. In the multi-objective case, while searching for solutions and when attempting to improve one further, other objectives may suffer as a result. In the case of multi-objective evolution, three approaches are conventionally applied. The first - and the simplest - one is that of "aggregating functions," because it combines all the objectives of the problem into a single one with weighting coefficients representing the relative importance of each objective [149]. Another one is that where a population is used to diversify the search performing proportional selection of sub-populations against each objective [148]. And the final technique involves the simultaneous optimisation of a set of objective functions with the use of Pareto optimality [31], [134] [135].

The last one became quite popular in recent decades. A tentative solution is called non-dominated or Pareto optimal if it cannot be eliminated from consideration by replacing it with another solution which improves an objective without worsening another one. Currently, most multi-objective evolutionary systems apply Pareto-based ranking schemes [134], [135], [31]. In [31] they used multi-objective EA for analogue circuit design in order to perform the optimal sizing of two mixed-mode circuits. As objectives they set 12 functions, defining the parameters of the components to be optimized.

In the frame of the current work, during *ranking* the additional objective of "component economy pressure" is suggested to be applied. In the framework system, a single-objective fitness function and ranking is utilized for the sake of unconstrained approach evaluation. However, the rest of the system's sub-versions utilized a double-objective evolution, where the second objective is the component parsimony. The "range-dependant" approach [150] with dynamic "weight" has been applied to combat the bloat.

2.3 Evolvable Hardware

In the following section, the main features that accompany the analogue EHW will be discussed. These terms are the essential attributes of any EHW. The following discussion is important since one has to make a choice among the options that will define the overall purpose of the system one is constructing. The brief explanations are given of the choices that are used during this work.

2.3.1 Intrinsic vs. extrinsic EHW

Extrinsic evolution is a case where individuals are evaluated by simulation software. For the analysis of analogue circuits, almost every existing simulation software is based on the freely distributed SPICE, which was developed at the Electronics Research Laboratory of the University of California, Berkley [144]. One of the most advanced commercial versions of SPICE is PSPICE, which is currently under development at Cadence Design Systems [91]. PSPICE was the first version of UC Berkeley SPICE available on the PC, and was released in January 1984 to run on the original IBM PC. *Intrinsic evolution* uses the real hardware, including the programmable integrated circuits Field Programmable Gate Array (FPGA), the Complex Programmable Logic Device (CPLD), the Field Programmable Transistor Array (FPTA), and the Field Programmable Analogue Array (FPAA) (Figure 2-1). It should be noticed that Thompson [38] was the first to perform what is called "intrinsic" (in-chip) evolution, with very attractive if controversial results.

However, when one is about to choose between *intrinsic*/*extrinsic*, first of all, it should be noticed that simulation SW is much more *accessible* than the specialized HW. Some types of hardware to which the EHW system is tuned may have been stopped being produced anymore (i.e. Xilinx FPGA XC6200). With software, such problems do not look as feasible. The total extrinsic EHW system will cost nothing for those who utilize SPICE SW since is available for free in the Internet (excluding the cost of the PC). The depreciation of software is miserable in comparison with HW. For example, the last version of SPICE3 - written in 1989 - is still actual and widely used; while a

programmable circuit may become obsolete after an illegal connection is made. Furthermore, HW may not enable some arbitrary connections between components and may even limit the granularity of an evolution, enabling only blocks of the circuit to be as the smallest components of the evolution (course-grained evolution). This leads to potential unconventional solutions being uncovered. The use of software simulation is safe because unusable or poor designs may damage the hardware system. A SW-based system is also more analytical, since the simulation can be customized so as to provide feedback to the designer about any aspect of the state of the evolutionary process. In addition - for circuit design - any internal node or state information can be extracted from the simulation and incorporated into the fitness evaluations. Finally, the last advantage of the extrinsic method is that it easily enables us to struggle against the internal variations of a circuit; while HW does not seem amenable to the simulation of component variations, in PSICE [91] - for example - there is number of automatic tools for doing this.

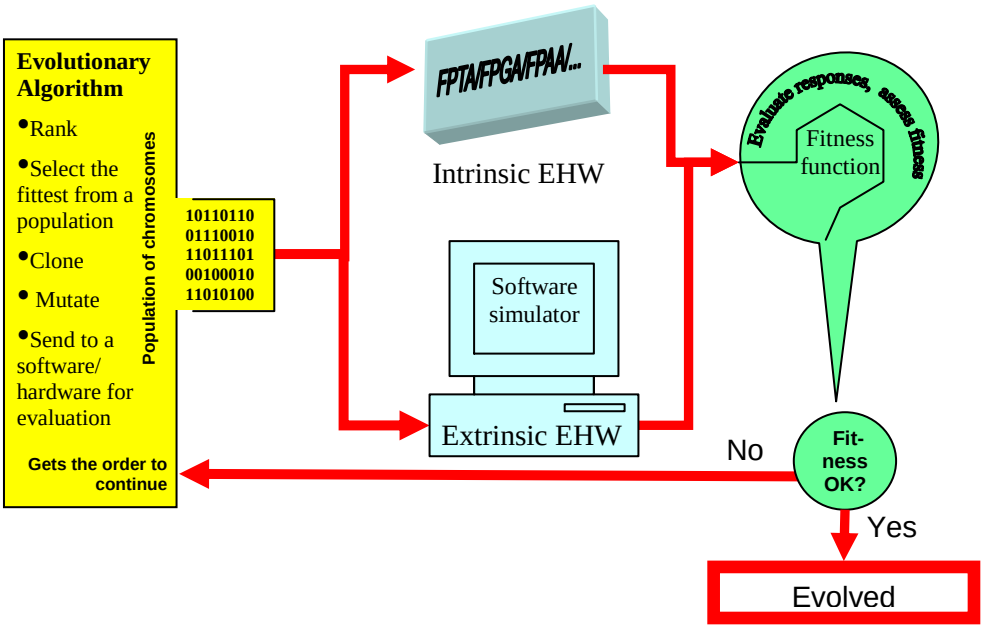


Figure 2-1. The principle difference between intrinsic and extrinsic EHW approaches.

On the other hand, the speed of extrinsic evaluation yields to its opponent to a significant degree. During experiments, Keymeulen et al. [39] confirmed that the time needed to evaluate one individual using the reconfigurable hardware (6.75 ms) is twenty

to fifty times faster than the SPICE simulator running on one processor. In earlier work [40] they proposed *mixtrinsic EHW*, where evolution takes place with hybrid populations in which some individuals are evaluated intrinsically and some extrinsically, within the same generation or in consecutive one. They developed their own chip based on FPTA. The amazing thing is that in order to find the efficient structural cell for the chip they used evolution, i.e. the evolution "taught" what the EHW prefers.

Furthermore, the most attractive advantage of using HW for circuit design is the fact that the resulted circuits are much closer to the commercial market then the ones represented in SW, for which there are a range of quite tedious procedures, like board design, layout, routing, etc.

This work utilizes the extrinsic evolution mainly due to its accessibility. Moreover, this valuable part of the thesis is due to the use of extrinsic method, which enabled many discoveries and novelties during the exploration and research.

2.3.2 Digital vs. analogue EHW

It has already been mentioned that the process of the design of analogue circuits constantly deals with continuous electric signals among conductors and semiconductors. This fact imposes fewer constraints on analogue circuit evolution and allows more interactions among components and - thus - has greater evolutionary potential than the evolution of digital circuits. Furthermore, the absence of systematic analogue design methodologies increases the necessity of automatic synthesis solutions.

2.3.3 Constrained vs. unconstrained EHW

The question of *unconstraining* the evolution - firstly - has appeared in [38], [51]. The frequency discriminator had been evolved intrinsically with the help of FPGA by Thompson in [51]. However, the FPGA had instead been treated as an analogue device rather than a digital one. The clock signal had been released and the connections among the cells and transistors had been freed up to create unconventional structures. As a

result, the fine-grained evolution at the component level had evolved a circuit that was characterized by a small number of components and an unconventional structure. The circuit was able to solve the task with an amount of components that was ten times lower than a conventionally designed circuit required for a solution to the same task. By realizing constraints, Thompson [51] had enabled an open-ended evolution to create an untrustworthy circuit that was not liable for verification. The latter caused researchers in the field of EHW to explore the ways of coping with untrustworthiness (see Section 2.3.5).

In this work, both approaches are regarded and compared in detail in the initial part of the research. Furthermore, the exploration of methodologies that are able to discover unconventional designs with the help of *unconstrained evolution* is set as one of the prior targets. There are four kinds of constraints that are distinguished and that can be set at the mutation stage of extrinsic evolution:

1. ***Constraints №1.*** Prohibiting some kinds of connections among units inside a circuit by applying circuit-structure-checking rules. For instance, transistor connections, emitter-to-collector, base-to-constant-voltage-source, etc. [13], [44], [49]. The extrinsic approach is only meant here, because in intrinsic evolution these kinds of connections are not allowed by default.

2. ***Constraints №2.*** Constraints that are set by evolvable substances due to their inherent properties. For example, FPTA and FPGA are limited by routing channels among configurable cells [39], [51], [84].

3. ***Constraints №3.*** Constraints that are set by emulation software due to division by zero. For instance, the nodes that have no DC path to ground and loops that involve inductors and/or a voltage source [12], [17], [18], [59]

4. ***Constraints №4.*** These limit the number and types of substructures that may modify a circuit. For example, by establishing up the predefined substructure database [44], [86].

The Table 2-3 gives statistics on the constraints listed above from among the various works on analogue circuit evolution.

Table 2-3. Statistics on constraints in analogue circuit evolution

Researchers	**EA type**	**Circuit-structure-checking rules Const. No.1**	**Constrains set by hardware Const. No.2**	**Constraints set by software Const. No.3**	**Limit for sub-structures Const. No.4**
Koza et al. [12]	GP, Extrinsic	No	No	Partially	No
Thompson [38]	GA, Intrinsic	No	Yes	No	Not used
Mydlowec et al. [98]	GP, Extrinsic	No	No	Partially	No
Keymeulen [39]	GA, Intrinsic	Data n/a	Yes	No	Yes
Streeter et al. [99]	GP, Extrinsic	No	No	Partially	No
Lohn et al. [13]	GA, Extrinsic	Yes	No	Yes	Not used
Goh, Li [17]	GA, Extrinsic	Yes	No	Yes	Not used
Zebulum, et al. [28]	GP,GA, Extrinsic	Yes	No	Yes	Not used
Grimbleby [18]	GA, Extrinsic	Data n/a	No	Yes	Not used
Dastidar, et al. [86]	GA, Extrinsic	Yes	No	Yes	Yes
Ando, et al. [78]	GP,GA, Extrinsic	Yes	No	Yes	Yes
Sripramong et al. [15]	GP, Extrinsic	Yes	No	Yes	Yes
Walker et al. [49]	GA, Intrinsic	Partially	Yes	No	Not used
Chang, et al. [81]	GP, Extrinsic	Yes	No	Yes	Not used
Trefzer [148]	GA, Intrinsic	Yes	Yes	No	Not used
Mattiussi et al. [43]	GA, Extrinsic	Yes	No	Yes	No
Gan, et al. [80]	Clonal Selection, Extrinsic	Yes	No	Yes	Not used
McConaghy et al. [44]	GP, extrinsic	Yes	No	Yes	Yes
Kim et al. [54]	ES, Extrinsic	Data n/a	No	Yes	Not used
Das et al. [89]	GA, Extrinsic	Yes	No	Yes	Not used

Attempts at unconstraining the evolution of analogue circuits have been made in [12], [98] and [99] where they utilized the special procedure providing the DC path to ground from each node of a circuit by adding the giga-Ohm resistance, allowing any kind of connections among capacitors. This let them avoid the most of the "node floating" errors and reach an amount of invalid circuit graphs up to 2% [99].

It should be noticed that constraining the evolution on one hand reduces the solution space, but on the other hand it closes the pathways to unconventional solutions. The last notion is especially important for unconventional tasks that have never been regarded or approached before; for example, the designs of analogue circuits that may replace digital logic in applications that conventionally belong to a digital domain [52].

In this work, by choosing *extrinsic evolution*, Constraints №2 imposed by the hardware and described in the current section are overcome. By abolishing the *circuit-structure-checking rules* and introducing the REMP in Section 3.5, Constraints №1 are avoided. The introduction of *R-Support* elements in Section 3.7.1 helps to cope with Constraints №3. And, finally, the automatically defined *substructures* described in Section 4.1 release Constraints №4. Thus, by releasing all the constraints the approach presented in this thesis claims to be fully unconstrained.

2.3.4 Robust vs. unrobust circuits in EHW

Today the open-ended methods of evolutionary analogue circuit synthesis are questioned (e.g. in [53]) and raise an important issue, namely that in being realized in silicon are they able to create solutions that are valid enough? *Robustness* is known as the ability of a circuit to cope with the environmental, internal variations and the ability of a circuit to continue to operate despite any damage that may occur to some of its parts.

In [54], a set of experiments proved that open-ended techniques enable the design of low/high-pass filters with topology-based robustness. In [55], a frequency discriminator that was robust to wide temperature range was evolved with an open-ended GA intrinsically in FPGA. [56] describes experiments that allowed adaptive *in situ* circuit

reconfiguration in extreme temperature and radiation environments. In [57], unconstrained evolution successfully created analogue variability-tolerant CMOS circuits performing XOR and XNOR functions. The literature review on that subject allows the distinguishing of two approaches:

1. The first and traditional approach follows the paradigm whereby evolution is initially set to discover the unconventional design, and where the circuit is later tuned to improve its robustness ([28], [55], [56], [58] and [59]);

2. Another approach suggests the evolutionary system that was originally proposed for robust designs ([53], [54] and [57]).

In the current study, the first approach has been adopted, focusing on the exploration of the technique's capabilities to create novel designs and leaving the evolution of robustness for the future stage.

2.3.5 Trustworthy vs. non-trustworthy circuits in EHW

The open-ended fine grained EHW may produce circuits that are *untrustworthy* [51]. *Trustworthiness*, or design verification, of a circuit is called such the circuit's feature that enables for one to fully understand how the circuit works. That is, one is able to define the contribution of each component to a circuit's functionality [53]. The appearance of *untrustworthy circuits* during evolution is caused by the increasing role of physical characteristics that are exploited by circuit during functioning. These characteristics are not contemplated by the conventional design approach. And it seems impossible to verify all the feedback loops and electromagnetic influences among the component [82].

The non-trustworthy approach to circuit exploitation is supported in this work. Furthermore, the overall thesis target is based on the idea of the unconventional features of the analogue circuits.

In [53], McConaghy criticized the circuits made by open-ended evolution techniques with no apparent logic behind them, recommending his own trustworthy structural synthesis approach where the closed-ended methodology may suggest the limited number of the topology count (for instance 3528 topologies in [53]).

On the other side, Miller et al. [50] strongly advocate the view according to which the exploitation of the physics has the advantage: "we should not expect artificially intrinsically evolved circuits to be decomposable and understandable." They stated that the genome doesn't specify how to make a cell, let alone the organism: "Once the components are made, they obey their own physics." Thompson, who was the first to have discovered and explored the physical effects in FPGA [51], noticed that evolution should take advantage of the properties inherent in the very physics of the medium being evolved. Furthermore, he advocates that "if the most promising application domains are the ones problematic for conventional design, then the exploration of new strategies is appropriate."

In this sense and in support of the second viewpoint - it should be taken that the applications that refer to such a kind of system are "situated," in the sense that they continuously adapt themselves to constantly changing constraints. There is no solution, but only the best possible adaptation to the current set of constraints. Typically, both the evolutionary system and the constraints are evolving in response to each other. The instance of such a kind of system is Evolutionary Robotics (ER). Since ER systems have to adapt themselves to a countless number of situations within the real world environment, it is logical to suppose that their evolutionary devices have to be able to synthesise the same number of circuit topologies. In ER, the adaptation (evolution) time is crucial for their application [60]. Thus, the role of verification becomes unfeasible.

2.4 Enhancing Techniques

In this section, the following techniques that help to enhance the basic evolutionary system and successfully deal with such challenges as the scalability and stalling effect problems in EHW will be given an overview: *Parallel computing, Divide and Conquer,*

Adaptation. All of these approaches will be helpful in the further enhancement of the proposed system.

2.4.1 Parallel evolution

EA are inspired by biological evolution, which is a massively parallel system where every individual is independent from every other. Thus, countless number of individuals can be evaluated simultaneously. This feature becomes increasingly important as multi-processors and cluster computers become more powerful. There are three approaches to parallel EA that are dominant in EHW: the *master-slave approach* [49], the *diffusion approach* [67], and the *migration approach* [68].

The *master-slave approach* is the simplest form of parallel EA. A master node implements all aspects of the EA itself, other than calculating the fitness; this has the advantage of introducing no new parameters. This approach is used when calculating the fitness function, which is a very costly operation compared to ranking and mutation. The string representation of parameters makes mutation very simple, and can thus be easily run on the master node. For instance, in [49] they used parallel evolution to design a multi-output digital circuit. Each slave node designed a particular sub-circuit with a single output selecting a particular part or a so-called multi-chromosome. Then, the master node joined the sub-chromosomes and provided the evaluation of the multi-chromosome. There are two types of selection in the *master-slave approach.* In a *sequential selection* model, a master processing unit waits for the ending of computation in all slave nodes so that there is a clear distinction between successive generations [137]. This makes the implementation simple, yet the cost in the idle time of slaves is a potential bottleneck for this approach. In a *parallel selection* model, a master unit does not wait for other slaves but if anyone finishes the work it is immediately allocated a new task. In this case, selection may be done according several variants, for example in the form of a tournament.

The *fine-grained approach* (also called the *diffusion approach*) concentrates on producing a large, interacting population over a number of nodes, often with just one or few individuals per node [67].

The *migration approach*, often called the *coarse-grained approach* (and also called the *island-model approach*), involves the running of a number of largely independent evolving systems - each on a separate processor - which occasionally exchange information with each other. Whereas the *diffusion approach* has much in common with mainland population biology, this approach is inspired by island population biology, with populations connected together by migrations. In [67], the parallel island model evolution was applied towards a synthesis of analogue circuits, namely a "hierarchical fair competition model." Since the purpose of the work was a comparison of different techniques, the circuits evolved were not complex enough to probe the power of the parallel EA.

There is a common problem for all of the parallel evolution approaches described: communication costs become a bottleneck. The majority of the communication costs are spent on migrating individuals from one subpopulation to another. The factors affecting the communication costs are the processor network topology, the migration scheme and the migration intensity (rate and frequency).

It has been noticed that ES would be particularly well-suited to the *migration approach* [68]. Work [68] noted that an island-ES could solve problems that a standard ES could not. In [69], a parallel ES was used in the determination of a protein structure, which behaved mostly like a *master-slave* ES but with *coarse-grained* elements. Selection was done in parallel, with each slave node evaluating a subset of the population. However, the view was expressed in [70] that no extensive study has been done on the implementation and efficiency of a purely *migration*-based parallel ES.

2.4.2 Adaptation

To get successful results using an EA, one needs good parameters such as the mutation rate, the SR, the crossover rate, etc. Often parameters have to be predefined or

tuned manually and are only optimal for a specific problem [61]. In order to find optimized parameters for a certain application, researchers tune them by hand, i.e. experimenting with a multitude of values and selecting one that exhibits the best performance [13]-[18]. For instance, different values of the mutation rate are desired at different stages of the evolutionary process [138]. Tuning the rates manually is very time-consuming where the tuned result is only efficient for some particular instance. The space of the operators and parameters is large. Therefore, hand-designed adaptive mechanisms have had relatively little success, and there has been natural interest in the application of adaptive techniques [34], [36]. The important feature of adaptation is that the algorithm can be adjusted to the particular task while solving that task. In general, there are three levels where the adaptation may take place inside evolution:

- Population-level adaptation adjusts control parameters that apply for the entire population. This approach is the most studied in the literature;

- Individual-level adaptation focuses on adapting of parameters of every chromosome. For instance, each chromosome has its own mutation rate [62]. They may be varied depending on the convergence state of the population, the fitness value of the chromosome, the average fitness value of the population and whether the population tends to get stuck at a local optimum or is scattered in the solution space. The convincing example here is in [63], where the automatically defined function is an individual-level adaptive genetic program, where each individual adapts its definitions for a predetermined set of subroutines.

- Component-level adaptation dynamically alters how the particular gene of each chromosome will be manipulated independently from each other [34].

There is another classification of the adaptation. According to [74], the dynamic adaptation could be *deterministic*, *adaptive* or *self-adaptive*, depending on the mechanism of adaptation. The *deterministic adaptation* is a case where the rule - according to which the adaptation takes place - is predefined and does not depend on the EA. The *adaptive* type is a case where the EA in some way causes the adaptation rule to

change. Furthermore, the *adaptive* type is a case where some coefficients for the adaptation rule are co-evolved or else evolved in parallel.

There are several cases where adaptation is used in this work. The most simple is the deterministic adaptive weight called the "pressure-constant," introduced during ranking in Chapter 4. In addition, the novel differentiated mutation is suggested with the adaptive mutation strategy. The most intriguing type of adaptation is the selection rate adaptation during VNFE in Chapter 5, where a self-adaptive selection rate helped the parallel WDWC strategy.

2.4.3 Divide and conquer (Incremental evolution)

Incremental evolution is regarded as one of the techniques for conquering the scalability problem. One of the first attempts to apply it was undertaken in [75] towards digital circuits. Since then, many approaches have been developed in the digital domain [48], [76], [49]. In the analogue area, few works distinctly utilized these approaches [46], [47]. Furthermore, the targeted circuits were not complex enough to exploit the potential of the technique. The basic idea of *incremental evolution* was first introduced by Torresen in [139] as a "divide and conquer" method, when the task is decomposed into subtasks and then the subtasks are solved step-by-step.

Today the method has been used widely and diversely, and within it can be distinguished three categories of incremental evolutionary techniques: *divide and conquer*, *staged evolution* and *fitness shaping* (Figure 2-2). The first one is quite popular in digital circuit evolution [48], [76], [49], when subtasks may be evolved independently and then joined together for further evolution if required. During the second case of incremental evolution, the initial population is evolved to complete the first sub-task. Based upon the end of the first, the fitness function is changed to the second subtask's fitness function and the second sub-stage runs, and so on. This technique was first introduced by Harvey et al. [167], where in order to evolve a vision-based target location task he used three sub-stages from locating a large immobile target to tracking a moving smaller one. The last type is not often seen as an incremental evolution scheme because it does not suppose the task decomposition [169], [168]. Instead, for the initially

complex task, it applies the fitness function (FF) that changes its parameters with increasing pressure in response to the achieved progress in results. For example, Gomez et al. used competitive co-evolution with a predator-prey task. He modified FF parameters related to the prey's speed and the delay before starting the pursuit ten times, i.e. ten sub-stages were run.

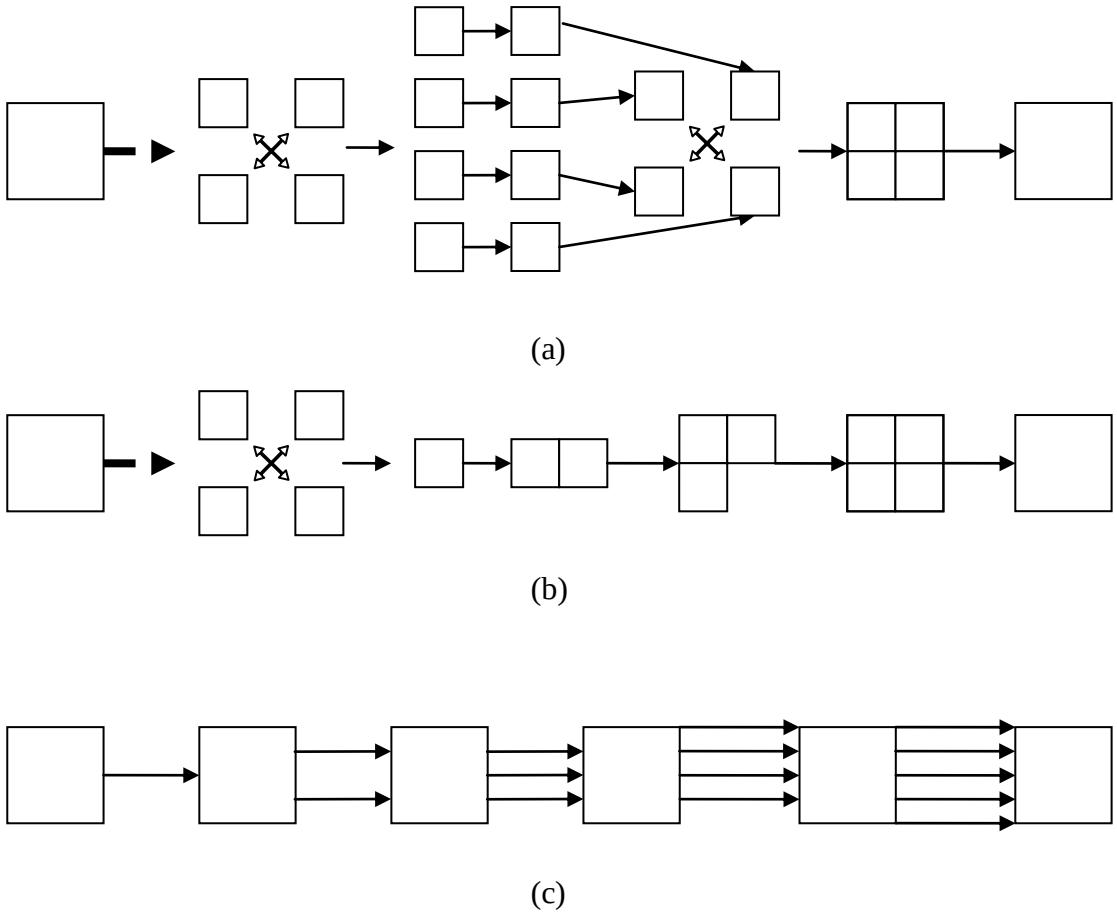


Figure 2-2. Three types of incremental evolution: (a) divide and conquer, (b) staged evolution, (c) fitness shaping. The squares are the tasks and subtasks. The arrows are evolutions. In case (a) and (b), the first sub-stage is the task decomposition. In case (c), multiple arrows between sub-stages express FF modification.

In analogue electronics, the sub-solutions of the subtasks cannot be easily connected to get the proper functioning solution. That is, two perfectly working circuits, when connected to a common input, are not guaranteed to perform in the same way, and it is more likely that each circuit will disturb the functioning of its neighbour. This comes from the physical nature of the electronic components that interact with each other by

means of potentials and currents. This situation differs from that of digital circuits, where the rules of *Boolean algebra* and the complex task could be decomposed by *Shannon's expansion theorem* or *output decomposition* [48]. For instance, in [49] a digital circuit with multiple outputs was broken down into many smaller sub-circuits (each encoded by a single chromosome) with a single output. Due to the digital nature of the target, the "divide and conquer" approach enabled parallel evolution for each sub-circuit. The final solution was built by a simple jointing of sub-solutions together.

In this regard, and in this work, the task decomposition has been applied and the *staged evolution* is found to be most suitable. Thus, when *staged incremental evolution* is mentioned in this work, it means - first of all - not the independent evolution of subtargets but rather the evolution of the current subtarget together with all the subtargets evolved previously. That is, if one has the already evolved a subtarget, when evolving the second one the first solution must participate in that evolution, being encoded in the chromosome together. This fact decreases the benefit in comparison with "divide and conquer." However, as will be shown, to tackle this issue a special technique has been proposed in Section 5.1.2.

2.5 Problems of EHW

In the following, there is a discussion of three major problems in EHW: *generalization, scalability* and *the stalling effect*. The first one is met during Experiment 16: Evolution of Time Interval Meter Circuit (TIMC). The second one is met and even has brought about the failure of Experiments 8: Evolution of Cube Root circuit and 14: Evolution of 8-Output Voltage Distributor circuit (VDC). The third problem is met at every experiment, starting from Experiment 8: Evolution of Cube Root circuit.

2.5.1 The generalization problem

In EHW, the prospective circuits are tested by exposing them a range of input signals and observing the circuit outputs in order to evaluate their fitness. In most of the cases, it would be unfeasible to expose and observe all the possible training examples. Therefore,

the HW has to generalize beyond the cases it has observed. The notable notion relating to this is in [41]: “Modern real-world circuits can process hundreds of input signals, and to observe each possible combination of these just once, even at millions of training cases a second, would take longer than the age of the universe. For sequential circuits the number of training cases is infinite.” So, it is obvious that the ability of EHW to generalize is vital. In Figure 2-3, there is an example how an evolved solution at a good fitness value cannot generalize for the rest of the cases beyond the five test samples.

The chart in Figure 2-3 shows a fitness function of the best Time Interval Meter Circuit (TIMC) against the 1000 test pulses ranging from 0 to 4V. There are five visible test samples used during the evolution, applied at 0.1, 0.6, 1.5, 2.3 and 3.2V. Beyond these five cases, the evolved circuit does not function properly, which means that the circuit has not generalized. In this thesis, the solutions to the problem of generalization are met and solved in the frame of Experiment 16: Evolution of TIMC. The solution for the problem is the increasing of the number of input samples during evolution.

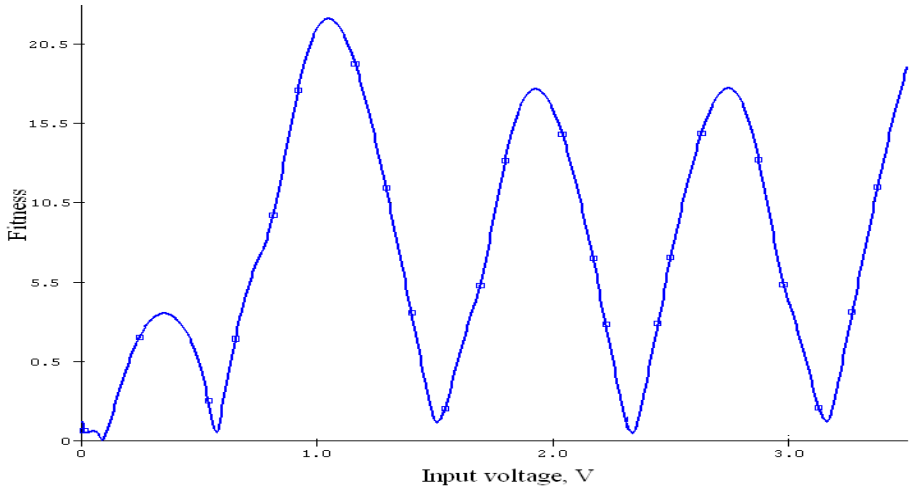


Figure 2-3. The chart fragment of a fitness function of the best TIMC against the different test pulses. The set of 5 test samples were applied (at 0.1, 0.6, 1.5, 2.3 and 3.2V) during evolution, but beyond these cases the circuit has not generalized.

2.5.2. The scalability problem

In contrast to digital circuit design, most analogue circuits are handcrafted and the process is regarded as "a knowledge-intensive, multiphase, iterative task, which usually stretches over a significant period of time" [140]. The evolution of analogue circuits, - both their structure and their parameters - is considered to be a difficult task as well [12]. This is because - unlike a digital component - an analogue component brings to the circuit a high number of new variable dimensions, including: a component's type (up to 5 in this work) and the connection numbers for each of 2-3 pins and parameters (up to 90 in this work). This exposes the requirement of tackling the problem of scalability in EHW. Until recently, the most advanced EHW techniques were able to deal with analogue circuits sized up to 70 components (Table 2-1) (the evolution of both the structure and parameters). This number may be increased if one possesses some knowledge of the targeted circuit, for example the structural topologies or the limits of the parameters that belong to the components of the evolved circuit. However, and in any case, the solution space that it is necessary to search is tremendous. Yao and Higuchi consider the fundamental limiting factors in producing large-scale, complex systems, and in [111] they point out that the solution space circuit evolution is proportional to $\boldsymbol{O(2^n)}$, where $\boldsymbol{n}$ is the number of functional components in the EHW and $\boldsymbol{O}$ is some constant. However, it is more likely concerned with the solution space of digital circuits, because the thoughtful estimation of the solution space of analogue circuits - given in the next Section - exceeds this value considerably.

To tackle the scalability problem, and according to [43], "designers have introduced various approaches that can be divided into three classes: functional level evolution, incremental evolution and development." Most of the attempts to tackle the scalability problem use the first option, namely improving the system at the EA level by exploiting the novel techniques developed by EA theory, such as *representation, multi-objectiveness, co-evolution* and *adaptation*. For the second option, few works have distinctly utilized this approach in the analogue domain [47], [78], [120] (see column 5 of Table 2-4). Furthermore, as it can be seen from column 6 of Table 2-4 that the targeted circuits were not complex enough to exploit the potential of the techniques. This is explained in Section 2.4.3, i.e. in the analogue electronics, the sub-solutions of

the subtasks cannot compose a final solution by simply jointing "bricks into a wall," and in most of the cases every sub-solution has to be coded in the chromosome during the evaluation procedure.

Table 2-4. Developers in evolution of analogue circuits: scalability

Researcher	EA type	Genotype length varying strategies	Scalability Method	Input/ output No	Circuit size
Koza et al. [24]	GP	ILG	D&C	3/3	64
Mattiussi et al. [43]	GA	OLG	development	2/1	55
McConaghy et al. [44], [53] [179]	GP	ILG	structural homotopy	2/1	48
Sripramong et al. [83]	GP	Fixed	represent.	1/1	41
Shibata et al. [84]	GA	Fixed	represent.	1/1	36
Trefzer [148]	GA	OLG	M-O	2/1	34
He et al. [170]	DEA	ILG	represent.	1/1	28
Hu et al. [45]	GP	ILG	M-O	1/1	26
Lohn et al. [59]	GA	ILG	represent.	1/1	23
Ando et al. [78]	GP,GA	n/a	D&C	1/1	22
Kruiscamp et al. [87]	GA	Fixed	No	1/1	22
Zebulum et al. [77]	GP,GA	ILG/OLG/ UDIP	represent.	2/2	19
Dastidar, et al. [86]	GA	OLG	No	2/1	18
Chang et al. [81]	GP	UDIP	represent.	1/1	17
Das et al. [89] [174]	GA	UDIP	represent.	1/1	15
Ohe et al. [171]	GP	ILG	M-O	1/1	15
Langeheine, et al. [121]	ES	Fixed	M-O	2/1	15
Yuan et al. [173]	DEA	ILG	DE	1/1	14
Conca et al. [177]	GA	ILG	development	1/1	14
Kim et al. [54]	ES	ILG	co-evolution	1/1	14
Wang et al. [47]	Gene Expression Programming	ILG	D&C	1/1	13
Goh et al. [17]	GA	ILG	No	1/1	12
Xia et al. [85]	GA	Fixed	adaptation	1/1	11
Grimbleby [79]	GA	ILG	No	1/1	10
Sabat et al. [172]	DEA	ILG	DE	2/1	10
Gan et al. [80]	GA	OLG	represent.	1/1	7
Kuyucu et al. [120]	ES	Fixed	D&C	4/4	n/a

D&C - divide and conquer, DEA is Differential Evolution Algorithm, M-O – multi-objectiveness, DE – differentiated evolution

And finally, the third way of tackling the scalability problem is the *development*. Over the past decade, increasing attention has been targeted toward modelling the biological developmental mechanism in artificial evolutionary synthesis [23], [43], [45], [123]. Since the search space grows exponentially with the genotype size, the evolution of large phenotypes should benefit from *development*. *Development* uses a genetically encoded growth program in several recursive steps. Parsimony arises from the fact that rewriting rules can be applied an arbitrary number of times so that the genotype size should be highly independent of the phenotype size.

In 2001, Layzell [93] pointed out that, to his knowledge, "no circuit with 100+ functional basic elements has yet been evolved; the greatest number so far attained seems to be around 30-40," since this time there has been little progress in scaling EHW to more complex analogue circuits. In this regard, it is suggested the following classification of analogue circuits along the levels of complexity that will be helpful for automatic analogue circuit synthesis, at least within the frame of this thesis.

1. The first or initial complexity level circuits are the ones that contain only 2-pin components. With regard to the number of inputs and outputs, these circuits contain only one input and one output; with regard to the number of components, these circuits - as usual - do not need more than 20-30 components to perform a function that is within 1% deviation from the ideal function. This class of circuits is usually used as an initial test targeted for novel evolutionary systems, since there are a lot of them available for comparison and the making of the decision as to whether the system is powerful [17], [59], and [79].

2. The second complexity level circuits are the ones that may contain 2- and 3-pin components. With regard to the number of inputs and outputs, these circuits may contain a maximum three inputs and outputs, in sum. With regards to the number of components, these circuits - as usual - need about 30-70 components to perform a function that is within 1% deviation from the ideal function. To this level of class belongs the largest circuits evolved until recently [24], [43], [44], [84].

3. The third complexity level circuits are the ones that may contain any kinds of components. With regard to the number of inputs and outputs, these circuits may contain 3 or more inputs and outputs, in sum. With regard to the number of components, these circuits - as usual - need more than 70 functional components to perform functions that are within 1% deviation from the ideal function. This type of circuit marks the future of EHW, those that aims to tackle the scalability problem as their main subject, including the current thesis.

In the above classification, only the functional components are supposed, i.e. each of them by its functioning improves the individual's fitness. There are examples with a greater element count - for example [2], [55] and [110] - but tests have not been conducted on these examples to determine how many of the elements have a functional role.

2.5.3. Solution space for analogue circuit evolution

The automatic synthesis of analogue circuits from high-level specifications is treated as a challenging problem. For example Alpaydın, et al. [16] stated: "Design in the analogue domain requires creativity because of the large number of free parameters and the sometimes obscure interactions between them." To the author's knowledge, no attempts have been undertaken to make estimation of the solution space of analogue circuit evolution. However, some works refer to this question with regard to an exponential dependence on circuit size [17], [23], [28].

In this regard - and in this section - the function of the volume of the solution space from the number of circuit components and its parameters will be defined. The total solution space $\boldsymbol{S}$ depends on structural solution space $\boldsymbol{S_s}$ and parametric solution space $\boldsymbol{S_p}$. They are bound by the *Product Rule* [112]: $\boldsymbol{S = S_s \times S_p}$.

Let is regard $\boldsymbol{j}$ floating pins of some number of unconnected components; then, the minimum number of the pin-to-pin connections - supposing that there are no floating pins left - is $\boldsymbol{j/2}$. On the other hand, no pin is limited to connect to all other pins at the

same time; thus the maximum number of connections enabling each pin to connect to every another pin except itself is defined by combinatorial formula for the *combination* [112]: ***j!/(k!(j-k)!),*** where the *set* has in total ***j*** pins and the *subset* ***k*** equals to ***2*** (connecting 2 pins). So the maximum number of pin-to-pin connections is ***j!/(2!(j -2)!) = j (j-1)/2.*** For simplicity, suppose that all of the pins in a circuit are represented by two-pin components (resistors, capacitors, inductors, etc), then the number of components is ***z*** and the number of two-side connections in this circuit - as defined above - may vary from ***z=j/2*** to ***z(2z-1).*** For every fixed number and structure of connections of a circuit within the defined interval, adding or changing just one additional connection may bring about a significant redistribution of currents and potentials; therefore, this circuit can be regarded as a separate solution.

For simplicity, the combinations among the connections of minimum number ***z*** are not regarded, i.e. the circuit has some kind of fixed structure with a minimum number of connections that make all the pins joined and the circuit proper. This assumption, while it simplifies the task, reduces the solution space by the number of combinations that ***z*** connections may connect to ***z*** components in a circuit. Thus, possible combinations should be counted from the first connection, following ***z*** until ***z(2z-1)-z= 2z(z-1)***.

Thereby, and firstly, for every fixed number of connections the total amount of combinations within the maximum possible ways ***2z(z-1)*** can be defined. An ***a****-combination* of size ***a*** from a *set* ***2z(z-1)***, where order does not matter, is given by a sequence of ***a*** *distinct elements* [112] and is equal to the binomial coefficient: $\frac{(2z(z-1))!}{a!(2z(z-1)-a)!}$. Secondly, all combinations should be summarized along the number from ***1*** to ***2z(z-1)***:

$$S_s = \sum_{2z(z-1)}^{a=1} \frac{(2z(z-1))!}{a!(2z(z-1)-a)!} \qquad [2\text{-}1]$$

Unlike the structural solution space, the parametric solution space S_p depends on component parameters. If the total number of components with different parameters that are available for evolution is ***f***, then according to the *Product Rule* [112] the number of

ways to select z components - provided that each choice has no effect on any subsequent choice - is $S_p=f^z$. Together with the structural solution space, formula [2-1] is the last one and according to the same *Product Rule* [112] it composes the solution space for the evolution of an analogue circuit:

$$S = f^z \sum_{2z(z-1)}^{a=1} \frac{(2z(z-1))!}{a!(2z(z-1)-a)!} \qquad [2\text{-}2]$$

As applied to different circuit sizes, this last formula instantiates the solution spaces listed in Table 2-5.

Table 2-5. Solution spaces for circuits of different sizes

Component number in a circuit	Structural solution space	Parametric solution space	Total solution space	Reasonable structural solution space	Reasonable solution space
3	4095	1,839E+7	7,53E+10	4016	7,39E+10
4	1,68E+7	4,86E+9	8,15E+16	9,74E+6	4,73E+16
5	1,10E+12	1,28E+12	1,41E+24	8,46E+10	1,08E+23
9	2,23E+43	6,23E+21	1,39E+65	1,66E+29	1,03E+51
13	≈6E+91	≈3E+31	≈2E+123	≈2E+52	≈7E+83
15	≈4E+126	≈2E+36	≈2E+161	≈8E+60	≈2E+97
20	≈2E+227	≈3E+48	≈5E+275	≈8E+89	≈2E+138
40	≈1E+863	≈7E+96	≈1E+960	9E+222	≈7E+318

The total number of parameters, equal to f=264, is taken from Experiments 3-7, where there are 84 values for inductors and capacitors, while for resistors there are 96 values.

Formula [2-2] does not claim to offer an exact calculation of the solution space because it does not take into account three-pin components, number of input/output pins, ground and power connections, useless circuits where components may short themselves or else create a non-functional group of components. On the other side, it accounts for the meaningless circuits where every component is connected to every another component or most of the components in the circuit.

The last problem may be tackled to replace the upper limit ***2z(z-1)*** of the sum [2-2] with a more reasonable value. If - to take as an example - the circuits evolved in the past, (including ones that are commercialized [44], [53]) the average numbers of connections per component are presented in Table 2-6. As can be seen from Table 2-6, this number does not exceed ***4***, i.e. the total number of connections in a circuit exemplified does not exceed ***4z,*** where ***z*** is a number of components. Taking into account the previous assumption that the first ***z*** connections are fixed, the solution space S_r for analogue circuits with a maximum of ***4z*** interconnections could be counted according to the formula:

$$S_r = f^{z} \sum_{3z}^{a=1} \frac{(2z(z-1))!}{a!(2z(z-1)-a)!} \qquad [2\text{-}3]$$

Table 2-6. Brief analysis of previously evolved circuits on the average number of pin-to-pin connections per component

Researcher	Circuit name	Component No	Total No of pin-to-pin connections	Avg. No of pin-to-pin connections per component
McConaghy et al. [44], [53]	Current conveyor	15	45	3.0
	Current-feedback opamp	25	80	3.2
	Opamp	48	129	2.7
Koza et al. [24]	Cubing	56	127	2.3
	Four-way source identification	24	52	2.2
Mattiussi et al. [43]	Temperature sensing	55	114	2.1
	Voltage reference	47	107	2.3

The last two columns of Table 2-5 displays the solution space with a more reasonable number of circuit connections, which is called - for simplicity - a “reasonable solution space” S_r. Figure 2-4 shows that for circuits with less than 5 components the S_p dominates the S_s, but for larger circuits the S_s gives the main contribution to the solution space. Despite this, the formula [2-3] does not define the exact value and works with some assumptions, giving the idea that the potential solution space for evolutionary

analogue circuit synthesis grows according to a factorial law rather than an exponential law [17], [23], [28].

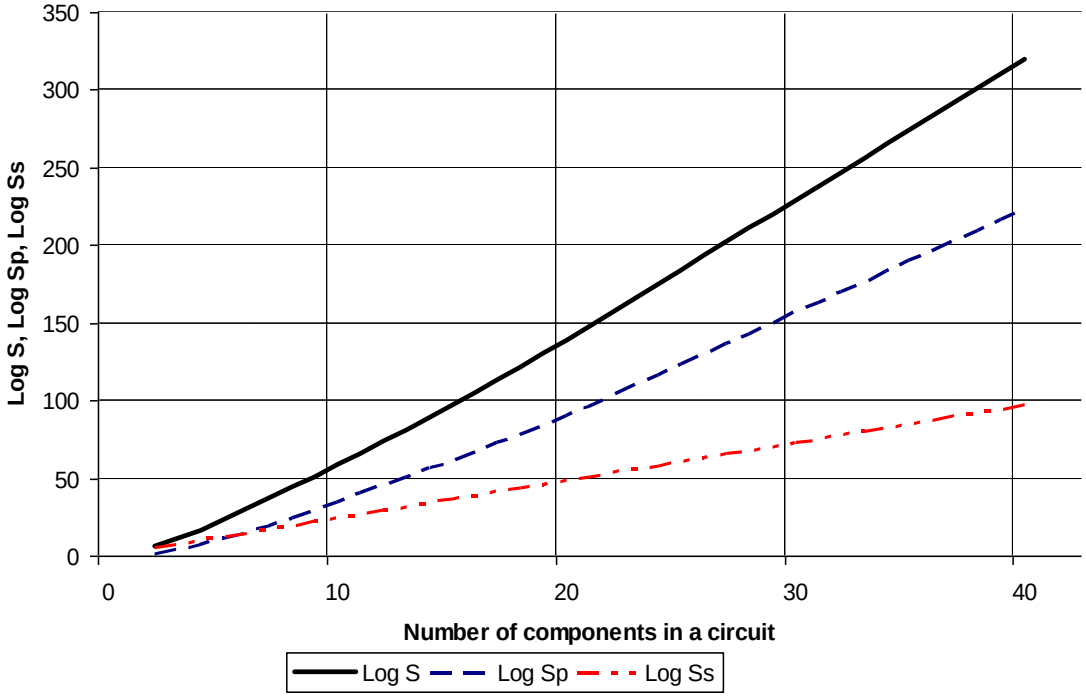


Figure 2-4. Dependence of the solution space on the number of components: parametric $\boldsymbol{S_p}$, structural $\boldsymbol{S_s}$ and their product $\boldsymbol{S}$.

2.5.4 The stalling effect problem in the evolutionary process

The EA displays a tendency towards stalling effects, or local optimums. There are two main reasons for such a problem, as pointed out in [119]. The first one typically occurs when an inappropriate technique is applied towards the task. That is, the technique enables the diminishing of the gene pool; in other words, it leads to the depletion of gene diversity inside a population. It may be caused by an application of a non-optimal mutation parameter or the wrong selection operation. Another reason is the scalability problem [107], [123], when the search space becomes too large for relatively few evaluations. While the first problem is usually solved by the application of adaptive evolution parameters [62], [67], the second problem is solved by the methods

exemplified in Section 2.5.2. All the methods are developed and tested in the frame of this thesis.

2.6 Targets for evolution

In this section, the motivation for evolutionary targets and a brief description of the targeted circuits are given. The experimental work presented in this thesis can be divided into 3 parts, the testing of the framework system, the testing of the intermediate system and the testing of the final system.

2.6.1 Targets for the framework system

The framework system - described in Chapter 3 - has been mainly applied to the low-pass filters. Many of the works in analogue circuit design begin from evolving a passive low-pass filter [12], [13], [17], [28], [78] which is a convenient tool for the probation of an evolutionary technique and the tuning of the evolutionary algorithm parameters for the more sophisticated designs. The behaviour of low-pass filters between the frequencies of 1Hz and 100KHz - the cut-off frequency 1KHz and the transition band 1KHz - has been actively researched through in [12], [13] and [28]. Thus, the performance of a proposed evolutionary technique could be evaluated more precisely if the evolution target were to have exactly the same filter properties. Moreover, the filters could be of two types: LC and LCR. The evolution of LC filters was considered in [12], [17] and [18]. LCR filters were discussed in detail in [13], [18], [28], [45] and [78].

Considerable results were obtained by Koza et al. in [12]. They used Genetic Programming (GP) circuit-constructing program trees with four kinds of circuit-constructing and automatically defined functions that are as one with the mutation types described in this thesis. The last one let them got as results the filters with regular structures within the circuit. The main drawback of this experience is that the technique required large computing power and the methodology was very complex in view of its implementation. The larger computational efforts in a circuit evolution required by GP were proven by Zebulum et al. [28] and Ando et al. [78], where they have given a

comparison between GP and GA. In the first work, the low-pass filters operated as a testing task for the comparison of performance among different types of variable representation strategies. The second work concerned the evolution of real hardware for the purposes of robustness.

There are two important features of low-pass filter that define how good the filter is: the maximum absolute attenuation in the pass-band and the attenuation in the stop-band. The best evolved filter in terms of the first feature is presented by Lohn et al. in [13] (0.0144dB). In terms of the second feature, the filter evolved by Koza in [12] (-72dB) exceeds all others. When making comparisons, other evolutionary features should be taken into account, such as the number of components in the circuit and the computation effort spent.

Finally, the last notion relating to low-pass filters concerns the transition band. The transition band varies from wider [28] to shorter [12], but no one has ever attempted to evolve transition bands shorter than 1 KHz. This is because the shorter the transition band the closer the filter is to the ideal, and the harder the tasks therefore become [106].

2.6.2 Targets for the intermediate system

For an intermediate system that has been armed with a *differentiated mutation technique* the computational circuits (CC) have been chosen as the main targets for evolution. An analogue electrical circuit whose output is a mathematical function is called a computational circuit.

The CC is a circuit that converts the incoming voltage into outgoing voltage in accordance with some computational function, such as square, square root, cubing, etc. Analogue CCs have two main advantages over digital CCs [24]:

1. They operate faster and they are especially useful when the mathematical function must be performed more rapidly than is possible with a digital circuit (e.g., for real-time signal processing at high frequencies).

2. They are more cost- and size- effective. Analogue CCs are useful when the need for a single mathematical function in an analogue circuit does not require converting an analogue signal into a digital signal, with the help of an analogue-to-digital converter, performing the mathematical function through the digital processor and converting the result back to the analogue domain using a digital-to-analogue converter.

The design of computational circuits is one of the most protracted issues for any automatic circuit synthesis system, because it relies on the clever exploitation of some aspect of the underlying device physics of the components (e.g., the transistors) that is uniquely suited to the particular desired mathematical function. Because of this, the implementation of each different mathematical function typically requires an entirely different design approach [145]. It should be mentioned that among all the analogue circuits evolved by Koza, the largest one is a square root circuit with 64 components in [24]. In [24], [98] and [99], they used GP substructure reuse to evolve four types of analogue CC. These papers suggest an attractive opportunity to judge the effectiveness of the evolutionary tool. Targeting the same arithmetic functions, and utilizing an identical evaluation procedure (fitness function), one can directly compare the fitness values (average error), circuit size (economy) and PC time spent. In this work, this opportunity has been taken advantage of. In [98], two CCs were developed by a similar evolutionary technique, as in [24]; however, they used time-continuous signals in time-domain simulations. The transient analysis of a circuit in contrast to DC-analysis provides more robust circuits, despite the higher time-consumption in completing the analysis. The patent in [108] presents the conventionally designed cubing CC, which was improved in [99] by the iterative refinement method.

Cube root computational circuits are very hard to find in the literature, especially the schematics that are formed through evolution. With regards to deciding as to the efficiency of the proposed evolutionary technique, only one work was found that directly addresses the problem [24].

2.6.3 Targets for the final system

The evolutionary targets for the final system have been chosen based on the following criteria: they should belong to an unconventional application domain and thus be of higher complexity in comparison with previous targets. Since the CCs are one of the most complex circuits in the EHW domain [24], the next targets are those circuits that never been approached before.

Three circuits are defined as such targets: the 4-output Voltage Distributor Circuit [116], the 8-output Voltage Distributor Circuit [65] and the Time Interval Meter Circuit [146]. The first two of them are interesting since they are multi-pin circuits. The last one attracted attention because it represents a real world problem, suggested by a commercial entity.

Since these tasks are more specific, their descriptions are given in Sections 4.6.1, 5.2.1 and 5.6.1 where the corresponding system and experiment setups are given.

2.7 Summary of Chapter 2

One of the main purposes of this chapter was to uncover the area of research, and it was done in two stages. At the first stage, the location of EHW in relation to other fields was demonstrated, while second stage focuses on the concrete place to which this work mostly contributes to inside the EHW domain. This Chapter reviewed the field of EHW, focusing on the main techniques, features, questions and problems that are useful for a reader without spreading too thinly over every grain of the area. The special attention is paid to scalability problem, for which the solution space for analogue circuit evolution is investigated in detail. The comprehensive reference list is follows this charter. The next Chapter introduces the development of the ES-based EHW framework, with its limitations at the end of the Chapter.

Chapter 3. The EHW Framework for Analogue Circuits

The aim of this chapter is to describe the EHW framework system whose purpose is the automatic synthesis of analogue electronic circuits of initial complexity, according to the classification given in Section 2.5.2. It depicts - step-by-step - the entire system, starting from the chromosome encoding to the choice of some suitable strategies. The chapter details the analysis of several simulation results that revealed the most appropriate techniques. Finally, the research in this chapter develops an Unconstrained ES-based EHW system with OLG that is able to evolve highly effective LCR analogue circuits.

In Experiments 1-7 below, the methodology developed has been utilized towards the constrained and unconstrained extrinsic evolution for analogue LC and LCR circuit design. Experiments 1-3 utilize the simplest ILG strategy, while in Experiments 4-8 the OLG technique has been applied. In Experiment 8: Evolution of Cube Root circuit, it is considered the first circuit that contained - beside L, C and R components - p-n-p and n-p-n bipolar transistors, called in further LCRQQ circuits.

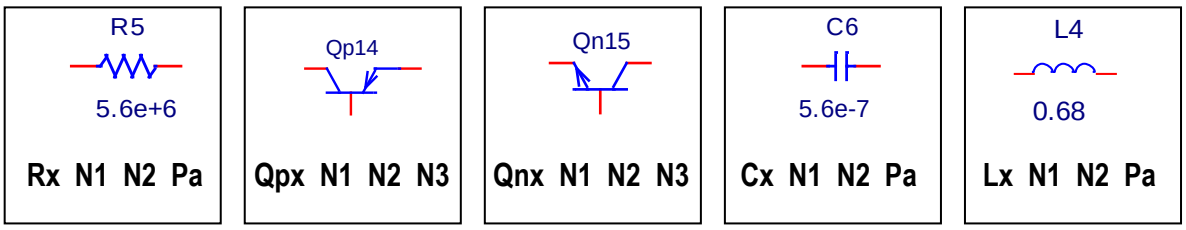
3.1 The Start: Encoding (Representation)

In initiating the system with the purpose of circuit synthesis, one first of all needs to decide as to what kind of genotype this form of the circuit phenotype properties should be coded in. For the framework system, the *direct coding* of the phenotype properties of a circuit is proposed.

Besides the initial components - called the embryo - it has been proposed that evolution should use five types of analogue components (Figure 3-1):

- L – inductor, a two-pin component;

- C – capacitor, a two-pin component;
- R – resistor, a two-pin component;
- Qn – the n-p-n bipolar transistor, a three-pin component;
- Qp – the p-n-p bipolar transistor, a three-pin component.



(a) (b) (c) (d) (e)

Figure 3-1. Genes coding: a resistor (a), a p-n-p bipolar transistor (b), a n-p-n bipolar transistor (c), a capacitor (d), an inductor (e). Rx, Qpx, Qnx, Cx, Lx are loci for names, where the letter "x" is a particular number. N1, N2, N3 are loci for the first, the second and the third pins; Pa-loci is the parameter.

As is usual, each component has four features that describe it. For 3-pin components (transistors), these four features are: three pin numbers and a name of a component. For 2-pin components (resistor, capacitor or inductor), these four features are: two pin numbers, a component's name and a parameter. These four features are to be coded into four loci of a gene. On Figure 3-1, there are five components and five corresponding genes with four loci in each. Thus, every component of a circuit is directly represented as a particular gene, and each gene consists of exactly four loci corresponding to a component's features.

In such a way, the list of genes that describes a circuit composes a chromosome of that particular circuit. The genes on the Figure 3-1 look exactly the same as component lines in the PSPICE netlist on Figure 3-2; as such, there is no necessity to converting a genotype into a netlist. This type of coding simplifies the terminology (for example, it is

meant “circuit” when “chromosome” is mentioned; it is meant “component” when “gene” is mentioned; it is meant “population” when “netlist” is mentioned).

```
**CHROMOSOME No 0 of Gen 133
**Random No=770
**Best fitness 0.091650
.OPTIONS NOREUSE NOMOD NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN 125ms 5s
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  996   0     DC    15.0V
Vneg  995   0     1.5V
Vs 999 0 PWL (0,0) (3.5,5) (5,0)
Rs    999   1     30
.PRINT TRAN V(2) V(3)
R12   0     2     0.01K
R13   0     3     0.01K
R0     7     4     2.700e+004
R1     4     6     8.200e+002
Qn0    7     9     0     NBJT
Qp0    7     1     15    PBJT
C0     17    24    5.600e-003
Rc_1   17    24    1E+9
C1     10    12    2.200e-005
Rc_2   10    12    1E+9
...
C32    7     16    1.200e-009
Rc_5   7     16    1E+9
R29    1     28    6.800e+003
Qn23   25    26    2     NBJT
C33    20    5     2.200e-007
Rc_6   20    5     1E+9
.END
```

Figure 3-2. The PSPICE deck fragment of a computational circuit derived from the cir-file.

Table 3-1. The dimensions of potential values for each of the loci

	Name loci	**Node 1 loci**	**Node 2 loci**	**Node 3 loci**	**Parameter loci**
Dependence	Static	Depends on circuit structure	Depends on circuit structure	Depends on circuit structure	Preset
The number of values	5	From 5 to 100	From 5 to 100	From 5 to 100	From 84 to 96

Each locus has its own dimensions of potential values that differ from each other. These differences are shown in Table 3-1. If, for a parameter and a name loci, it is possible to set the particular number of values (say, one can choose the number of types of components and number of parameters) for component pins, the number of potential

connections cannot be predefined as it is a function of the circuit size (see Section 2.5.3).

3.2 The System Framework – Master Part

In order to create a system that will be able evolve electronic circuits, one has to be able to join two main parts: *the slave evaluation part* (i.e. simulation software) and *the master part* of the system.

Conventionally, the master part of the system must create the following functions and - accordingly – it must have the same number of subroutines:

1. *The initial conditions setting part.* Here, the limits of the system are set, such as:

    - The number and names of components that participate in evolution. A *component list* is initiated, where to every component is assigned its own position number;

    - The number and values of parameters that participate in evolution. For each component, the *list of parameters* is initiated based on *12 parameters per decade*, where to every parameter is assigned its own position number; for inductors (from 1E-9H) and capacitors (from 1E-12F), there are 84 values, for resistors (from 1.8Ω) there are 96 values used by evolution;

    - The PSPICE analysis options, such as the component models to be used, RELTOL, ITL1, ITL2, etc. (listed and described in Appendix B);

    - The EA parameters, such as the population size, the mutation rate and the SR;

    - Initiating the results summarizing *data-file*, where all the required by the researcher results are accumulated throughout the experiment;

- The number of testing points and the number of tests.

2. The *embryo circuit part*. This part of the system is responsible for initiating the embryo circuit. It defines the circuit terminals and the embryo's components, parameters and structure. It only works during the first generation.

3. The *initial circuit growth part.* Since the embryo circuit cannot be evaluated - due to floating pins - here the embryo has to be grown up until at least several elements are able to conduct the current from the source to the load of the circuit. Usually, this number depends on the size of a population. For example, for a population 20,000 the number of components in the *initial circuit* (embryo excluding) should be from 3 to 5 for the one input-one-output circuit, and it takes the corresponding number of initial generations. The lower the number of initial components that are initially used may lead to a large amount of identical circuits in a population to be evaluated. On the other hand, a higher number may cause the earlier appearance of the stalling effect. This part of a system works only during the first generation.

4. There must be a special interface that organizes *communication* with the *evaluation part* of the system. This interface runs and stops the evaluation software/hardware and downloads the *cir-file* and opens the *out-file*.

5. When running the *evaluation part* of the system, the special sub-program has to *monitor the process* of evaluation in order to interact in real time with the latter in case of any problem. There are two main problems here. Both of them happen due to the analysis of non-convergent circuits that may cause PSPICE to delay the analysis or even to stop the analysis.

6. When reading the *out-file*, a special *fitness assigning subroutine* must identify the valuable information - such as the chromosome number - circuit functioning characteristics and errors. Depending on the type of the information read, the subroutine has to assign a fitness value per chromosome. All the fitness values are associated with corresponding chromosomes and are stored inside the system

memory. Since this part of the process is the most time consuming, a great deal attention should be paid on it.

7. The *ranking part* of the system ranks the chromosomes based on their fitness values and other available properties. As has been mentioned in Section 2.2.4, this part of the system suggests an opportunity for the inclusion of additional objectives for evolution.

8. The *selection part* selects the parents of the next population and clones them to the complete population.

9. The *mutation part* of a system mutates each chromosome of a population. It creates the PSPICE netlist and writes it in the next cir-file.

10. The *process termination part* of a system checks the terms and stops or enables the further evolution with the report written in the *data-file*.

These 10 milestone procedures are regarded as obligatory for the *master part* of the framework evolutionary circuit synthesis system (Figure 3-3).

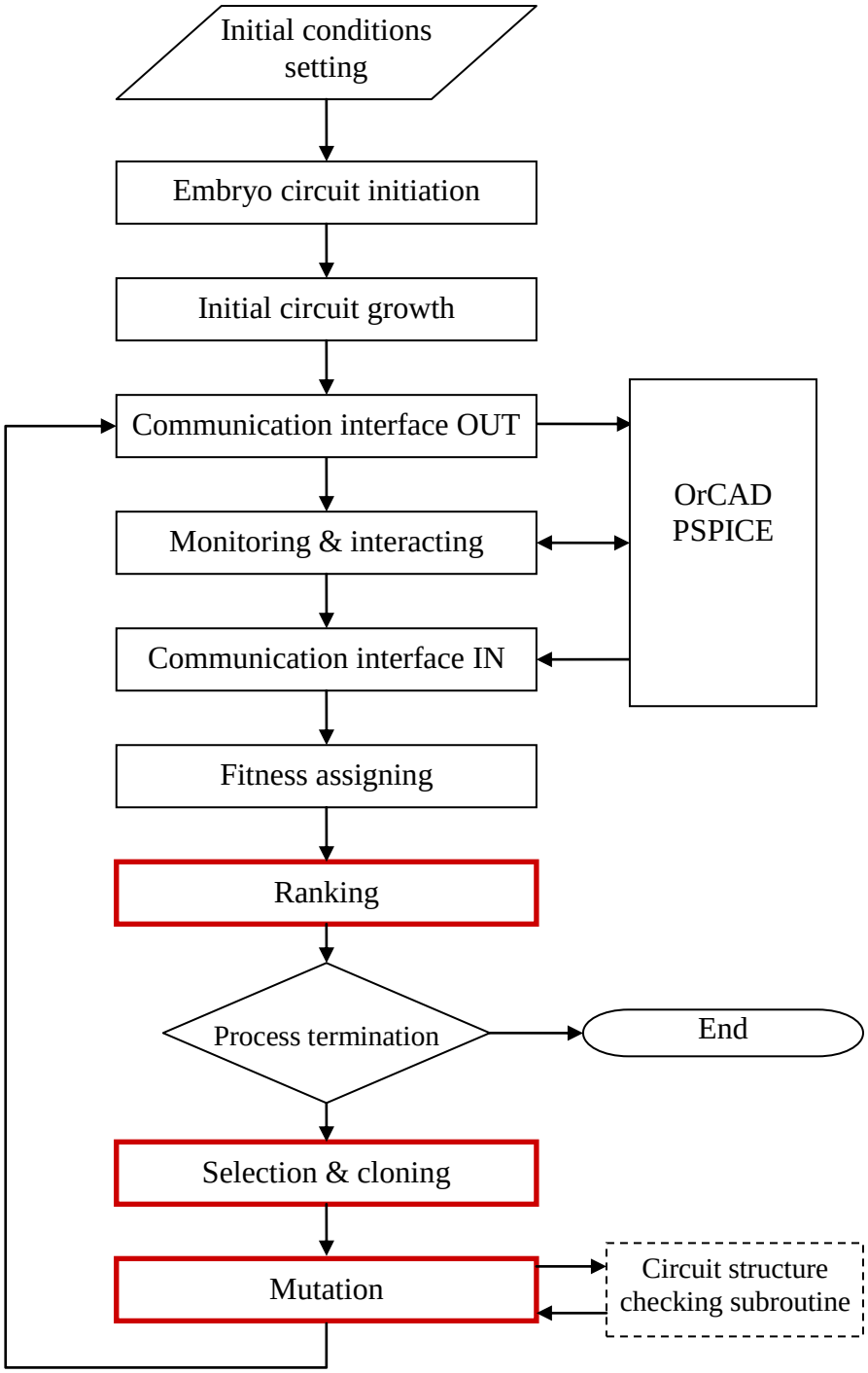


Figure 3-3. The flowchart of the proposed system. The ranking, selection and mutation stages of the system are squared in red bold, since they are the most modifiable parts in the frame of this thesis. In the dashed box is the subroutine that is usually presented in almost all the other approaches. However, within the frame of this thesis this kind of subroutine will be applied only during Experiments 1 and 4.

3.3 The System Framework – Slave Part (Simulation SW)

It has been discussed that there are two conventional options for evaluation: extrinsic and intrinsic (the third, unconventional one, is mixtrinsic). The advantages of the extrinsic method have become more valuable and feasible for in view of the current research.

In order to download a circuit into PSPICE, one has two options: interactive and non-interactive. If the first one requires the constant presence of a human, the second one implies the off-line communication with the program in a command-line regime. Since dealing with EA means dealing with a population of circuits (chromosomes) that have to be evaluated by PSPICE, the second - non-interactive - option is more suitable for utilization. The non-interactive mode of PSPICE enables it to run the batch regime, where the multitude of circuits - coded in the PSPICE netlist inside a cir-file - is downloaded into PSPICE. The last one produces the out-file upon the end of the analysis. The software enables the use of built-in library models or else the use of one's own.

Thus, each generation a population of chromosomes coded in the PSPICE netlist is listed in a cir-file and downloaded into PSPICE.

Being tuned, however, the simulation software may produce an error message if the analogue circuit under analysis has at least one of the following properties:

1. Has at least one floating pin of any of a component inside the circuit;

2. The circuit is not able to converge according to the formulas calculating the current and voltage of PSPICE;

3. If the voltage or current at any node or component exceeds the limits set in PSPICE;

4. If any orthographic mistake is made in the netlist.

If problems 1 and 4 are solvable in the frame of the master part of the system, problems 2 and 3 refer to the source-code of PSPICE. Since the source-code is closed, one has to find a way to tackle problems 2 and 3 by an additional subroutine. This has to be made because problems 2 and 3 in PSPICE default mode may lead to enormous time consumption during evolution. In default mode, the software - if it has met problem 2 or 3 - solves it during ten seconds in the best case. That is, the evaluation of a 10,000-population where there are 1000 non- converging circuits (10%) requires almost three hours (for a chromosome size equal to 10-12 genes, instead of 15 minutes), which is intolerable. In the worst case, PSPICE comes to a standstill, requiring manual interaction.

To tackle this problem, in this work the built-in Windows API functions have been used to recognize the appearance of problems 2 and 3 during PSPICE's running and are able to automatically interact with PSPICE in order to skip the various time consuming default procedures. When PSPICE ends the work, it automatically saves the results of an analysis in an *out-file*, which is then downloaded back into the system.

3.4 The Initial Circuit Growth Part

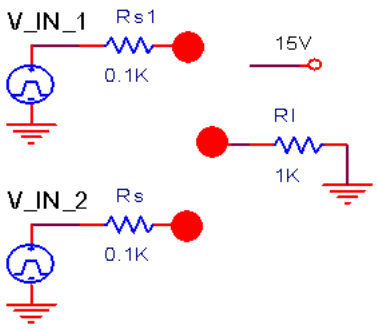


Figure 3-4. The embryo circuit for TIMC. The task of the initial circuit growth part of the system is to provide connections to floating pins that are labelled on the figure by red circles.

On Figure 3-4 is an image of the embryo for the TIMC circuit. The task consists of connecting each input with each output with the help of randomly chosen components.

Otherwise, if there is any floating pin, the circuit will be rejected by PSPICE as invalid. The initial circuit growth part of a system suggests an option to choose the initial number of elements to grow up. As mentioned above, this number varies from 3 to 5 components (per out pin). Since the structure of the resulting 3-5 component circuit is random, the first generation may bring about 30-50% crippled chromosomes. However, in further generations, with their being chosen, the right chromosomes only bring up to 1% of invalid offspring in the case of LCR circuits and up to 8% of crippled chromosomes in the case of LCRQQ[2] circuits,[3] where all the "cripples" belong to non-convergent circuits.

3.5 Mutation

For the framework system, only the static population-level mutation rate is utilized. For the first part of the framework system, the ILG varying strategy has been inculcated. This means that the mutation procedure consists of two main parts, which are applied depending on the current success of the search process: the *circuit-structure-mutation* (CSM) and the *add-new-element-mutation* (ANEM).

At the ANEM stage as well as at the CSM stage and in all other mutation types described in this work, the *Rule of Equal Mutation Probabilities* (REMP) is set. The REMP is established targeting the same general target, the unconstrained evolution of analogue circuits. According to this rule, any kind of mutation must have an equal probability of appearing at every component, node, pin, parameter and other attributes of a circuit. There are no prohibitions to any kind of connection. For instance, during ANEM, every component has an equal right to appear at any part of a circuit disconnecting any kind of connection and creating any type of a new structure with an equal probability. In intrinsic approaches, due to the inherent features of hardware, the application of REMP is limited. However, most of the extrinsic approaches constrain the evolution intentionally. When mutating by a single gene, they suggest for a new/mutated component a limited choice of connections/mutations. For instance, in

[2] LCRQQ is a circuit composed of inductors, capacitors, resistors and two types of transistors

[3] In this context, "up to" means "maximum." The amount of invalid circuits depends on the average chromosome length of a population. The longer the chromosome is the higher the amount of invalid circuits.

[94] they suggest for the n-p-n bipolar transistor only 52 variants of integration inside the circuit (nodes to connect to), regardless to the circuit size. The statistics constraining analogue circuit evolution are presented by Table 2-3. Some typical prohibitions on transistor connections are listed by Table 3-3.

3.5.1 CSM

CSM is like any conventional mutation in that it is an operation that concerns per loci replacement inside a chromosome. This is to say that the value of a randomly chosen locus has to be replaced by another randomly chosen value. If only CSM is applied, the procedure repeats as many times as is required by fitting the mutation rate. For instance, for a 20-gene chromosome, consisting of 80 loci, at mutation rate set at 5%, the procedure has to be repeated four times. Other examples instantiating the procedure are shown in Table 3-2.

Table 3-2. Examples of the 5% mutation rate for 5 different chromosomes.

Chromosome size, genes	**Locus number**	**No of locus to be mutated at mutation rate 5%**
10	40	40×5%= 2
20	80	80×5%= 4
50	200	200×5%= 10
80	360	360×5%= 16
100	400	400×5%= 20

For the framework system, the circuit growth methodology and the genotype varying strategy are the simplest conventional ILG. The flowchart of the mutation procedure, including the triggering of one or another mutation type, is shown on Figure 3-5. If the evolution has got into a local optimum (stalling effect), the add-new-element-mutation (ANEM) is applied at every stalling generation, otherwise CSM takes place.

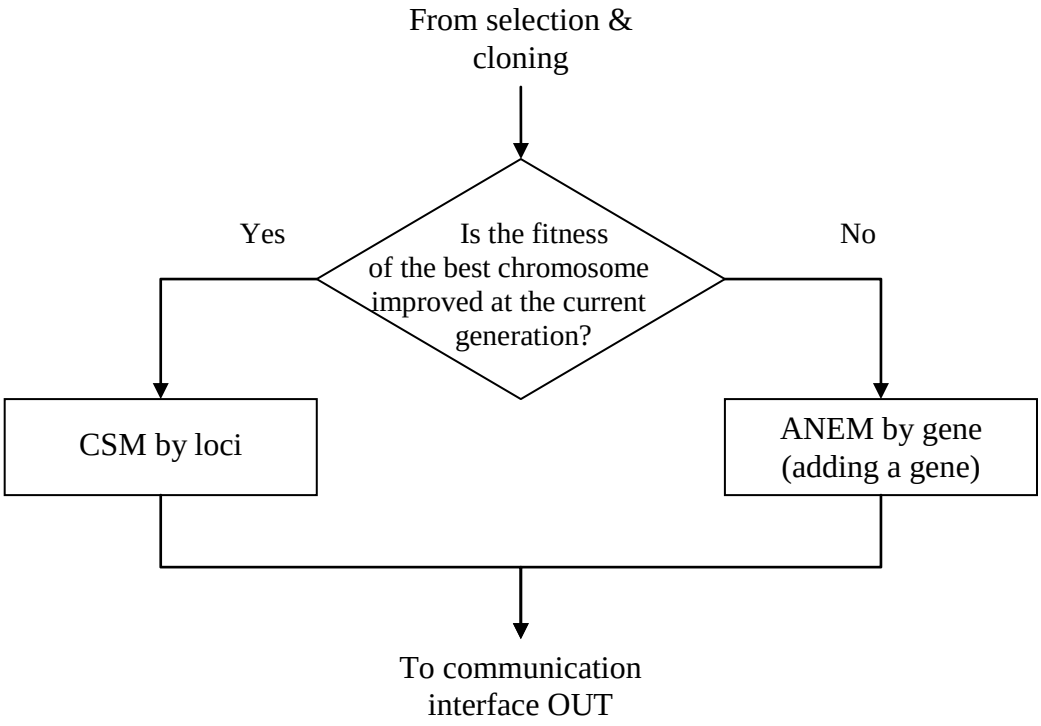


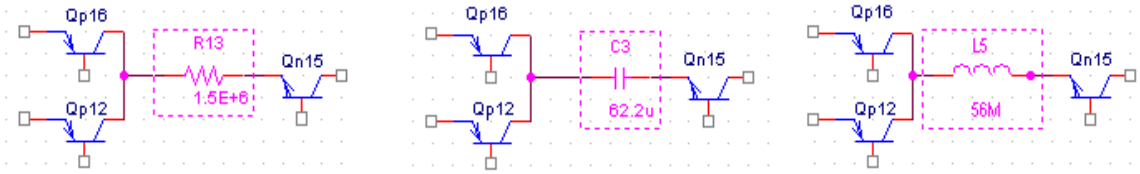
Figure 3-5. The flowchart of ILG-based mutation in the framework system.

The following circuit modification rules take place during CSM. The design decisions below are made in accordance with REMP, as stated at the beginning of Section 3.5; i.e. all of the options below suggest the freedom of connections for a new/mutated component and represent all the possible variations of connections that may occur in a circuit.

1. *Component name mutation rules.* There are four possible options that may appear when changing one component to another inside a circuit, depending on the component type combinations:

    - A 2-pin component changes another 2-pin component. Since each 2-pin component has a symmetry towards a current and voltage direction, the change of one component to another is the simplest procedure among all the others. For instance, a resistor can be changed by a capacitor or inductor without any probability of the appearance of any floating pin (Figure 3-6). A *parameter inheritance rule* takes place when components with parameters change each other: a parameter of a new component is chosen by its *position*

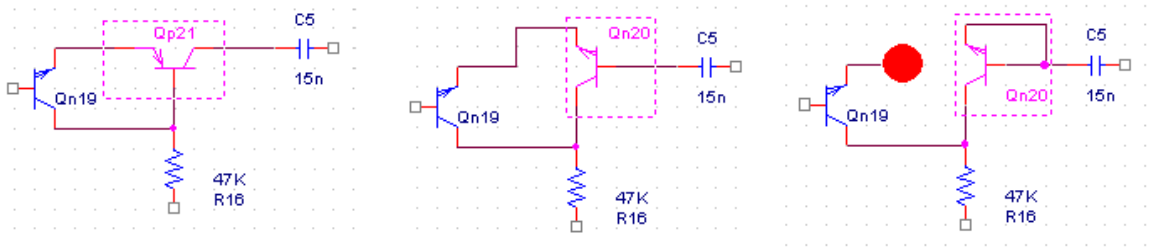
number that is proportional to a *position number* in a *parameter list* of the previous component's parameter. That is, if a *parameter list* of a component to be changed consists of ***n*** parameters and the parameter's number in this list is ***a***, a new component whose *parameter list* consists of ***m*** members, will have a new parameter ***b***: $b = \frac{a \times m}{n}$. For example, if in a *parameter list* of a resistor there are 50 parameters, while in a *parameter list* of a capacitor there are 80 parameters, and a capacitor with a *position number* 40 is changed to a resistor, the parameter of the latter should be under a *position number* 25.



(a) (b) (c)

Figure 3-6. The mutation of the 2-pin component to another 2-pin component. (a) The component with a name "R13" is replaced by the component "C3" (b) and "L5" (c).

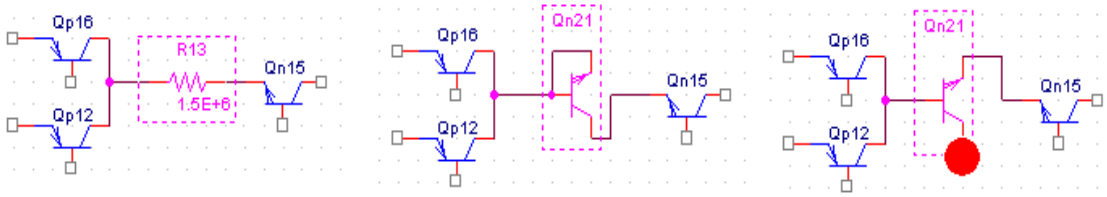
- A 3-pin component changes another 3-pin component. In this case, after the removal of the component, there are 3 pins left floating. Each pin of the new component is randomly assigned to one of the floating pins with equal probability. However, there is only one prohibition - it is not allowed to assign all three pins to one node. As a result of an application of this rule, it may appear as a circuit where there are no floating pins left (Figure 3-7b) or where is one floating pin left after the operation (Figure 3-7c). The last problem is solved by connecting a floating pin to any of the circuit's nodes with equal probability.



(a) (b) (c)

Figure 3-7. Mutation of the 3-pin component by another 3-pin component. (a) The component with the name "Qp21" is replaced by the component "Qn20" (b) without floating nodes left and (c) with one floating node left. By the red circle is indicated a floating pin at Qn19.

- A 3-pin component changes a 2-pin component. In this case, each pin of a new component is randomly assigned to one of the floating pins with equal probability. However, there is only one prohibition - it is not allowed to assign all three pins to one node. As a consequence of an application of this rule, there may appear a circuit where there are no floating pins left (Figure 3-8b) or where there is one floating pin left (Figure 3-8c). The last problem is solved by connecting a floating pin to any of circuit's nodes with equal probability.

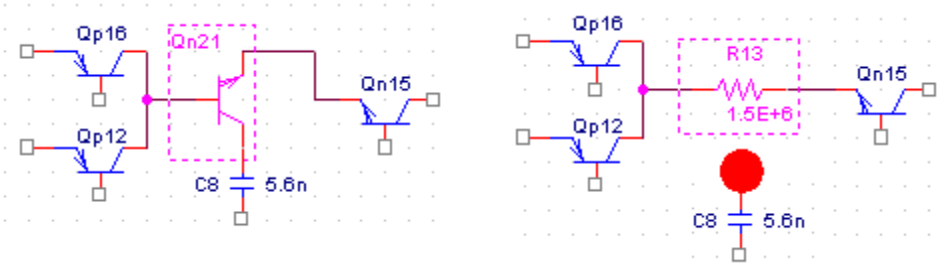


(a) (b) (c)

Figure 3-8. Mutation of the 2-pin component by a 3-pin component. (a) The component with a name "R13" is replaced by the component "Qn21" in two ways: without floating pins (b) and with floating pins (c)". By the red circle is indicated a floating pin at Qn21.

- A 2-pin component changes a 3-pin component. In this case, each pin of a new component is randomly assigned to one of the floating pins with equal probability. However, there is only one prohibition - it is not allowed to

assign all two pins to one node. Due to an application of this rule, a circuit with one floating pin appears (Figure 3-9b). The last problem is solved by connecting a floating pin to any of circuit's nodes with equal probability.

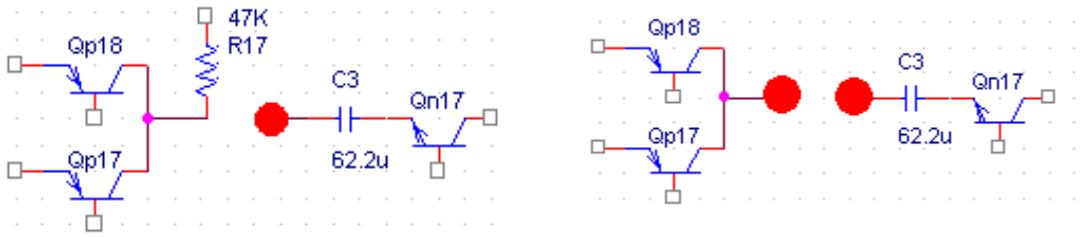


(a) (b)

Figure 3-9. Mutation of the 3-pin component by the 2-pin component. (a) The component with the name "Qn21" is replaced by the component "R13" (b). This kind of replacement always leads to a floating pin. By the red circle is indicated a floating pin at C8.

2. *Node connection mutation rules.* There is no difference as to which pin of a component is set to be mutated, whether first, second or third; for all of them the rule is the same. The randomly chosen pin of a randomly chosen component is disconnected from the current node and randomly connected to another node. The last one is randomly and equiprobably chosen from the all nodes of a circuit. The floating pins - if they appear - obey the same rule, i.e. they are connected to a randomly chosen node of a circuit. In Figure 3-10 there are two cases where a pin is disconnected from a 3-pin node (a) and a 2-pin node (b).

3. *Parameter mutation.* The parameter mutation is when a randomly chosen component with a parameter changes its parameter to a new one. The latter is chosen equiprobably from the list of available parameters set by the *initial conditions setting part* of the system. If the randomly chosen component does not possess the parameter feature, the choosing procedure continues until the required component is found or some limit of for search iterations is reached. In the latter case, the parameter mutation is randomly replaced by another mutation type.



(a) (b)

Figure 3-10. The node connection mutation. After breaking a randomly chosen connection, there are two kinds of possible cases: with one floating pin to be mutated (a), and two floating pins (b), one of which is supposed to be mutated and another one is caused to find another connection (b). For each floating pin, the new node is found randomly among all the nodes with equal probability.

3.5.2 ANEM

The add-new-component-mutation (ANEM) plays a crucial role in dynamic representation: during the initial circuit growth stage, as a mutation tool it struggles against stalling and for permanent circuit growth [127].

In common with other works ([2], [12]-[18]), ANEM consists of a procedure where one component has to be randomly chosen from a *component list* and connected to a circuit. For this, randomly chosen nodes and the parameter are selected with equal probability and assigned to a new component. When connecting to pins (Figure 3-11a), a new component may not disturb the existing circuit structure (Figure 3-11b) or change it in different ways (Figure 3-11c-j).

In case all the pins of a new component are assigned to the same node, the component disconnects the connections and again randomly chooses new "neighbours." The floating pins - left after this procedure - have to search for new connections. The example of a new 2-pin component connecting to a node with two and three pins is shown by Figure 3-12.

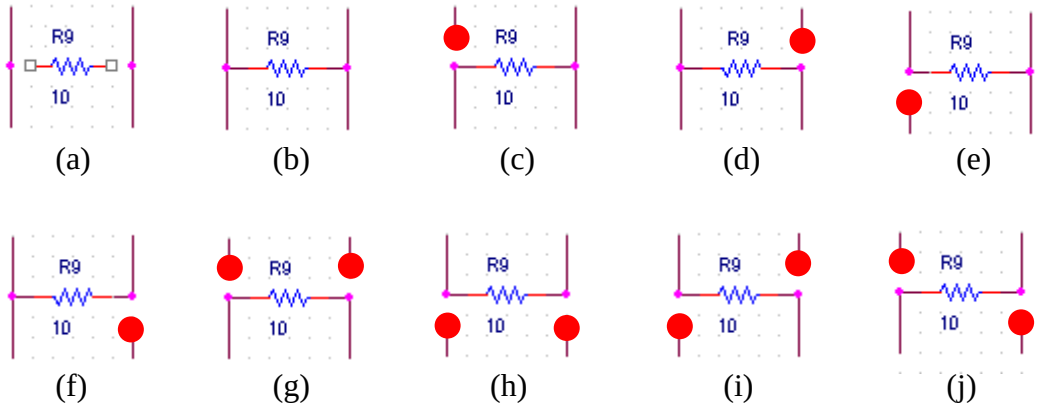


Figure 3-11. An example of an ANEM mutation. (a) For a resistor R9 with a parameter 10Ohm two nodes are chosen. (b) The new component is connected to a circuit without changing the overall structure. (c)-(j) There are eight variants where a new component is connected to a circuit while changing the overall structure. The floating pins are marked by red circles.

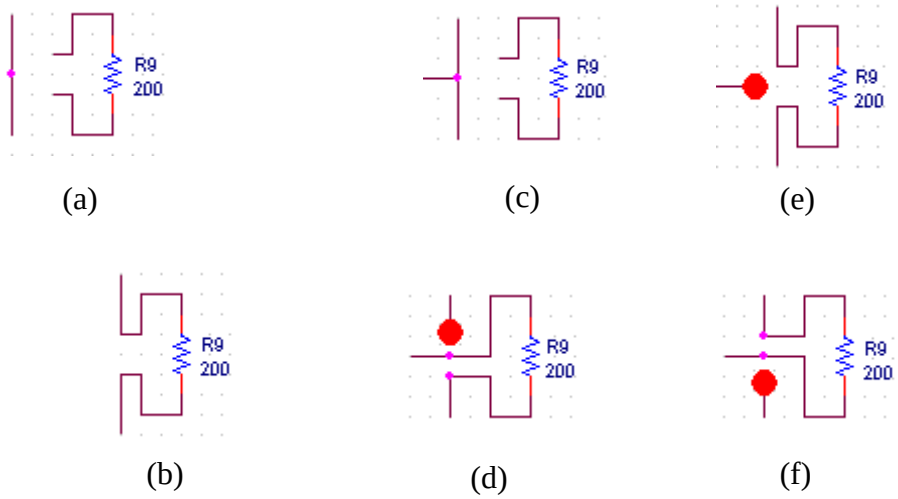


Figure 3-12. An example of an ANEM mutation when the same node is assigned to both pins of a resistor R9 with a parameter 200Ohm. (a) The node is formed by two pins and the only means of connection is on (b). (c) The node is formed by three pins. (c)-(f) Three variants when a new component is connected to a circuit changing the overall structure. The floating pins that are marked by red circles have to search for other connections.

3.6 Fitness Assigning, Ranking and Selection Procedures

A *fitness assigning subroutine* searches for useful information inside an *out-file*, such as the number of the chromosome, the points of analysis and the corresponding values of analysis results (Figure 3-13). Based on this information, it calculates a fitness function and assigns it to a particular chromosome. The built-in PSPICE options - such as "NOREUSE", "NOECHO", etc. - significantly help to form the *out-file* and are convenient for searching for the necessary information inside the file. In Figure 3-13, there is a fragment of an *out-file* with the transient analysis result that refers to an evolution of TIMC.

```
**CHROMOSOME No 12
.OPTIONS NOREUSE NOMOD NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE ITL4=11

 ****     TRANSIENT ANALYSIS               TEMPERATURE =   27.000 DEG C

  TIME         V(3)
  0.000E+00    9.989E-03
  1.000E-03    9.976E-03
  2.000E-03    9.976E-03
  3.000E-03    9.976E-03
  4.000E-03    9.976E-03
  5.000E-03    9.977E-03
  6.000E-03    9.977E-03
  7.000E-03    9.977E-03
  8.000E-03    9.977E-03
  9.000E-03    9.978E-03
  1.000E-02    9.978E-03

 ****     JOB STATISTICS SUMMARY

  Total job time (using Solver 1)   =         .02
```

Figure 3-13. A fragment of an out-file that refers to a chromosome coding TIMC. The useful information here for a *fitness assigning subroutine* is "Chromosome No 12," "TIME," "V(3)" and 11 values of V(3).

At the framework system, the ranking is only made upon the fitness value, i.e. the better the fitness of the chromosome the higher the ranking.

Another point to note concerns the prevention of chromosome replication. The extrinsic EHW has an advantage in using the simulation software. The simulation software, when making the analysis of identical circuits, produces identical analysis values. Consequently, the FF calculates identical fitness values, even if with a precision

of 5 or more decimal digits. During ranking procedure, when comparing two or more chromosomes with identical fitness values and identical genotype length, this feature enables the assignation of a rank to only one individual and eliminates the others. This could be done with the confidence that the rest of the chromosomes, in terms of their functional parts, are replications of the ranked one. The aim of this operation is the increasing of the diversification of the gene pool. On the other hand, the non-functional components of such the circuits - called "introns" [126] - that can differ may carry the neutral mutations and are regarded as an important factor in the evolution of circuits [38], [101]. As a result of this trade-off, the first option is chosen because the diversification of the gene pool will be crucially important in the later part of this research during VNFE, when selection rates are minimized.

There are many alternatives as to how to select individuals for the next generation. There are two types of selection schemes that have been used in the frame of this thesis: the "roulette wheel" and the *disruptive selection scheme.* The "roulette-wheel" selection scheme is used with a selection strength of $\beta=\infty$. The *disruptive selection scheme* is used when 9% of the top-ranked and 1% of the bottom ranked are selected as parents for the next generation.

Throughout all the experiments, the fitness threshold is set to 0.3%, i.e. the evolution ranks the fitness of a new chromosome as the best if the relative fitness difference between one of the current best ones and one of the rival chromosomes is more than 0.3%. This barrier enables pressure to be applied during selection which stimulates an application of more radical mutations (ANEM). Furthermore, it prevents the appearance of chromosomes with negligible differences that any simulation software like PSPICE will inherently produce.

3.7 Unconstrained Evolution

As has been discussed in Section 2.5.3, unconstrained evolution provides a larger space for potential solutions to be explored and for the discovery of unconventional solutions for conventional tasks as well as for unconventional tasks.

In [82], unconstrained evolution - both spatially and temporally - is applied intrinsically to digital FPGA-based reconfigurable hardware. By releasing the full repertoire of the behaviour that FPGA can manifest - namely, allowing any connections among modules, letting the evolution evolve the granularity of modules as well as the regimes of synchronization - such evolution has been able to find a highly efficient electronic structure, which requires 1-2 orders less silicon area to achieve the same performance as a conventional design does.

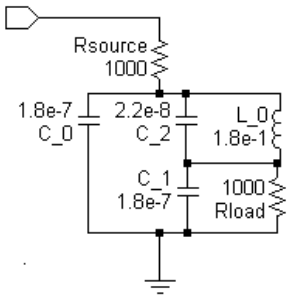
By analogy to this approach, unconstrained evolution could be applied towards the original analogue circuits. The *Rule of equal mutation probabilities* (REMP) introduced in Section 3.5 makes the first contribution to removing the constraints on the evolution. Furthermore, the analysis of earlier developments in the evolution of analogue circuit design reveals that all of the approaches so far developed are based on the *circuit-structure–checking rules* for avoiding invalid circuits (Table 2-3). In this sense, the range of circuit-structure-checking rules at the netlist composition stage - prohibiting invalid circuit graphs - are regarded as the main *constraints* for the design methodology.

In this regard, there are two terms that should be satisfied by the system to perform *the unconstrained evolution*. The first one is that no circuit-structure-checking rules should be applied and all the circuits should be counted as valid graphs except for the ones that have elements with floating pins and isolated sub-circuits. The second is the adaptation of the REMP. In what follows, the first term is regarded in detail.

3.7.1 Unconstraining the evolution of LCR circuits

There are two main kinds of invalidities in netlists that treated as errors by most of the simulation software: the nodes that have no DC path to the ground (tackled in [12]) and loops that involve inductors and/or a voltage source. By tackling these issues, the evolution is enabled so as to create structures with arbitrary connections and eliminate the constraints imposed by the simulation software. Most of the methodologies in the area simply prohibit such kinds of invalidities from appearing. In the case of LCR circuits, adding to each capacitor the Giga-Ohm resistance in parallel and adding to

each inductor the Micro-Ohm resistance in series, and at the stage of PSPICE cir-file generation, this allows us to avoid these invalidities. Such kinds of resistances are called *R-support*. Using R-support and avoiding floating pins makes almost any randomly generated LCR circuit valid, and indeed makes the evolution *absolutely unconstrained*. Figure 3-14b demonstrates how *unconstrained evolution* generates the circuits with R-support. The circuit depicted on Figure 3-14a, once it has been prepared for unconstrained evolution, will appear as shown by Figure 3-14b. Each element line describing inductor (L_0) is followed by an R-support element (Rl_1) in series with an infinitesimal parameter; and each element line describing the conductor (C_No) is followed by an R-support element (Rc_No) in parallel with an infinite parameter.



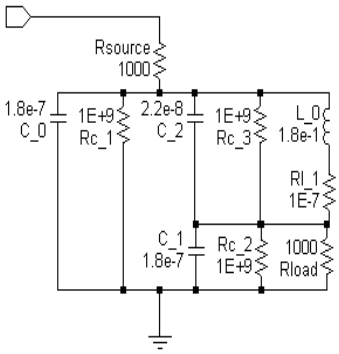

```
**GENERATION No 7
**CHROMOSOME No 19977
**Fitness 1.73055

.AC DEC 19 1 100000
.lib "nom.lib"

Vs          999    0      AC 2
Rsource     999    1      1000
Rload       0      2      1000

L_0     1      2      1.8e-1
C_0     0      1      1.8e-7
C_1     2      0      1.8e-7
C_2     2      1      2.2e-8
.END
```

(a)




```
**GENERATION No 7
**CHROMOSOME No 19977
**Fitness 1.73055

.AC DEC 19 1 100000
.lib "nom.lib"

Vs          999    0      AC 2
Rsource     999    1      1000
Rload       0      2      1000

L_0     1      2000   1.8e-1
Rl_1    2000   2      1E-7
C_0     0      1      1.8e-7
Rc_1    0      1      1E+9
C_1     2      0      1.8e-7
Rc_2    2      0      1E+9
C_2     2      1      2.2e-8
Rc_3    2      1      1E+9
.END
```

(b)

Figure 3-14. Two different chromosome representations of the circuit following constrained (a) and unconstrained (b) evolution.

If the circuit-structure-checking rules are applied to a circuit after a fitness assignment as a part of the pruning procedure - i.e. every R-support element is checked

as to whether it could be pruned out without damage to the current fitness value of a circuit - then the circuit may contain a very few R-support elements or even not contain any at all. For instance, on being evolved the circuit shown by Figure 3-14b and after such the pruning process may again become the circuit depicted by Figure 3-14a.

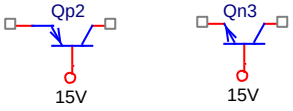
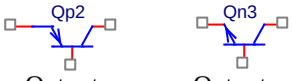
3.7.2 Unconstraining the evolution of LCRQQ-circuits

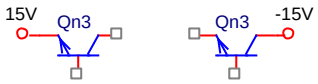
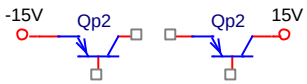
In the previous section, only those circuits that contained 2-pin components have been treated by unconstrained evolution. Now, the unconstrained evolution of circuits with 3-pin bipolar transistors will be considered.

In Table 2-3, some works in the area of evolutionary analogue circuit synthesis are listed with information as to the types of constraints applied. The analysis of the table reveals that most of approaches already developed are based on the circuit-structure–checking rules for avoiding invalid circuit graphs.

The literature review on the subject of LCRQQ circuit synthesis [15], [13], [44], [54], [94], claims that the main conventional prohibitions for connections during circuit evolution are as follows in Table 3-3:

Table 3-3. The list of those prohibitions that are popular for bipolar transistor connections during a circuit synthesis.

	Transistors are banned from joining their emitters to collectors
	Transistors are banned from connecting their base to a voltage source
	Transistors are banned from connecting their base to the ground
	Transistors are banned from connecting their base to outputs

	n-p-n transistors are banned from connecting their emitter to a positive voltage source and their collector to a negative voltage source
	p-n-p transistors are banned from connecting their emitter to a negative voltage source and their collector to a positive voltage source

As it has already been declared in Section 3.5, the REMP is set as fundamental. That is, any kind of connections must have an equal probability of appearing at every component, node and pin of a circuit. The releasing of bans in Table 3-3 is unconstraining on the evolution of LCRQQ circuits from Constraints №1 listed in Section 2.3.3.

3.8 Experiments 1-3: Constrained vs. Unconstrained Evolution

In the frame of low-pass filter experiments, a total of 103 circuits were evolved. The best seven of them have been chosen to be presented in detail in Chapter 3, because, according to the structure of this thesis drawn in Figure 1-4, seven are sufficient for the discovery of all the required features of the frame system. In Section 3.10, three circuits are presented which have been evolved in the frame of the ILG technique. Two LC filters are evolved by constrained and unconstrained evolution. In the end of the section, the first three-component circuit LCR low-pass filters were evolved with unconstrained evolution.

In this section, the framework system is tested with a low-pass filter and a computational circuit.

In three ILG-based experiments below, several issues have been tackled including:

- Testing the framework described in this section;

- Testing the constrained evolution approach;
- Testing the unconstrained evolution approach;
- A comparison of all three tests in terms of the example of a low-pass filter.

3.8.1 Task setting

A low-pass filter passes low frequencies fairly well, but attenuates high frequencies. An ideal low-pass filter completely eliminates all frequencies above the cut-off frequency while passing those below unchanged at the transition band infinitesimal (Figure 3-15). This can be realized mathematically by multiplying with the rectangular function in the frequency domain or - equivalently - convolution with a sync function in the time domain.

However, the ideal filter is not realizable; otherwise, the filter would need to predict the future and have infinite knowledge of the past in order to perform the convolution. Real filters for real-time applications approximate the ideal filter by delaying the signal for a small period of time, allowing them to "see" a little bit into the future. Greater accuracy in approximation requires a longer delay. Two types of filters are shown by Figure 3-15: an ideal one with no transition band at all and the real one with the transition band.

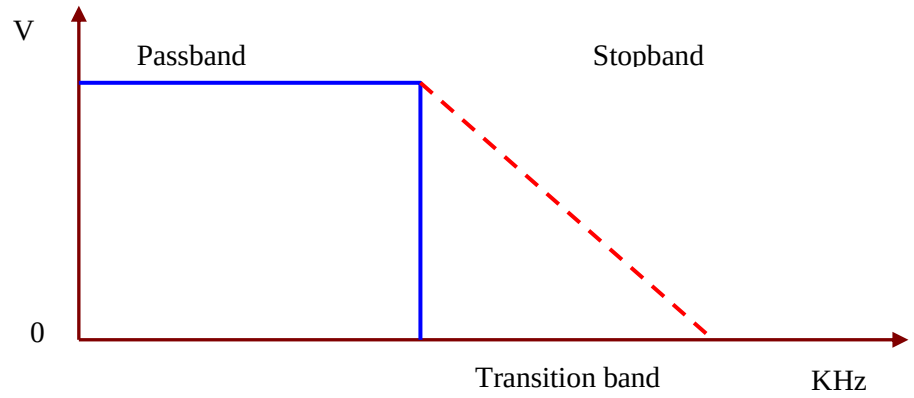


Figure 3-15. The circuit response of an ideal and a real low-pass filter. The last one is dashed with the transition band.

The goal for the framework evolutionary system thus constructed is to design a filter with an AC input signal with a 2 volt amplitude. The filter has a passband below 1kHz with voltage values between 970mV and 1 volt and has a stopband above 2kHz with voltage values between 0 volts and 1mV. This corresponds to a passband ripple of at most 0.3 decibels and a stopband attenuation of at least 60 decibels. The circuit is to be driven from a source with a source resistance of 1 kΩ and terminated in a load of 1 kΩ.

The embryo-circuit refers to the elements that are definitely known as essential for the target circuit, that stay unchangeable during all the evolution and take place in each circuit netlist. In the case of a low pass filter as a target circuit, there are three such kinds of elements: the AC voltage source, the source resistance and the load resistance. The embryo circuit is defined in a similar manner to the most popular case, where the circuit is driven by an incoming AC voltage source with a 2V amplitude: it has a source resistance R_{source}=1kΩ and a load resistance R_{load}=1kΩ (Figure 3-16). The output voltage is measured on the pins of the Rload.

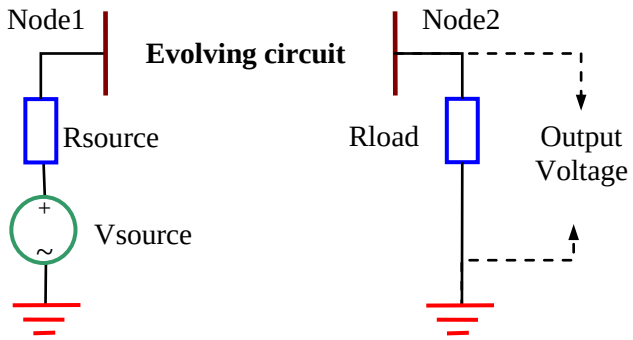


Figure 3-16. Embryo circuit for a low-pass filter.

3.8.2 Fitness Function

An AC-analysis is performed along 96 points between 1 Hz and 100 kHz (19 per decade), measuring the absolute deviation voltage between the ideal value and the value produced by PSPICE. The fitness evaluation is set in the analogy with [12], i.e. it is

distinguished as an acceptable voltage in the passband between 970 mV and 1 V and a voltage in the stopband between 0 V and 1 mV:

$$F_1 = \sum_{i=0}^{p} | V_{ideal}^{i} - V_{measured}^{i} |,$$

where V_{ideal}^{i} is the voltage at the *i*-th point for an ideal filter and $V_{measured}^{i}$ is the voltage at the *i*-th point obtained for the evolved filter; *p* is a number of points evaluated in both the stopband and the passbands, equalling 96. The voltage at any other location is as unacceptable, punishing it as it follows $F_2 = a \times F_1$, where ***a=10***; for reasons of comparison, it is taken as the same as in [12], where multitude of low-pass filters are evolved.

The transition band - consisting of five points between 1 kHz and 2 kHz - is regarded as the "don't care" band, where the fitness value is supposed to be equal to zero.

3.8.3 Initial settings

Experimentally, it has been established that the disruptive selection scheme [97] suits well: only 9% of the best chromosomes and 1% of the worst ones are to be chosen for the next generation. Being chosen for the next generation, each chromosome contributes 10% of the next population size, i.e. a total of 10% of the selected chromosomes generate 100% of the population of the next generation. A static mutation rate of 5% is then applied to each chromosome, randomly changing with equal probability at every loci of a gene. The evolutionary strategy is the simpler evolutionary algorithm, because it does not contain the recombination stage.

A population size of 20,000[4] chromosomes is set. It was decided to use a larger-population-size approach because it provides the advantage in speed: PSPICE requires some time for starting up, downloading the cir-file and web-licensing. This is to say that

[4] Despite the good results have been received with this population size, there are no reasons why this size should not be increased. In fact, in here the author is only driven by convenience of processing the PSPICE out-file, which size in 20,000 population case reaches 210MB. It is used PC Pentium-4, 3GHz, RAM 1GB.

10-20 seconds are required for each generation to be loaded, regardless of the population size.

The termination criteria are set as either exceeding the chromosome length of 28 genes or the running of 20 contentious generations without any improvement of the fitness value of the best chromosome.

For Experiment 1 - as the constraint for the evolution - two special subroutines have been created for the framework system. The first one checks circuits for invalidities up to five nodes in the chain around the new/mutated element on the floating nodes. The second rule checks up to four elements around the new/mutated element, whether or not they are involved in the inductor/voltage source loop. If the chromosome is found to be "crippled" - i.e. either floating nodes or inductor loops or other invalidities are found - it is not sent for evaluation.

3.8.4 Experiment 1-2: Unconstrained vs. constrained evolution of LC circuits

The purpose of the experiments below is to compare constrained and unconstrained evolutions on the example of a LC low-pass filter using inductors and capacitors. For both constrained and unconstrained cases, thirty nine experiments have been run with different seeds for a random number generator (RNG): 1-39. All 39 experimental results are shown in Table 3-4.

The best result for constrained evolution has been obtained at chromosome 11,863 of generation No.61 (20,000×60+11,863=1,211,863 individuals), with 26 elements (without embryo) with a best fitness value of 0.0041. The schematic and the voltage response of the best circuit are shown by Figure 3-17. The non-monotonic filter is received with the following features: the maximum absolute attenuation in the passband is 0.0015dB and the maximum attenuation in the stopband is -66dB.

Table 3-4. The experimental results for 39 different seeds of a random number generator

(RNG). Against each RNG seed, there are two values: the best fitness value reached and the generation number when this fitness value appeared.

Seed for RNG	Constrained evolution		Unconstrained evolution		Seed for RNG	Constrained evolution		Unconstrained evolution	
	Best fitness	**Gen. No**	**Best fitness**	**Gen. No**		**Best fitness**	**Gen. No**	**Best fitness**	**Gen. No**
39	0,0278	55	0,0052	69	**19**	0,0181	53	0,0247	61
38	0,0230	91	0,0279	61	**18**	0,0226	54	0,0137	56
37	0,0220	63	0,0083	62	**17**	0,0118	60	0,0147	56
36	0,0201	63	0,0046	69	**16**	0,0087	63	0,0192	61
35	0,0161	57	0,0081	73	**15**	0,0120	62	0,0093	67
34	0,0066	55	0,0180	65	**14**	0,0220	57	0,0139	64
33	0,0217	64	0,0182	57	**13**	0,0230	51	0,0168	59
32	0,0130	62	0,0184	68	**12**	0,0236	62	0,0132	70
31	0,0095	56	0,0120	62	**11**	0,0246	57	0,0129	62
30	0,0049	62	0,0053	65	**10**	0,0261	62	0,0209	69
29	0,0105	65	0,0143	63	**9**	0,0221	61	0,0134	57
28	0,0081	61	0,0082	62	**8**	0,0119	56	0,0118	59
27	0,0313	54	0,0064	68	**7**	0,0090	56	**0,0029**	**60**
26	0,0080	64	0,0067	61	**6**	0,0055	56	0,0101	57
25	0,0082	55	0,0108	61	**5**	0,0119	58	0,0099	64
24	**0,0041**	**61**	0,0124	60	**4**	0,0103	57	0,0179	59
23	0,0133	60	0,0034	55	**3**	0,0239	55	0,0124	58
22	0,0216	57	0,0031	66	**2**	0,0219	59	0,0059	70
21	0,0244	57	0,0295	67	**1**	0,0244	53	0,0166	63
20	0,0109	56	0,0113	58	**0**	**0,0164**	**59,2**	**0,0126**	**62,7**

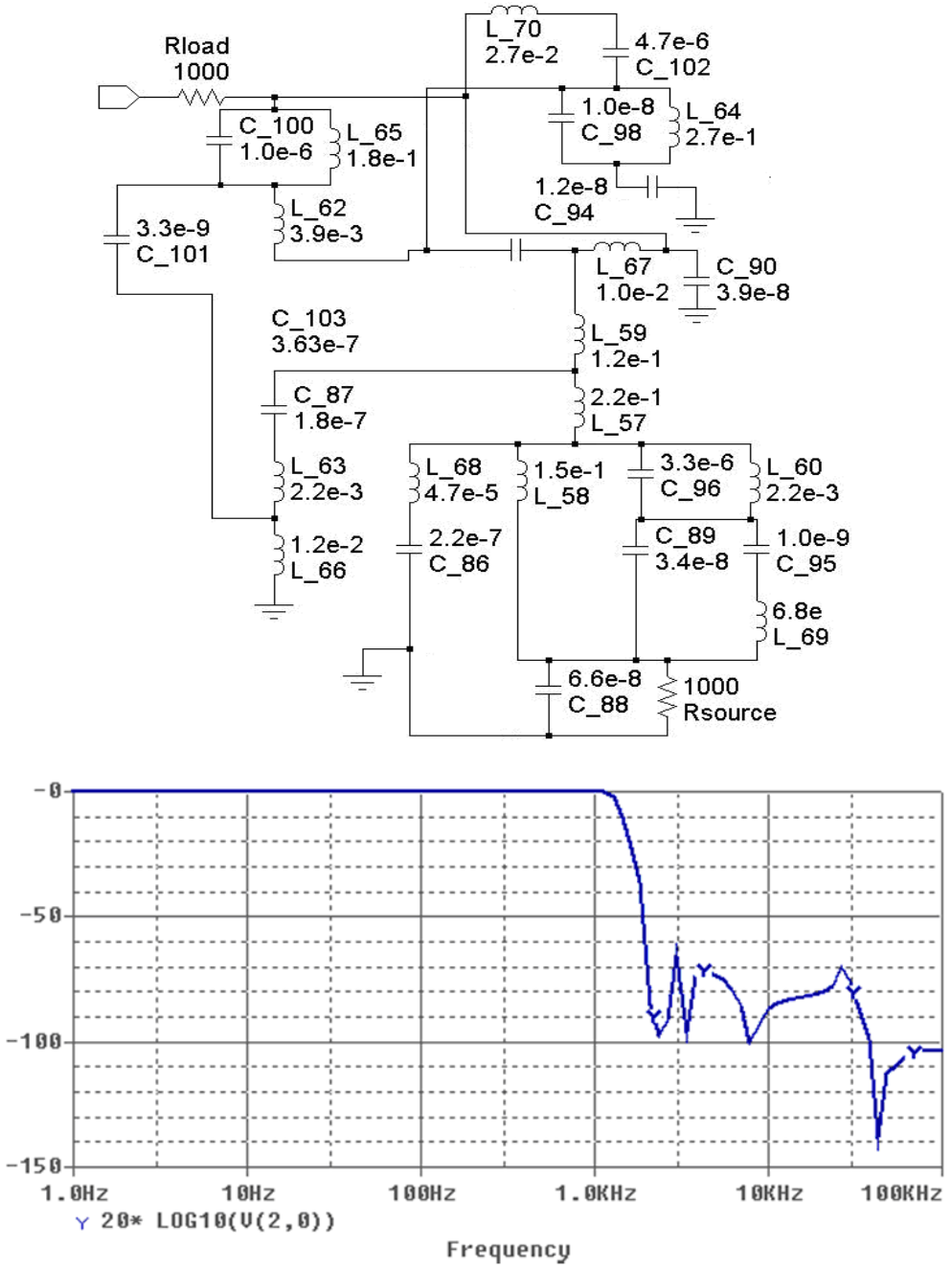


Figure 3-17. The schematic after pruning and the voltage response of the best low-pass filter evolved with constrained evolution in Experiment 1.

The best result for *unconstrained evolution* was obtained at chromosome 19,993 of generation No.60 (1,199,993 individuals) with a best fitness value of 0.002855, which is 44% better than the best achieved through constrained evolution. The schematic after pruning and the voltage response of the best circuit are shown by Figure 3-18. As can be

seen, a non-monotonic filter is evolved, consisting of 27 elements (without embryo) among which 3 are the R-support and with the following features: the maximum absolute attenuation in the passband is 0.00118dB and the maximum attenuation in the stopband is -69dB.

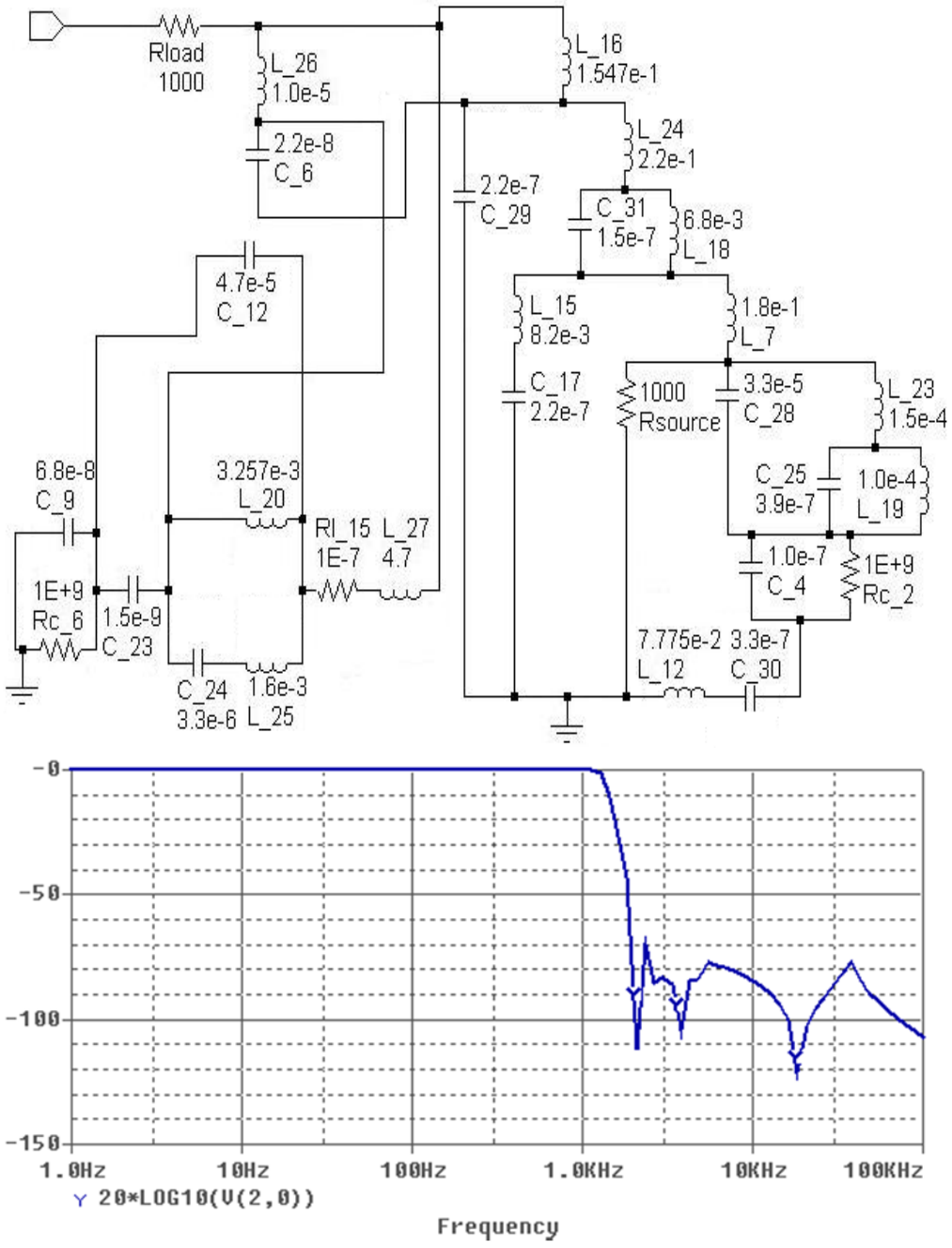


Figure 3-18. The schematic after pruning and the voltage response of the best low-pass filter evolved with unconstrained evolution in Experiment 2.

3.8.5 Experiment 3: Unconstrained evolution of LCR circuits

Additionally, with regard to the previously evolved filters in this experiment, the purpose is set so as to apply the newly developed methodology of unconstrained evolution for the evolution of the LCR low-pass filter using inductors, resistors and capacitors. The task is more sophisticated due to the larger space of potential solutions to search in. The latter is due to the third dimension which is now added and which is available for the evolution on the place of the *name loci* in each gene of a chromosome. In this and in all other experiments until the end of Chapter 3 (except for "Experiment 8: Evolution of cube root circuit"), only LCR circuits will be considered.

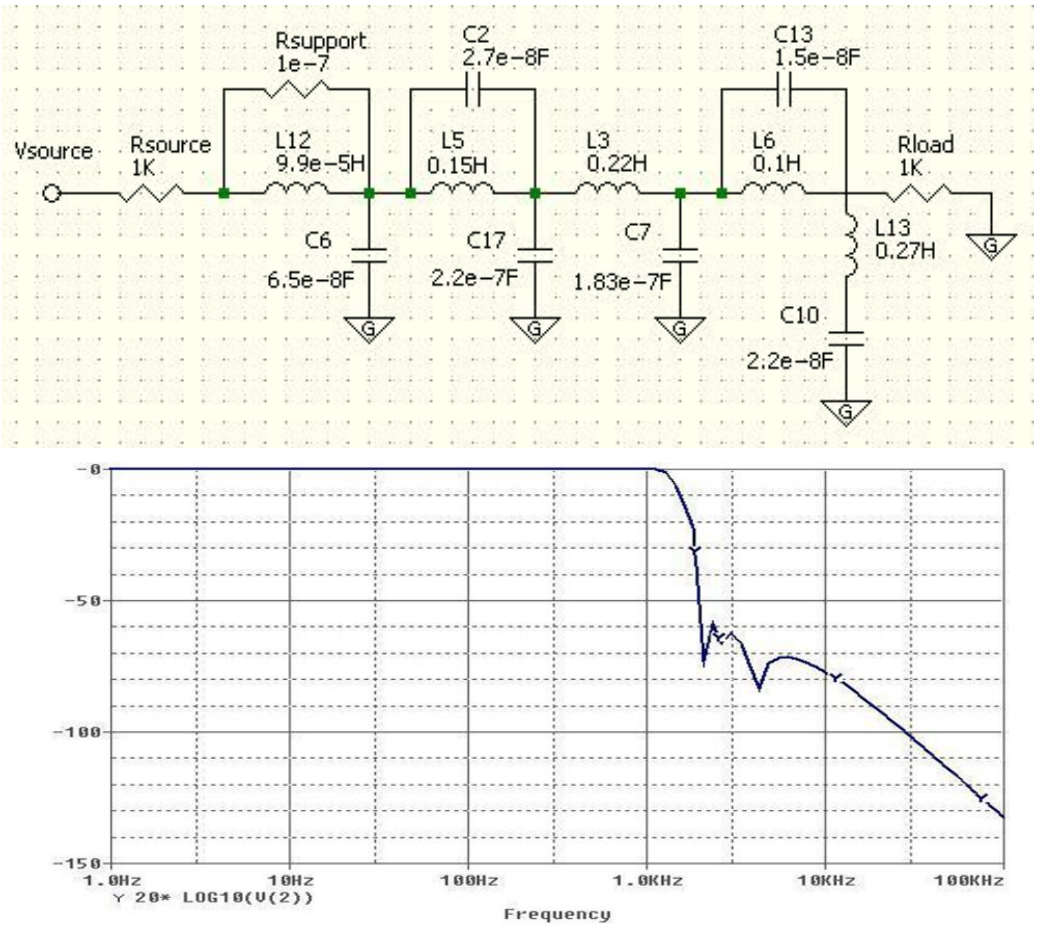


Figure 3-19. The schematic after pruning and the voltage response of the best LCR low-pass filter evolved in Experiment 3.

The best result out of five attempts has been obtained at chromosome 9,790 of generation No.42 (20,000×41+9,790=829,790 individuals), with 12 elements before pruning (without embryo), among which one is R-support and with a fitness value of 0.009693. The schematic after pruning and the voltage response of the best circuit are shown by Figure 3-19. The non-monotonic filter has the following features: the maximum absolute attenuation in the passband is 0.012dB and the maximum attenuation in the stopband is -59dB.

It should be noted that after pruning no resistors are left inside a circuit (except for R-support). This fact can be explained as that the evolution has "decided" that the low-pass filter is absolutely sufficient, with only two types of elements ("L" and "C").

During the experiments above, the amount of invalid circuits among all those randomly generated did not exceed - on average - 0.03%.

3.8.6 Results comparison

In this section, a comparison of the results of Experiments 1-3 is presented as well as the results received by others on the low-pass filter with the same properties.

In order to provide a fair comparison between those results obtained and those previously published, the author has validated each result using PSPICE. For this, each previously published schematic has been manually netlisted and the netlists have been run on PSPICE. The same fitness function is applied to all of the compared filters. By doing this, the filter characteristics for each circuit and its fitness values have been received, and all are summarized in Table 3-5. The proof of the correctness of this operation is verified by the perfect match between the fitness value obtained by the operation described and ones published in [12].

The experimental data on the LC filters in Table 3-5 shows that on average unconstrained evolution requires 62.7 generations, which is 5% longer than that of constrained one; however, the average fitness value (0.0126) is 30% better. A similar situation arises with the chromosome producing the best fitness: the unconstrained

evolution at RNG seed 7 runs out 60 generations and reaches a fitness value of 0.002855, which is 44% better than that of a constrained one (0.0041, reached at generation No.61 at RNG seed 24).

Table 3-5. Comparison table of filter and evolution characteristics among works published before and presently.

	Ideal filter	10order Chebyshev filter [102][12]	Koza 1 et al, [12] elliptic	Koza 2 et al, [12] ladder	Koza 3 et al, [12] bridge-T	Lohn et al, [13]	LCR-filter Unconstrained Experiment 3 [96]	LC-filter Constrained evolution Experiment 1 [33]	LC-filter Unconstrained evolution Experiment 2 [33]
Filter Characteristics									
Pass band, V	1	1	1	1	1	1	1	1	1
Stop band, V	0	0	0	0	0	0	0	0	0
Transition band, KHz	0	1	1	1	1	1	1	1	1
Maximum absolute attenuation in the pass-band, dB	0	0.035	0.179	0.0175	0.137	0.0144	0.012	0.0015	**0.0012**
% of improvement	-	***2866***	***15069***	***1383***	***11510***	***1120***	***917***	***27***	-
Maximum attenuation in the stop band, dB	- ∞	-83	-72	-61	-60	-59	-59	-66	**-69**
% of improvement*	-	*-20*	*-4*	***12***	***13***	***14***	***14***	***4***	-
Evolution characteristics									
Fitness value		0.0259	0.0805	0.0071	0.0502	0.0134	0.00969	0.0041	**0.0029**
% of improvement*	-	***809***	***2725***	***149***	***1661***	***369***	***240***	***44***	-
No. Elements		10	25	14	15	24	12	26	27
No. Individuals		-	N/A	2,048,000	N/A	997,000	829,790	1,211,863	1,199,993
Gen.No./Individ.No. at which the fitness is reached in Exp.2		25/ 520,000	16/ 340,000	34/ 700,000	20/ 420,000	31/ 640,000	-	-	-
Circuit simulator		-	SPICE				OrCAD PSPICE		

* The value "% of improvement" shows the correlation of the difference between the value above in the same column and the corresponding value in the column "LC Unconstrained evolution,"

This result can be explained as a result of two reasons. The analysis has shown that exploiting the circuit-structure-checking rules in a constrained evolution still allows a significant amount of mistaken circuits to be sent to simulation SW. In the case of the *structure checking rule* applied in Experiment 1, up to 15% of error circuits - on average- are allowed to be generated, which generates up to roughly 3% of the invalid chromosomes. In contrast, during the running of unconstrained evolution, the number of invalid graphs among all those randomly generated never exceeds 0.03%. In other words, the effective population size with unconstrained evolution was 15% larger than

that of a constrained one.

The second reason explaining the result is that each generation produced by unconstrained evolution contains more diverse chromosomes. The usage of R-support enabled the unconstrained evolution to create chromosomes which a constrained evolution will never allow to appear. For instance, the best circuit reached at RNG seed 7 contains three R-support elements, without which the circuit loses proper functionality and - thus - could never be replicated by constrained evolution.

Despite the fact that the evolution of the LCR-filter in Experiment 3 is worse than both of the LC-filters evolved in Experiments 1 and 2 - according to all of the characteristics in Table 3-5 - it displays features that are significantly better than the LC-filters designed in [12] and the LCR-filter designed in [13]. More attention will be paid later on for the evolution of LCR-filters.

In Table 3-5, all comparison characteristics are of two types: filter or evolution. The first three lines of the table show that those filters that are compared are of the same nature, and it is correct to make a comparison among them. The last two lines of the filter characteristics - attenuations in the stopband and the passband - are two major features that directly define the fitness value. The number of elements and individuals do not influence a fitness value; however, they are presented for a comparison because this property is regarded as an important feature of the unconventional design of the circuit that has been evolved by unconstrained open-ended evolution [51].

The last idea concerns the comparison of the system's ability with those developed by others. Line 15 of Table 3-5 shows the generation number and the number of the individual evaluated during Experiment 2 against each case. It should be noted that it took 3-times less individuals to evaluate than in [12], and 35% less individuals than in [13], to reach the same fitness.

Through Experiment 3, the first important version of the framework system is concluded. The results of the experiment are inspiring, showing that the methodology initiated is arranged in an appropriate manner. The framework system created will

become a skeleton for bringing up the more powerful system. The next section will suggest the OLG strategy as the first modernisation.

3.9 Oscillating Length Genotype (OLG) Varying Strategy

The main reason for introducing OLG is for chromosome length control so as to struggle against bloat [126], [127]. As previous experiments show, during the pruning there were several components (genes) that were removed. Furthermore, it is felt that 26-27 components for a circuit is quite a large size for a low pass filter. Therefore, the introduction of OLG should lead to more compact circuits, since OLG enables the deletion of single genes from chromosomes.

For this purpose, a new kind of mutation is introduced, namely "Delete element mutation" (DEM) which will be responsible for the deletion one gene if the best chromosome is shorter than the one viewed by some preset threshold value (for instance two genes, Figure 3-20). This is to say that after two fruitless consequent generations the difference becomes one gene, while after three fruitless consequent generations the difference becomes three genes and the DEM begins to work.

Now the whole picture of the mutation procedure is as follows. The "circuit structure mutation" (CSM) performs mutation over any of four loci of a randomly chosen gene. If the mutation comes to a pin connection, the whole structure of a circuit is changed. However, the total amount of components stays unchangeable during CSM. The number of components may only increase during ANEM. If the best chromosome has not been changed for several generations, the ANEM procedure may cause a difference between the size of the best individual and the size of the chromosomes in the rest of a population. If the difference exceeds two components (genes), DEM starts to randomly choose a gene in a chromosome for deletion.

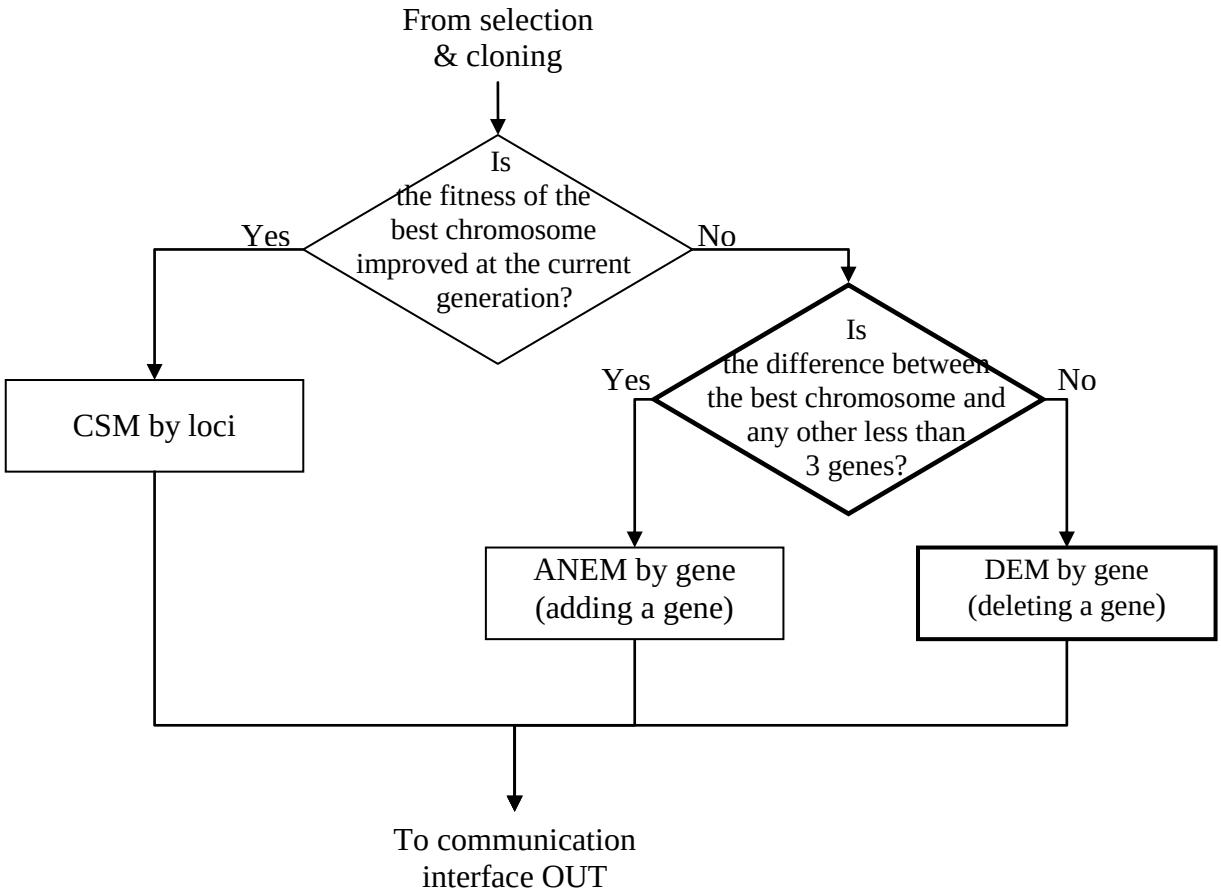


Figure 3-20. The flowchart of OLG-based mutation in the framework system. New terms are in bold.

When DEM removes a randomly chosen component, every new potential connection made by the floating pins has an equal probability of appearing. However, the last procedure may cause some pins to become unconnected, and then these floating pins connect to any of the circuit's nodes equiprobably. The example of the deletion of a 2-pin component may cause in total seven possible cases of structural recovery, as shown by Figure 3-21.

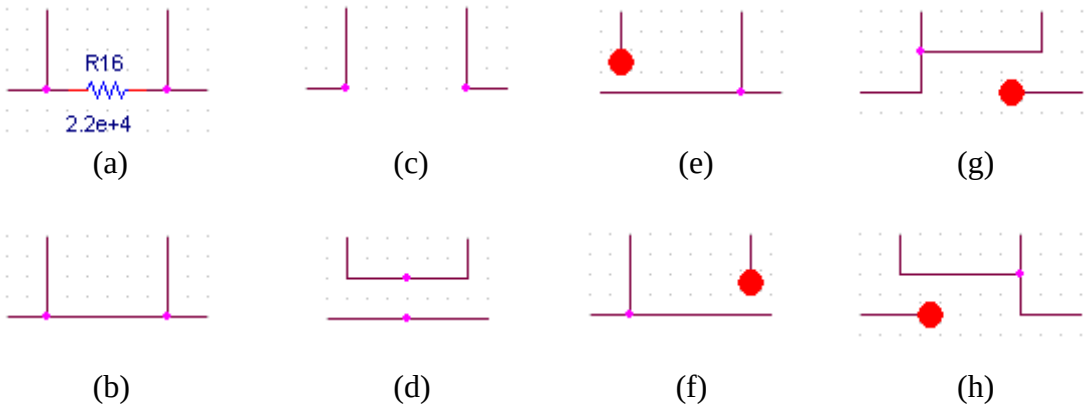


Figure 3-21. An example of a "delete element mutation". A resistor R9 with a parameter of 22kΩ (a) is removed and seven different cases of circuit structuring are shown (b-h). The floating pins - marked by red - have to search for other connections. A similar picture arises in the case of the removal of a 3-pin component and in the case of different circuit topologies.

The influence of OLG with the help of DEM on the fitness history of the best chromosome during evolution is shown by Figure 3-21. It represents the fragments of the circuit size and its fitness during evolution with OLG. The size of the circuit gradually grows up from 5 to 20 components, while the fitness value of the best circuit falls (improving). It can be seen that ANEM improved the fitness at generations 1-2, 6, 10, 12, 15, 18, 21, 25-27 ,30 and 33; DEM worked out at 8, 13, 17, 23-24, 28-29 and 31. The rest of the evolution is ruled by CSM (3-5, 7, 9, 11, 14, 16, 19-20, 22, 32 and 34-35). In general, the behaviour of the chromosome's length during evolution corresponds to the "oscillating length genotype strategy" mentioned in [28], where the chromosome's length can grow-up as well as shorten-down.

It also should be noticed that the proposed OLG technique differs from that presented in [28]. In [28], the genotype length was oscillating according to a sinusoidal function, whereas in the case presented here, the oscillation order becomes adaptive and is self-defined by evolution (Figure 3-22).

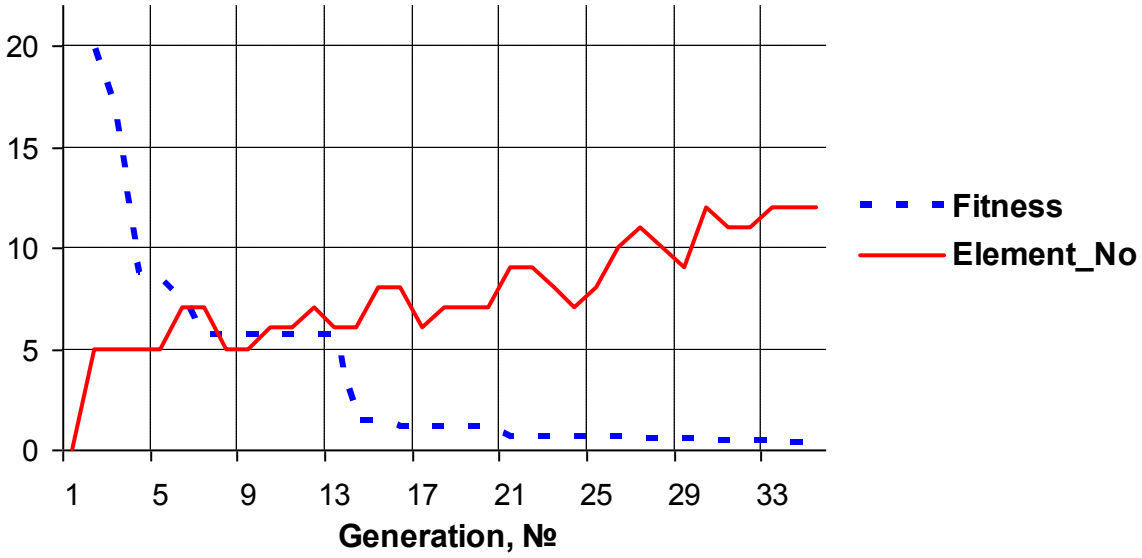


Figure 3-22. Chart fragments of the fitness value and size of the best circuit vs. generation. The ridges at the chromosome length curve are caused by the DEM procedure.

3.10 Experiments 4-5: Constrained vs. Unconstrained Evolution of LCR Circuits

The results of the evolution of two LCR circuits are presented in this section [103]. All of them are made with help of the OLG technique. In Experiment 6, the constrained evolution has been running. In Experiment 4, the unconstrained evolution is applied to the same target under the same conditions as in Experiment 5. The results are then compared.

3.10.1 Introduction

The low-pass filters for the experiments below have the same properties as before, namely the embryo and the stopband and the passband. Furthermore, the ES properties also are set as the same for facilitating further comparison, namely the FF, the SR, the mutation rate, and the population size, etc.

In the three OLG-based experiments below, several issues have been tackled, including:

- Testing the new OLG-based system which possesses a DEM operation;
- Testing the framework described in this section on the LCR circuits;
- Testing the constrained evolution approach;
- Testing the unconstrained evolution approach;
- Comparison of the two tests.

3.10.2 Experimental results

Termination criteria are set as either the running of 20 unfruitful consecutive generations or the reaching of a fitness of 10^{-3}. The results presented below are the best out of five attempts for both of the experiments performed, with five different seeds for the RNG.

3.10.2.1 Experiment 4: Constrained evolution of a LCR circuit

The purpose of the experiment is to evolve the LCR low-pass filter by means of constrained evolution and OLG. All the target properties and evolution conditions are used as in Experiments 1. The best result has been obtained at chromosome 17,308 of generation No.62 (20,000×61+17,308=1,236,308 individuals), with 27 elements (without embryo-circuit) with a best fitness value of 0.008084. The schematic and the voltage response of the best circuit are shown by Figure 3-23.

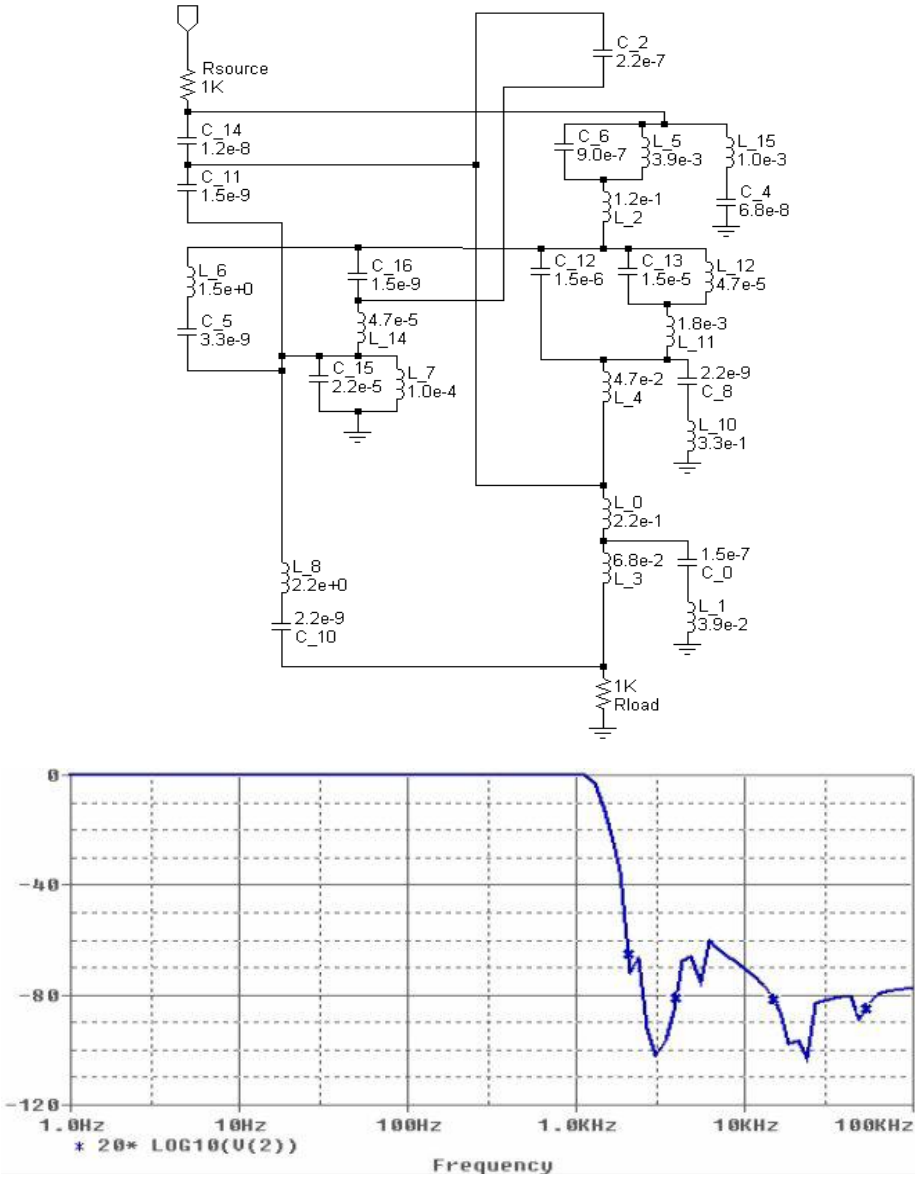


Figure 3-23. The schematic after simplification and the voltage response of the best low-pass filter evolved by Experiment 4.

As can be seen, in Figure 3-23 the non-monotonic filter has the following features: the maximum absolute attenuation in the passband is 0.0023dB and the maximum attenuation in the stopband is 60dB.

A special subroutine has been set that checks invalid circuits up to five nodes in the chain around a new/mutated element - whether or not they are floating - and a rule that checks up to four elements around a new/mutated element, whether or not they are

involved in the inductor/voltage source loop. This subroutine has been detaining the invalid circuits to be evaluated. Once any of the invalidities are found by this subroutine, the latter one sends back the chromosome to the mutation part of the framework system.

It should be noticed that, after pruning, no resistors are left inside a circuit as with Experiment 3.

3.10.2.2 Experiment 5: Unconstrained evolution of a LCR circuit

The purpose of this experiment is the unconstrained evolution of a low-pass filter in contrast to the one evolved previously by constrained evolution. All of the target properties and evolution conditions are as those used in Experiments 2 and 3. The best result has been obtained at chromosome 9,958 of generation No.75 (20,000×74+9,958=1,857,453 individuals), with 28 elements before simplification and with a best fitness value of 0.003916, which is two times better than that achieved through constrained evolution (Experiment 4). The schematic after pruning and the voltage response of the best circuit are shown by Figure 3-24.

As can be seen from Figure 3-24, there is a non-monotonic filter consisting of 16 elements (without embryo) among which 1 is R-support and with the following features: the maximum absolute attenuation in the passband is 0.0043dB, the maximum attenuation in the stopband is -69dB.

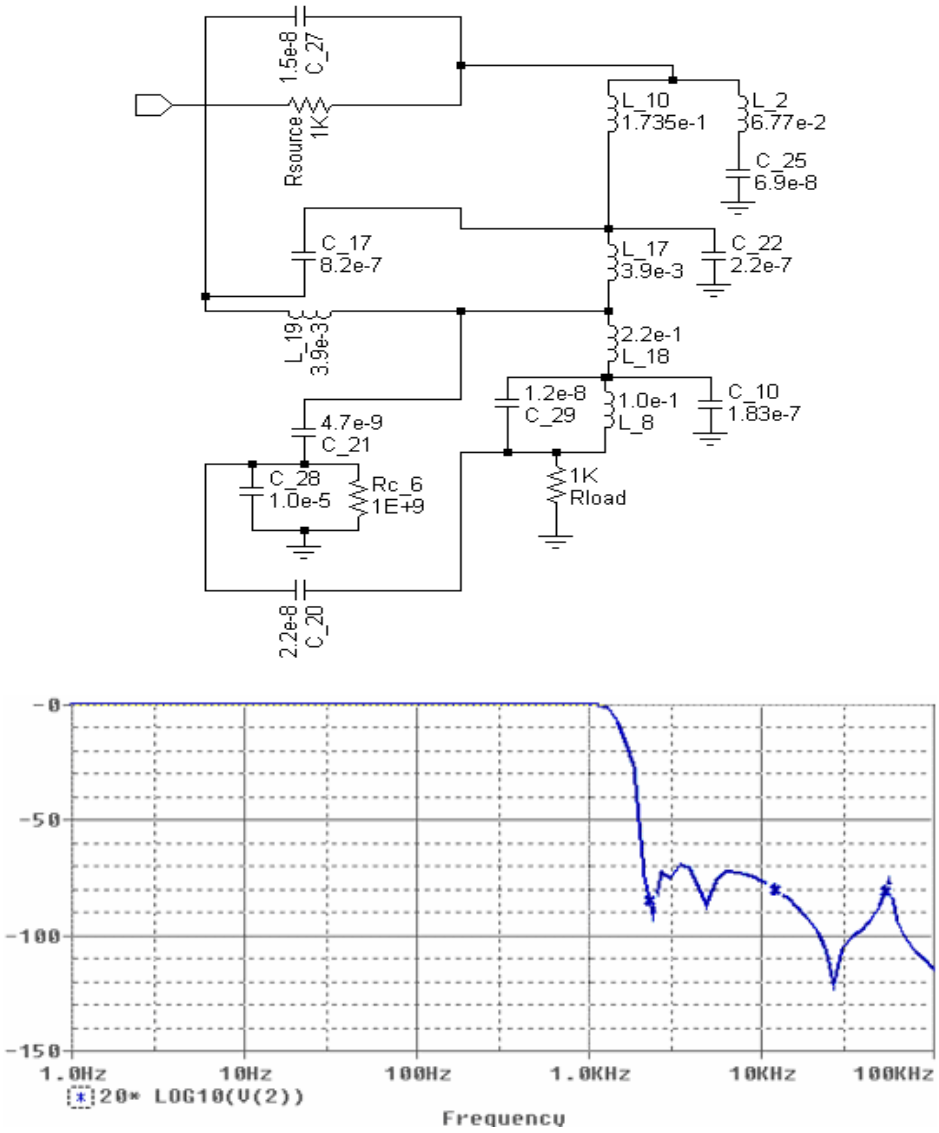


Figure 3-24. The schematic after pruning and the voltage response of the best low-pass filter evolved at Experiment 5.

3.10.3 The comparison of constrained and unconstrained evolutions

The experimental results show that exploiting the circuit-structure-checking rules in Experiment 5 still allows a significant amount of mistaken circuits to be sent to the simulation software. For example, in those cases where the rules are they still allowed up to around 15% of error circuits to be generated, which reduces the effective population size. In contrast, during the running of Experiment 5, the amount of invalid

graphs among all randomly generated ones did not exceed - on average - 0.05%.

In order to provide a fair comparison between the results obtained and those previously published ([12], [13], [17], [28]), each result is validated using PSPICE. By this, one can get the filter characteristics for each circuit and its fitness values, all of which are summarized by Table 3-6. The correct performance of the fitness function is verified by the perfect match between the fitness value received and the fitness value published in [12]. As can be seen by Table 3-6, in having only 16 elements the filter from Experiment 5 exceeds one that of Experiment 4, as well as by its other characteristics and its fitness value. In comparison with the best filter from [12], the fitness is improved by 82% at lower number of evolution attempts (generations) by 37%.

Table 3-6. Comparison table of the filter and evolution characteristics among works published before and present. N/A means that the data is not available.

	Ideal filter	10order Chebyshev filter [102]	Koza 1 et al, [12] elliptic	Koza 2 et al, [12] ladder	Koza 3 et al, [12] bridge-T	Lohn et al, [13]	Goh, et al . [17]	Zebulum et al. [28]	Constr. evolution Exper. 4 [103]	Unconstr. evolution Exper.5 [103]
Filter Characteristics										
Pass band, V	1	1	1	1	1	1	1	2	1	1
Stop band, V	0	0	0	0	0	0	0	0	0	0
Transition band, KHz	0	1	1	1	1	1	1	1	1	1
Maximum absolute attenuation in the pass-band, dB	0	0.035	0.179	0.0175	0.137	0.0144	0.042	0.188	0.0023	**0.0043**
% of improvement	-	714	4063	307	3086	235	877	4272	-47	-
Maximum attenuation in the stop band, dB	$-\infty$	-83	-72	-61	-60	-59	-34	-24	-60	**-69**
% of improvement	-	-20	-4	12	13	14	51	65	13	-
Evolution characteristics										
Fitness value	-	0.0259	0.0805	0.0071	0.0502	0.0134	0.186	N/A	0.0081	**0.0039**
% of improvement		564%	1964	82	1187	244	4664	1.5e+8	108	-
No. Elements		10	25	14	15	24	12	10	27	16
No. Individuals		-	N/A	2,048,000	N/A	997,000	20,200	320,000	1,236,308	1,489,958
Gen.No./Individ.No. at which the fitness is reached in Exp.5		24/ 500,000	13/ 280,000	55/ 1,120,000	17/ 360,000	32/ 660,000	5/ 120,000	-	-	-
Circuit simulator		-	MicroSim			SPICE	Micro Sim	SMASH	OrCAD	

The value "% of improvement" shows the correlation between the value above in the same column and the corresponding value in the column "Unconstrained evolution."

The comparison of the system ability with those developed by others again provides a significant advantage, except with [17]. Line 15 of Table 3-6 shows the generation number and the number of individuals evaluated during Experiment 4 against each case. It should be noted that it took almost twice less individuals to evaluate than in [12] and 34% less individuals than in [13] to reach the same fitness.

The analysis of the results provides another two discoveries. First, in both circuits there is only one R-support element left after simplification (in Experiment 5). Second, despite the fact that the resistor had an equal probability of being chosen as an inductor or a capacitor the best circuits that were evolved do not contain resistance at all. Further analysis show that throughout Experiment 5 the amount of R-support elements in the best circuits on average does not exceed 2% and in the rest of the circuits does not exceed 1.5%. On one hand, neither the R-support nor the resistors are essential for the functionality of the low-pass filter and they drastically increase the search space; on the other hand, both of them improve the characteristics of the filters thereby evolved.

The only - and the obvious - explanation of both these discoveries is that the evolution has used the resistance and R-support elements in neutral networks. Neutral networks have been already applied and are regarded as the crucially important factor in avoiding the local optimums and in reaching successful results [38], [17], [78], [82], [100], [101].

3.11 Experiments 6-7: Long and Short Transition Band LCR Low-Pass Filters

Within the following experiments, two LCR low-pass filters have been evolved. The first one - with a longer transition band of 1 KHz - is intended to tune up the methodology of unconstrained evolution. The second experiment is targeted towards the more sophisticated task of a low-pass filter with a shorter transition band of 0.4 KHz. This is one of the first attempts in this area to evolve a “close-to-ideal” low-pass filter [106].

The low-pass filters could be of two types: LC and LCR. The following works have concentrated on the design of LC low-pass filters: [12], [17] and [18]. The works that have considered LCR low-pass filters are: [13], [18], [28], [45] and [78]. The low-pass filter, if the proper R-load and R-source are provided, is absolutely sufficient with only two types of elements (L and C). However, in this experiment the more complex tasks for the evolution of LCR filters are considered, which corresponds to the overall strategy of the thesis, according to which the system under development aims towards the synthesis of highly complex analogue circuits. Contrary to a filter with the 0.4KHz transition band, the evolution of a filter with a 1kHz transition band has been investigated by a number of researchers ([12], [17], [18], [13], [28] and [78]) whose results have been taken for comparison.

In the two OLG-based experiments below several issues have been tackled, including:

- Testing the OLG-based framework described in this section;
- Testing the unconstrained evolution approach on an LCR low-pass filter with a standard transition band;
- Testing the unconstrained evolution approach on an LCR low-pass filter with a short transition band;
- Comparison of all the tests.

3.11.1 Experiment 6: LCR Low-pass filter with a transition band of 1KHz

The best result has been obtained at chromosome 17,453 of generation No.93 (20,000×92+17,453=1,857,453 individuals) with 28 elements before pruning, and with a best fitness value of 0.002445. The schematic after simplification and the voltage

response of the best circuit are shown by Figure 3-25. It should be noticed that after pruning no resistors are left inside a circuit, as with Experiments 3 and 4.

As can be seen, the non-monotonic filter has been received, consisting of 21 elements (without embryo) with the following features: the maximum absolute attenuation in the passband is 0.0028dB and the maximum attenuation in the stopband is -80dB.

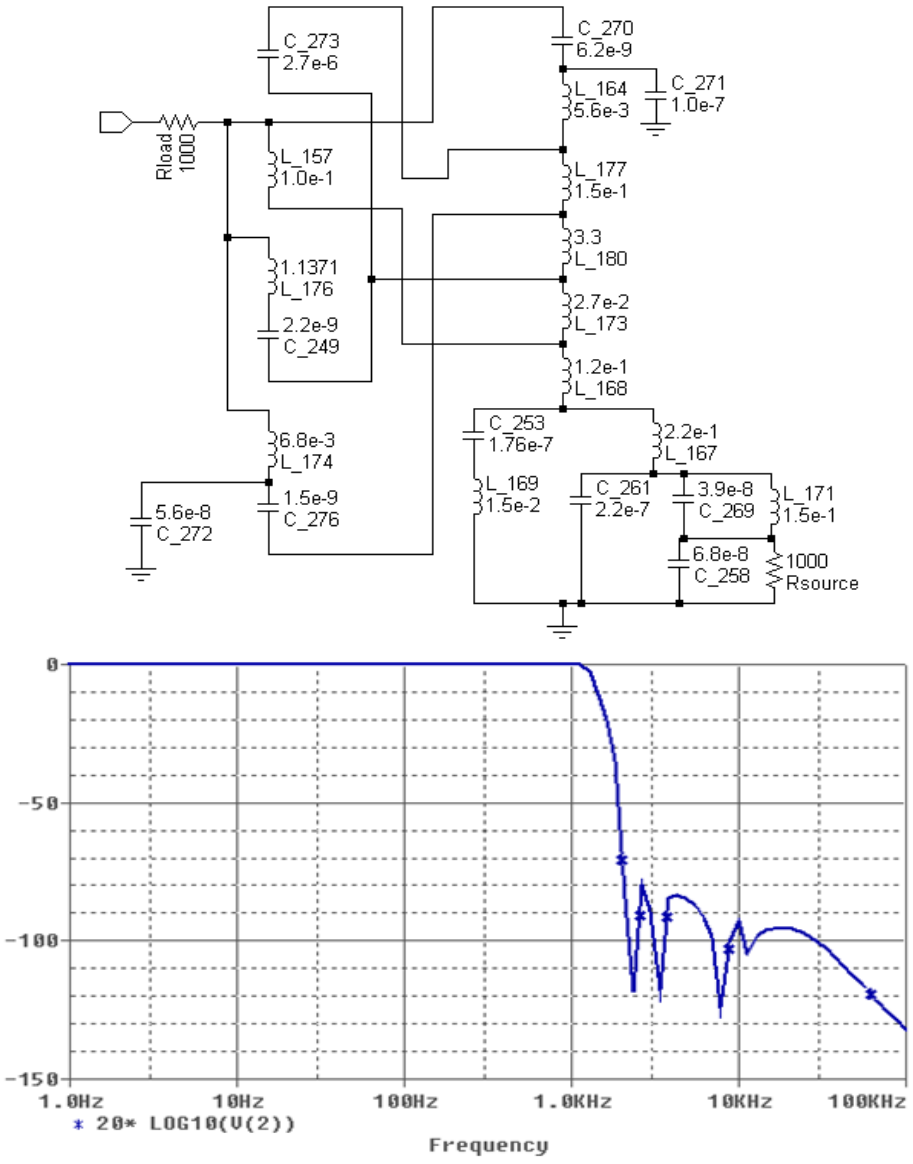


Figure 3-25. The schematic after pruning and the voltage response of the best low-pass filter evolved with a transition band of 1KHz in Experiment 6.

3.11.2 Experiment 7: LCR Low-pass filter with a transition band of 0.4 KHz

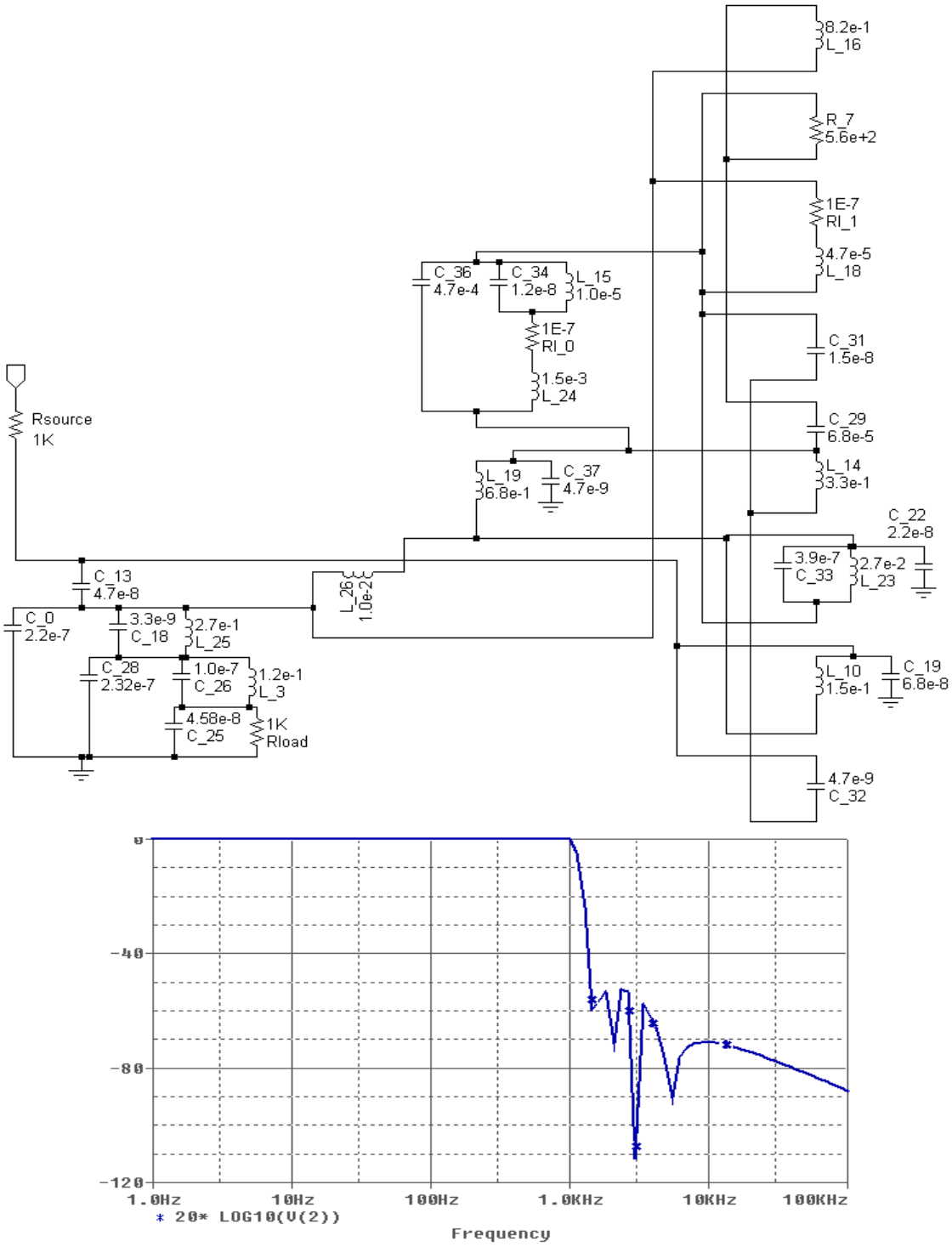


Figure 3-26. The schematic after simplification and the voltage response of the best low-pass filter with a transition band of 0.4KHz in Experiment 7.

The best results for the 0.4 KHz transition band filter have been obtained at chromosome 19,275 of generation No.85 (20,000×85+19,275=1,699,275 individuals), with a best fitness value of 0.021018. The schematic after simplification and the voltage response are shown by Figure 3-26. The non-monotonic filter consists of 29 elements (without embryo), among which two R-support elements and one resistor show the following features: the maximum absolute attenuation in the passband is 0.01513dB and the maximum attenuation in the stopband is -53dB.

3.11.3 Results analysis

In order to provide for a fair comparison between the circuits obtained and those previously published ([12], [13], [17], [18], [28] and [78]), each result has been put through PSPICE. By this, the filter characteristics for the circuits and their fitness values which have been obtained are all summarized by Table 3-7. The correct performance of the fitness function is verified by a perfect match between the fitness value which has been obtained and the fitness value published in [12]. As can be seen by Table 3-7, the unconstrained evolution exceeds the constrained evolution by the filter and the evolution characteristics.

The comparison of the system's ability with those developed by others - as in previous experiments - provides a significant advantage, except as with [17]. Line 15 of Table 3-7 shows the generation number and the number of individuals evaluated during Experiment 6 against each case. It should be noted that it took twice less individuals to evaluate than in [12] and 18% less individuals than in [13] to reach the same fitness.

Table 3-7. Comparison table of the filter and evolution characteristics among the results published before and within this work. N/A means that the data is not available.

	Ideal filter	10order Chebyshev filter [102]	Koza 1 et al. [12] elliptic	Koza 2 et al. [12] ladder	Koza 3 et al. [12] bridge-T	Lohn et al. [13]	Goh, et al. [17]	Zebulum et al. [28]	Ando, et al. [78]	Exp. 6, 1KHz filter [106]	Exp. 7, 0.4KHz filter [106]
Filter Characteristics											
Pass band, V	1	1	1	1	1	1	1	2	1	1	1
Stop band, V	0	0	0	0	0	0	0	0	0	0	0

Transition band length, KHz	0	1	0.934	0.934	0.934	1	1	1	0.3	1	0.4
Maximum absolute attenuation in the pass-band, dB	0	0.035	0.179	0.0175	0.137	0.0144	0.042	0.188	N/A	**0.0028**	0.0151
% of improvement		*-1350*	*-6493*	*-725*	*-4993*	*-614*	*-1600*	*-6814*			
Maximum attenuation in the stop band, dB	- ∞	-83	-72	-61	-60	-59	-34	-24	N/A	**-80**	-53
% of improvement		*-4*	*10*	*24*	*25*	*26*	*58*	*70*			
Evolution characteristics											
Fitness value		0.0259	0.0805	0.0071	0.0502	0.0134	0.1858	N/A			
% of improvement		*959*	*3192*	*190*	*1953*	*447*	*7499*	*2E+6*	N/A	**0.0024**	0.021018
No. Elements	-	10	25	14	15	24	12	10	20	21	29
No. Evaluations		-	N/A	2,048,000	N/A	997,000	20,200	320,000	100,000	1,857,453	1,699,275
Gen.No./Individ.No. at which the fitness is reached in Exp.6		30/ 620,000	17/ 360,000	45/ 920,000	60/ 1,220,000	40/ 820,000	14/ 300,000	-	-	-	-
Circuit simulator		Spice, MicroSim				SPICE	Micro Sim	SMASH	Intrinsic EHW	OrCAD PSPICE	

The value "% of improvement" shows the correlation between the value above in the same column and the corresponding value in the column "Experiment 6, 1KHz filter."

3.11.4 Conclusion of Experiments 1-7

The process of the extrinsic evolutionary design of analogue circuits has always been constrained in the generation of only valid circuit graphs. However, the introduction of R-support elements can significantly reduce these constraints. The proposed technique is based on ES in combination with an OLG sweeping strategy. In Experiments 1-7, the developed system has been applied towards the LCR and LC low-pass filters, and it showed the superiority of the method over the conventional constrained evolutionary ones which were earlier applied towards analogue circuit design. The results obtained are in order to improve the previous attempts in the area through the characteristics of filters as well as through the features of evolution.

Thus, the instinctive wish to reduce the potential solution space for an evolutionary search, by which the circuit-structure-checking rules are usually justified, is not always

the best strategic manoeuvre for obtaining unconventional circuit designs.

Next, the developed methodology has been applied towards a more sophisticated task - the design of a filter with a shorter transition band of 0.4 KHz. The filter displays excellent characteristics.

An OLG sweeping strategy in conjunction with the capability of evolution to focus on limited genotype lengths has been developed. The results of the experiments agree with ones in [28]; here, the OLG strategy is one of the best for a low-pass filter design.

Further analysis of results reveals the implicit tendency of evolution to minimize the usage of resistance and R-support elements - such that the final solution could not contain them at all - and use them as the neutral elements inside the neutral networks. These two discoveries again emphasize the importance of neutral networks in the evolutionary search.

3.12 Experiment 8: Evolution of computational circuit

In this section, the first attempt is undertaken to evolve the RCQQ circuit [109]. It is made with help of the OLG technique and unconstrained evolution. This experiment is the last made by means of the framework system and in the frame of Chapter 3.

3.12.1 Introduction

Despite the difference between the previous and the following targets, they have the same embryo and the same ES properties, such as the SR, the mutation rate, etc.

In the three OLG-based experiments below several issues have been tackled, including:

- Testing the framework described in this chapter;

- Testing the unconstrained evolution approach;

- Testing the unconstrained evolution approach on a more challenging circuit that contains 4 components, including 3-pin components.

3.12.2 Fitness function and termination term

The target for the evolutionary search is to evolve an analogue circuit whose output voltage is the *cube root* of its input voltage. To enable ourselves to make an estimation of the final results from the experiment, the same fitness terms as in [12] have been set. That is, the PSPICE simulator is made to perform a DC sweep analysis at 21 equidistant voltages between –250 mV and +250 mV for the cube root. The fitness value is set to the sum, over these 21 fitness cases, of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit. The smaller the fitness value, the closer the circuit is to the target. It is set so that a given fitness will be penalized by 10 if the output voltage is not within 1% of the target voltage value. The error circuits are not analyzed by a simulator and are assigned to the worst fitness value.

For the termination criteria, they are set as the achievement of either of the following conditions: the fitness value does not improve over 20 generations or the best circuit reaches more than 100 elements, or else the best fitness value reaches 0.5, which corresponds to an average voltage deviation from the target of 0.02V per point.

3.12.3 Experimental results

The results presented are the best out of 20 runs on 10 different PCs with different seeds for the RNG. The ES with linear representation and OLG are utilized. The total population consisted of 30,000 individuals, with a mutation rate of 5%.

At generation No.3, the best individual (No.24,999) with three genes (in addition to embryo elements) showed a fitness of 65.57. The circuit that this chromosome describes is presented by Figure 3-27 and has two transistors and one resistor.

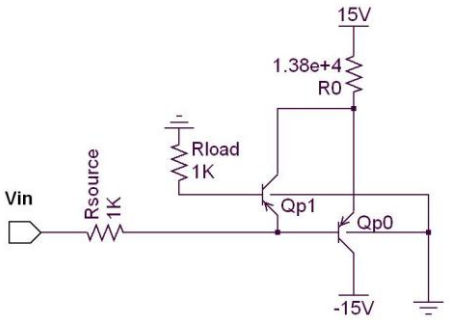


Figure 3-27. The best cube root circuit from generation No.3 of Experiment 8.

The next notable result appeared at generation No.15 (No.23,882) with 14 genes (in addition to embryo elements), which describes a circuit with 7 transistors, 1 diode (a transistor whose collector is connected to the base) and 6 resistors. This circuit, pictured by Figure 3-28, has a fitness of 5.53.

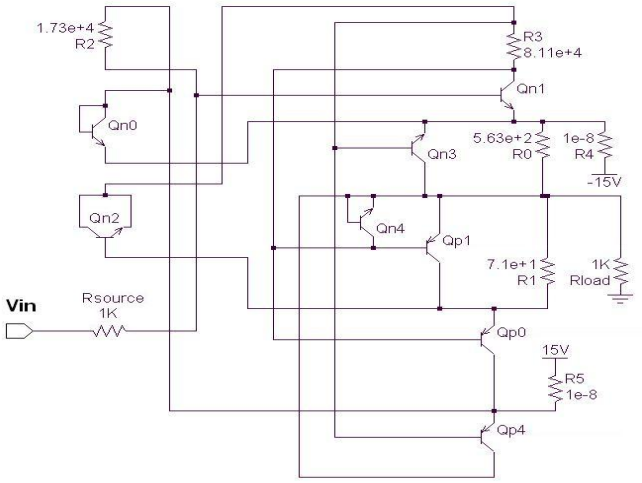


Figure 3-28. The best cube root circuit from generation No.15 of Experiment 8.

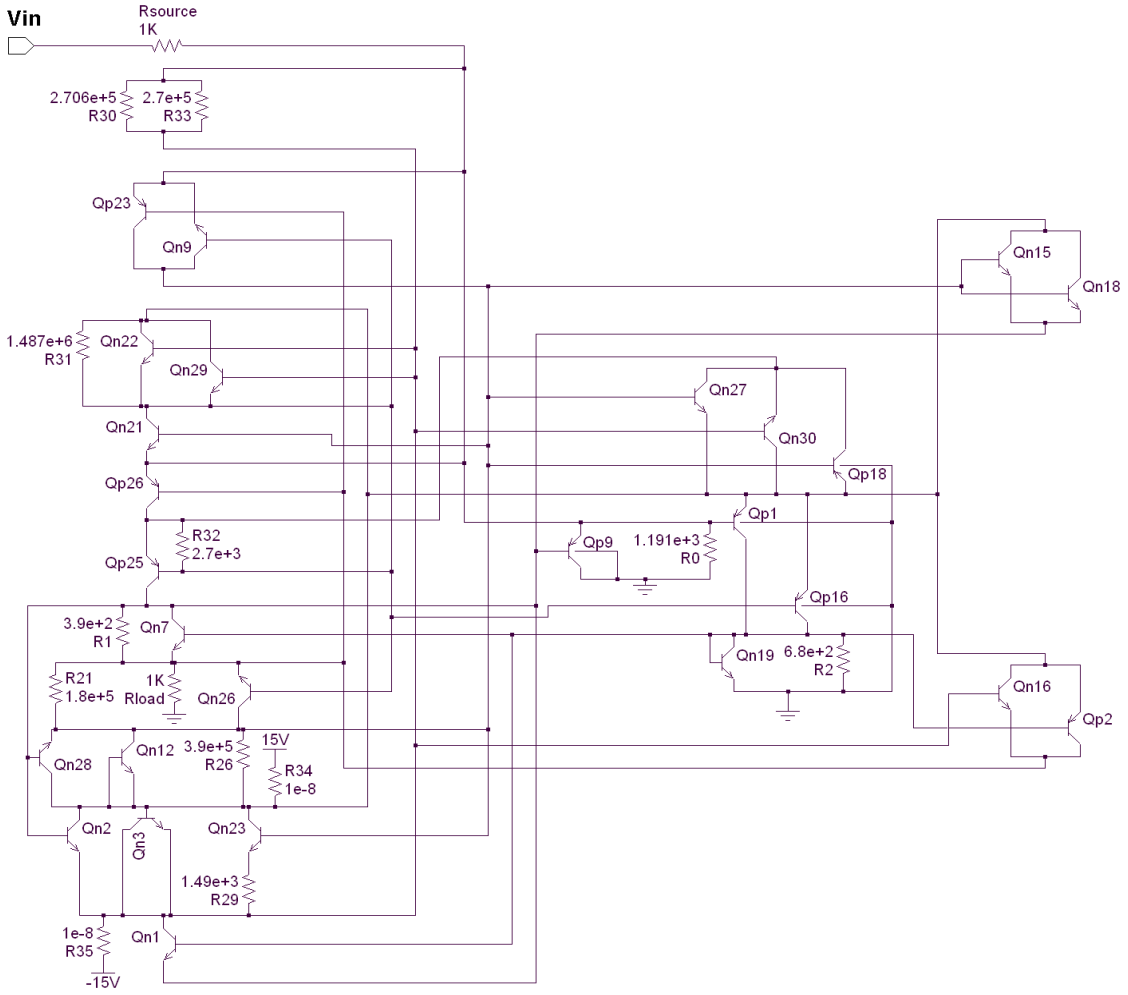


Figure 3-29. The best cube root circuit from generation No.133 of Experiment 8.

Finally, the circuit evolved at 133 generations (No.34318), and for which evolution had not been able to improve during the following 20 generations, appeared with a total of 38 elements (in addition to embryo elements): 24 transistors, 12 resistors and 2 diodes (Figure 3-29). The fitness of this circuit achieved 2.37.

The average number of invalid circuits per population is 4-5%, and most of them are non-convergent.

It is very rare to find the cube root computational circuit in the literature, and especially the schematics that were made by evolution. To decide on efficiency of the

proposed evolutionary technique, one work is directly addressed, namely [12]. Since the same fitness function is set, one can directly compare the circuits and their corresponding fitness values. The result of the comparison is presented by Table 3-8.

As can be seen from the Table 3-8, the results indicate that the stalling effect took place over the last part of the evolution, while at the beginning and in the middle the performance exceeded the results made by Koza et al. [12], finding more effectively functioning solutions with the same (3 genes) and with less (14 against 18) genes. One of the main reasons for the stalling effect is the weakness of the technique as applied towards the task [119]. This requires further development of the proposed technique.

Table 3-8. The comparison of the results received with those published in [12]

	1-st circuit		**2-nd circuit**		**3-d circuit**	
	Element number_1	**Fitness value_1**	**Element number_2**	**Fitness value_2**	**Element number_3**	**Fitness value_3**
Achieved in [12]	3	77.7	18	26.7	50	1.68
Achieved in this experiment [109]	3	65.6	14	5.53	38	2.27

3.12.4 Conclusion of Section 3.12

In this section, the unconstrained evolution with OLG is applied towards the analogue circuit design of the QR computational circuit performing the cube root function.

The method utilized here is much easier than that applied in [12]. While the last approach - with help of reusable sub-constructions - has successfully evolved circuits with a large amount of elements, the proposed method (as can be seen by Table 3-8) succeeds for small- and middle-sized circuits.

The computer resources in the attempt just presented are much lower, and equal to about 0.5mln chromosomes for a final solution (17*30000) against to about 11mln (for the second circuit).

The proposed method has shown its potential for further improvement by getting the fitness close to that in [12]. The shortage in fitness is almost the same (26%) as the gain in the element economy (24%). Moreover, the gain in computer resources is tremendous, at 90%. This comparison is encouraging because the computational cube root circuit is one of the largest circuits evolved by Koza [12], and - despite its attractive advantages - it is quite difficult to find another example of this circuit in the open sources.

The OLG sweeping strategy developed with the capability of evolution to focus on the limited genotype length dispersion has proved its powerful search capacity. Experiment 8 has indicated that the further strengthening of the framework system is required.

3.13 Summary of Chapter 3

With Chapter 3, the start of the proposed evolutionary system has been triggered. First of all, it has presented the direct representation technique, which is something of a standard in the area. Next, two basic mutations have been introduced, namely CSM and ANEM: these are the base for the ILG varying strategy. With ILG, the direct dynamic representation is concluded.

A very important principle has been introduced, namely the *Rule of equal mutation probabilities* (REMP). This basic principle represents the true spirit of the proposed approach, unconstraining the extrinsic evolution of analogue circuits. According to it, and during mutations, any kinds of connections are allowed without limit, having an equal probability of occurrence. To support this idea, the R-support elements have been introduced so as to tackle the issues caused by inductors and capacitors. With the last procedure, the evolution is becoming fully unconstrained. On the other hand, the

constrained approach has been created by enabling a circuit-checking-structure-rule in a form of a specially created subroutine so as to analyse and filter out every circuit with an invalid structure. Experiments 1-3 have shown that with ILG, unconstrained evolution gives more promising results than constrained evolution.

The analysis of the circuits derived during evolution has shown that one of the reasons why REMP may help to succeed in the described experiments is that REMP enabled creation of unconventional connections inside the circuits which brought the creation of neutral components and even sub-circuits and which in turn participated in neutral mutations. There are a lot of works in the area of EHW devoted to the importance of neutrality for the success of evolution [38], [82], [101].

The further development of the system was introduced through the DEM procedure. With DEM, the OLG varying strategy became available for utilisation by the system, enabling the creation of circuits that are potentially more economic in terms of their components than ILG. OLG allows evolution to decrease the number of genes in a chromosome. The second part of the experiments has proven that unconstrained evolution is more valuable than constrained evolution. Experiments 4-7 created low-pass filters that were even better than before in terms of their functioning and the economy of their components (Table 3-9).

As a test task, the low-pass filter has been chosen from the quite high number of circuits with the *initial level of complexity* (see Chapter 4), including high-pass filters, band pass and band stop filters, etc. The choice made was dictated by the wide range of papers dedicated to the low-pass filter. Moreover, the behaviour of low-pass filters between the frequencies of 1 KHz and 100 KHz - the cut-off frequency 1 KHz and the transition band 1 KHz dominating all the exemplified circuits - suggests an opportunity for a comparison.

It is notably that the evolutionary system of both approaches – both unconstrained and constrained - during the evolution of the seven low-pass filters presented and another 38 circuits (in the frame of Experiment 1-2) prefers only two types of elements as building blocks, namely L and C (ignoring resistors). The statistics show that only

1.5% of the total amount of components are resistors. But on the other hand, the filters evolved with the use of resistors have better features, even if they do not contain any resistor. This fact could be explained in that resistors can play the role of neutral components, composing neutral networks and carrying neutral mutations [38], [82], [101].

These comparisons enable us to make a strategic decision as to what kind of technique it is better to use and to trust. Both the OLG and the unconstrained techniques will be utilized as default techniques.

Figure 3-30 gives a general view of how the evolution takes place. All seven curves are put in one chart; among them the curve corresponding to the close-to-ideal filter stands out by its gentle slope. On average, up to generation No.20 evolution reaches a fitness of 0.5% of its initial value, while up to generation No.40 it reaches a fitness of 0.1%.

The failure of the system to improve the fitness of the cube root circuit during Experiment 8 - shown in the last column of Table 3-8 - pushes for further explorations of new techniques that, when integrated into the system, could help to succeed not only over the cube root circuit, but also in evolving any other circuit that belongs to a second complexity level.

Table 3-9. The aggregated comparison table of low-pass filters evolved by others and evolved in Experiments 1-7.

										Evolved in the frame of the Thesis						
	Ideal filter	10order Chebyshev filter [102]	oza 1 et al. 12] elliptic	oza 2 et al. 12] ladder	Koza 3 et al. [12] bridge-T	Lohn et al. [13]	Goh, et al. [17]	ebulum et al. [28]	Ando et al. [78]	ILG Con. LC Exp.1 [33]	ILG Unc. LC Exp.2 [33]	ILG Unc. LCR Exp.3 [96]	OLG Con. LCR Exp.4 [103]	OLG Unc. LCR Exp.5 [103]	OLG Unc. LCR Exp.6 [106]	OLG Unc. LCR Exp.7 Close-to-ideal [106]
Filter Characteristics																
Pass band, V	1	1	1	1	1	1	1	2	1	1	1	1	1	1	1	1
Stop band, V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Transition band, KHz	0	1	1	1	1	1	1	1	0.3	1	1	1	1	1	1	0.4
Maxim. absolute attenuation in pass-band, dB	0	0.035	0.179	0.0175	0.137	0.0144	0.042	0.188	N/A	0.0015	0.0012	0.012	0.0023	0.0043	0.0028	0.0151
Maxim attenuation in stop band, dB	- ∞	-83	-72	-61	-60	-59	-34	-24	N/A	-66	-69	-59	-60	-69	-80	-53
Evolution characteristics																
Fitness value		0.0259	0.0805	0.0071	0.0502	0.0134	0.1858	585.7665	N/A	0.0041	0.0029	0.00969	0.0081	0.0039	0.0024	0.021018
No. Elements	-	10	25	14	15	24	12	10	20	26	27	12	27	16	21	29
No. Individuals		-	N/A	2e+6	N/A	1e+6	20,200	320,000	1e+5	1,211,863	1,199,993	829,790	1,236,308	1,489,958	1,857,453	1,699,275
Circuit simulator		-	SPICE				Micro Sim	Smash	Intrinsic	PSPICE						

"Unc." is for Unconstrained evolution; "Con." is for Constrained evolution; "Exp." is for Experiment;

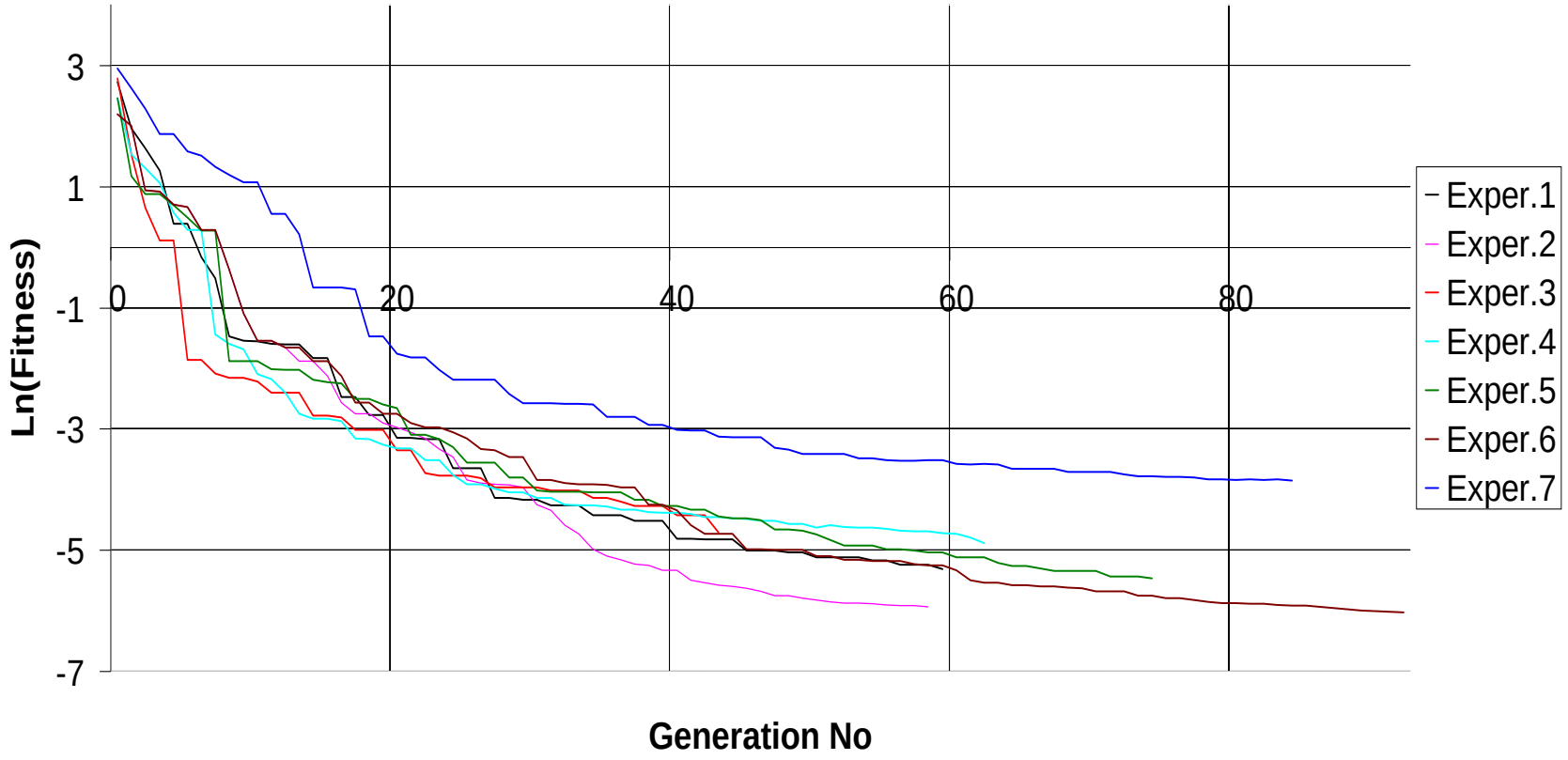


Figure 3-30. Evolution of seven low-pass filters in Experiments 1-7.

Chapter 4. The Individual-Level Differentiated Mutation Technique

In this chapter, a novel paradigm is suggested to deal with the *second level of complexity* circuits. According to the first part of this paradigm, the novel approach is suggested towards substructures and their utilization. A new mutation approach is formed into a new adaptive individual-level mutation technique. This important development of the proposed system is called *substructure reuse mutation* (SRM). SRM will be smoothly integrated inside the mutation procedure which makes the contribution to the building of *differentiated mutation technique*. It will play an important role in the evolution of the second level of complexity circuits.

The second part of the novel approach describes the development of SRM into a more global technique called the *differentiated mutation technique*, which will unite all the types of mutations into one highly-organized and well-coordinated procedure.

The range of the experiments tested both of the methodologies on the examples of the computational circuits and the 4-output voltage distributor circuit (VDC).

4.1 The Substructure Reuse Mutation

4.1.1 Introduction

The SRM - also called as the topological reuse [86] or memory paradigm [28] - is quite a popular technique in EHW [2]. The memory paradigm requires the existence of a central database, which will behave as a genetic memory; as fitter genes come up in the beginning of the evolution, they are stored and then reused as the genotypes grow in size.

One of the first attempts to reuse the substructures was undertaken by Koza in [12] when evolving analogue electronic circuits. Since then, almost every work in the area

utilizes this technique. The essence of the technique lies in the idea that it is not only random evolutionary per-locus mutations and the per-gene growth of a chromosome that are able to bring improvement to the search process, but also the "specially selected" fragments of chromosomes that may consist of between one and several genes. The questions that arise here relate only to from where to get these structures and how to use them. There are two approaches that are pertinent to the first question:

1. The first approach may be called a "long-term database approach" or a "knowledge-based approach" [2], [53], [86], [87]. Here, the creation of the substructure database involves almost the same laborious work as the evolutionary system's development. The multitude of sorted circuits from the past may go through attentive analysis for the sake of generalized substructures. The substructures even may be prepared manually [2]. The building blocks are stored in classified databases designed for particular evolutionary targets. This kind of approach mostly aims at the synthesis of trustworthy circuits and uses course-grained topologies such as current mirrors, differential pairs, active loads and cascade stages, etc.

2. On the other hand, what can be called as "heuristic approach" mostly uses substructures [12]-[13] that are produced by evolution and during evolution. For every particular task, evolution creates building blocks for that particular case. The designer does not participate in this process and does not try to understand how the topology works. The evolution uses the substructures as *ad hoc* building blocks.

If the first approach targets designs that are close by their structure to conventionally designed circuits, the second one is able to create unconventional solutions. In this work, the second approach is utilized.

4.1.2 The sources for substructures

When targets become more complex, an evolutionary search becomes difficult and the stalling effect becomes a problem. In this case, the necessity for an additional

process revival technique becomes essential. One such kind of technique is the SRM, which is applied in the frame of mutation operation. The idea is that the structures that were helpful during previous generations may help in the current one. Now, evolution acquires the memory and it is able to recall better building blocks when a “hard time” comes.

There are four main files that accompany the evolutionary process: 1) the *cir-file* which is formed by the system as an entry in PSPICE, 2) the *out-file* which is the result of the PSPICE analysis produced by PSPICE, 3) the *data-file* which is accumulates the useful information throughout the generations, including the best chromosome fitness (Figure 4-1), and 4) the *best-chromosome-file* which contains the PSPICE decks of the best chromosomes.

When it comes to applying the particular substructure, the system based on the information contained in the *data-file* turns to the *best-chromosome-file* to obtain the required chromosome. It finds the best individual among others of the same length. In Figure 4-1 the best individuals of lengths 2, 3, 4, and 5 are marked by red boxes. These are the best representatives in their classes. Recalled from the memory, the individual from a chromosome becomes a substructure after a special procedure. This procedure removes from a chromosome the components that belong to the embryo (source, input/output resistors, etc.). The floating pins that are left after that procedure are entitled to provide new connections of the substructure to a new main circuit. The details of that procedure for a one-input-one-output circuit are exemplified by Figure 4-2. The nodes for connections inside the main circuit are chosen with an equal probability from the total list of the circuit nodes.

When a substructure’s floating pins are connected to a circuit, the same rules as with ANEM take place, as described in Section 3.5.2. That is, each floating pin of a substructure could be associated with the pin of a new component to be added (Figures 3-11 and 3-12).

GenNo	BstNo	BstFt	ElNo	Crip	EvrFt
2	913	5,13E+02	2	16774	2687
3	29999	5,13E+02	2	214	1960
4	7	4,68E+02	3	156	1410
5	28625	3,28E+02	3	127	1234
6	28506	2,74E+02	3	107	923
7	29999	2,74E+02	3	129	523
8	29968	2,73E+02	3	91	462
9	15881	1,74E+02	4	371	389
10	29999	1,71E+02	4	161	323
11	29976	1,68E+02	4	138	284
12	29999	1,68E+02	4	130	268
13	29913	1,51E+02	5	252	258
14	29999	1,51E+02	5	170	235
15	29999	1,16E+02	5	115	195
16	29999	1,16E+02	5	168	167
17	24042	9,58E+01	6	509	156
18	25922	9,23E+01	6	359	155
19	29999	9,23E+01	6	342	153
20	29995	8,63E+01	7	188	99
21	29999	8,63E+01	7	198	98
22	29999	8,63E+01	7	489	96
23	29999	8,63E+01	7	537	95
24	23779	7,63E+01	8	357	94

Figure 4-1. The fragment of a *data-file*. By columns: 1-“Generation number”, 2- “The best chromosome number”, 3- “The best chromosome’s fitness value”, 4- “Gene (component) number”, 5- “The number of crippled chromosomes”, 6- “The average fitness of a population”. By the red rectangular boxes are indicated the chromosomes that have been taken as substructures.

The effectiveness of SRM directly depends on the size of the circuit to which the SRM is going to be applied to. Since the junction points for the substructure inside a circuit are under the choice of a random process, a substructure with a larger amount of floating pins for connection ***E1*** has more possible ways (N_{E1}) of being connected to a circuit containing ***N*** nodes than a substructure with fewer floating pins ***E2*** has to be connected to the same circuit: $N_{E1} > N_{E2}$, where ***E1*** > ***E2***. Thus, the higher the number of inputs and outputs of the evolving circuit, the higher the floating pins of the substructure and the larger the population size required for containing enough diversity by which two structures could bring via their junction.

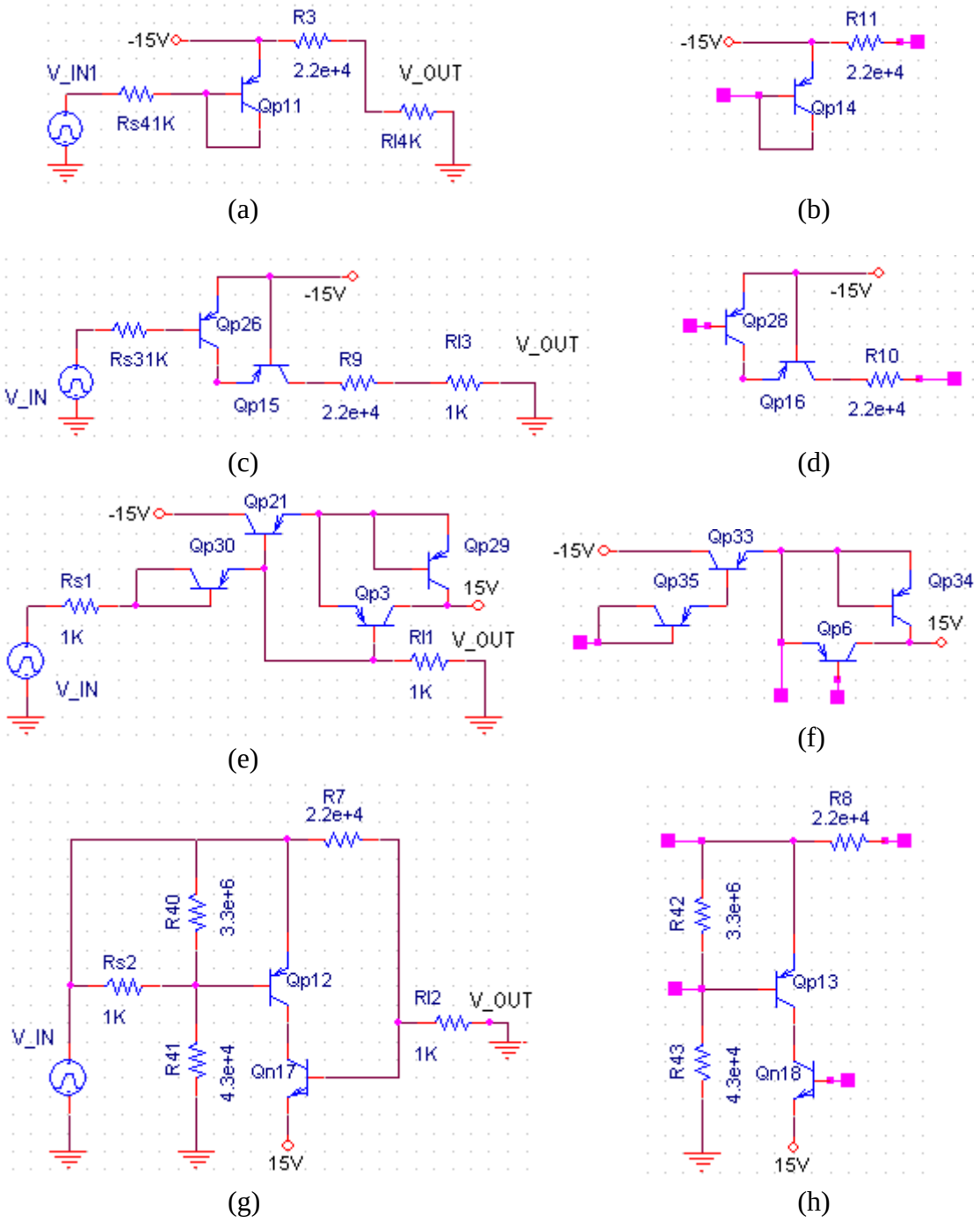


Figure 4-2. The example of circuits of different sizes and the substructures derived from them. a), c), e), g) are the chromosomes of 2-, 3-, 4- and 5-component circuits. b), d), f), h) are the corresponding substructures. Red squares mark the pins by which the substructures are going to connect to the main circuits.

On the other hand, those circuits with a larger node number has more possible ways (N_1^E) to connect to themselves a substructure with E floating pins than a circuit with a lower node number N_2 could to connect to itself the same substructure: $N_1^E > N_2^E$, where $N_1 > N_2$. Thus, the larger the circuit, the higher number of nodes for a connection, and the larger population size it requires for containing enough diversity which two structures could bring by their junction. The limit for a substructure size of up to six components, at maximum population size of 30,000 chromosomes, followed from the series of experiments and has verified that substructures of larger size rarely help in evolution. The analyses showed that - as usual - the chromosomes that received larger substructures rarely survived, showing low fitness values. This caused the evolution to steadily slow down and finally to experience the stalling effect. There are three possible reasons of this.

The first one is caused by the number of connecting pins rather than by the number of substructure components. Unlike the known example of incremental evolution, when a chromosome may join to another comparable in terms of the size of its genetic part, the place for substructures inside the chromosome is not defined. The growth of substructure size may cause the growth of the number of pins by which the substructure should connect to the circuit. This, in turn, causes the solution space to search to grow considerably. As has been discussed in Section 2.5.3, the maximum number of connecting variants where each pin of a substructure may connect to different nodes of a circuit (except connecting to itself) is defined by combinatorial formula for the *combination* [112]: $j!/(k!(j-k)!)$, where the *set* has a total of j nodes and the *subset* k equals the number of substructure pins. According to the SRM strategy, it is applied at later stages of the evolution where j may equal 15 or more. For $k=4$, the total number variations is 1365, for $k=6$ this number is 5005. Taking into account that - for example - an 8-gene substructure has an equal right to be applied to about ten others and that not every chromosome in a population needs the SRM, the probability of success from the SRM with a larger size falls down. A possible way out for this problem is the “know-how” of where the multiple pins of a substructure should be connected to.

The second reason does not concern the size of the substructures but rather the competition between them. With the appearance of a new substructure, the potential

benefits for the rest decreases. Evolution has more choice but under the same amount of resources, i.e. the population size is not enough for diversifying this choice.

And the third reason why the larger substructure size causes a stalling effect is bloat (Section 2.2.2). The sudden increase of genotypes in chromosomes will defocus the evolution, spreading its efforts to a larger solution space. This requires reconsideration of the DEM strategy and possibly additional measures against bloat.

4.1.3 Triggering the SRM procedure

To implement the triggering of the SRM procedure it has been set as the threshold for a number of unsuccessful generations. If, for DEM procedure, this number is set to 3, for triggering the SRM this number is set to between 4 and 6. The details of triggering the SRM are depicted on Figure 4-3.

To avoid the situation where the SRM procedure leads to the overgrowing of the chromosomes' length - or bloat - an additional condition is introduced, namely one that limits of application of SRM by one within a minimum of four generations (the same number as the maximum size of substructure applied in the experiments below in this section). If the last four generations did not bring about improvement and the last time that SRM was applied - say, two generations ago - the system checks whether the difference in length between the best and the current chromosomes are within the three genes. If the difference is large, DEM will reduce the length, otherwise ANEM works. To fight bloat, CGP was found to be an effective technique [147]. However, as it is discussed in Section 2.2.1, the limit of CGP is the restrictions imposed on feed-forward interconnectivities, which contradicts the unconstrained nature of the proposed technique in this thesis.

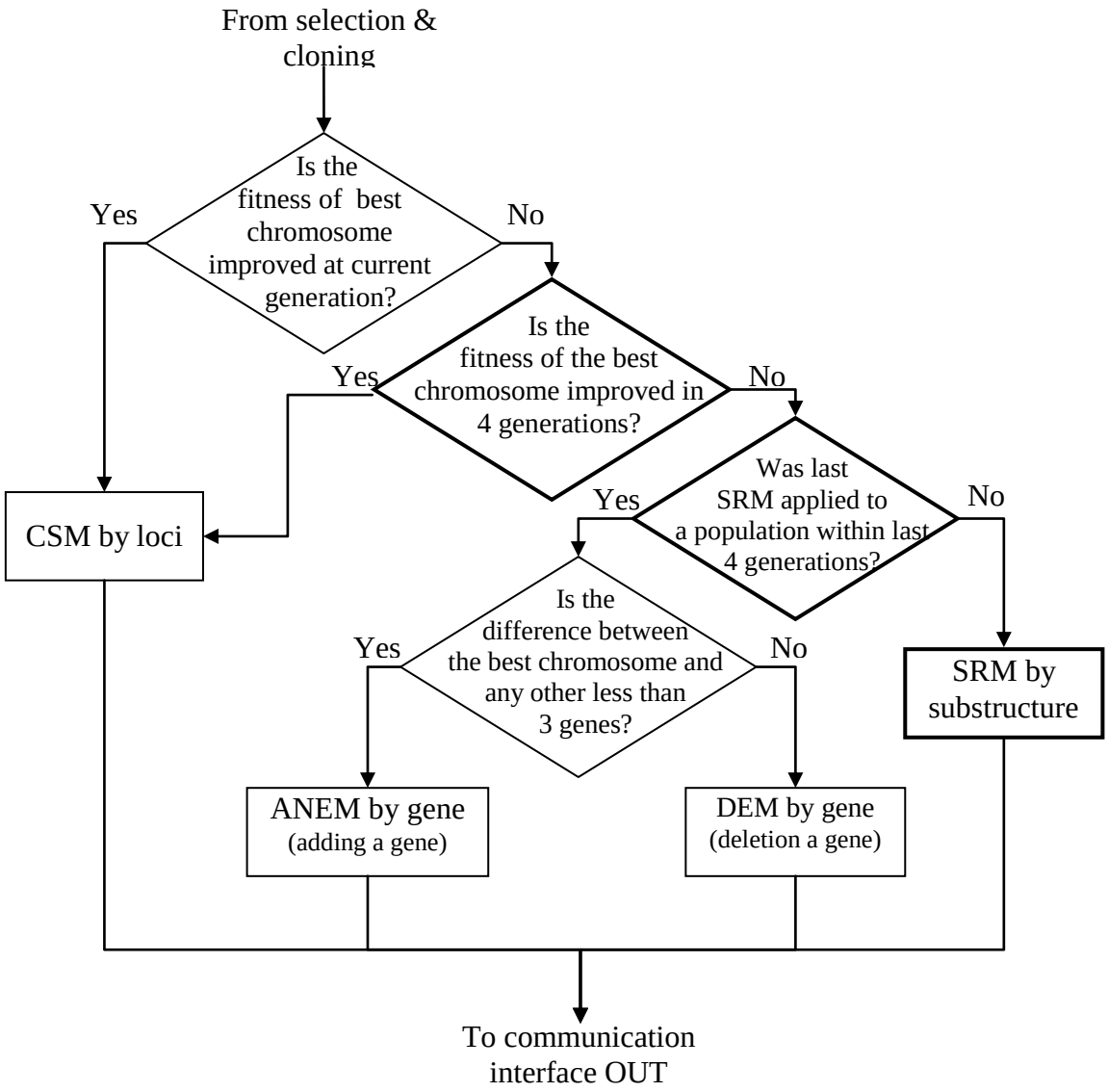


Figure 4-3. The flowchart of mutation procedure with the SRM at the population level. The new terms are in the bold frames.

4.2 Individual Level Mutation

If previously the mutation operator was applied towards the total population without differentiating among individuals, now it will approach each chromosome individually. The reason why such an approach is suggested now was the introduction of SRM in the previous section. Indeed, if the system decided to apply SRM, for each chromosome it will randomly choose among the blocks available, but the blocks are of different sizes. That is, after an application of SRM, the chromosomes tend to be of a different length.

The further applications of SRM may lead to significant differences in length among individuals. For example, a chromosome with a length of five genes went via SRM four generations ago and was modified by a 4-gene substructure. If, now, the same individual with nine genes goes through the same 4-gene modification, it will consist of thirteen genes, while its neighbour will have grown from five to nine genes for the same number of generations.

The individual approach requires the control of two of the chromosomes' features: 1) their length, and 2) their fitness. For this, the additional function inside the system is enabled so as to associate each chromosome with its length and fitness histories. The view of the standard print out of the memory buffer looks like that presented by Table 4-1, where at generation N+6 and N+13 two substructures of sizes 4 and 3 genes correspondingly have been connected to the chromosome; at generations N+2 and N+9 ANEM has been applied; at generation N+7 DEM has been applied; and at generations N+3, N+4, N+5, N+8, N+10, N+11, N+12 CSM has been applied.

Table 4-1. The fitness and length stories of a single chromosome throughout 15 generations.

Generation	N	N+1	N+2	N+3	N+4	N+5	N+6	N+7	N+8	N+9	N+10	N+11	N+12	N+13	N+14
Length	8	8	9	9	9	9	13	12	12	13	13	13	13	16	16
Fitness	46.2	46.2	41.4	41.4	41.4	41.4	41.4	29.3	29.3	26.5	26.4	26.4	26.3	22.1	20.3

According to the new paradigm, when every chromosome has to go through a procedure as depicted by Figure 4-3, the new mutation flowchart will look as is shown by Figure 4-4.

It should be noted that the choice among the 4 types of mutations (CSM, ANEM, DEM and SRM) is only defined by the chromosomes' fitness and length stories, and the choice among the different types of substructures is defined randomly.

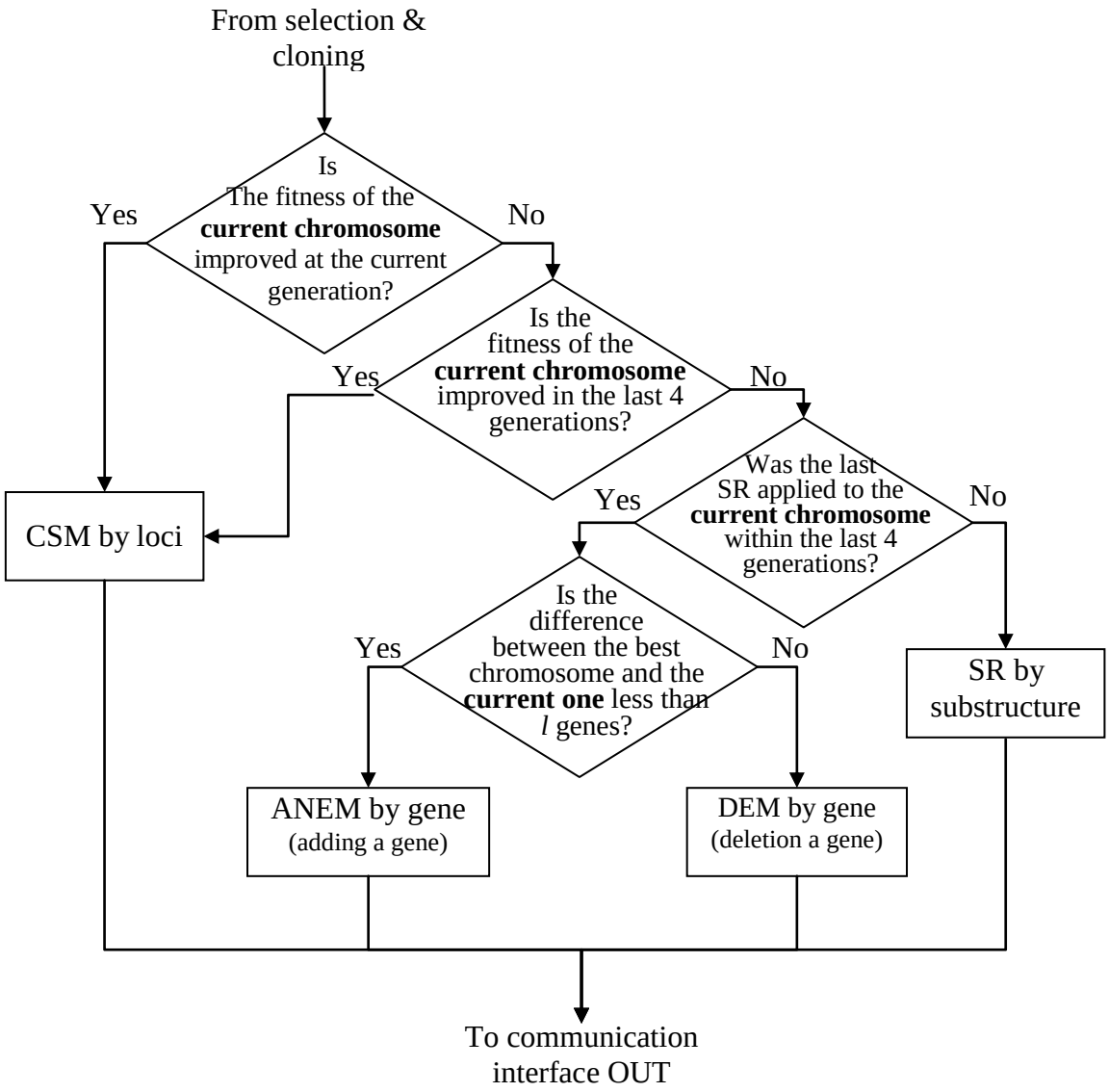


Figure 4-4. The flowchart of the mutation operation at the individual level.

4.3 Trade-off between fitness and size

With the individual approach during mutation procedure, it becomes possible to more closely control the second most important of the chromosomes' features (after fitness): *the length*. The problem of the *overgrowing* of mutation size is a well-known problem in evolutionary electronics [2], [38], [78] and the pruning procedure is the standard technique for tackling this issue. However, as experiments have revealed, pruning in the middle of the evolution process does not bring any valuable benefits in

the long term. Furthermore, in most cases, when pruning is applied at every generation it causes the appearance of the stalling effect at earlier stages in the evolutionary process. The reason for this is that with the deletion of the components the neutral network shrinks, and neutral mutations have less of an effect on the circuit, which decreases the effectiveness of evolution. More details on neutral effects are described in [17], [38], [78], [82], [100] and [101].

The problem of *overgrowing* affects the evolutionary process in two ways: 1) it increases the search space, and 2) it increases the evolution time. Both consequences are unpleasant, especially when one cannot be sure at any given moment as to how deep the *overgrowing* problem is. Indeed, without any additional procedure there is no way defining of how many redundant components are in the circuit, but that requires stopping the process.

The introduction of the DEM procedure does not solve this problem, since DEM - during its operation - chooses the component for deletion *randomly*. This means that the components that take part in the circuit's functioning have the same chance of being removed as those components which do not have an influence on the circuit's functioning.

It is necessary to introduce another kind of mechanism that will control the chromosome's size against *overgrowing*. Therefore, in this section, the introduction of the pruning procedure and then the second objective are described.

4.3.1 Pruning

The idea behind the pruning procedure is to prune those components that have no influence to the circuit's functionality and, thus, seemingly reduce the solution space. Since the procedure is time-consuming, it is meaningless to apply it towards every chromosome of a population at each generation, but there is reason to apply it towards the top-ranked chromosome (selected) or towards a single best individual at each generation. The last approach has been tried, and a brief description is given below.

After evaluation and ranking, each chromosome gets into the special sub-system, which tries to eliminate, one-by-one each gene from a chromosome. The floating pins - after elimination - are connected to each other in all possible variations, as is described in relation to the DEM; each time a new variant of a chromosome is tested. Finally, the gene is eliminated and the new chromosome is adopted if any variant has a fitness value equal or better than that which was present prior to the procedure.

The described operation has been variously applied towards the best-ranked individuals during Experiment 11. The experiments show that applying this procedure either at each, or one of two, three or four generations towards the best-ranked chromosome did not bring an acceptable solution due to the stalling effect. The experimental results are shown at the beginning of Section 4.4.2. After getting these results, it was decided to introduce - instead of the direct deletion of introns - parsimony pressure during ranking.

4.3.2 The second objective

The pressure on evolution towards more compact solutions could be applied by means of the second objective. If the first objective is the fitness value - which refers to the functionality of a circuit - the second objective is the chromosome length, which refers to the size of a circuit.

Besides the techniques described previously, during the experiment it was found that it is also convenient to apply the second objective, namely pressure at the ranking procedure. That is, when ranking chromosomes - along with their fitness values - their length is taken into an account.

4.3.3 Ranking

First of all, we should distinguish a chromosome with the best fitness from the best chromosome. In the first case, an individual attains the best fitness value according to the fitness function. In the second case, an individual has gone through ranking and has

been ranked as number one among the population. In most instances, these two are represented by the same chromosome, though not always; this is because the chromosome length is taken as the second objective during ranking. Thus, if one looks at the graph of the fitness function of the best chromosome, it will not always be slowing down (improving) and there may appear some ridges (Figure 3-22, from generation No.7 to No.10 or from No.22 to No.24).

Through ranking, one has an opportunity to apply selective pressure: along with the functionality of the circuit, the shorter chromosomes are preferred over the longer ones. The fine-grained, open-ended evolution of analogue circuits with dynamic encoding has a side-effect when a resulting circuit integrates into its structure along with functional components, namely the ones that have no effect to the circuit's behaviour [82]. During the ranking procedure, it is applied as a simple *pressure-constant* that behaves adaptively, i.e. depending on the progress of the evolution, it varies the pressure.

To enable the longer length genotypes to compete with the shorter ones, the longer individual should have a better fitness value than that of one of the compared shorter chromosomes. The established trade-off rule between size and fitness uses a function that combines fitness and size so as to yield a single measure of quality. The idea is similar to the notion Adaptive Parsimony Pressure in [151], where the size penalty is based on the size of the best individuals. The ranking procedure consists of two stages: the adjustment of fitness values of chromosomes according to their sizes, and the simple ranking of them in descending order. In order to get a new adjusted fitness of the current chromosome f_{ad}, to its current fitness f, the adjustment value k is added: $f_{ad}=f+k$, where k represents the normalized difference between the sizes of the current individual and the best one: $k=m(l-l_{best})$. If the length of the best chromosome is shorter than that of the current one, k is positive and the new fitness of the current chromosome is f_{ad} and increases: $f_{ad}>f$, otherwise $f_{ad}<f$. The normalization coefficient m represents the fitness per gene of the best individual: $m=f_{best}/c$, where the coefficient c is a *pressure-constant*, the meaning of which is a predicted number of genes (components) in the target; the smaller the number, the higher the pressure that is applied. The adaptive features of it are described further. As such, k determines how much of the best chromosome's fitness is related to the difference in size, and the final fitness is:

$$f_{ad} = f + f_{best} \times (l - l_{best}) / c \qquad [4\text{-}1]$$

According to the classification set by Bentley [150], the proposed technique belongs to the "range-dependent" ranking methods and the system is not a "general-purpose multi-objective" one. The main reason for the choice of this type of ranking is its simplicity. The *pressure-constant* is a deterministic dynamic parameter that is adapted by the evolution. The *pressure-constant* should be set to some initial value. However, if two successive generations do not bring forth better individuals, this number is increased by two,[5] which causes the component-reducing pressure to decrease. Conversely, if the two successive generations have brought a fitness improvement, the *pressure-constant* is reduced by two. This strategy leads to the inevitable gradual weakness of the selective pressure due to the permanent growth of a chromosome's length, complexity and solution space.

4.4 Experiments 9-12: Evolution of Computational Circuits

The CC is a circuit that converts incoming voltage into outgoing voltage in accordance with some computational function. As is mentioned in Section 2.6.2, an analogue CC is useful when there is a need for a single mathematical function; it does not require the conversion of an analogue signal into a digital signal with the aid of an analogue-to-digital converter, performing the mathematical function by a digital processor, and converting the result back to the analogue domain by using a digital-to-analogue converter, as is shown by Figure 4-5 (right).

As has already been mentioned in Section 4.1.1, the substructures are not pre-scribed but are automatically created during evolution. It is used as the limit for the substructure size, to up to six components at a population size of 30,000 chromosomes. The limit for the substructure size is discussed in Section 4.1.2.

[5] This and other specified numbers used in description of methodologies are suggested since they have been used during experiments.

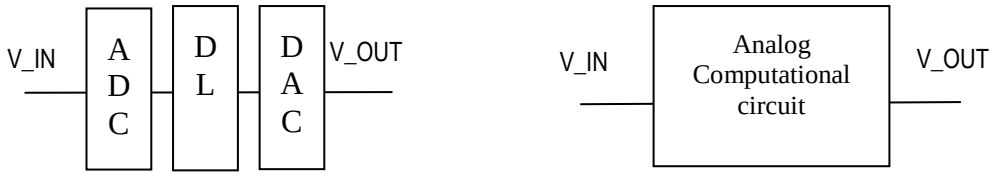


Figure 4-5. A digital (left) and an analogue representation of a computational circuit.

In the beginning, the brief results of the experiments with pruning are presented. Next, four experiments will be described without pruning. All of them are challenging tasks since very few attempts have been made before in relation to the evolution of computational circuits (CC). Moreover, the CCs that have been developed so far are known as some of the largest circuits that have ever been automatically synthesized [24]. The use of unconstrained evolution armed with individual-level mutation and SRM will attempt to evolve the cube root circuit (which never been synthesized through previous techniques). Furthermore, three new CCs are set as targets in this section.

In Experiments 9-12, several issues are tackled, including:

- Testing the technique armed with individual-level mutation and SRM;
- Testing the new technique on the *second level of the complexity* task;
- Testing the pruning technique without *parsimony pressure*;
- Testing the *parsimony pressure* without pruning;
- Testing the new technique on other tasks, some of which are more sophisticated (cubing function);
- Testing the system on the tasks that belong to the *third level of complexity* 4-output VDC according to the classification introduced in Section 2.5.2.

4.4.1 Fitness function

The goal for the evolutionary search is to evolve four CCs whose output voltages are: the cube root, cube, the square root and the square of their own input voltages. To enable ourselves to make a comparison of the final results, the same fitness terms are set as in [12] for all four cases. These are:

- The PSPICE simulator is made to perform a transient analysis of a source signal of a length of 0.2 seconds at 21 equidistant time-points;

- The voltage source forms a pulse signal arising from –250 mV to +250 mV for the cube root, cubing and squaring; and from 0 mV to +500 mV for the square root;

- A fitness value is set to the sum, over these 21 fitness cases of the absolute weighted deviation between the target value and the actual output value: $F = \sum_{i=0}^{p} | V_{ideal}^{i} - V_{measured}^{i} |$, where V_{ideal}^{i} is the voltage at the *i*-th point for the ideal response and $V_{measured}^{i}$ is the voltage at the *i*-th point obtained for the evolved circuit; *p* is the number of points evaluated equalling 21;

- The fitness penalizes the output voltage by 10 if it is not within 1% of the target voltage value;

- The smaller the fitness value, the closer the circuit is to the target.

The circuits that treated by PSPICE as “error circuits” are assigned the worst fitness. The termination criterion is set where either the fitness value has not improved over 20 consecutive generations or else the best circuit exceeds 70 components in size.

The embryo circuit is a component or a number of components (including the voltage source) which can be predetermined for the particular targeted circuit so as to ease further circuit growth. Regardless of the cube root circuit evolved earlier in Section

3-12 with an unsatisfactory fitness of 2.27, the results of that experiment have not been used here. The embryo circuit is defined for all targets as the same: a pulse voltage source, the source resistance Rsource=1kOhm and the load resistance Rload=1kOhm. These three components in Figure 4-6 compose the embryonic circuit. The embryo also has two sources of direct voltage, allowing the evolution to choose between them (or use both) +15V and -15V, so that the initial node number at the beginning is five (with ground).

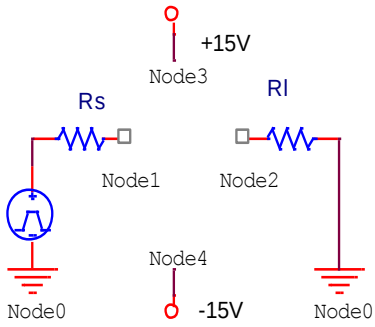


Figure 4-6. Embryo circuit for CC.

4.4.2 Experimental results

Four attempts are described here for evolving a cube root circuit that have been undertaken with a pruning procedure prior to parsimony pressure. Each attempt consists of two runs. The best of the runs are shown by Figure 4-7. The first approach was undertaken when the pruning was applied towards the best individual at each generation ("Pruning 1" in Figure 4-7). During the second experiment, the pruning was applied towards the best individuals at every second generation ("Pruning 2" in Figure 4-7). During the third and fourth experiments, the pruning was utilized at one of three and at one of four generations towards the best ranked chromosomes respectively ("Pruning 3" and "Pruning 4" on Figure 4-7).

The best of the runs out of all eight experiments reached a fitness of 2.68 ("Pruning

4”), which cannot be treated as successful because the targeted fitness was 1.68 from [24]. The resulting circuits have not been analyzed deeply, because the goal was to tackle the bloat in the simplest manner rather than to perform deep research into the anti-bloat technique.

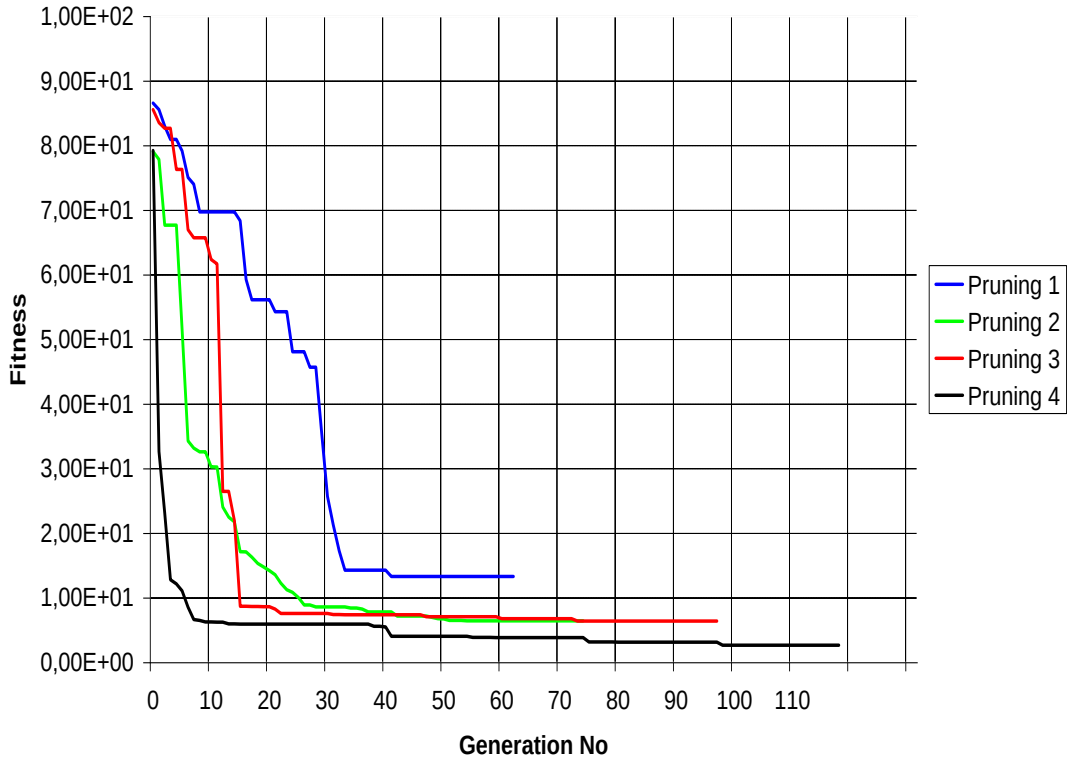


Figure 4-7. Experimental results of the evolution of cube root with pruning.

The results of the experiment described could be explained by the crucial role of neutrality in evolution [38], [82], [101]. Moreover, it is claimed that the basis of self-adaptation is the use of neutrality [152]. In the absence of external control, neutrality allows a variation of the search distribution without the risk of fitness loss.

The results presented in the rest of the Section are related to the work of the system without pruning, but with the utilization of parsimony pressure as described in Section 4.3.3. They are out of five runs for each of the target cases, with different seeds for the RNG. The data for all 20 runs is presented by Table 4-2, where the best runs are marked in bold. 10 PCs are used with a Pentium-4/3GHz/RAM2GB processor running at the

same time and independently of each other. The average time per run is 43 hours, which is comparable with the duration of the evolution of analogue circuits in [93], [38]. A total population of 30,000[6] individuals, a mutation rate of 5% and a selection rate of 10% are utilized.

Table 4-2. Statistics for the evolution of the 4 targeted circuits

No.	Fitness	Component No.	Generation No.	Fitness	Component No.	Generation No.
	Square Root			**Squaring**		
1	0.283	43	119	**0.0302**	**35**	**92**
2	**0.194**	**23**	**123**	0.0459	43	309
3	0.443	50	208	0.0563	48	143
4	0.798	38	97	0.0951	38	97
5	0.255	50	200	0.0776	50	135
	Cube Root			**Cubing**		
1	0.764	44	115	0.0095	50	195
2	1.060	49	179	0.0205	38	72
3	**0.251**	**39**	**152**	0.0079	49	109
4	0.268	50	201	**0.0061**	**44**	**78**
5	0.643	40	294	0.0101	37	98

[6] Despite the good results that have been received with this population size, there are no reasons why this size should not be increased. In fact, here the author is only driven by convenience in processing the PSPICE out-file.

4.4.2.1 Experiment 9-10: square root circuit and squaring circuit

The best-of-run circuit (Figure 4-8) for the problem of designing a square root circuit had 23 components with a fitness of 0.194. The best-of-run circuit (Figure 4-9) for the problem of designing a squaring circuit had 35 components with a fitness of 0.0302.

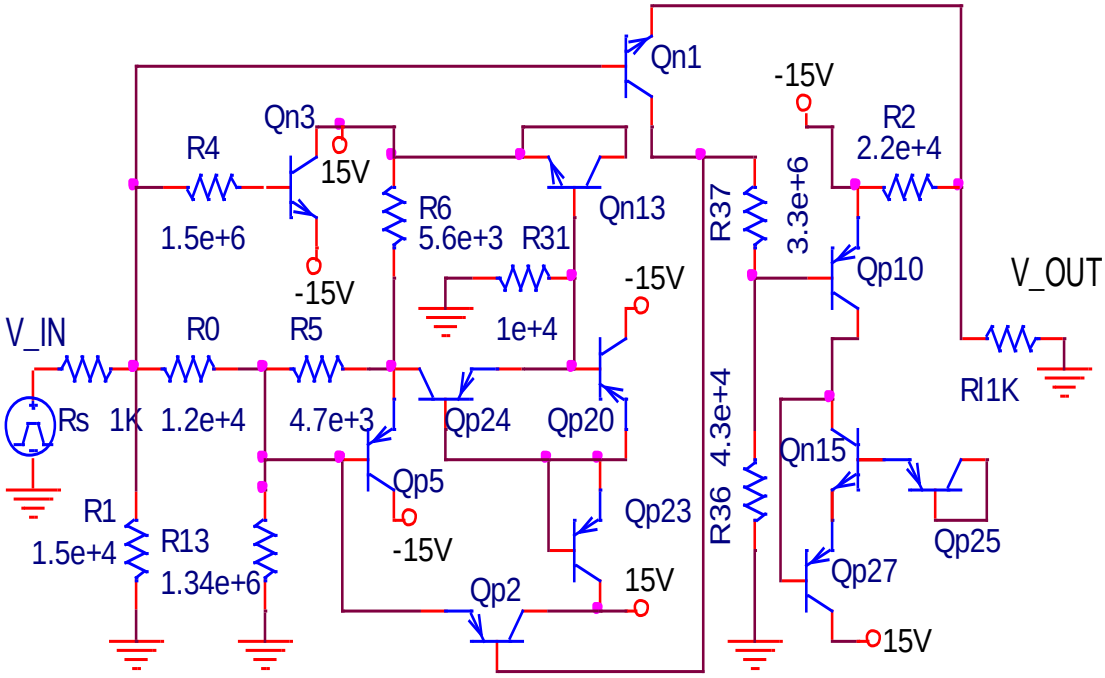


Figure 4-8. The evolved square root circuit in Experiment 9.

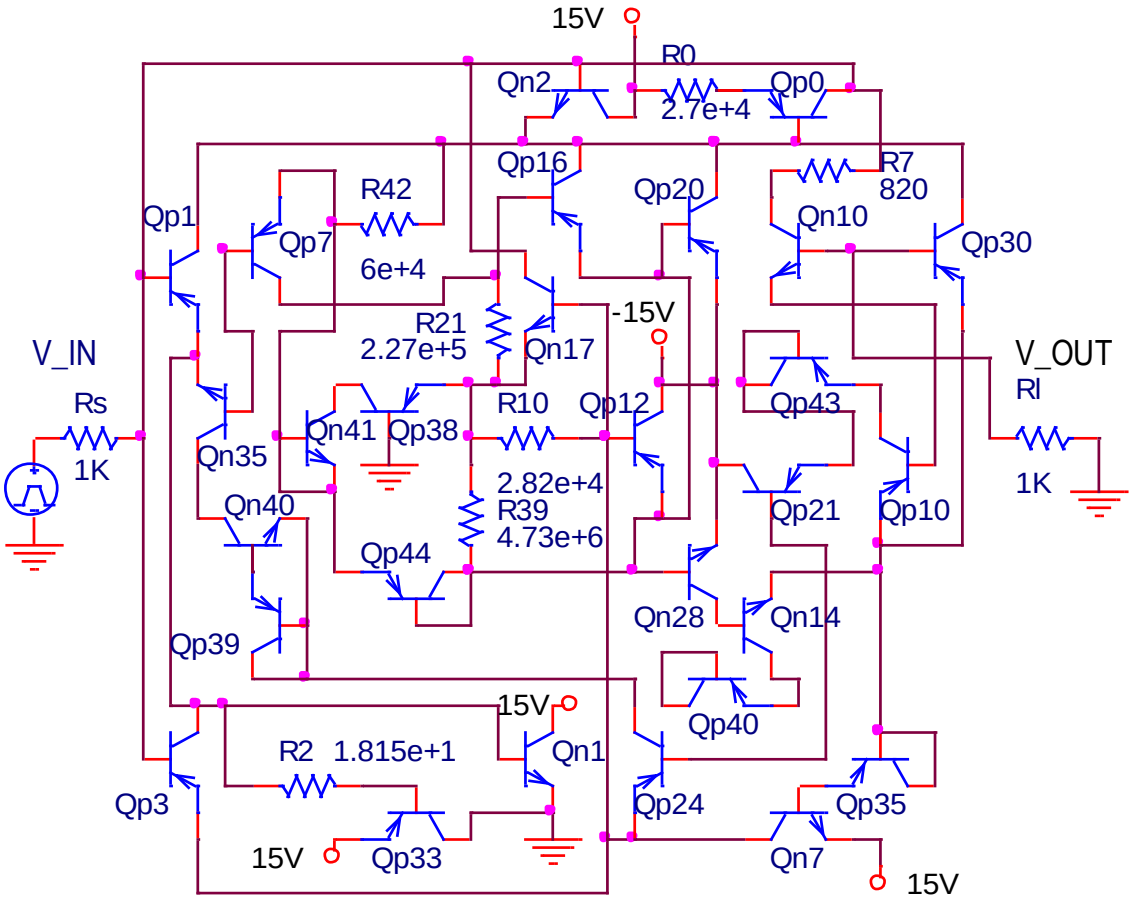


Figure 4-9. The evolved squaring circuit in Experiment 10.

4.4.2.2 Experiment 11-12: cube root circuit and cubing circuit

The best-of-run circuit (Figure 4-10) for the problem of designing a cube root circuit appeared at generation No.152 and had 39 components with a fitness of 0.2508. The best-of-run circuit (Figure 4-11) for the problem of designing a cubing circuit appeared at generation No.78 and had 44 components with a fitness of 0.00614.

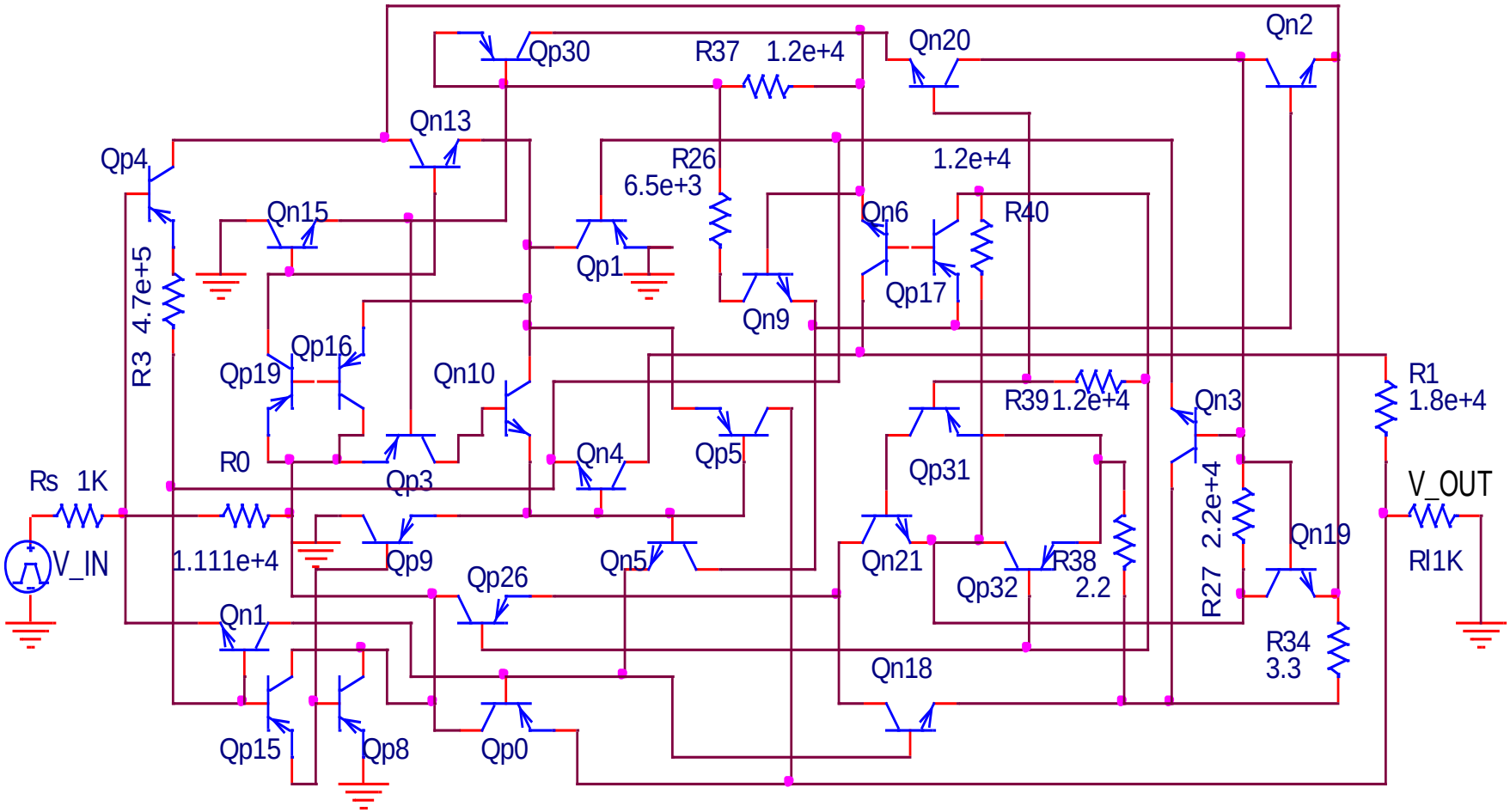


Figure 4-10. The evolved cube root circuit in Experiment 11.

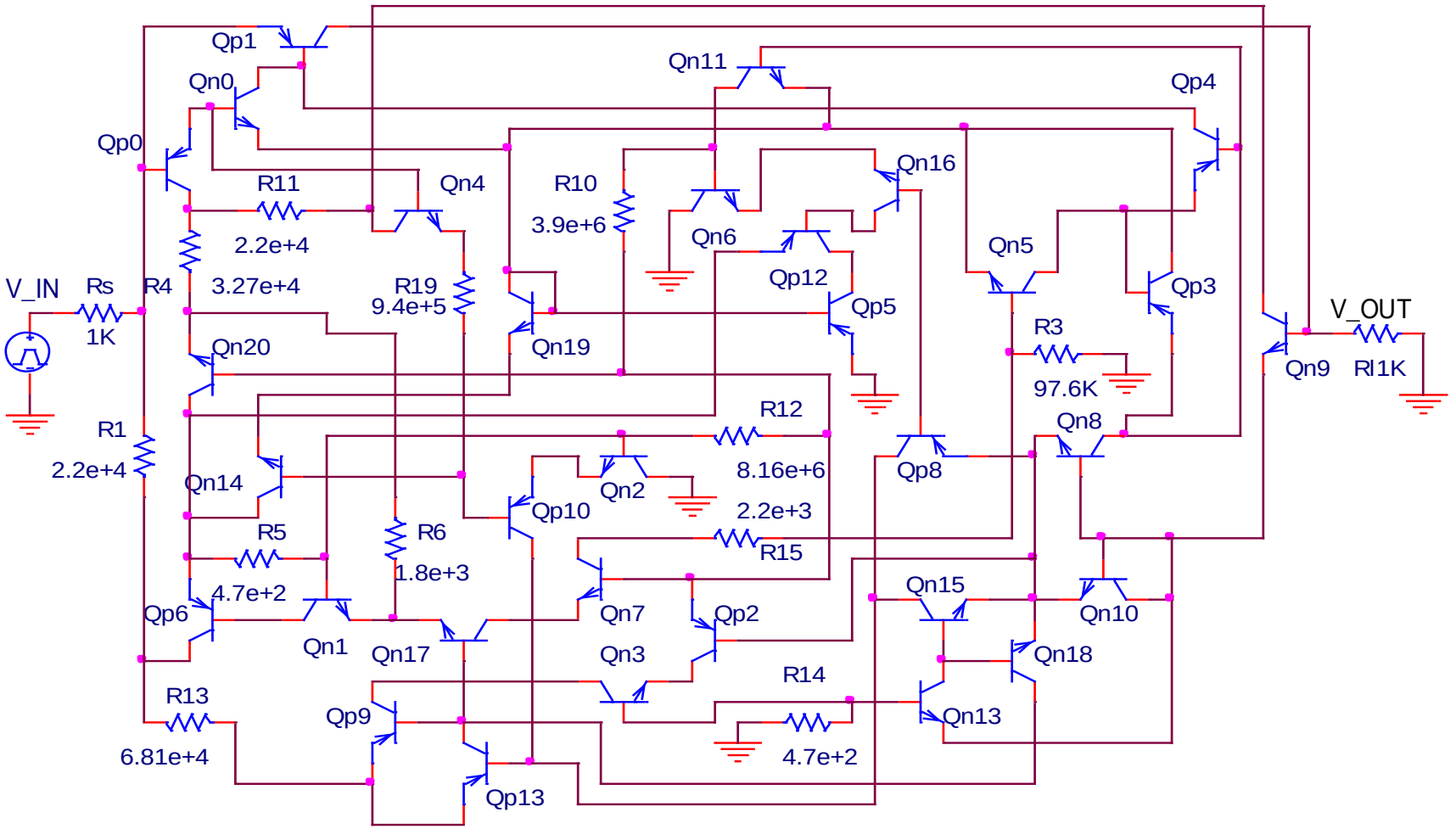


Figure 4-11. The evolved cubing circuit in Experiment 12.

Table 4-3. Comparison with circuits published previously

Feature \ Author	Koza et al. [24]	Mydlowec et al. [98]	This work [115]	Improvement, times
Square root				
Average error, mV	183.57	20.00	9.23	2.2
Fitness value	3.855	70.403	0.194	18.9
Component No.	64	39	22	1.8
Evaluation No.	Data n/a	6,7E+9	3,7E+6	1800
Squaring				
Average error, mV	Data n/a	27.00	1.44	18.7
Fitness value	Not converged	4.812	0.0302	159.3
Component No.	39	37	35	1.1
Evaluation No.	Data n/a	1,1E+9	2,7E+6	407
Cube root				
Average error, mV	80.00	-	11.90	6.7
Fitness value	1.68	-	0.2508	6.7
Component No.	50	-	39	1.3
Evaluation No.	3.8E+7	-	4.5E+6	8.4

The improvement values in column 5 are received from the division of the best corresponding values from columns 2-3 and the values from column 4.

Table 4-4. Comparison of the evolved cubing circuit with ones published previously

Feature \ Author	Koza et al. [24]	Streeter et al. [99]	Cipriani et al. [108]	This work [115]	Improvement, times
Cubing					
Avg. error, mV	1.04	0.99	7.13	0.29	3.4
Fitness value	0.0219	Data n/a	Data n/a	0.0061	3.6
Component No.	56	47	12	44	0.3
Evaluation No.	Data n/a	2.94E+6	-	2.34E+6	1.3

The improvement values in column 6 are received from the division of the best corresponding values from columns 2-4 and the values from column 5.

The schematics published in [24], [98], [99], [108], enable the netlists in PSPICE to get the fitness values appropriate for comparison. Both DC and transient analysis give

identical results for each schematic, together with the other published data, presented by Tables 4-3 and 4-4. For some of the circuits from [12], exactly the same fitness values have been received, the last fact ensuring that the correct transistor models (PSPICE default models) and other simulation parameters have been chosen. The right-most column of the tables suggests the relative comparison between the value received in this work and the best corresponding values from the past. As can be noted, the received results are considerably better. Notably, the best by size (12 components) conventionally designed cubing circuit from [108] has an average error of 7.13mV which is 25 times larger than that (0.29 mV) of the cubing circuit (44 components) evolved in this work. Moreover, during evolution the intermediate result with a fitness of 7.27 was obtained at generation No.20, but with a component number of 11. The next generation of the cubing circuit with 13 components gave a fitness of 6.64.

In the above experiments, unconstrained evolution with OLG was applied along with SRM and individual-level mutation towards the design of analogue computational circuits, to the examples of cube root, cubing, square root and squaring functions. This was one of the first successful attempts of the application of ES at the synthesis of analogue circuits of the *second level of complexity*, according to the classification introduced in Section 2.5.2. In all four experiments, circuits with fewer numbers of components with much less computer effort and with significantly better fitness have been successfully evolved.

4.5 Differentiated Mutation of Analogue Circuits

In the previous sections, attention was paid to a mutation procedure of the system. Such mutation operations are introduced as CSM, ANEM, DEM and SRM. The SRM consisted of up to six operations with different mutation parameters. The mutation application is shifted from the population-level to the individual-level. All these operations are organized and united in this section into one *differentiated mutation* (DM) operation. DM is the logical conclusion of the mutations introduced earlier. The DM technique is a novel method that is applicable to the evolution of analogue circuits. It

enables the application of the mutation procedure in a smoother, more intelligent and purposeful manner.

Four types of mutation are described in previous sections:

1. CSM modifies a chromosome per locus without influencing an individual's length. This is the most minimal type of mutation in terms of degree of influence;

2. ANEM modifies a chromosome by adding a gene (+4 loci);

3. DEM modifies a chromosome by removing a gene (-4 loci). ANEM and DEM are middle-sized mutations in terms of degree of influence;

4. SRM modifies a chromosome by adding a group of genes united in a substructure. This is the most influential mutation (+8, +12, +16, +20, +24 loci).

The main disadvantage of the existing mutation method is that the triggering of different types of mutations depends on the subjective values of the number of non-successive generations. Furthermore, all of the mutation types are independent of each other and thus cannot be assured as being effective. If we look at a typical single chromosome's mutation history during the evolution of the computational circuit in Table 4-5, it might be noted that the sequence of the mutation rates as applied is not smooth. When SRM is applied, it brings very considerable changes in the genotype (generations N+6 and N+13 of Table 4-5).Therefore, the instantiated average mutation rate over 14 generations results in an average of 11.4%. Therefore, it is necessary to reorganize the existing mutation method.

Table 4-5. The fitness and length stories of a single chromosome throughout 15 generations.

	N	N+1	N+2	N+3	N+4	N+5	N+6	N+7	N+8	N+9	N+10	N+11	N+12	N+13	N+14
Length	8	8	9	9	9	9	13	12	12	13	13	13	13	16	16
Fitness	46.2	46.2	41.4	41.4	41.4	41.4	41.4	29.3	29.3	26.5	26.4	26.4	26.3	22.1	20.3
Mutation type	-	CSM	ANEM	CSM	ANEM	ANEM	SRM	DEM	CSM	ANEM	CSM	ANEM	ANEM	SRM	SCM

Effective mutation rate	-	5%	12.5	5%	11.1%	11.1%	44.4%	7.7%	5%	8.3%	5%	8.3%	8.3%	23.1%	5%
The average effective mutation rate is 11.4%.															

4.5.1 The essence of Differentiated Mutation

This method has been developed gradually from experiment to experiment. In earlier works, the author has already reported on some aspects of DM [109], [115] where the substructure reuse operation is regarded as a case of a more general mutation procedure, but never has it been approached from the quantitative point of view. The idea of the DM technique lies in a *quantitative approach* towards every type of mutation. If - to measure every mutation by a number of loci - it actually modifies inside the circuit, it would be possible to manage the procedure in a much the smoother manner than was shown in Table 4-5.

While different mutation rates are associated with different types of mutations, the reference mutation rate is associated with a static predefined value, e.g., 4%.[7] This rate operates as a reference mutation rate and is set as the minimum level of the modifications that can take place in a chromosome. The reference mutation rate gives the quantitative value of mutation as a percentage. However, under the DM technique, the unit for measure of mutation is the suggested locus. The modification of the locus is the minimum volume of a chromosome that can be mutated. Thus, every type should be calibrated along a new unit.

The essence of the DM approach is based on four basic concepts: *mutation types, virtual mutations, mutation ways* and *mutation strategy*, each of which is described below.

[7] This numeric value is exemplified due to only this number has been used within Experiments described in the thesis. However, it does not mean that there may be another value instead.

4.5.1.1 Types of mutations

It has been noted already that each mutation type modifies a specific number of loci. If we sort all the mutation types in ascending order of their modification ability, they will appear as presented by Table 4-6.

Table 4-6. The types of mutation, where CSM is a circuit structure mutation, ANEM is adding a new element mutation, DEM is deleting an element mutation, and SRM is a substructure reuse mutation.

No		Mutation Type
1	**CSM**	*Node_number_*, *Parameter_* or *Component_name_* mutation phenotypically means the reducing, adding or replacing only 1 locus. There are no limitations on where to use it inside a circuit. In most cases, it is applied in combinations with *Component_mutation* and *Substructure_X_mutation*.
2	**ANEM & DEM**	*Component_mutation* phenotypically means reducing (DEM) or adding (ANEM) a component. It concerns 4 loci at once.
3	**SRM1**	*Substructure_1_mutation* concerns 8 loci. It adds 2 genes at once to a chromosome. These two genes compose the first substructure.
4	**SRM2**	*Substructure_2_mutation* concerns 12 loci. It adds 3 genes at once to a chromosome. These three genes compose the second substructure.
5	**SRM3**	*Substructure_3_mutation* concerns 16 loci. It adds 4 genes at once to a chromosome. These four genes compose the third substructure.
6	**SRM4**	*Substructure_4_mutation* concerns 20 loci. It adds 5 genes at once to a chromosome. These five genes compose the fourth substructure.
7	**SRM5**	*Substructure_5_mutation* concerns 24 loci. It adds 6 genes at once to a chromosome. These six genes compose the fifth substructure.

These kinds of mutations modify the numbers of loci that are given in an *ad hoc* manner, based on their *definitions*. However, each mutation may lead to another factual number of loci that is modified. So, there should be *virtual mutation rates* for each kind of mutation.

4.5.1.2 Virtual mutations

When some type of mutation should take place - being intended to modify one locus (CSM) or four loci (ANEM) - it may cause the modification of more than the intended number of loci. This is to say that the actual number of loci modified may differ from the number of loci that were intended to be modified.

It should be noted in Chapter 3 that all types of mutation may result in the effect that *virtually* the number of loci actually modified differs from one that was initially intended, because all of them may cause a change of circuit topology whereby some pins are floated. These floating pins, as described in earlier sections, have to find new nodes, which in turn virtually creates an additional number of loci to be unintentionally modified.

The types of loci that could be unintentionally modified may only belong to a *node_connection* type of mutation, and the maximum number of such a kind of loci is limited by two in addition to the loci that were intended to be mutated. Thus, a new summary table of mutation types will look that provided by Table 4-7.

There are in total 47 mutation subtypes that are listed together in Table 4-7. As might be noticed, each mutation type now covers a longer interval of mutations and there are only two gaps left uncovered in the total range of mutations: 3 loci and 7 loci. In Figure 4-12, two graphs compared. The first one is associated with the old mutation approach, while the second graph refers to the new technique. A novel mutation approach enables a much smoother transition among the *types* and wider coverage of the *virtual mutation rates*, thus providing a more powerful tool for the mutation procedure.

It should be explained that the system must have an additional subroutine that maintains statistics on the types of mutations possible per node in order to avoid the *unintended loci mutations*. This subroutine enables a virtual mutation procedure "to do what was asked." This is viewed as another advantage of the proposed technique because the subroutine is selecting the particular nodes for mutation, which reduces the randomization of the search process.

It should also be noticed that - to the author's knowledge - nobody has mentioned the idea of *virtual mutation rates* before.

Table 4-7. The calibration of mutation types among virtual mutations, where CSM is a circuit structure mutation, ANEM is an adding a new element mutation, DEM is a deleting an element mutation, and SRM is a substructure reuse mutation.

No	Mutation Type		Virtual mutation value
1	**CSM**	**Parameter**	Modifies only 1 locus.
2		**Node_ number**	a) Modifies 1 locus. Example is in Figure 3-10a. b) Modifies 2 loci. Example is in Figure 3-10b.
3		**Component_ name**	a) Modifies 1 locus. Example is in Figures 3-7b, 3-8b. b) Modifies 2 loci. Example is in Figure 3-7c, 3-8c, 3-9b.
4	**ANEM**	**Component mutation**	a) Modifies 4 loci. Example is in Figure 3-11b, 3-12b. b) Modifies 5 loci. Example is in Figures 3-11c, 3-11d, 3-11e, 3-11f, 3-12e, 3-12d, 3-12f. c) Modifies 6 loci. Example is in Figures 3-11g, 3-11h, 3-11i, 3-11j.
5	**DEM**		a) Modifies 4 loci. Example is in Figures 3-21b, 3-21c, 3-21d. b) Modifies 5 loci. Example is in Figures 3-21e, 3-21f, 3-21g, 3-21h. c) Modifies 6 loci. (Not instantiated, but possible.)
6	**SRM**		Substructure_1_mutation modifies the following number of loci: a)8; b)9; c)10; d)11; e)12; f)13; g)14.
7			Substructure_2_mutation modifies the following number of loci: a)12; b)13; c)14; d)15; e)16; f)17; g)18.
8			Substructure_3_mutation modifies the following number of loci: a)16; b)17; c)18; d)19; e)20; f)21; g)22.
9			Substructure_4_mutation modifies the following number of loci: a)20; b)21; c)22; d)23; e)24; f)25; g)26.
10			Substructure_5_mutation modifies the following number of loci: a)24; b)25; c)26; d)27; e)28; f)29; g)30.

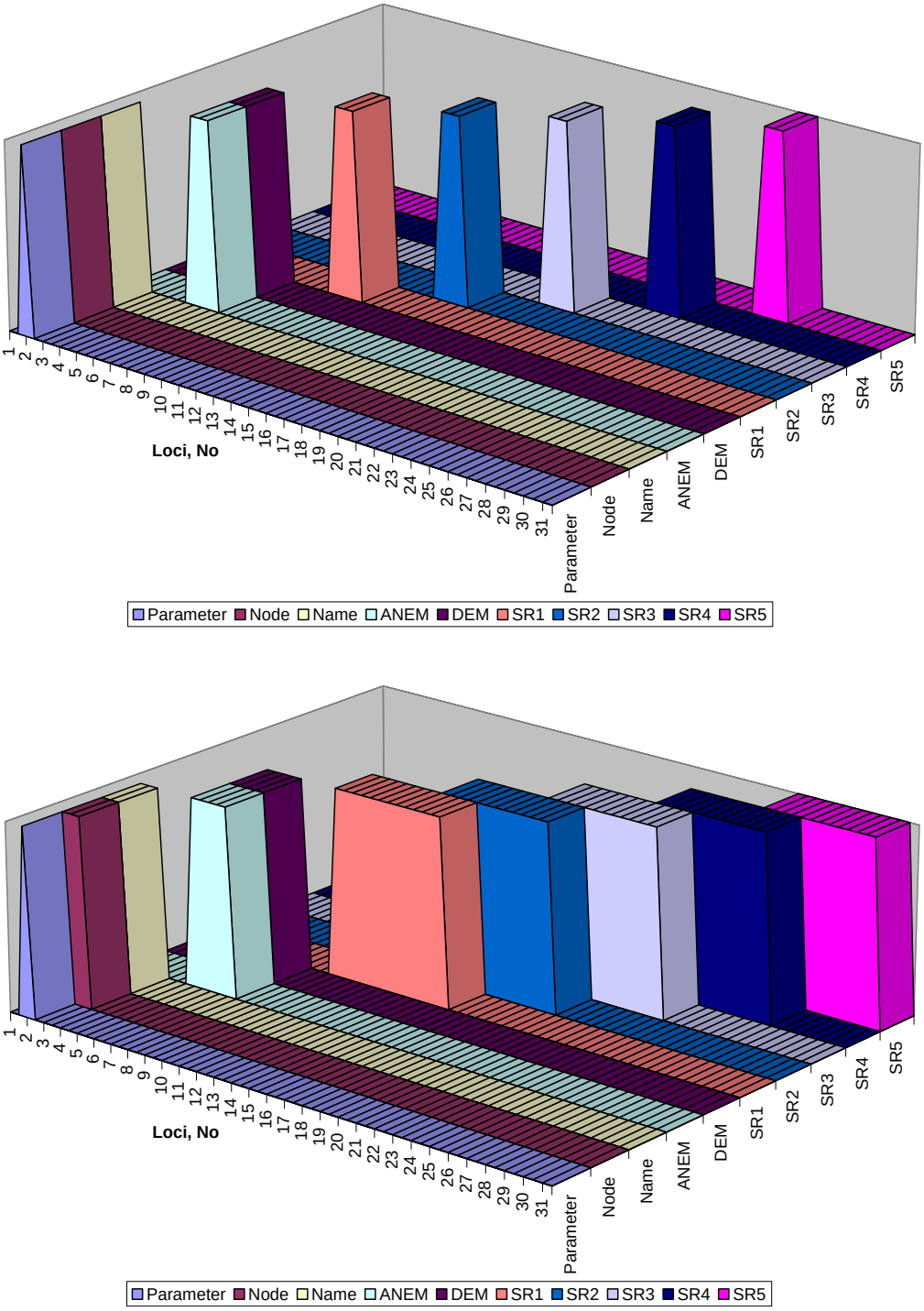


Figure 4-12. Two graphs that show the loci coverage by different types of mutation. The upper graph refers to the old approach, while the lower one shows the loci coverage of the new technique. Only 3 and 7 loci are left uncovered, but they may be reached by a combination of mutation types. The *virtual mutation rates* (lower graph) suggest a more diversified mutation field.

4.5.1.3 Mutation Ways

Combined with each other, the mutation types suggest a variety of *mutation ways* in the frame of the *reference mutation rate*. In Table 4-8, there are five examples of chromosomes against the list of different *ways* of mutations which may be applied to a corresponding individual. For instance, the 50-gene chromosome can be mutated in 112 different ways, each of which may be composed of six different mutation types. The choice of the particular *way* is set as a random procedure, if it is not specified elsewhere.

Table 4-8. Examples of the ways of mutation for 5 different chromosomes

Chromo-some size, genes	**No of loci mutated at reference mutation rate 4%**	**The mutation way list applied to different chromosomes (by combining mutation types named by No.1-No.7 in accordance with Table 4-7)**
10	2	1) No.1×2; 2) No.2a×2; 3) No.3a×2; 4) No.2b×1; **5)** No.3b×1;
20	4	1) No.1×4; 2) No.2a×4; 3) No.3a×4; 4) No.2b×2; 5) No.3b×2; 6) No.1×2+No.2b×1; 7) No.1×2+No.3b×1; 8) No.2b×1+No.3b×1; 9) No.1×3+No.2a×1; 10) No.1×3+No.3a×1; 11) No.2a×3+No.3a×1; 12) No.3a×3+No.2a×1; 13) No.2a×2+No.3b×1; 14) No.3a×2+No.2b×1; 15) No.2a×2+No.2b×1; 16) No.3a×2+No.3b×1; 17) No.4a×1; 18) No.5a×1;
50	8	Totally, **112** possible combinations with involvement of mutations No.1, No.2, No.3, No.4, No.5 and No.6
80	13	Totally, about **1000** possible combinations with involvement of mutations No.1, No.2, No.3, No.4, No.5, No.6, and No.7.

Thus, here is suggested the *differentiation* of mutations in analogue circuit evolution not just through rates but also through *types* and *ways*. In the following sections are described the aspects of the fourth concept - the *mutation strategy*.

4.5.1.4 Mutation strategy

The following rules are actuated if the stalling effect begins to appear:

- *The diversification of a mutation history*. Each individual carries its own history of the mutations its ancestors have gone through. If the chromosome is ranked within 10% of the worst of the selected to the next generation, the random choice of mutation is replaced by the following rule: the most seldom mutation type from the individual's history should be applied in the first place at the current generation.

- *The mutation pressure.* If the chromosome does not improve its fitness over the previous two generations the following rule is activated: the lowest mutation way number temporarily leaves out the potential mutation way list (Table 4-8), increasing the probability of the other *ways* to be chosen. This brings more radical changes to a genotype by joining bigger substructures. The mutation rate continues to stay in the frame of the *reference mutation rate*. The mutation pressure may continue until there is only one mutation way left. The pressure disappears once a chromosome has improved its fitness.

- *The Radical mutations.* If the chromosome has not improved within the last *k* generations (for instance, 3) the next that should be applied is the mutation that modifies the higher number of loci (according to Table 4-7) than is allowed within the frame of the *reference mutation rate.* Suppose a 10-gene chromosome with a *reference mutation rate* of 4% goes through the mutation of only two loci. Thus, it allows only mutation numbers 1, 2 and 3, according to Table 4-7 (*Node_number_*, *Parameter_* or *Component_name_*). However, due to its fitness being stuck, this chromosome should go through the mutation of three loci, which is possible by using the same mutation types (Table 4-9). Furthermore, if the chromosome has not even improved within the next generation, it should go through the modification of four loci, and now its mutation may also include mutation type numbers 4a and 5a (*Component_mutation*). With each "unfruitful"

generation, the virtual mutation grows by one loci and the mutation rate increases.

Table 4-9. Examples of the first and second radical mutations for 4 different chromosomes

Chrom.	Mutation at reference value 4%		The first radical mutation			The second radical mutation		
size, genes	Loci No. mutated	Mutation types, No.	Loci No mutated	Mutation types, No.	%	Loci No mutated	Mutation types, No.	%
10	2	1-3	3	1-3	7.5	4	1-4a, 4b	7.5
20	3	1-3	4	1-4a, 4b	5	5	1-4b, 5b	5
50	8	1-6a	9	1-6b	4.5	6	1-5	4.5
80	13	1-6c, 7a, 7b	14	1-7c	4.4	15	1-7d	4.4

Thus, the general algorithm of the mutation procedure now has a view of the flowchart in Figure 4-13. The *radical mutation* may be applied as many times and to as many generations of a chromosome that has not improved its fitness. The *virtual mutation rate* may reach a value many times higher than the *reference mutation value*. The *radical mutation* is a very important part of a general concept called *Very Narrow Focused Evolution* (VNFE), which will be introduced in Chapter 5. It provides the essential modifications to uncommonly homogenous individuals, especially during stuck periods.

In this chapter, a novel and feasible *individual-level* mutation scheme is proposed for the analogue circuit synthesis system based on the novel concepts of *mutation types, virtual mutations, mutation ways* and *mutation strategy.* In the approach presented here, there is a feature of the *adaptation* of the mutation rate, which is based on the idea that the particular rate value of each mutation is defined by evolution itself. Evolution defines the particular *mutation type* and randomly chooses the *mutation way* for every chromosome using the current and past features of the chromosome and the population it comes from. The researcher sets only the increasing direction in the loci modification number in the case of idle generations. In the method, the parameters that control the mutation rate of a chromosome are not encoded into their corresponding chromosome as additional genes [62] but are represented by such chromosome and population

characteristics as: the chromosome length's story, the chromosome mutation's story, the chromosome and population fitness' stories.

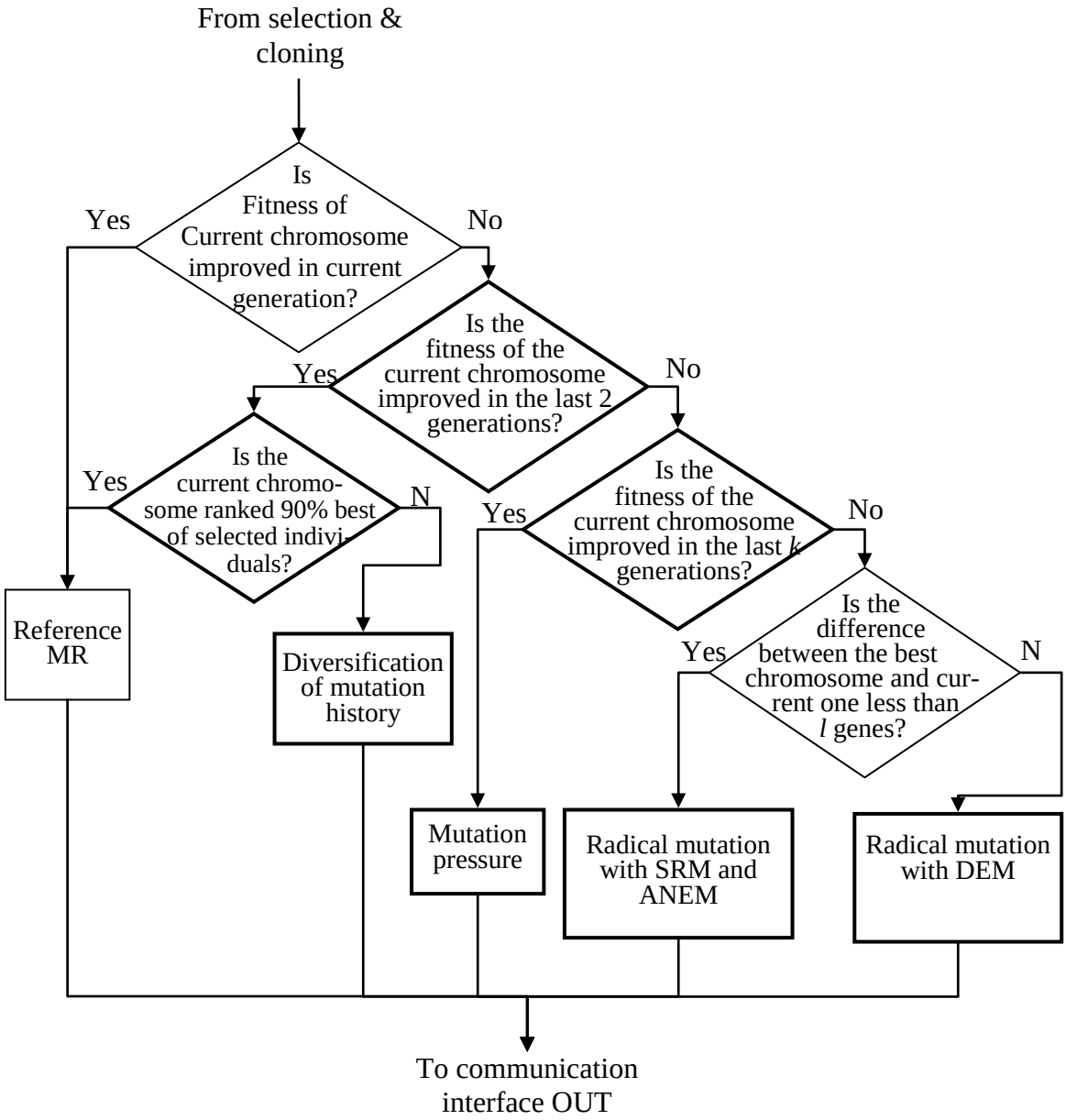


Figure 4-13. The flowchart of differentiated mutation. The new terms are in bold rhomboids and squares.

The idea of an adaptive mutation operator to improve GA performance has been employed earlier and it has been both empirically and theoretically demonstrated that different values of mutation might be optimal at different stages of the evolutionary process [34], [61],[62], [67] and [133]. According to the trade-off paradigm between

exploration and exploitation [153], [156], [154], [155], when evolution gets into a local optimum, the exploration is lacking, and the balance between the exploration and exploitation should be re-established by increasing the mutation.

4.6 Experiment 13: Evolution of 4-Output Voltage Distributor

As the number of inputs and outputs increases, the difficulty level of a given task increases exponentially [119], since additional dimensions are added to the search space. Hence, the evolution of multi-input/output circuits becomes considerably more complex. In Experiment 13, the following issues have been tackled, including:

- Testing the newly developed system with the DM technique described in this section;

- Testing the system on the example of a circuit that belongs to the *third level of complexity*, according to the classification introduced in Section 2.5.2.

4.6.1 Task description

The essence of the *voltage distributor circuit* (VDC) becomes simpler if one looks at a single-source divergent neuron (SSDN) that has one dendrite and many axons with similar functionality [113]. The work of the SSDN does not just include transporting the same signal from a single source to different locations, but also in disintegrating the incoming signal and distributing the result among the outputs.[8] Reasoning for the choice of analogue over digital for the VDC, one should mention that in the natural neural network all (graded and impulse) signals are essentially analogue [114]. Moreover, most of the up-to-date industrial sensors receive stimuli and transduce them into electric potentials in a purely analogue form.

The conventional method of circuit design could easily model a neuron by utilizing

[8] The procedure is in common with the well-known convergent neuron which integrates different signals from different locations into one [7], but it has a backward direction for the signals' distribution.

the up-to-date digital signal processing units, such as controllers supplemented by analogue-to-digital and digital-to-analogue converters. However, a purely analogue circuit, in comparison with a digital one, can provide a considerably shorter delay in the circuit response because - as with any asynchronous circuit - its speed is not constrained by an arbitrary clock; instead, it runs at the maximum speed of electro-magnetic interaction. Furthermore, analogue circuits suggest an economy in power and components. The last argument becomes vital if the difference in the components between the competing circuits reaches multiples of a hundred and concerns such an application as a NN where the number of units (neurons) tends to be enormous.

Knowledge about neurons mostly concerns those convergent ones that integrate multiple signals from dendrites into a single signal to an axon. Divergent neurons are not as widespread in natural neural systems. This last fact is due to the convergent nature of NN, which is mostly caused by a vast diversity of receptors that sense stimuli at a molecular level. That is, any stimuli comes into a natural NN at such a fine-grained level that the networks are left only with converging the mosaic into the pictures, thus solving higher-level intelligent tasks like cognition. On the other hand, most of the up-to-date industrial sensors do not possess such a feature. Thus, and practically, it is reasonable to target a circuit that simulates a divergent neuron that has the ability to disintegrate the incoming voltages from the sensors and distribute them among multiple outputs. Hence, the circuit is called a voltage distributor (VDC). Figure 4-14 gives a general view of a neuron model consisting of three digital circuitry units. Our task is to replace all three units with one analogue circuit.

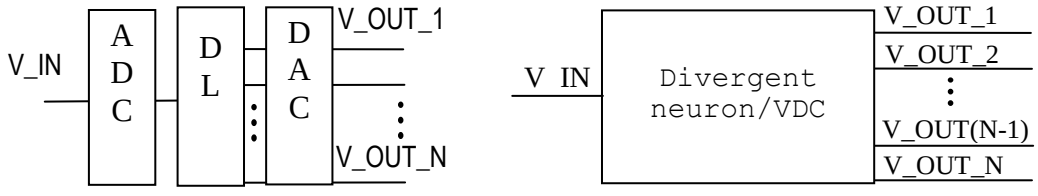


Figure 4-14. A digital (left) and an analogue representation of a one-input multi-output voltage distributor/divergent neuron circuit.

The disintegration task for each output of the VDC involves working in a filter-like

mode that passes the input signal located within a particular voltage band without any change in the form of the signal. For a 4-out VDC, the band-pass for each output equals 5V/4=1.25V: the first output passes the voltages from 0 to 1.25V, the second from 1.25-2.5V, the third from 2.5-3.75V and, for the fourth, the band-pass is 3.75-5V. Figure 4-15a demonstrates separately the transient analysis of every pin of the targeted 4-out VDC. As it can be seen by Figure 4-15a, the aggregated signals from all the outputs must exactly repeat the form of the input piecewise signal without gaps between the signals.

For a VDC, the author was unable to trace any existing device or published work that described an analogue or a digital circuit performing a similar task. The last fact gives an alluring opportunity to challenge the potential of the evolutionary technique.

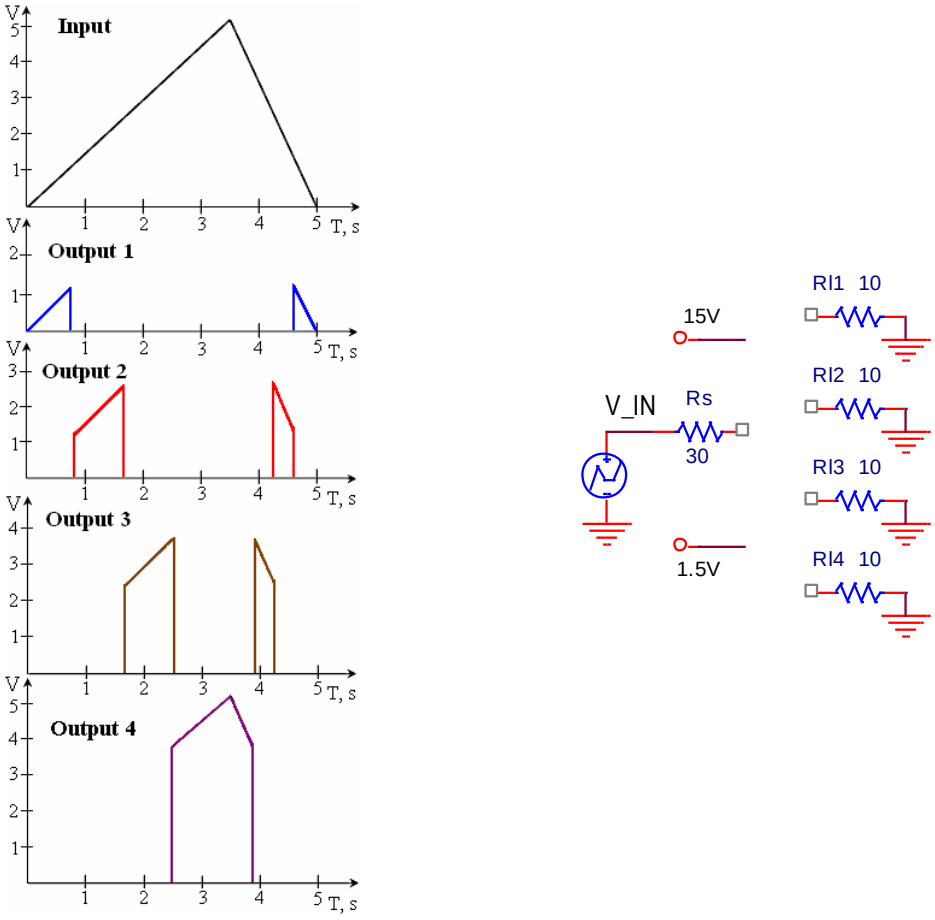
In Figure 4-15b there is an embryo for a 4-out VDC. The embryo consists of a source of piecewise input signal (V_IN), a source resistor (Rs) and four load resistors (Rl1…Rl4). The embryo can also have two sources of direct voltage, allowing the evolution to choose between (or use both) 15V or 1.5V.

4.6.2 Fitness Function

For all design cases, a fitness value is set to a sum over *p* fitness cases of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit:

$$F = \sum_{i=0}^{p} | V_{ideal}^{i} - V_{measured}^{i} |, \qquad [4\text{-}2]$$

where V_{ideal}^{i} is the voltage at the *i*-th point for the ideal response and $V_{measured}^{i}$ is the voltage at the *i*-th point obtained for the evolved circuit; *p* equals 81 time-points. The smaller the fitness value is, the closer the circuit is to the target. The fitness penalizes the output voltage by 10 if it is not within a specified percentage range of the target voltage value.



(a) (b)

Figure 4-15. a) Transient analysis of potentials at the input and four outputs of the targeted 4-out VDC. b) Embryo for the 4-out VDC.

The ES with linear representation and OLG is utilized. Different selection schemes are tried, ranging from 10% to 0.05%, and it is defined that a 1%-selection scheme is the optimum, i.e. 1% of the best chromosomes are selected to be parents in the next generation. Being chosen, each chromosome contributes 100 new chromosomes for the next generation. The ES is deserving of the name of the simplest EA because it does not contain the crossover operation: all the offspring chromosomes are identical to a corresponding parent. The reference MR of 4% is applied.

A population size of 30,000[9] chromosomes is set. Five PCs are used with Intel Core 2 Duo/2GHz processors running at the same time independently of each other. The results presented in the next section are the best out of five runs for each case, with different seeds for the random number generator.

4.6.3 Experimental results

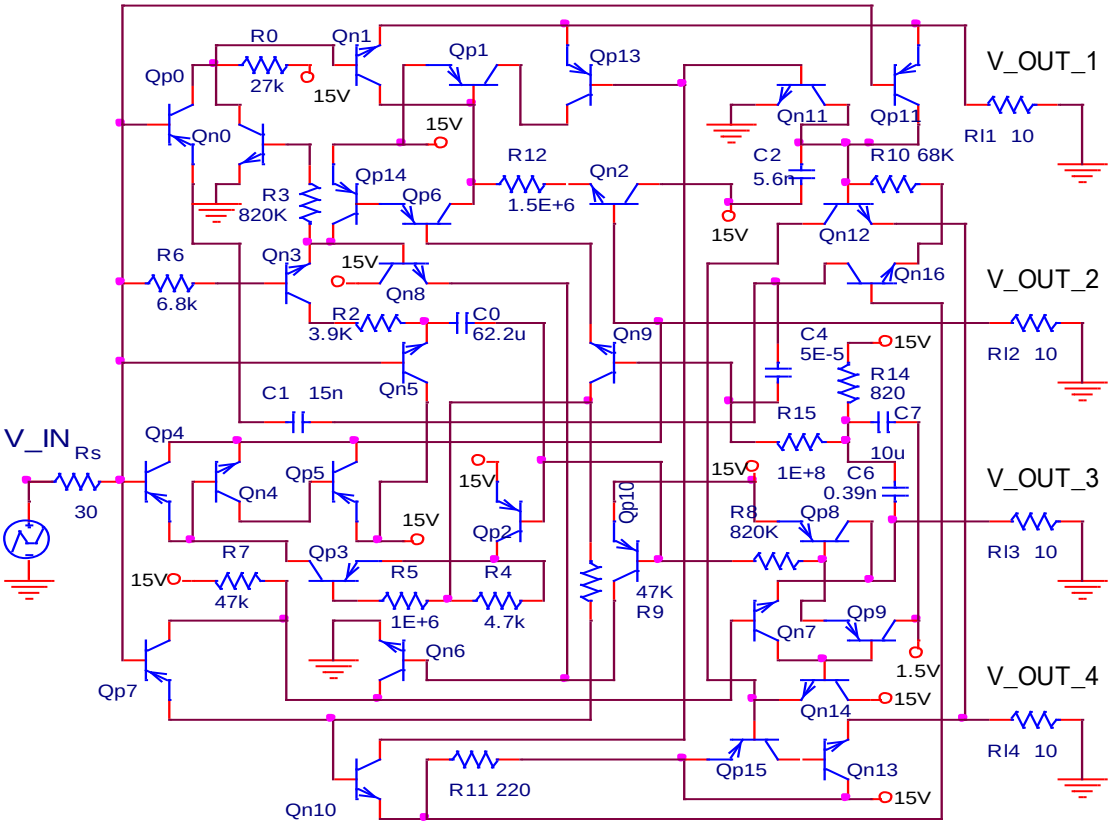


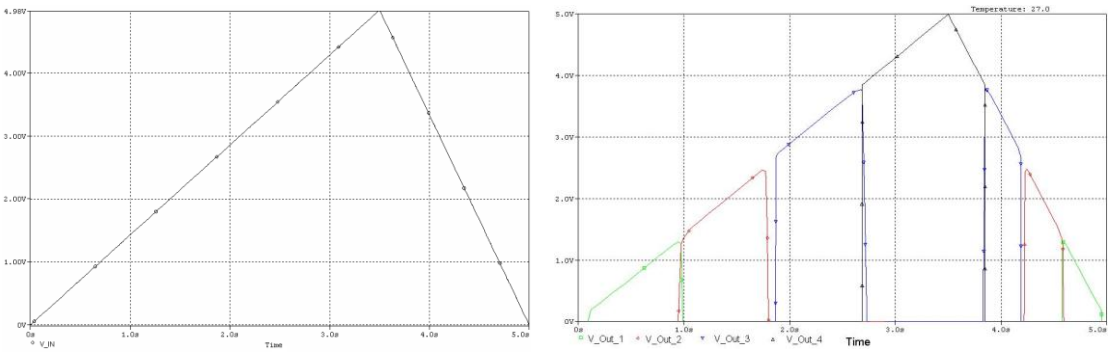
Figure 4-16. The evolved 4-output VDC in Experiment 13.

The average evolution time of the 4-out Voltage Distributor is 123 hours. The best-of-run circuit (Figure 4-16) appeared at the 120th generation and had 51 components (embryo excluded) among which there are 14 resistors, 6 capacitors, 0 inductors, 16 NPN transistors and 15 PNP transistors, with a fitness of 0.38 [116]. The aggregated transient response of the circuit to an incoming signal (Figure 4-17a) - as can be seen by

[9] Despite the good results have been received with this population size, there are no reasons why this size should not be increased. In fact, in here the author is only driven by convenience of processing the PSPICE out-file.

Figure 4-17b - almost exactly repeats the form of the incoming piecewise signal.

In this section, the proposed system evolves the unconventional nonlinear, multi-output and time-dependent functioning analogue circuits. The evolved example is a complex analogue circuit that is able to replace digital logic in its conventionally adopted applications. To succeed with the targets, unconstrained evolution with linear representation, OLG strategy, substructure reuse and *differentiated mutation* is applied.



(a) (b)

Figure 4-17. The transient analysis at the input and outputs of the 4-output VDC. (a) A piecewise signal used during evolution, and (b) the circuit response.

Despite the high strength of the methodology developed and the wonderful test results obtained, the recently developed evolutionary system has failed to evolve the 8-output VDC (presented in Section 5.2.4). This fact caused an author to continue enhancing the method that is described in Chapter 5.

4.7 Summary of Chapter 4

The main contribution that is made in Chapter 4 concerns the development of the *differentiated mutation technique*. It starts by introducing *substructure reuse* and adding this new option to such a mutation procedure as SRM. The new system has been successfully tested on the examples of four computational circuits (CC).

Next, the *qualitative approach* is applied towards mutation in general as well as towards every mutation type, i.e. the base of the measuring mutation is suggested as a locus instead of percentage rate as before. Since a locus is the minimum modification that is possible in analogue circuit evolution, it now plays the role of the unit of measurement. Based on the new unit, each type of mutation is calibrated. The last procedure caused each mutation type to now correspond to an exact number of loci to be mutated. However, after introducing the *virtual mutation*, each mutation type there corresponds to a *mutation interval*. The concept of *virtual mutation* enables us to significantly diversify the mutations applied to chromosomes. More choice gives more abilities. The final concept that concludes the *differentiated mutation technique* is the *mutation strategy*. The mutation strategy consists of three main operators, which are the *diversification of a mutation history*, *mutation pressure*, and *radical mutations*. All of the latter are proposed to revive the evolution in case of fitness becoming stuck, with the help of a new virtual mutation methodology. Finally, the new approach is successfully tested on the example of a 4-output voltage distributor circuit (VDC).

It should be noted that the novel approach based on the DM technique being applied towards the similar to 4-output VDC circuit - but with eight outputs - failed to evolve towards an acceptable solution.[10] This obstacle has motivated the continued further development of the system, starting from the notion of *incremental evolution*.

[10] The results of this experience are not presented in this work.

Chapter 5. Incremental Parallel Evolution with Adaptive Parameters

In this chapter, first of all, the technique of incremental evolution is introduced and then challenged by a task so to evolve the 8-output voltage distributor circuit (VDC).

The results from the experiment then saw the discovery of an original parallel evolution strategy that is characterized by very low selection rates. Parallel island-model evolution runs in a hybrid competitive-cooperative interaction throughout two incremental sub-stages. The adaptive population size is applied for the synchronization of the parallel sub-evolutions. The novel system is tested on a familiar 8-output VDC and the challenging Time Interval Meter Circuit (TIMC) that performs the functions of several digital circuits.

5.1 Incremental Evolution

5.1.1 Types of incremental evolution for analogue circuit synthesis

As has been already mentioned in Section 2.4.3, *staged incremental evolution* is regarded as one of the main techniques for tackling the scalability problem. It is also noted that the physical nature of analogue circuits limits the application of a "divide-and-conquer" approach in comparison with digital circuits. Therefore, under the *incremental evolution* of analogue circuits it is further meant, first of all, not the independent evolution of subtargets but rather the evolution of the current subtarget together with all the subtargets evolved previously. That is, if one already has the evolved subtarget when evolving the second one, the first solution must participate in that evolution, being encoded in the chromosome. The last proposition means that the length of the chromosome after the incremental procedure will continue growing.

However, despite the fact that the chromosome length significantly increases with each sub-stage, the need to involve every gene of the previously evolved sub-solutions into every *evolution operation* is not necessary. There are two operations that are involved in this regard:

- The *evaluation* process, which requires all of the parts of the chromosomes to participate, i.e. it is important to get the adequate fitness value of the whole chromosome.

- Since ES is utilized - where the recombination is not used - another important evolution operation is that of *mutation*. Three methods are considered in terms of the degree of the involvement of the genotypes of the previous sub-solutions into the mutation:

    1. *Non-involvement*. This is when the fragments of the chromosome that belong to previous sub-solutions do not participate in all kinds of mutation. On one hand, this option keeps the solution space constrained and saves computing effort. On the other hand, removing the opportunity for the previous sub-solutions to adapt their structures and parameters to a new more general solution may obstruct the finding of a current sub-circuit and even leave the process out of any solution. In this case, the whole "responsibility" for the adjustment of the sub-solutions to each other lies on the structure of the currently evolving sub-solution. This means that with each sub-stage the current sub-task become increasingly complex.

    2. *Partial involvement*. To avoid the problems that might appear in the first option, the second option suggests a "partial evolution," enabling some loci of the previous sub-solutions to participate in the evolution along with the rest of the chromosome. These loci are represented by such as the components' names and nodes and the parameters of the evolved sub-circuits that are located at the junctions between the currently evolving sub-circuit and the previous ones. These junctions are predefined when one is decomposing the task into subtasks at the start. One can expand the parts of

the chromosomes that participate in the current sub-stage mutation by adding not only the genes coded for the components at junctions but also by adding the genes coded for their neighbours and the neighbours of their neighbours, etc. (Figure 5-1). In this case, the "responsibility" for adjusting the sub-solutions to each other is distributed among the structures that belong to all the sub-solutions, which should facilitate the evolution for each sub-stage.

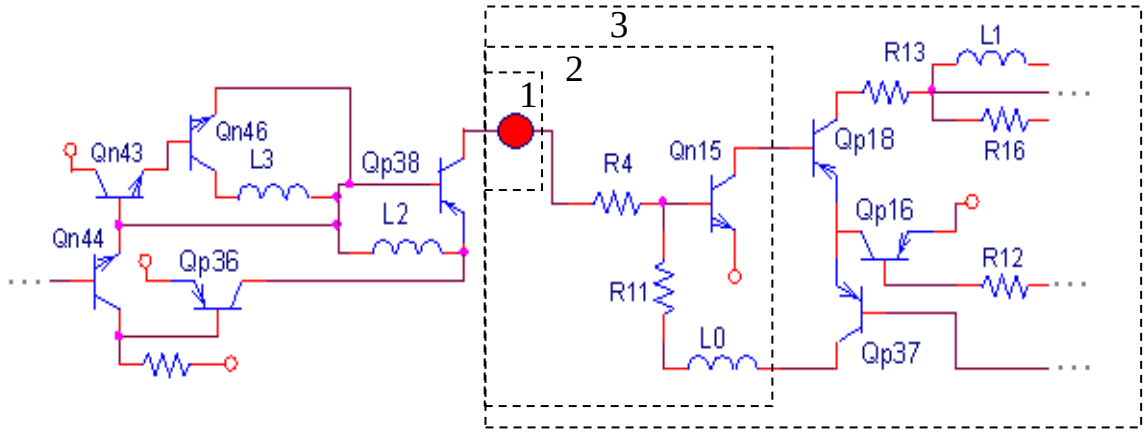


Figure 5-1. The incremental approach: two sub-circuits that are jointed in a point marked by a red circuit. On the right is a sub-circuit that was evolved first, and to the left is a currently evolving one. When evolving a sub-circuit, there are three degrees of involvement of the parts of the previous solution(s) in the current process: non-involving (marked by the dotted square 1), partially, where there are components neighbouring to a junction point (square 2) and full, where every component of a previous sub-circuit(s) takes a part in the evolution along with the components on the left side of the figure (square 3).

3. *Full involvement*. The third case is when the whole genotypes of the previously completed sub-tasks have the same rights to participate in all kinds of evolutionary operations as the genotypes related to the current subtask. In this case, the power of the "divide and conquer" method drastically falls down due to the extreme expansion of the chromosome length and the search space. However - and on the positive side of such the approach - while it is applied to deliberately easy problems where

scalability is not an issue, the unconventional designs and the foreseen component economy of the solutions resulting are. In the frame of this work, which has as its goal the construction of a system that will be able to synthesise complex analogue circuits by the method of *full involvement*, it is less helpful.

Since the rest of the targets are seen as the most complex circuits and are going to be decomposed into subtasks in order to diminish the scalability problem, only methods 1 and 2 from above are found to be suitable and only they will be considered any further. As such, Experiments 14 and 15 both use the *non-involvement method*, Experiment 16: Evolution of TIMC uses the *partial involvement method*.

5.1.2 Types of incremental coding

Another very important aid that is helpful for incremental evolution is *incremental coding*. When one is utilizing extrinsic evolution with the help of PSPICE (or any other xSPICE), one could utilize the PSPICE built-in function for the sub-circuits that are instantiated by using the letter "X". This causes the referenced sub-circuit to be inserted into the circuit "using the given nodes to replace the argument nodes in the definition" [91]. It allows a block of circuitry to be defined once and then used in several places. This built-in PSPICE coding can be handy when one would like to easily identify and implicitly protect some fractions of a chromosome as well in easing up the operations over those individuals that are too lengthy. It will be called "*X-coding*" from now on.

There are two kinds of analogue circuit decompositions that may exist. The first one is the decomposing of the circuit into sub-circuits that are in *series* to each other. In this case, if the partial involvement method is used, each next evolving sub-circuit will involve in its synthesis a part of only that sub-circuit which it is connected to serially. Figure 5-2 demonstrates this idea generally.

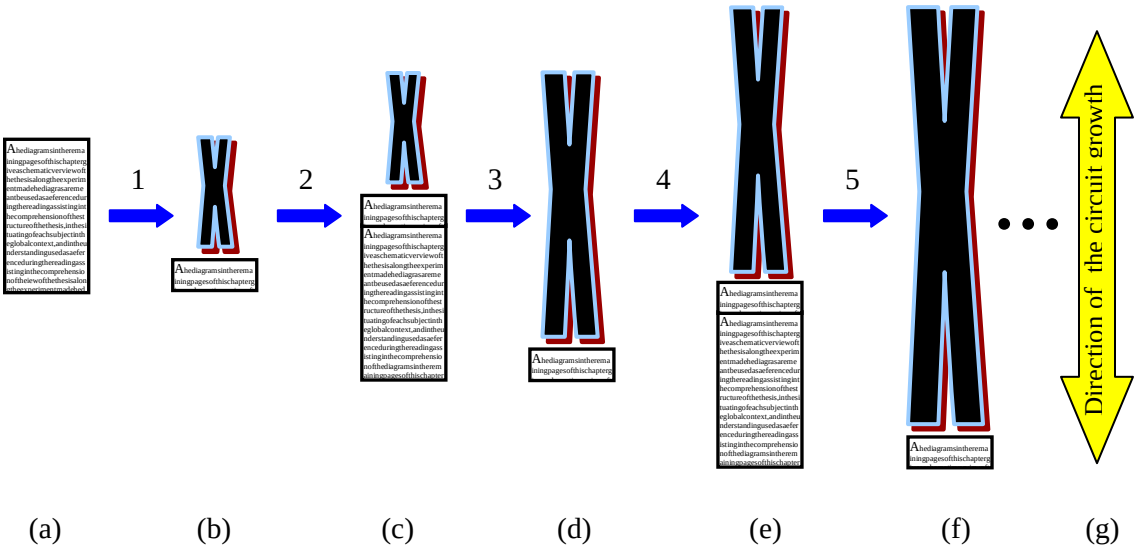


Figure 5-2. The general view of *series incremental coding* in cases of the partial involvement method. Only three sub-circuits are shown. By the letter "X" the X-coded part of a chromosome is indicated and which represents the non-mutated part of the chromosome. By the text box, the deck part of the chromosome is indicated. From left to right: (a) the first sub-solution deck after the first operation becomes partially X-coded (b). Being incremented and evolved on the second sub-stage (operation 2) it has a view on (c). The result of the second sub-stage is again partially coded (operation 3) and is evolved (operation 4) towards the third sub-target (e). The evolved third sub-circuit (e) is X-coded (operation 5) to (f). On (g) the sequential direction of circuit growth is shown. In the case of the non-involvement method, the small squares right under the X symbol will not be presented. In the case of the full involvement method, there will not be any X symbols in the figure, i.e. there will not be any parts of the chromosomes that are protected from mutations.

The second case is when the decomposed sub-circuits are in parallel to each other. Here, if we consider the partial involvement method, the currently evolving sub-circuit will involve in the mutation procedure those structures of all the previously evolved sub-circuits. Figure 5-3 demonstrates how the development of the *parallel incremental coding* takes place.

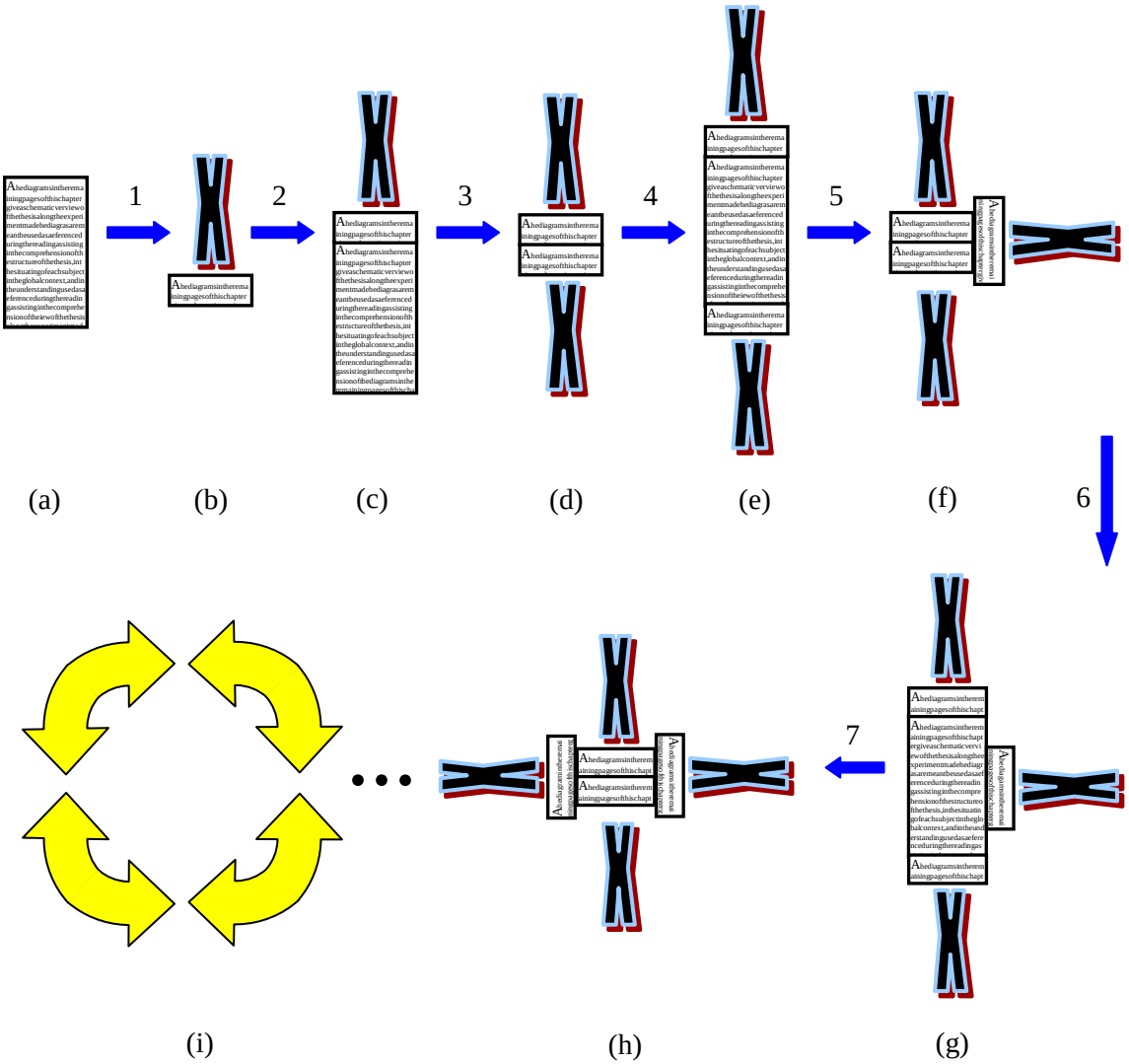


Figure 5-3. The general view of *parallel incremental coding*. Only four sub-circuits are shown. By the letter "X" the X-coded part of a chromosome is indicated and which represents the non-mutated part of the chromosome. By the text box the deck part of the chromosome is indicated. From (a) to (f): (a) the first sub-solution deck after the first operation becomes partially X-coded (b). Being incremented and evolved on the second sub-stage (operation 2) it has a view on (c). The result of the second sub-stage is again partially coded (operation 3) and is evolved (operation 4) towards the third sub-target (e). The evolved third sub-circuit (e) is X-coded (operation 5) to (f). On (g) and (h), the coding of the circuit with fourth sub-circuit and its X-coding are shown. On (i) the direction of the circuit growth is shown. In the case of the *non-involvement* method, the small squares right under the X-symbol will not be presented. In the case of the *full involvement* method, there will not be any X-symbols in the figure, i.e. there will not be any parts of chromosomes that are protected from mutations.

Regarding the already mentioned *full involvement method*, Figures 5-2 and 5-3 will not contain the X-coded parts of chromosomes, i.e. there is no need for the protection of any of the chromosomes' parts from mutation.

Conversely, in the *non-involvement method* the partial decks will not be presented because they will be covered over by X-coded parts of the chromosomes.

In the experiment below, the parallel decomposition of the main circuit is utilized and the *non-involvement method* has been applied.

In the next section, the developed system armed with *staged incremental evolution* will be applied towards the challenging task of the 8-output VDC. This target is much more sophisticated than any of the others already tried in this work. Furthermore, it looks like an ideal task for probing incremental evolution since an eight parallel output circuit suggests the decomposition of the main circuit to eight parallel sub-circuits.

5.2 Experiment 14: Evolution of 8-Output Voltage Distributor (Phase 1)

The VDC has already been introduced in detail in Section 4.6.1. However, here eight outputs instead of four are set for the target. If in the earlier case the target had evolved without task decomposition, for current problem this approach has failed. Five attempts - the results of which are presented in Section 5.2.4 - to evolve the 8-output VDC without incremental evolution have failed at earliest stages of the experiments. Thus, this failure assures us that the introduction of the incremental technique to the system provides the necessary step in going forward.

The motivation for the choice of an 8-output VDC as the target for evolution is the circuit's functionality in that it performs as the single-source divergent neuron (SSDN), which has one dendrite and many axons with similar functionality [113]. In Section 4.6.1 this has already been described in detail along with the perspectives for the evolution in targeting such circuits as VDCs.

According to the methodology described, eight subtasks corresponding to eight parallel sub-circuits are set. Each sub-circuit is responsible for receiving an incoming signal and producing an output signal to its own output pin. If the first task is a design of the first sub-circuit, the second task is a design of the first and the second sub-circuits, the third task is a design of the first, the second and the third sub-circuits, and so on. Finally, the 8-th task is a design of all eight sub-circuits constituting the whole of the VDC. The evolution starts from the first sub-circuit and upon its completion moves to the next one. It is by the degree of the involvement of the genotypes of the previous sub-solutions into the mutation of the current sub-solution that the *non-involvement method* is utilized.

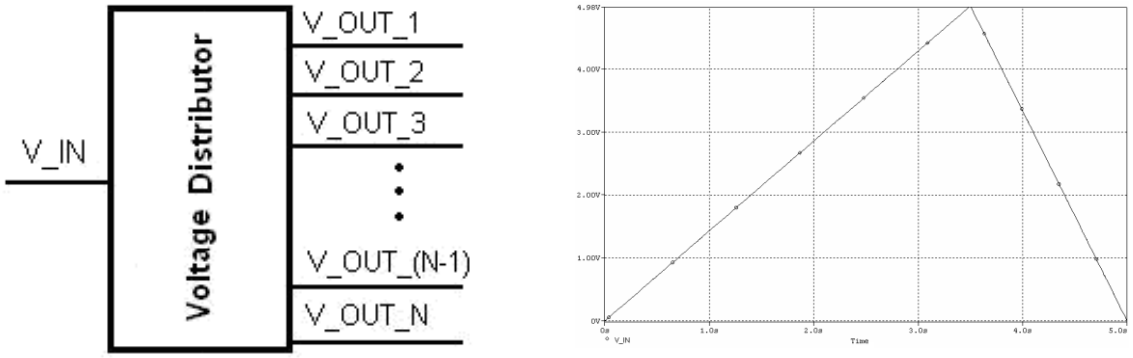
Experiment 14 below aims to tackle several issues, including:

- Testing the incremental method proposed in this chapter;
- Testing the system on a task that belongs to the 3rd level of complexity, according to the classification introduced in the beginning of Chapter 4;
- Testing the system on an unconventional application analogue circuit.

5.2.1 Task description

The general view of the *N*-output Voltage Distributor is presented by Figure 5-4a, where in our case *N=8*. As an input signal for the targeted circuit the same piecewise voltage pulse has been taken as was the case with the 4-output VDC: starting from 0V, going up to 5V for 3.5sec and down to 0V for the last 1.5sec (Figure 5-4b). The task for each output was working in a filter-like mode which is to pass through the input signal that is located within a particular voltage band, saving the form of the input signal. This is to say that the band-pass width for each of the outputs was the same and equalled 0.625V: the 1st output passes only the voltages from 0 to 0.625V, the 2nd output passes only the voltages from 0.625-1.25V, the 3rd band-pass is 1.25-1.875V, the 4th is 1.875-2.5V, the 5th is 2.5-3.125V, the 6th is 3.125-3.75V, the 7th is 3.75-4.375V, and finally the 8th is 4.375-5V. The summary of the transient analysis at the input and eight output

pins of the targeted ideal 8-out Voltage Distributor is presented by Figure 5-5. As it could be seen from Figure 5-5, the aggregated signals from all outputs must exactly repeat the form of the input piecewise signal in Figure 5-4b, without gaps between the signals.



(a) (b)

Figure 5-4. (a) The general view of the proposed *N*-out Voltage Distributor. (b) The asymmetrical input piece-wise voltage signal.

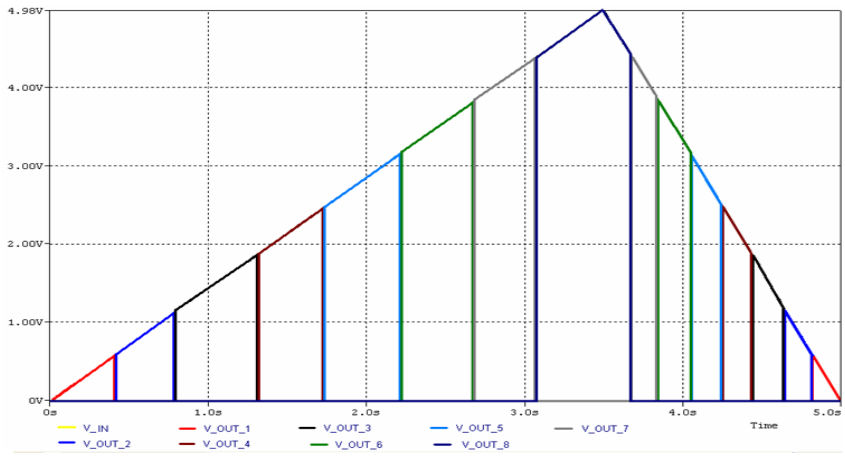


Figure 5-5. The ideal united transient analysis of potentials at the input and eight output pins of the targeted 8-out Voltage Distributor.

As mentioned above, incremental evolution was introduced for the design of the 8-output VDC. The fitness function was incremented each time, whenever the current task was fulfilled. Eight tasks corresponding to eight sub-circuits were set. Each sub-circuit

was responsible for acquiring the incoming signal and for provide the outgoing signal to the corresponding output. While the first task was the design of the first sub-circuit, the second task was the design of the first and the second sub-circuits, the third task was a design of the first, the second and the third sub-circuits and so on. Finally, the eighth task was the design of all eight sub-circuits that is the whole VDC. Each time the current sub-circuit is finished the system X-codes it, protecting it from mutation in further sub-stages. For each task, a new fitness function was introduced which incrementally counted the fitness of all the sub-circuits evolved at that time.

5.2.2 Dedicated topological reuse

The use of SRM has been discussed before, where the choice of one or another kind of substructure fully depends on a random choice among the *mutation ways*. However, in the case of an 8-output VDC, as well as in most cases of circuits with a multitude of outputs/inputs, it is possible to apply another type of topological reuse – the *dedicated topological reuse*. Due to the similarity of the functions that sub-circuits perform independently, in the case of the VDC and in order to pass through the particular voltage band and to stop the rest, the evolution's task (except for the first sub-circuit) is just to reprocess the previously evolved sub-circuits into a new sub-circuit with new properties - what for VDC is a new pass band. That is, the substructure to be reused is quite definite and the junction points for it are also known.

The main advantage of such an approach is the possibility of starting the evolution of the next sub-circuit (e.g. the 3rd) based on the reuse of the previously evolved sub-circuits (i.e. the 1st and 2nd).

Thus, there are two types of substructure reuse that are implied in the frame of an 8-output VDC. The first one - mentioned in Section 4.1 - is the SRM and is a part of differentiated mutation operation. Another kind of reuse is suggested as being applied during the increment in between the transition from one sub-circuit to another. If the first type of the substructure is limited to six components, the size of the second is unlimited; if the place for the first substructure is randomized, the place for the second one is

definite: between the corresponding source and load resistors in Figure 5-6. Another kind of difference between these two is where if the first one can be labelled “building blocks,” the second one is closer to the “framework skeleton,” since it aims to be used only once in the current sub-stage. It is reasonable to suppose that during evolution the DEM procedure may help to remove those components that were essential particularly for the functioning of the previously evolved (reused) sub-circuit, while the other mutation procedures contribute to finding components that help the functioning of the current sub-circuit.

The additional database is proposed to store the dedicated substructures so if it comes to an 8th sub-circuit evolution, the database should already store seven topologies. The system tries to reuse the substructure that was taken from a neighbouring sub-circuit first. Then, if the evolution is stuck, the system tries to utilize the other available topologies.

For an 8-output VDC as well as a 4-output one, the author was unable to trace any existing device as well as any published work that described an analogue or digital circuit which performed a similar task. This gives an alluring opportunity to challenge the potential of the purposed evolutionary technique, because the proposed circuit is going to function in a single analogue mode instead of a number of digital operations (Figure 4-14). Starting from CCs and then looking further, this tendency for targets will be kept.

5.2.3 Fitness function and embryo

Since the aim of the experiment is to run non-stop throughout all the sub-stages, a dynamic fitness function is introduced similar to “adaptive fitness schedule” from [59]; i.e. the fitness function is incremented “whenever the current fitness threshold is reached by at least one chromosome in a population.” The fitness function scheduled to each incremental sub-stage as a simple sum of the fitness values of all the sub-circuits evolved at the time. The final fitness function of the 8-output Voltage Distributor is:

$$F=\sum_{i=1}^{i=8} F^i ,$$

where F^i is a fitness value of the sub-circuit i and which is calculated in the following way: the PSPICE simulator performs a transient analysis at each output for five seconds at 81 equidistant time-points; a fitness value F^i is set to the sum and over these 81 fitness cases of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit; the fitness penalizes the output voltage by 10 if it is not within 50% of the target voltage value. The smaller the fitness value is, the closer the circuit to the target.

In Figure 5-6 there is an embryo for an 8-out VDC. In accordance with all previous tasks, it consists of source of input signals (V_IN), eight source resistors (Rs) and eight load resistors (Rl1…Rl8). The embryo also has two sources of direct voltage, allowing the evolution to choose between (or use both) 15V or 1.5V. This embryo is created by way of analogy with the 4-output VDC, but with inputs split apart for the comfort of the incremental evolution.

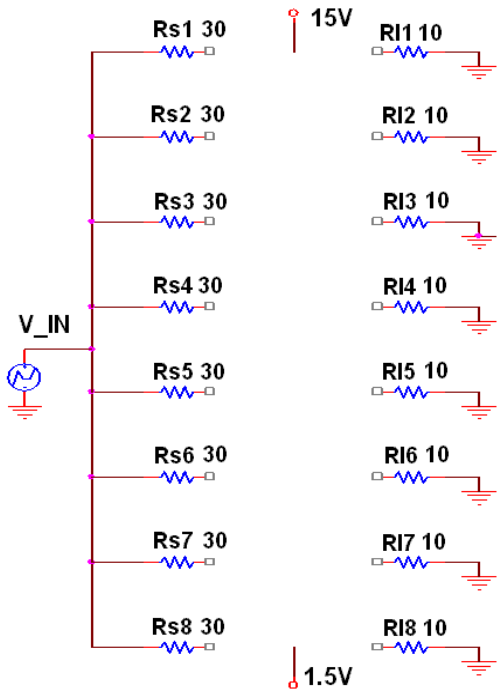


Figure 5-6. The embryo circuit

Five types of components are suggested that the targeted circuit should consist of: an NPN bipolar transistor (Qn), a PNP bipolar transistor (Qp), a resistor (R), an inductor (L) and a capacitor (C).

The termination criteria for the whole circuit is the consecutive summary of the termination criteria of each sub-circuit, namely that if the fitness value did not improve over 20 consecutive generations or else if the best sub-circuit reaches more than 100 components.

The ES with a 1%-SR scheme is applied, i.e. S=1% of the best chromosomes are chosen for the next generation. Upon being chosen, each chromosome contributed 100 new chromosomes for the next generation. A population size of 30,000 chromosomes is set. Five PCs have been used with Intel Core 2 Duo/2GHz processors running at the same time independently of each other. The results presented in the next section are the best out of five runs for each case with different seeds for the RNG.

5.2.4 Experimental results

At the beginning of this section, the brief results of the experiment to evolve an 8-output VDC without task decomposition will be presented. The evolution has been run 5 times with the same experimental settings as described in Section 5.2.1, and without *incremental evolution.* The FF and the embryo were similar to the FF and embryo of the 4-output VDC (formula [4-2] and Figure 4-15b), but with an extension to 8 outputs. All 5 experiments are shown by Figure 5-7.

As can be seen by Figure 5-7, the reaching of the 20-generation-limit termination criteria (i.e. the stalling effect problem) at the early stages of the evolution was the reason for the experiment's failure. It has been claimed before in [48], [119], [139] that with an increase of the input/output pins of the targeted circuit, the complexity level becomes extremely high and the most effective tool for tackling this problem is "incremental evolution." The fact that the stalling effect took place at the early stages of the evolution indicates that the targeted 8-output VDC is too hard a task for an existing

evolutionary approach. Therefore, incremental evolution has been applied to this task later on.

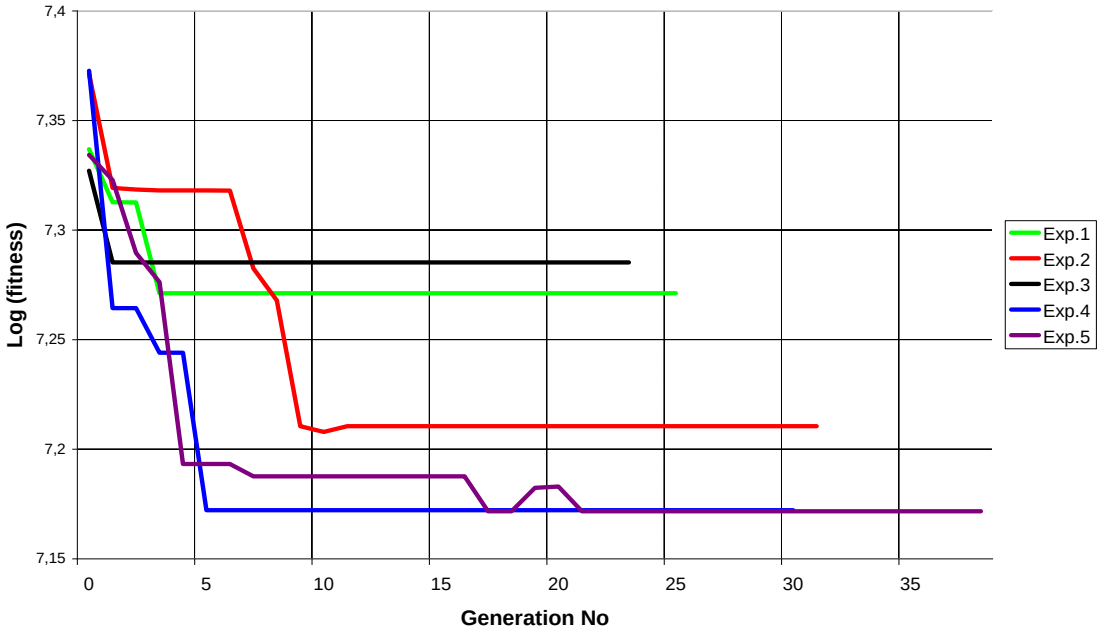


Figure 5-7. Five runs for the evolution of an 8-output VDC without "incremental evolution." For better visualization, the logarithmic values of fitness are presented.

The rest of the section relates to the incremental evolution of an 8-output VDC decomposed into 8 subtasks. The first sub-circuit is evolved relatively quickly, within 10 hours of the start and resulting in a 10-component circuit with a fitness of 0.095 after 76 generations. The second sub-circuit that was evolved with the *dedicated reuse* of the first one took twice as much time (19 hours), with 22 components and 132 generations, converging with a record fitness of 0.028. At the third sub-stage, it began to reuse the second sub-circuit, as was set by the rule. However, after a long term run and 20 consecutive generations without improvement, the system reused the topology derived from the first sub-circuit - and this try has succeeded. After 110 generations (at the second attempt) the sub-circuit resulted in a 16-gene chromosome with a 0.174 fitness value.

Since then, all attempts to evolve the next sub-circuit have failed. All three substructures were tried during evolution, with several attempts per substructure. The resulting 48-component circuit is shown on the Figure 5-8. The details of the whole experiment are presented in Table 5-1.

This result has stimulated the continuing development of the method in order to finish the current task and deal with even more complex targets.

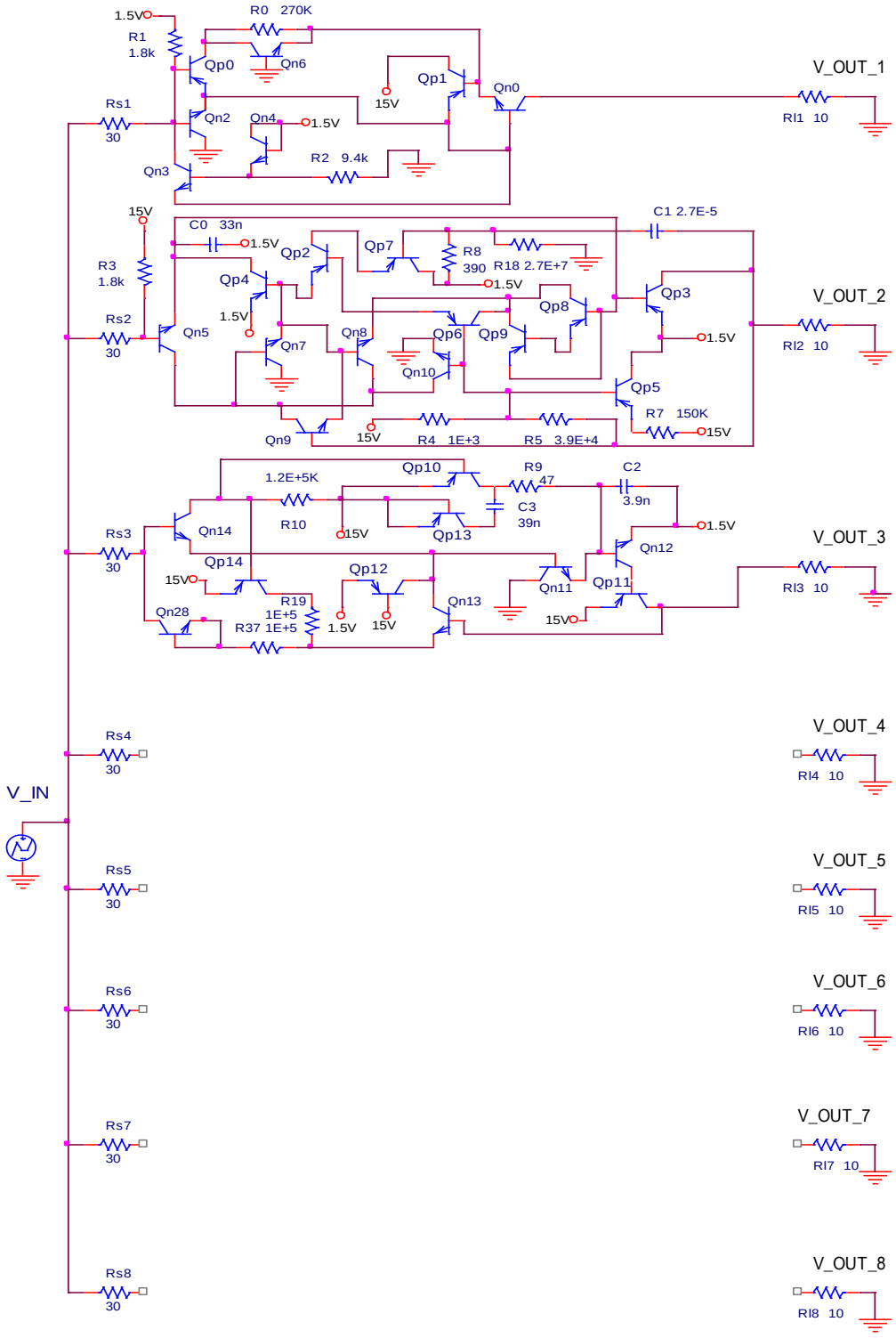


Figure 5-8. The result of the evolution of the 8-output VDC that failed to evolve at the 4th sub-stage in Experiment 14.

Table 5-1. The summary for the sub-circuits of the 8-output Voltage Distributor

	Fitness	**Component No.**	**Generation succeed**
Sub-circuit 1	0.095	10	76
Sub-circuit 2	0.028	22	132
Sub-circuit 3	0.174	16	110
Sub-circuit 4			
Sub-circuit 5			
Sub-circuit 6			
Sub-circuit 7			
Sub-circuit 8			
Total	**0.297**	**48**	**318**

5.3 Parallel Evolution with Migrations at Incremental Stages

5.3.1 Introduction

In this section, the power of the ES-based EHW system will be verified through two challenging tasks: the *8-output VD*C that has failed to evolve and the *Time Interval Meter Circuit* (TIMC), which is the core of the modern laser rangefinder. To solve these problems, the ES-based system is upgraded with a combination of the novel adaptive individual-level *Differentiated Mutation* (DM) and the *Winner-Dominates-Winner-Cooperates* (WDWC) parallel evolution strategy, formed by means of parallel island-model evolution.

When combined, these two techniques make the evolutionary system as "very narrow focused search tool," which is called - for simplicity - *Very Narrow Focused Evolution* (VNFE). The literature review provides - on the subject of the "optimal selection rate" - the idea that this mode of selection is a mechanism which increases the mean fitness of a population while "having the least deleterious effect" [95] on the genotypes, and thus is directly proportionate to the size of a population. In this section,

relatively large populations are used (from 15,000 to 35,000 individuals evolving in parallel), for each of which it is suggested that use is made of very low selection rates (SR): from 0.2% to 2% and, aggregately the SR of the system reaches 0.048%. Furthermore, the proposed technique theoretically enables all the parallel evolutions to "focus" on a single chromosome at one generation, which in our experimental case can bring up and of SR of 0.0006%.[11] As can be seen, premature convergence has not become a problem due to the DM technique and VNFE.

5.3.2 Parallel evolution with migrations at incremental stages

It has been already mentioned that since the evolution of low-pass filters several systems evolved in parallel towards the same target. In the previous section, parallel systems have been tried so as to evolve the fourth sub-stage of an 8-output VDC, but this failed. The utilization of parallel processors until now has always played the simplest role of alternate evolutions, providing results from which to choose the best; i.e. there were different populations, each of which acted as an independent ES with each one separately initializing, ranking, selecting and cloning and with mutation performing only within populations. Each population ran on a separate processor. The final results of the evolution were always been manually compared, and the best solution had was chosen. No kinds of migrations have ever been tried. Figure 5-9 generally presents the procedure thus described.

The difficulties in the previous experiment have led to the search for new techniques that may bring better results for the same amount of resources. The further strengthening of the parallel methodology was decided as the first choice for enhancing the system. Furthermore, in Section 4.5.1.3 we see presented the ways for differentiated mutations, with number for a 50-gene chromosome at 112. This number means that each chromosome has 112 forms of mutation to choose from. It is obvious that a good methodology is that one which will suggest enough individuals for the probation of the maximum mutation ways, i.e. the SR should be decreased. For instance, for an average population size 30,000 individuals, the SR should be 1% in order to enable 300 clones

[11] While during an experiment we have achieved a selection rate equal to 0.048%, theoretically the value 0.0006% is reachable.

from one chromosome, but it is so far not enough for an 80-component circuit to apply about 1000 mutation ways. Furthermore, this number does not reflect all of the diversity of the mutations possible if we want to take into account the random choice of the particular place inside a circuit where the mutation must occur. Therefore, the reasonable selection rate of about 0.2% - which brings about 1500 clones per individual - is the target. However, by lowering the SR, another problem arises, namely the depletion of the gene pool. To tackle this latter problem, the population size is increased.

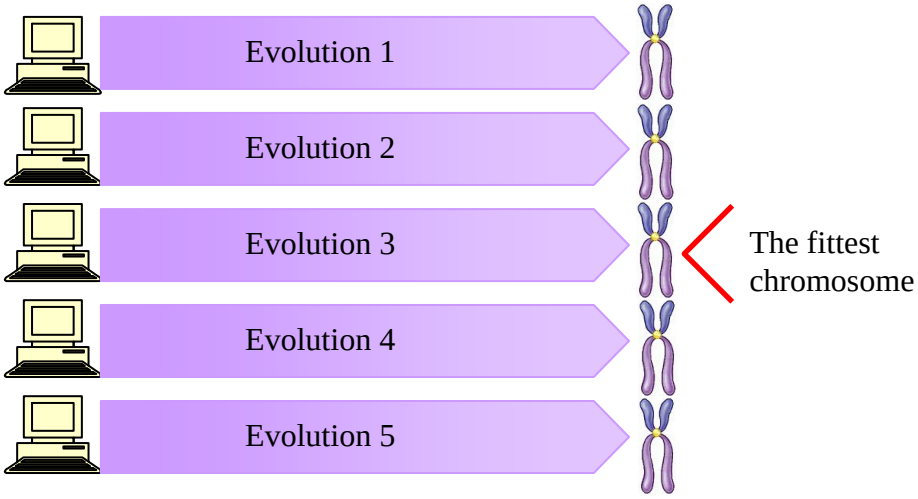


Figure 5-9. Utilization of parallel sub-systems in past approaches.

The idea of a narrow SR comes from experiments rather than from the theory of EA. In earlier works, the author experimentally noticed during work over computational circuits [115] that the system that integrates SRM as a part of mutation procedure performs better with an SR 10 times lower than that conventionally adopted before (from 10 to 1%). Despite the current system, and armed with the DM technique which successfully evolved the 4-output VDC [116], it has failed when evolving the final part of the 8-output VDC [65]. And one of the main reasons for this failure is held to be the lack of computer resources and the high selection rate.

Therefore, the following algorithm is suggested to continue the evolution of an 8-output VDC:

1. The resulting 8-output circuit with three designed sub-circuits becomes a single chromosome to start with for five independent sub-systems (Figure 5-10);

2. Each sub-system is evolving the first (fourth) and every next sub-stage independently of the others;

3. All the sub-systems work over the particular sub-target at the same time. The termination criteria must be applied to each evolution simultaneously in order for all sub-systems to switch to the next sub-stage.

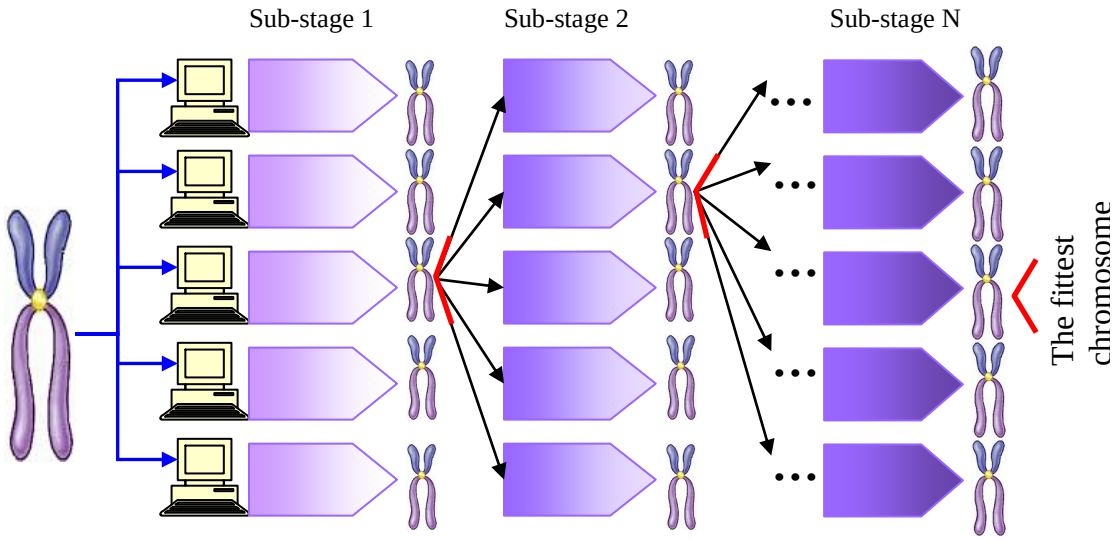


Figure 5-10. The utilization of parallel sub-systems in the second phase evolution of an 8-output VDC. A single chromosome produced in Experiment 14 (Phase 1) is an embryo for every Sub-system. There are migrations by black arrows from the sub-system produced the best individual after every sub-stage.

4. After each sub-stage, the best chromosome is chosen automatically among all the resulting individuals of the sub-systems. The fittest is downloaded to every sub-system. The last ones are restarted;

5. The process continues until the last sub-stage is terminated at all the sub-systems and the fittest is selected.

It should be remembered that for every sub-system the following are the most important features of the proposed technique which have been set:

1. The linear (direct) circuit representation is proposed for use, similar to that exploited in [28], where each component in a circuit was coded in a gene. Whether it is a 2-pin or 3-pin component, the component's features (nodes, parameter and name) are coded into four loci;

2. When evolution passes from one incremental sub-stage to another it applies X-coding.

3. For resistors and capacitors, there are 84 and 96 values of E-12 series, i.e. there are seven and eight decades corresponding with 12 parameters for each chromosome available for evolution.

4. The OLG strategy has been utilized in a manner similar to that described in [28], where different genotype varying strategies have been compared. In OLG, the chromosomes are allowed to increase as well decrease their lengths with the help of DEM and parsimony pressure. However, in the long term prospective genotypes grow up, which is in accord with natural evolution: the more complicated the behaviour of an individual, the longer the chromosome it requires. Due to OLG - during these experiments - the growth difference has resulting in a size difference of 7-8 genes in one population (Figure 5-11).

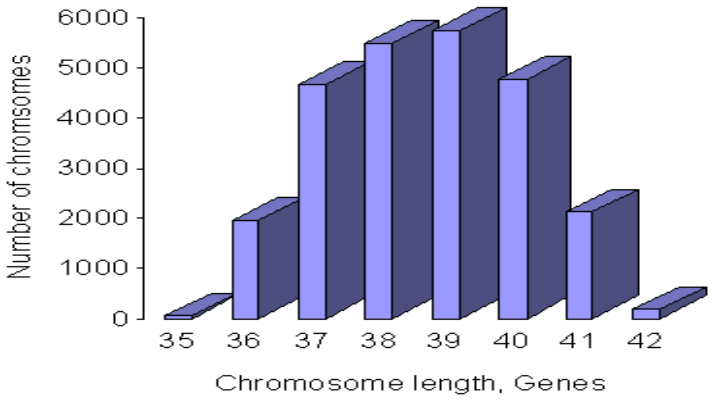


Figure 5-11. Distribution of chromosomes after generation No.46 among different lengths in a

25,000-population located in the 4-st PC.

5. The roulette-wheel selection scheme is used with a selection strength of $\beta=\infty$ [95]. From each population, only 0.2% and above of the best individuals are chosen as parents for the next generation. The single best chromosome with all its properties always stays as a reference for the individuals of future generations until a new one appears with features exceeding it;

6. During the ranking procedure, when comparing two or more chromosomes with identical fitness values (with a precision of 5 decimal digits) and genotype length, only one goes to the next generation. This is because it is supposed that any chromosomes with the same properties must have identical genotypes and thus replicate each other;

7. Each descendant inherits the mutation and fitness story of its ancestors from previous generations, stored in the RAM in the form of decimal values corresponding to the types and ways of mutations and the fitness story for the last *k* generations. These stories are helpful for the mutating operation;

8. Each individual of a new population is mutated according to an individual-level adaptive DM technique with an adaptive mutation parameter;

9. During the ranking procedure, the second objective with the help of a *pressure-constant* is introduced, i.e. the parameter of a genotype length. According to the classification given in [61] to the control parameters, the *pressure-constant* is a *deterministic dynamic penalty parameter*. It does not depend directly on the parallel evolutions, but it is adapted by the local evolution. At each sub-system, the *pressure-constant* is set to the same initial value 40. However, if two generations following each other do not bring better individuals, this number is increased by two, which causes the component-reducing pressure to decrease - ten generations without the best chromosome update make the pressure constant equal to 60, etc. A limit of 64 is set in case the stagnation period continues for more than 12 generations, i.e. the *penalty* release can

reach 60%. Conversely, if two successive generations have brought a fitness improvement, the *pressure-constant* is reduced by two. This strategy leads to an inevitable gradual weakness of the selective pressure due to the permanent growth of a chromosome's length, complexity and solution space. The main reason for introducing it is that it is the simplest technique to implement: it requires only the straightforward modification of the evaluation function;

10. The pruning procedure is enabled so as to be applied only after each sub-stage of an incremental evolution;

11. There are two kinds of substructure databases. The first one is when the system is enabled so as to memorize each of one substructure of sizes 4, 5 and 6 genes and, each of two substructures of sizes 2 and 3 genes for the evolution to have a choice. Here - for each such sub-stage - the substructure database is built independently. The substructures of the second type involve *dedicated topological reuse* where the substructures are made of previously evolved sub-circuits;

12. The first rule of the mutation strategy is the diversification of the mutation history. The second is the mutation pressure rule and the third rule is the radical mutation. The *radical mutation rate* depends on the length of the chromosome to which it should be applied and it may vary, reaching 80% for the short length individuals. *Radical mutation* is a very important part of VNFE. It provides the essential modifications to uncommonly homogenous individuals peculiar to VNFE, especially during stuck periods;

13. It has been allowed for the VDC that the evolution use the *non-involvement type* of the *staged incremental evolution* as described in Section 5.1.1, i.e. no genes at junction and any of its neighbours from previously evolved sub-circuits that take a part in the evolutionary processes. All of the sub-chromosomes evolved during the first phase of evolution are frozen up with the assistance of X-coding.

5.4 Experiment 15: Evolution of 8-Output Voltage Distributor (Phase 2)

5.4.1 Introduction

The task description has already been presented in Section 5.2.1. However, here the general information should be given again. There is one input and eight outputs. An input signal has been taken with the same piecewise voltage pulse as in the last case and also with the case of the 4-output VDC: starting from 0V, going up to 5V for 3.5sec and down to 0V for the last 1.5sec (Figure 5-4b). The task for each output involved working in a filter-like mode which passes through the input signal that is located within the particular voltage band, saving the form of the input signal. That is, the band-pass width for each of the outputs is the same and equals 0.625V: the 1st output passes only those voltages from 0 to 0.625V, the 2nd output passes only the voltages 0.625-1.25V, the 3rd band-pass is 1.25-1.875V, the 4th is 1.875-2.5V, the 5th is 2.5-3.125V, the 6th is 3.125-3.75V, the 7th is 3.75-4.375V and finally the 8th is 4.375-5V. The summary of transient analysis at the input and eight output pins of the targeted ideal 8-output Voltage Distributor is presented by Figure 5-5. As can be seen, the graph in Figure 5-4b must exactly repeat the form of the input piecewise signal.

As before, the FF was incremented each time whenever the current task was fulfilled. Eight tasks corresponding to eight sub-circuits were set. For each task, the new FF was introduced which incrementally counted the fitness of all the sub-circuits evolved at that time. Due to the similarity of functions that the sub-circuits perform, the *dedicated topological reuse* is utilized.

The fitness function of the 8-output Voltage Distributor is:

$$F=\sum_{i=1}^{i=8} F^i$$

where F^i is a fitness value of the sub-circuit *i*, which is calculated in the following way: the PSPICE simulator performs a transient analysis at each output for five seconds at 81 equidistant time-points; a fitness value F^i is set to the sum over these 81 fitness cases of the absolute weighted deviation between the target value and the actual output value voltage produced by the circuit; the fitness penalizes the output voltage by 10 if it is not within 50% of the target voltage value. The smaller the fitness value is, the closer the circuit is to the target.

As an embryo, the previous resulting circuit has been set in Figure 5-6. The same five types of components continue to participate in the evolution and the same termination criteria for the whole circuit are set, i.e. if the fitness value did not improve over 20 consecutive generations or else if the sub-circuit reaches more than 100 components.

Five PCs have been used in parallel with Intel Core 2 Duo/2GHz processors running at the same time. The systems have been connected via a hub to each other so as to enable migrations according to the proposed scheme in Figure 5-10. Five different SR have been set to each system, starting from 1% and going to 5%. The ES with a differentiated mutation technique is utilized with a population size of 30,000 chromosomes set for each PC.

5.4.2 Experimental Results

The experiment has been running throughout five sub-stages. Figure 5-12 presents the details of the sub-stages and migrants along the evolution. After each sub-stage, the best individual migrates to every sub-system and restarts the latter.

The average time of the evolution of the 8-out Voltage Distributor was 344 hours, which is about 43 hours per sub-circuit. The best-of-run circuit (Figure 5-13) appeared at the 629th generation and had 138 components (embryo excluded), among which there were 38 resistors, 8 capacitors, 7 inductors, 46 NPN transistors and 39 PNP transistors, with a best overall fitness of 1.757 [65]. The most ideal signal, with a fitness of 0.028 was provided by out-pin No.2, which was responsible for the band 0.625V-1.25V; the

worst reply - with a fitness of 0.797 - was at out-pin No.7 in the band 3.75V-4.375V. Figure 5-14a shows the transient reply of the circuit for the incoming piecewise signal. Table 5-2 contains the details of each of the sub-stage results per sub-system. As to how human-competitive this result is - this is discussed in Section 5.4.3.

Table 5-3 highlights the detailed information per incremental sub-stage from the start (Phase 1): the best fitness, the component number of the evolved sub-circuit and the successful generation number.

Table 5-2. The details of Experiment 15 (2nd phase). The best values are in bold.

PC No.	SR, %	4th sub-stage			5th sub-stage			6th sub-stage		
		Fitness	Gene No.	Gene No.	Fitness	Gene No.	Gene No.	Fitness	Gene No.	Gene No.
1	1	**0.323**	**37**	**23**	0.092	33	17	0.980	49	43
2	2	0.301	62	60	**0.049**	**26**	**14**	0.197	51	28
3	3	0.480	54	72	0.291	20	33	1.403	27	22
4	4	1.422	35	39	2.027	73	62	**0.200**	**107**	**23**
5	5	1.239	58	91	1.362	62	41	3.001	47	69

PC No.	SR, %	7th sub-stage			8th sub-stage		
		Fitness	Gene No.	Gene No.	Fitness	Gene No.	Gene No.
1	1	**0.797**	**104**	**22**	**0.089**	**37**	**8**
2	2	2.917	59	36	0.481	46	20
3	3	1.092	43	29	0.93	67	22
4	4	1.548	78	37	1.032	53	35
5	5	3.488	59	55	1.431	71	43

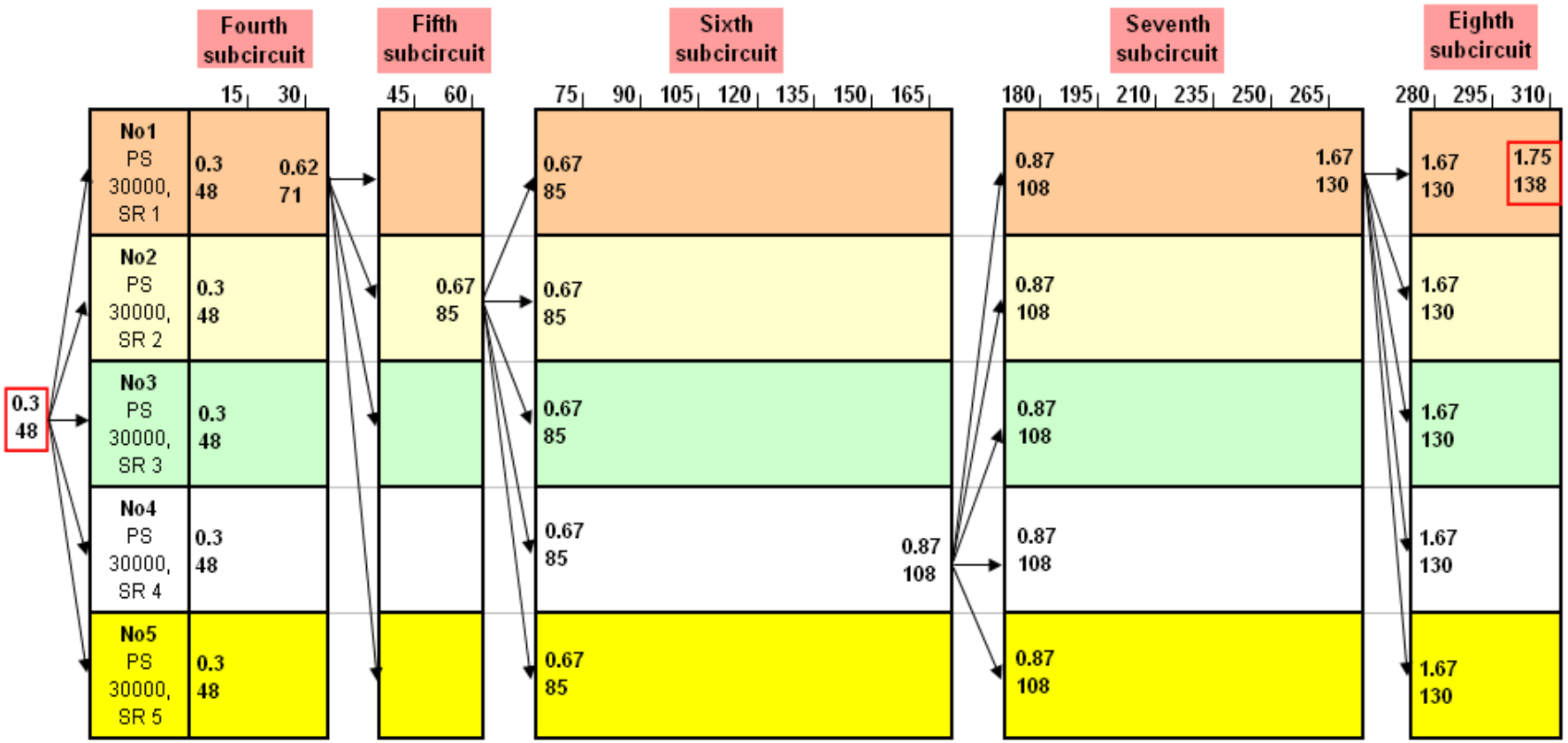


Figure 5-12. The migrant schedule for Experiment 15: Evolution of 8-Output Voltage Distributor (Phase 2). The diagram shows when and how the migrant takes place along a horizontal axis representing generation numbers. Five sub-systems with different SRs evolve in parallel from left to right. The arrows indicate which sub-system is a receiver, from where and at which generation. Each migrant is described by the fitness value of the migrant individual and its length in genes. The initial and final chromosome attributes are in red boxes.

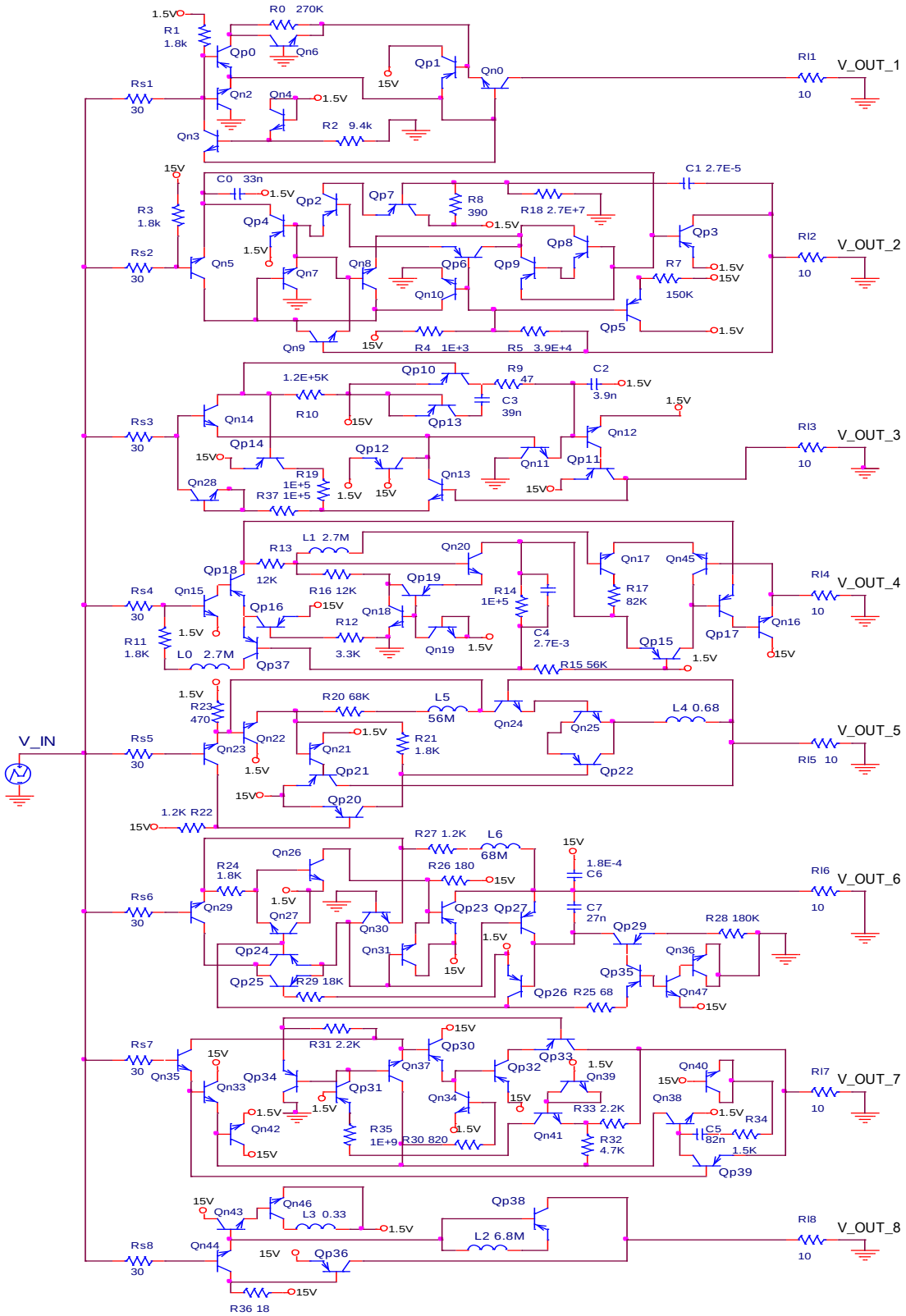


Figure 5-13. The evolved 138-component 8-output VDC in Experiment 15.

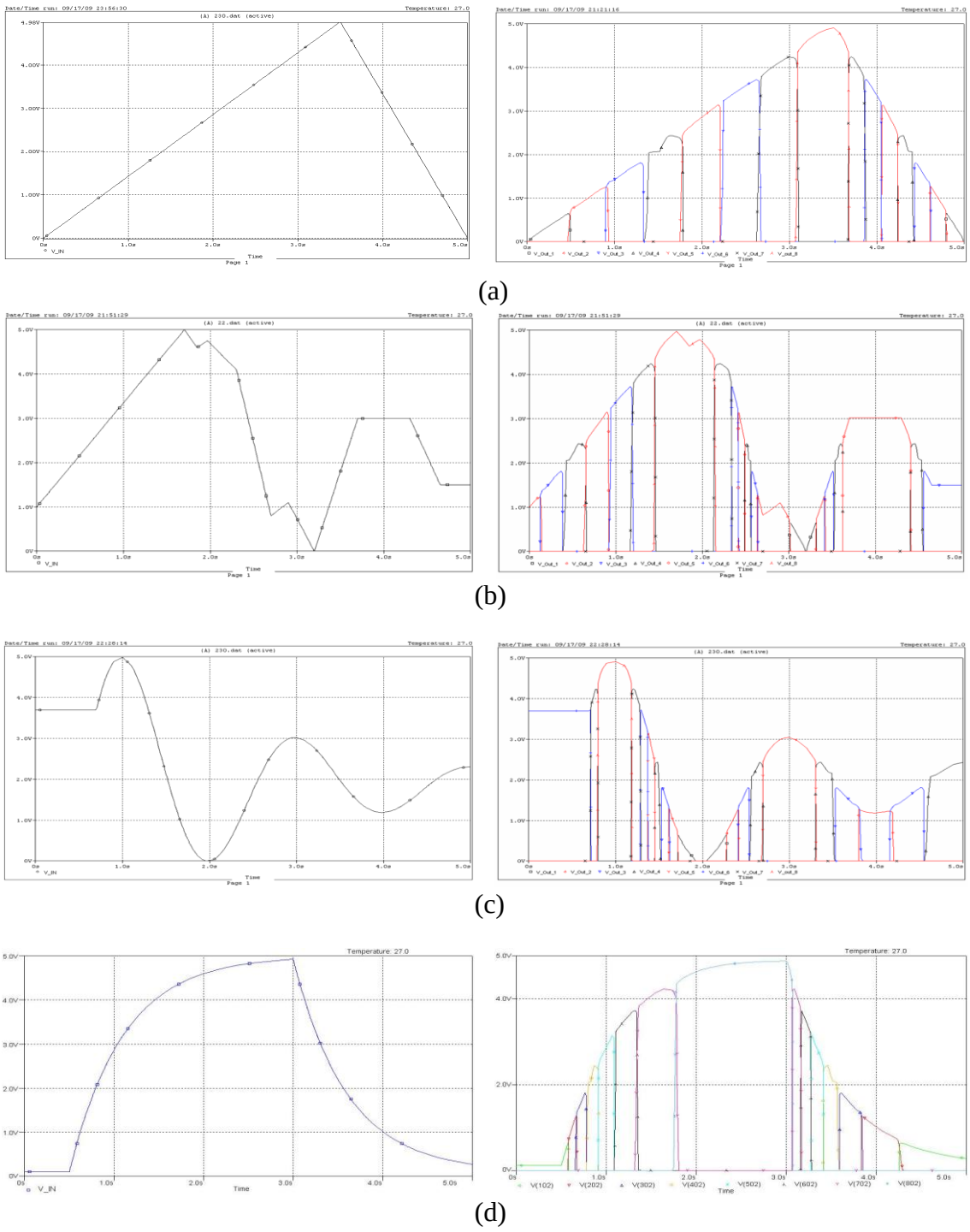


Figure 5-14. The transient analysis of the input and outputs of the 8-output Voltage Distributor. (a) A piecewise signal used during the evolution and the circuit's response. (b) The incoming arbitrary piecewise signal and the circuit's response. (c) The response to an arbitrary piecewise signal and the circuit's response. (d) The incoming arbitrary exponential signal.

To verify that the problem of generalization is overcome, different arbitrary signals are applied to the resulting 8-output Voltage Distributor. Figure 5-14b shows the more complicated piecewise signal and the transient reply at the outputs. Figure 5-14c and Figure 5-14d show the arbitrary sinusoidal and exponential signals applied and the transient replies of the circuit.

Table 5-3. Summary for the sub-circuits of the 8-output Voltage Distributor

	Fitness	**Component No.**		**Generation succeed**
		Before pruning	**After pruning**	
Sub-circuit 1	0.095	12	10	76
Sub-circuit 2	0.028	25	22	132
Sub-circuit 3	0.174	18	16	110
Sub-circuit 4	0.323	27	23	37
Sub-circuit 5	0.049	15	14	26
Sub-circuit 6	0.200	24	23	107
Sub-circuit 7	0.797	26	22	104
Sub-circuit 8	0.089	11	8	37
Total	**1.757**	**158**	**138**	**629**

As can be visually seen, each particular sub-circuit provides accurate replies throughout the different examples, which enables us to assume that the relative fitness of each of the sub-circuits as well as the whole circuit - least of all - depends on the characteristics of the incoming signal.

5.4.3 Discussion

In this experiment is described the application of the ES-based analogue circuit synthesis system with the differentiated mutation and incremental evolution techniques towards the design of an analogue multi-output circuit 8-output Voltage Distributor. The first phase of one experiment has failed during the fourth sub-stage. However, after the application of the parallel island-model evolution technique during phase 2, a novel methodology was able to succeed with the target and to finally evolve the analogue

circuit with 138 components. To the author's knowledge, this is the largest analogue circuit in terms of component number in the area of automatic analogue circuit synthesis.

It is obvious that this circuit is very special by its modular structure, but this last fact does not depreciate its high complexity. The human designer with substantial practical experience in the design of analogue and digital circuits has been attempting to design the 8-output VDC. Considering the task, the designer draws the conclusion that it is possible to design this circuit purely with analogue components (shown on Figure 5-15), but it may take an unduly significant amount of time and effort. The voltage controlled oscillator (VCO) modulates the incoming voltage signal through frequency. The modulated signal comes in the bandpass filters (BPF), each of which is tuned to its own pass band. Particular signals which passed through the BPFs then are demodulated by analogue demodulators. The drawback of such a purely analogue circuit is that each path starting from the BPF input up until the circuit output is independent of the other. This makes the signals at all N outputs asynchronous. This last fact may bring problems if someone further utilizes signals from outputs, for example in trying to recreate the original signal. Therefore, synchronization is required at the circuit outputs, which could be set as digital or as analogue. While the first one requires the introduction of additional digital devices and comes at the cost of increased complexity in timing analysis, the last one requires cumbersome transformers. In any case, even without synchronization, the circuit requires a much higher number of components than 138. As a final idea concerning the human design of analogue circuits, the "the analogue dilemma" from [140] should be mentioned: "Analogue circuit design… usually stretches over a significant period of time and is performed by designers with a large portfolio of skills. It is therefore considered by many to be a form of art rather than a science."

Considering the technique, one of the features which make this approach unique is the single-chromosome migration which has been allowed to happen during parallel evolution. That is, only a single best chromosome is defined as a start up embryo for every system that had been failed after each sub-stage. This idea is adopted due to the multitude of experiments that have proven its feasibility. This is only because the

conventional knowledge on optimal SR never suggests the usage of such small values as have appeared during Phase 2.

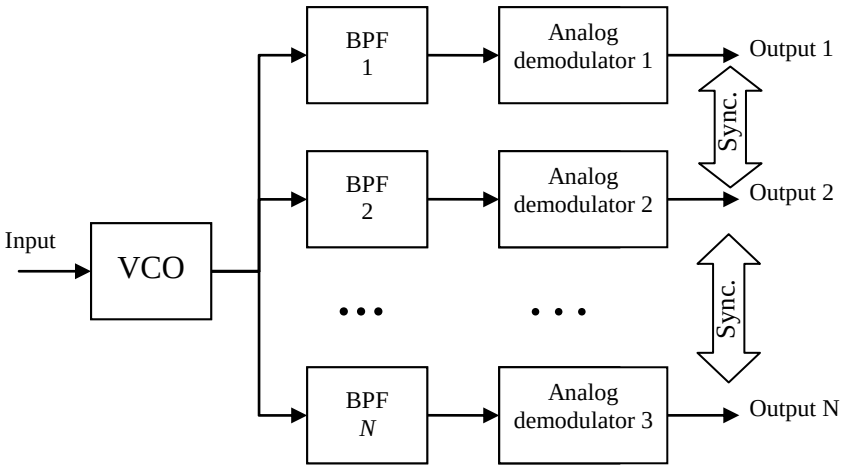


Figure 5-15. The human designed N-output VDC with synchronization.

During Experiment 15, different SRs are used at 5 different sub-systems, with the most successful sub-system being No.1, with a minimum SR=1% which has produced two of the five best individuals, including the final one. It is notable that the low SR sub-systems dominated by producing the best individuals; however, the making of any stronger propositions would require more statistical data.

If we regard all the subpopulations at the sub-systems as one large population and follow the evolution during Phase 2, it will be possible define the aggregated SR of the system. By doing this, it is possible to follow the idea of VNFE, where differentiated mutation requires smaller SRs, and clarifies which SRs have brought about any success. Table 5-4 represents the aggregated SR of the system at the beginning of each sub-stage.

Table 5-4. The aggregated SR of the system. The SRs are given for the beginning of each sub-stage at each population size of 30,000.

	Sub-system 1, SR=1%	**Sub-system 2, SR=1%**	**Sub-system 3, SR=1%**	**Sub-system 4, SR=1%**	**Sub-system 5, SR=1%**	**Aggregated**

4th sub-stage	**Chrom.selected, No.**	1	1	1	1	1	5
	SR, %	0.0033	0.0033	0.0033	0.0033	0.0033	0.0033
5th sub-stage	**Chrom.selected, No.**	300	1	1	1	1	304
	SR, %	1	0.0033	0.0033	0.0033	0.0033	0,203
6th sub-stage	**Chrom.selected, No.**	1	600	1	1	1	604
	SR, %	0.0033	2	0.0033	0.0033	0.0033	0,403
7th sub-stage	**Chrom.selected, No.**	1	1	1	1200	1	1204
	SR, %	0.0033	0.0033	0.0033	4	0.0033	0,803
8th sub-stage	**Chrom.selected, No.**	300	1	1	1	1	304
	SR, %	1	0.0033	0.0033	0.0033	0.0033	0,203

As can be seen from Table 5-4, the minimum SR is at the beginning of the initial 4th sub-stage. In the middle of each sub-stage, 4500 individuals are selected aggregately at every generation at all the sub-systems, which gives the aggregated SR of 3%. Next, at the incremental moments between the sub-stages, the SR again falls down.

The observed behaviour of the SRs makes a contribution towards an understanding of the WDWC strategy that will be described in detail since Section 5.6.5.2.

One of the targets of this section was to confirm the potential strength of the developed technique in the synthesis of complex unconventional-application analogue circuits. It also has been confirmed that the analogue circuits synthesized by the proposed methodology are able to take the place of digital ones in the solution of complex tasks.

5.5 Parallel evolution based on the WDWC strategy

During Experiment 15, it was noticed that the methodology of parallel evolution may be improved significantly. Some sub-systems in the previous experiment had running idle during the 20 stalling generations required by the termination criteria for the sub-stage termination. Another disadvantage of the previous method is that the systems have to wait while the last sub-system finishes all the work. The main idea that

lies under the WDWC strategy is to make use of the migrations more often, using them as a main tool against the stalling effect during each sub-stage. The migrations since that point are regarded as those following from *radical mutation* in the DM technique, proposed so as to stimulate the evolutionary search at local optimums. In Section 5.5.1, the algorithm and the proposed strategy are presented. At the same time, the general structure of the previous methodology will be preserved.

5.5.1 Migrant strategy

The populations are connected together by a migration operation, which is performed by communication between processors. There is no centralized ("master-slave" mode) schedule set for the communication frequency or the magnitude of migration. On the contrary, each evolution "decides itself," i.e. when to start migration and what it needs for that migration. There is only one condition set, namely when a communication among the parallel sub-systems can take place: each evolution is allowed to run without communication as long as the best chromosome improves (for the term of the best chromosome improvement see Section 3.6. As soon as any population does not improve for at least N (in the following experiment $N=7$) generations, the built-in migration operator activates and makes the sub-system search for help from the other sub-systems. First of all, the stagnated sub-system collects the data-files from all the parallel nodes, analyzes them and decides what the most successful evolution among all until now is. It applies the ranking rules - including formula [4-1] - to rank out the best chromosome among the latest of each evolution. As such, the sub-system gets a ranking list consisting of six members, a top-member of which becomes a "winner" chromosome from a "winner" sub-system. Next, the sub-system gets a clone of the "winner" with all its history, checks the substructures (see the paragraph below), and updates its SR and population size. It clones a single individual to the total population and continues with its further isolated evolution procedures. Here, our approach differs from that of the others. Usually the migrant strategy implies the highest-ranking individual to replace the lowest-ranking individual without dumping out the rest of the genotype material.

The advanced feature of the migrant strategy so described is the ability of the whole system to

adapt the activity of the migrant operator, varying it from 0-power - when no one sub-system gets in the local optimum during evolution - until the full *n*-power regime (*n* being the number of parallel sub-systems), whereby every processor gets a *N*-generation stuck period where within *N* generations *n* migrations are happening.

In the previous paragraph, a general migrant operation is described which is liable during all of the evolution. However, there is a migration of the substructures. The substructures are limited to six genes, as has been described in Section 4.1. When a stagnated sub-system analyses the data-files, it makes two independent rankings for the individuals with lengths of five or six genes. The best are decomposed and stored in a substructure database. The sub-systems that donate the best chromosomes may differ. The procedure of substructure-checks is performed every time before the start of a new generation.

This last statement together with the migration approach enables us to draw the general view of a novel parallel evolution in Figure 5-16.

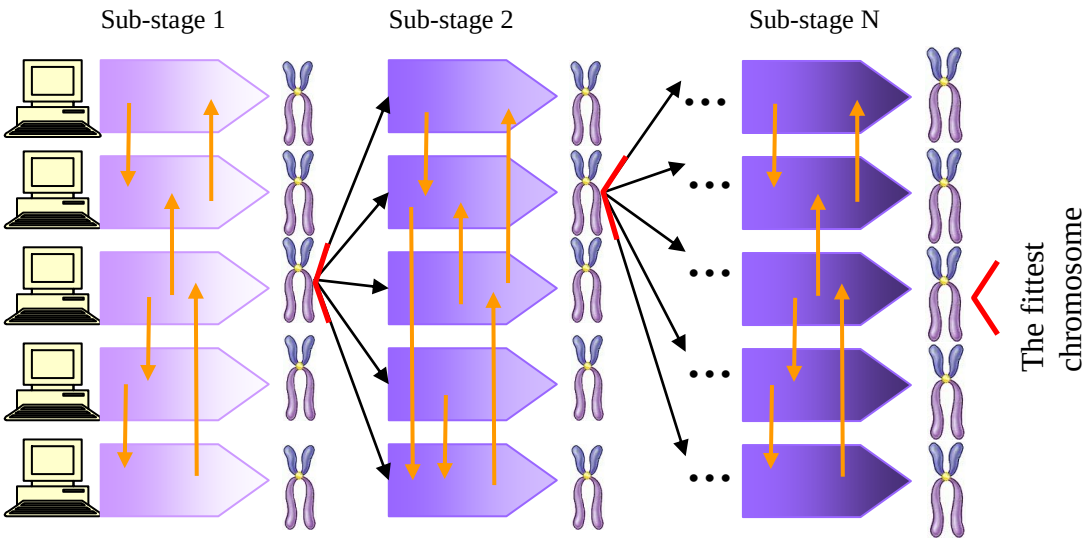


Figure 5-16. Utilization of parallel sub-systems in the WDWC strategy. There are two kinds of migrations: the black arrows show the ones when the best individuals are migrating after every sub-stage and the yellow arrows indicate the migrants inside the sub-stages.

5.5.2 Parallel evolution

In general, the proposed approach is very similar to a competitive co-evolution approach, where as long as some evolution produces "winners" more of these "winners" become parents in other parallel evolutions. Notice that while migrating to a new sub-system the best chromosome destroys all of the previous genotype material of the sub-system. Despite the different initial conditions set for separate evolutions - due to migration - it may be that the descendants of one ancestor compete with each other. Furthermore, the approach theoretically allows that all seven sub-systems during our experiment (i.e. in total about 169,000 individuals) evolve the same chromosome at the same time. Together with the extremely narrow SR (0.2-2.0%), it seems quite contradictory to the conventional opinion in EA theory which hails the diversity of genotypes. The literature review provides an explanation of why this happens.

There should be a balance in evolution between exploration and exploitation. If the level of exploration is too high, the search may quickly become random; however, if the degree of exploitation is too high, it may result in premature convergence [153], [156], [154], [155]. In other words, the diversity created by mutation must be paired off with the reduction performed by the selection operator. The reduced SR has been indirectly used previously by Altenberg [158] and Soule et al. [159]. They tried to use a method whereby offspring are only kept if they improve upon their parents' fitness against the growth of inoperative code through "neutral" crossover events. By doing this they applied the higher mutation rate towards the parent population at the increased selection pressure. In [156], Smith et al. used a similar approach called Improved Fitness Selection (IFS). To test whether IFS suffers from too low a selection rate, they have shown how "IFS with high selection pressure gains an extended life to the evolutionary search process by avoiding the stagnation that is inevitable with the standard method over the course of many generations, and thus managed to achieve a very high success rate on a problem that is normally intractable to standard GP even with large population sizes." Thus, the DM technique described in Section 4.5 - with its adaptive mutation rate becoming higher with each unfruitful generation - is balanced by VNFE. Experiment 16 has proved that the WDWC technique has found the trade-off between exploitation and exploration; moreover, it appears to be the only way to find a solution.

Thus, the competition aspect consists of the constant threat of the destruction of the whole population when a sub-system is failed. Even a sub-system that imported in the past the genotypes from a "winner" sub-system is regarded as a competitor to the "winner." Due to the radical increase of the mutation rate by the DM strategy, these two become different much faster than was the case during a conventional evolution with a standard mutation rate.

The competition aspect also plays an important role when the termination criteria appear during the first sub-stage: the most competitive sub-system at that moment becomes a "winner" and only "it" is allowed to save and transfer its population into the next sub-stage. All the others must obey a migrant operation.

On the other hand, the described competitive approach has some features of cooperation. The most obvious one is when a "winner" shares the successful SR and genotypes with the sub-systems that lost their populations' individuals (hereinafter called "losers"). A "winner" does not allow a stagnated sub-system to stop. It shares with that last one the best that it has, in spite of the fact that later the "looser" may become its competitor. Another aspect that makes all the sub-systems become cooperative is a substructure database that may consist of genotypes (in total seven substructures) from different evolutions and is accessible by everyone. Moreover, all the sub-systems focus their work towards the same target and have the same FF. This parallel evolution approach is called a *Winner-Dominates-Winner-Cooperates* (WDWC) strategy.

In the following section, the proposed technique is demonstrated as when applied to the most sophisticated task in this thesis.

5.6 Experiment 16: Evolution of the Time Interval Meter Circuit

5.6.1 The problem's description: TIMC for the laser rangefinder

The TIMC is proposed to function in a single analogue mode instead of the number of digital operations inside the up-to-date laser rangefinder DAQ-2 [118], where the time interval metering function is performed by several digital circuits when the rangefinder uses a laser beam to determine the distance to an object. The targeted analogue TIMC belongs to a class of devices that are known as "time to amplitude converters" (TAC): "TAC generates a rectangular output pulse whose peak amplitude is linearly proportional to the time interval between a START and STOP input pulse pair" [161]. To the author's knowledge, this is the first attempt towards the automatic synthesis of a TAC circuit.

A laser rangefinder is a device which uses a laser beam to determine the distance of an object. The most common laser rangefinder operates on the time of flight principle by sending a laser pulse in a narrow beam towards an object and measuring the time taken by the pulse to be reflected off the target and returned to the sender. The distance is given by:

$$S = \frac{cT}{2},$$

where ***c*** is the speed of light and ***T*** is the amount of time for the round-trip between the device and the target. The typical laser rangefinder has two main parts: one optical and the other electrical. The optical block sends the laser beam and receives the reflection, providing the electrical block with two voltage pulses, based on which of the electrical blocks calculates the distance.

As a prototype, we take the artillery quantum rangefinder "DAQ-2" [118] with the following data:

- Working at range is 0.2÷100km;
- Measurement accuracy is 6÷30m;
- The width of both pulses is 50ns; the fall/rise time of the pulse is up to 5ns; the first pulse has a 9V amplitude while the reflected one has 6V;
- Power supply required is 29V.

The core part of the electrical block of the device is a time interval meter sub-block (TIMSB). The working principle of a conventional TIMSB consists of three functional stages:

1) At the first stage two electrical pulses received from an optical block should be reshaped into the voltage gate pulse, where the first incoming pulse is caused by the laser beam sent towards a target and the second one is caused by the beam reflected off the target. The gate pulse is a pulse of some constant potential that should have the same time-width as the interval between two narrow pulses caused by a laser beam;

2) At the second stage, the gate pulse (i.e. the time interval of the gate pulse) is filled up by the clock signals from a crystal oscillator. According to (1), the gate pulse width varies from about 0.667us for the minimum measured distance of 0.1km to 0.667ms for the maximum measured distance of 100km;

3) Finally, to count the number of pulses contained in the packet, the result of counting in binary code should be sent to a decoder for further conversion into a decimal code.

In Figure 5-17 is a general schematic of the TIMSB of the up-to-date laser rangefinder. Based on a description available to the public, the goal is set to synthesize the analogue circuit which is able by its functioning to unite stages 1), 2) and 3) described above, and replace the five digital units from Figure 5-17. A new circuit receives two pulses from an optical block and produces the particular constant voltage.

The linear correlation between the time gap and the voltage produced is set, ranging between the maximum of 5V (against the maximum 100km) and 5mV (for a distance of 0.1km). The proposed TIMSB based on an analogue circuit is shown in Figure 5-18a. The decomposition method is shown on Figure 5-18b, where it is suggested that two sub-circuits should be evolved during two incremental sub-stages. The decomposition into 2 parts is made by simply splitting Figure 5-17 in the middle and it did not require any special knowledge of the device. An attempt is made to evolve both the whole device and its decomposed variant. The results are presented in Section 5.6.5.

The ideal circuit response of the evolved circuit is shown on Figure 5-19.

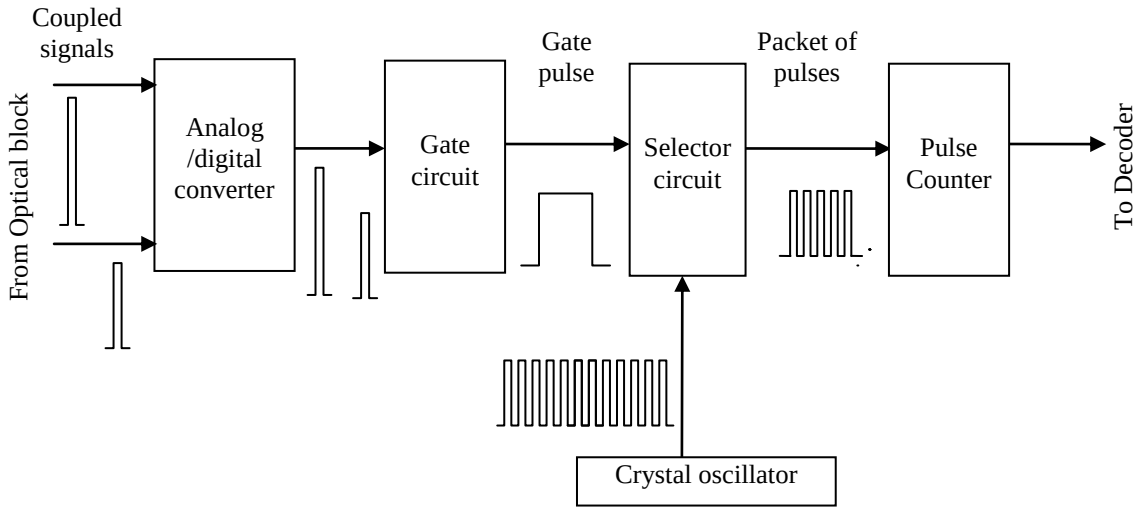


Figure 5-17. The TIMSB of an up-to-date laser rangefinder made of digital logic. The shapes of the signals are shown under each pin. From left to right: there are two pulses coming in from an optical block - 9V and 6V - separated by the time taken for the beam to be reflected and returned; they are converted into digital form by ADC. Next, they are transformed into a gate pulse by *gate circuit*; a *selector circuit* fills up the gate with clock pulses generated by a *crystal oscillator*; a *pulse counter circuit* gets the packet of pulses and counts the clock pulses; a *decoder* converts that count into decimal form.

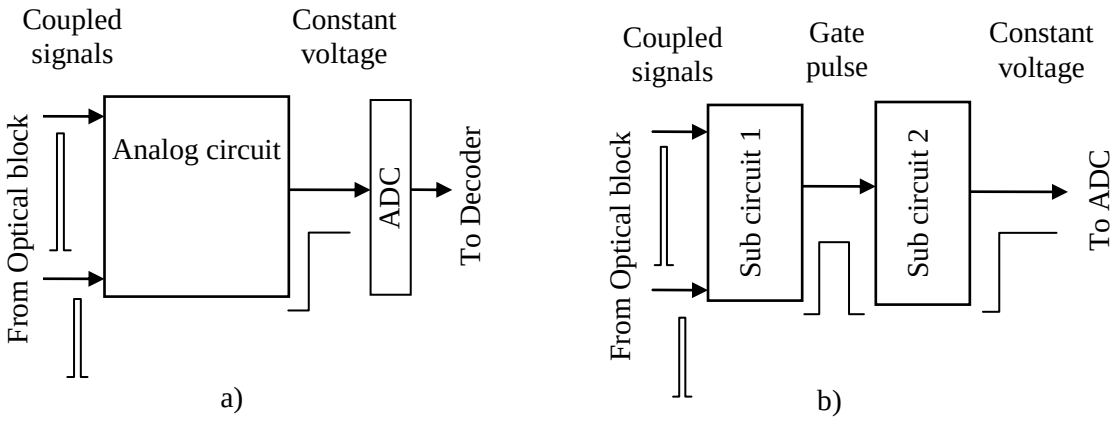


Figure 5-18. a) The proposed TIMSB with the targeted analogue circuit. The shapes of the signals are shown under each pin. From left to right: two pulses are converted into a constant voltage; the voltage level is in linear proportion to the time interval between the two pulses; the ADC converts the voltage into the binary code for further decoding. Due the preference that the resolution of the circuit should be at least 50uV (corresponding to 1 meter) - i.e. in total 1e+5 discrete values - the 18-bit ADC with 262144 quantization levels will meet the requirement. b) The proposed decomposition of the targeted analogue circuit. The first sub-circuit's task is to form the gate pulse based on a coupled signal. The aim of the second is to produce a constant voltage.

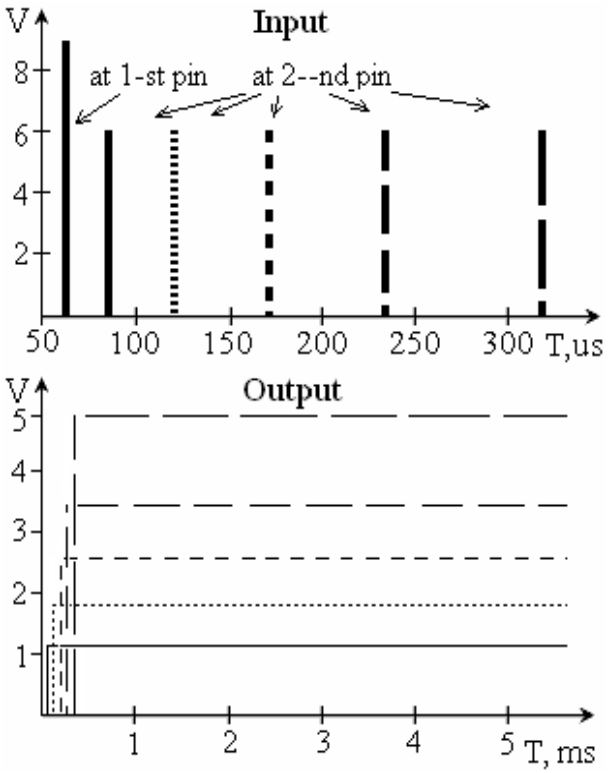


Figure 5-19. The top graph shows 2 pulses at 2 input pins of the TIMC (both are of 50ns width): The 1st is of 9V at 60us, while for the 2nd we took 5 arbitrary pulses at 85, 120, 170, 230 and 333.4ns. These coupled signals correspond to the distances to the target of 25, 36, 51, 69 and 100km. The bottom graph shows 5 transient replies at the output pin of an ideal TIMC.

5.6.2 Adaptive parameters

Besides the mutation, another three parameters are allowed for each evolution to adapt. Different initial populations, SRs and *pressure-constants* are set.

The strategy for population size adaptation is to maintain equal times per generation of different populations because the difference is as large the deeper the process. If, initially, the weaker processors keep pace with the others, later on – and starting from the average chromosome length of 10-15 genes - they need their cycle periods to be reduced. To enable this synchronization, the evolving period of the first population (at the 1st PC) is set as a reference for the others. Thus, during the migrant operator activation, each sub-system - except for the 1st one - adjusts its population size P to a

new one P_1 in order to keep pace with the population of the first processor according to the following simple formula:

$P_1 = t_1 \times P / t$, where t_1 and t are generation times at the 1st sub-system and the synchronizing sub-system.

Seven SRs from 0.2% to 2% (Table 5-5) are allowed to migrate from a "winner" evolution to a "loser" evolution along with a "winner's" genotype and its history. This enables statistics to be accumulated at the end of an experiment as to which SR is becoming the most frequent "winner."

Table 5-5. Initial conditions at 7 parallel PCs

No.	PC description	Initial pop. size*, individ.	Initial selection rate, %
1	Intel Core2Quad, 2.4Ghz, 4GB	35,000	0.2
2	Intel Core2Quad, 2.4Ghz, 4GB	35,000	0.5
3	Intel Core2Duo, 2.2Ghz, 2GB	25,000	0.8
4	Intel Core2Duo, 2.2Ghz, 2GB	25,000	1.1
5	Pentium4, 2.8Ghz, 2.0GB	18,000	1.4
6	Pentium4, 2.5Ghz, 1.0GB	16,000	1.7
7	Pentium4, 2.8Ghz, 0.5MB	15,000	2.0

* The population sizes are chosen in accordance with the operational powers of each PC, so that the times to be taken by the initial generations are approximately equal.

The *pressure-constant* is a "range-dependant" [150] deterministic dynamic parameter that differs from other adaptive parameters in that it does not depend directly on the parallel evolutions, but it is adapted by the local evolution. At each sub-system, the *pressure-constant* is set to an initial value of 40. However, if two generations following each other do not find better individuals, this number is increased by two, which causes the component-reducing pressure to decrease. Ten generations without the best chromosome update make the pressure constant equal to 60, etc. The limit of 64 is set if the stagnation period lasts for more than 12 generations, i.e. the ranking pressure

can be decreased by 60%. Conversely, if two successive generations have brought a fitness improvement, the *pressure-constant* is reduced by two. This strategy leads to an inevitable gradual weakness of the selective pressure due to the permanent growth of a chromosome's length, complexity and solution space.

5.6.3 Fitness function

A fitness function similar to one used in [71] is scheduled and is calculated by the following static fitness function set to a sum over *p* fitness cases of the absolute weighted deviation between the target value V^i_{ideal} and the actual output value voltage produced by the circuit $V^i_{measured}$:

$$F = \sum_{i=0}^{p} | V^i_{ideal} - V^i_{measured} | \qquad [5\text{-}1]$$

p equals 11 time-points for TIMC. The smaller the fitness value is, the closer the circuit is to the target. The fitness penalizes the output voltage by 10 if it is not within a specified percentage range of the target voltage value. For TIMC, where the output from the circuit is supposed to be a constant voltage, all 11 measured points are equidistant within the range from 1ms to 10ms (which is quite a long period of time for the ADC to catch up with the signal for further coding).

The fitness threshold is set to 0.3%, i.e. the evolution ranks the fitness of a new chromosome as better than the current one if the relative fitness difference between the best chromosome and the one under consideration is more than 0.3%. This barrier enables pressure to be applied during selection which stimulates the application of more radical mutations (see next section).

The problem of generalization was met during the experiment, which should be introduced here since it influences the FF. The problem of generalization appears when the validity of the circuit functioning is limited only by a case of source signals used during evolution, and it is not extending to arbitrary signals (see Section 2.5.1). Seven cases of coupled signals are suggested, corresponding to distances of 0.4, 2, 10, 30, 45,

65 and 95 km so as to tackle the problem of generalization. This number means that every chromosome in a population at each generation is tested seven times for seven different incoming signals, and the final fitness value for that particular chromosome is created by a simple sum of seven normalized fitness values: $F = \sum_{j=1}^{j=7} w_j F^j$, where F^j is defined by expression [5-1] and w_j is a weight that normalizes the contribution of each case. For the distances mentioned, the weights are: 237.5, 47.5, 9.5, 3.167, 2.11, 1.462 and 1 [5-1].

5.6.4 Termination criteria

First of all, it should be noted that the whole system terminates only if every sub-system terminates. The termination criteria should be distinguished for a first sub-stage and that for a second one which is a complete experiment. There are two events that may happen that could most probably cause the evolution to be terminated. The first one is the reaching of a goal in a form exceeding the preset fitness threshold. If this happens during the initial sub-stage, two main events are automatically triggered: the best chromosome that attained the threshold is X-coded (Section 5.1.2) and all other parallel sub-systems are forced to stop searches and activate a migrant operator which is to receive that individual as a migrant with all its data, initiate a new population as well as all of the other standard procedures after migration. If the same happens by the end of the second sub-stage, this means the end of the whole experiment. For both cases, the threshold is set as reaching a fitness of less than 1.0, which is equivalent to an average deviation from the ideal reply function per point of 0.031V for the first sub-circuit and 0.044V for the whole circuit.

The second termination criterion is when a sub-system is not able to update the best chromosome for over 15 consecutive generations. During 15 generations, every evolution - in the case of its local optimum - will get two migrants (one per seven generations, see Section 5.5.1), and if the migrants are worse or equal to the sub-system's best individual (i.e. all the other sub-systems did not improve as well) then the experiment stops. However, if some sub-system finds a better solution it will be able to

revive the others. In the last case, the sub-systems that stopped the search can join the process under the same conditions, as if they had just activated the migrant operator.

If the first term terminates all the sub-systems simultaneously, the second one acts independently for each sub-system.

5.6.5 Experimental Results

5.6.5.1 The circuit

Before making a decision on decomposition of TIMC into 2 sub-targets, an attempt has been made to try to evolve TIMC without incremental evolution. At the beginning of this section, the results of 5 runs of that experiment are presented. The same experimental settings as described in Sections 5.6.2-5.6.4 are applied, including the FF and the embryo. All 5 experiments are shown on Figure 5-20.

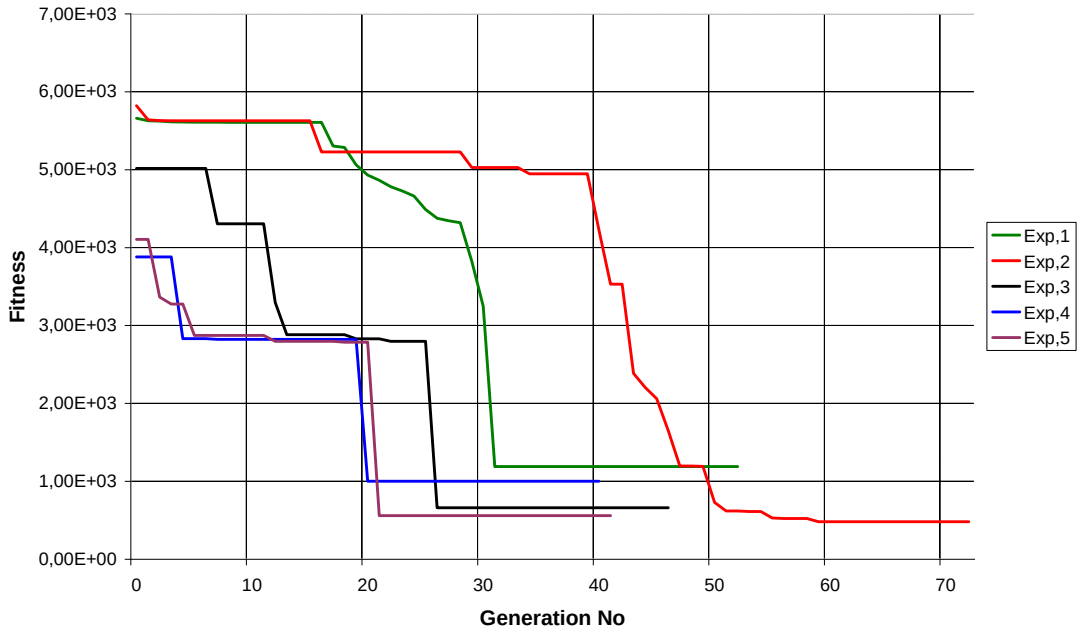


Figure 5-20. The results of five runs of TIMC without decomposition.

As could be seen from Figure 5-20, the reaching of the 15-generation-limit termination criteria (i.e. the stalling effect problem) was the reason for experiment

failure. There is a visible improvement in fitness for all cases from about 4000÷6000 to 500÷1000. The analysis of the best circuit demonstrated that the circuit gives a constant voltage that weakly responded to the varying input signals. The evolution just found the optimal constant voltage that fitted to all seven input cases. Therefore, incremental evolution has been applied to this task for further experiments.

Thus, the problem has been decomposed into two subtasks (Figure 5-18b). The initial sub-stage is the evolution of a two-input-one-output gate pulse producing a sub-circuit and the next one is the evolution of the one-input-one-output sub-circuit, which is in series with the first one.

The experiment ran non-stop throughout all the sub-stages. To design the whole circuit took about one week, where 17% of the time was spent on the first sub-circuit and the remaining 83% for the second one. The discussion on competitive features of the resulting circuit in relation to human design is presented in Section 5.6.7.

The first sub-circuit with two inputs and one output, with a primary task of providing a gate pulse, consisted of 34 components. Before the next evolution began, the fitness of the best first sub-circuit was 0.906. The pruning procedure eliminated three components that have no influence on the circuit's behaviour. The second sub-circuit, with a task to accept a gate pulse and produce the required constant voltage, consisted of 61 elements. Five components were pruned. The final design consisted of 95 components before pruning and 87 components after pruning, among which are: 29 resistors, 26 p-n-p transistors, 17 n-p-n transistors and 15 capacitors. The second termination criteria stopped an experiment with a summary fitness of 1.137 at generation No.105 [146].

One run was made. The resulting device is presented by Figure 5-21, and its functionality is shown in six arbitrary instances by Figure 5-22a. One of the additional features of the circuit is the lower voltage supply required by the circuit in comparison with the "DAQ-2": 15V against 29V.

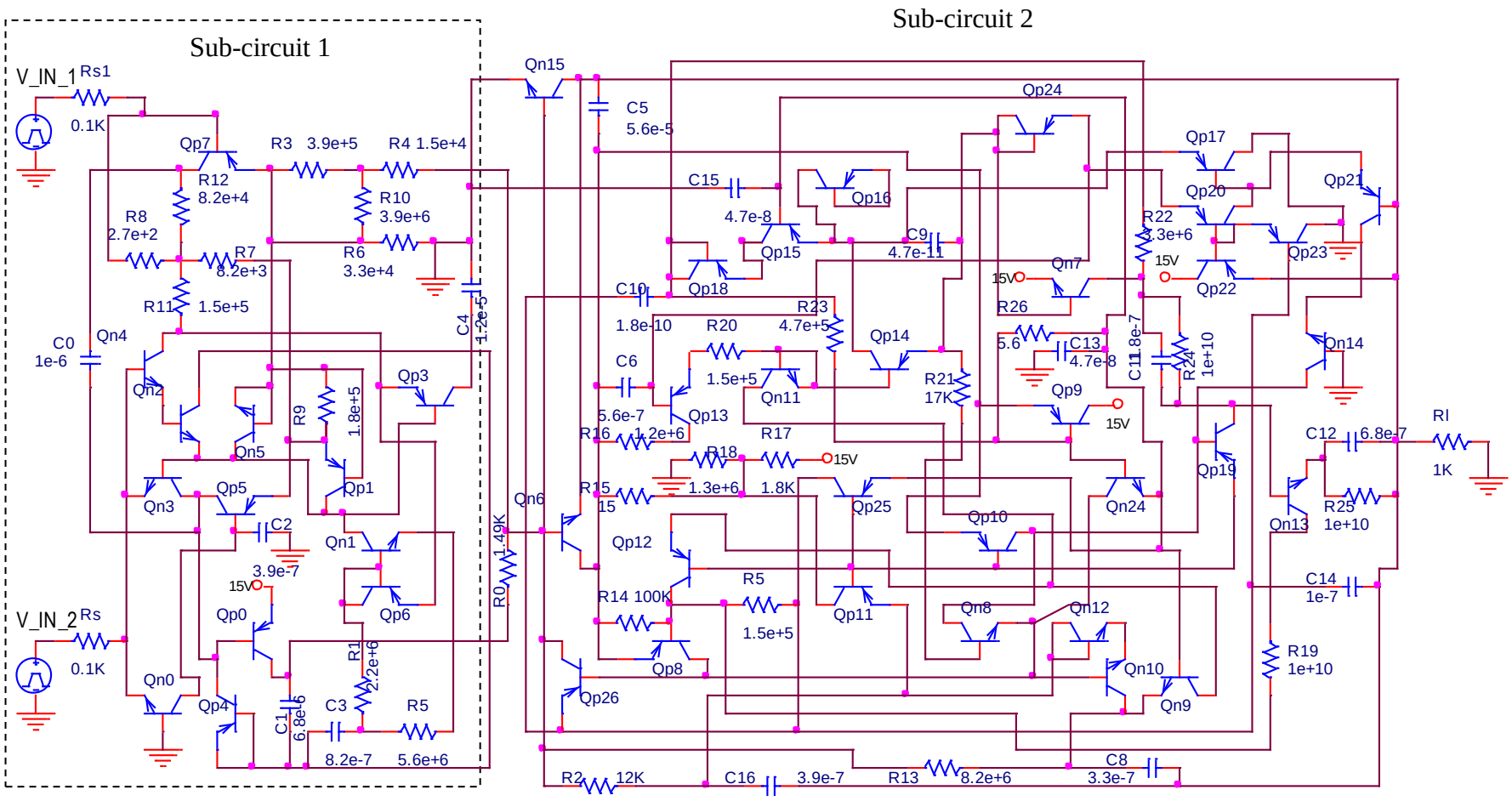
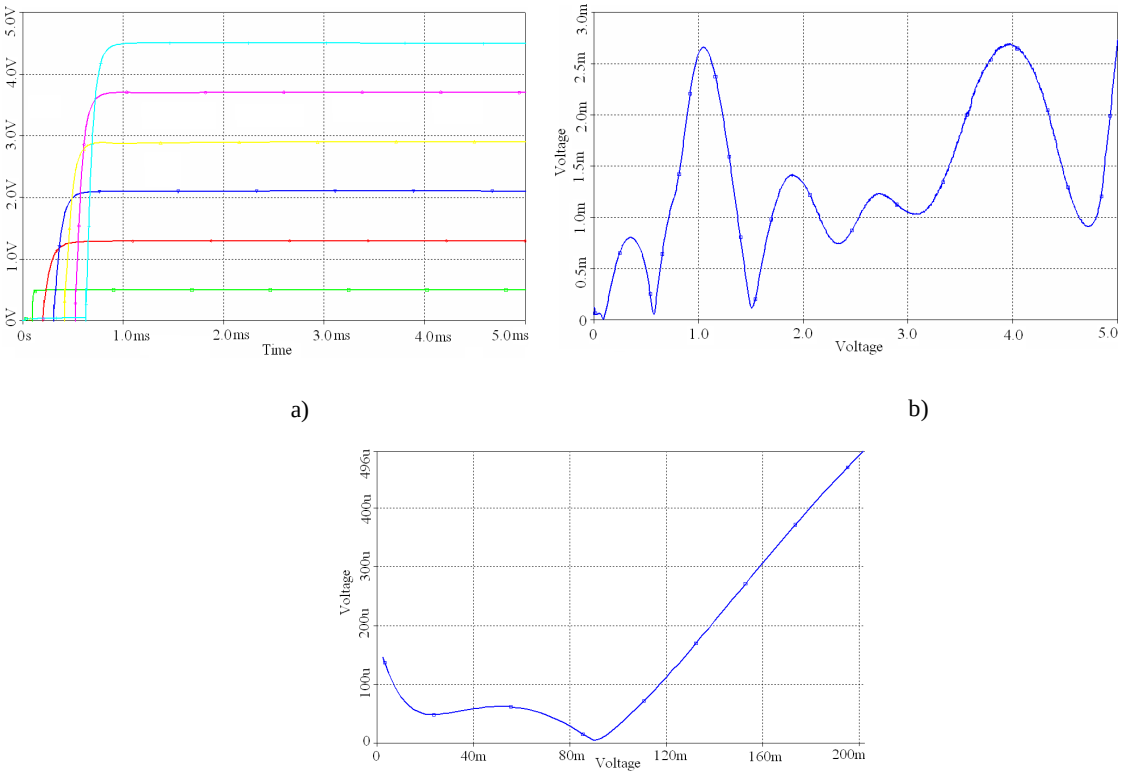


Figure 5-21. The evolved TIMC in Experiment 16 consisted of 2 sub-circuits: the first sub-circuit (dashed) passes the gate pulse, while the second one produces the required voltage.

PSPICE's performance analysis enables us to measure the generalization ability of the circuit by tracing the dependence of circuit replies on a swept parameter. If we take as a swept parameter the absolute average deviation from the ideal circuit response and apply it to a family of waveforms, PSPICE produces a trace that is a function of the variable that changed within the family. As it can be seen in Figures 5-22b and 5-22c, this represents the absolute average deviation along 2000 equidistant circuit replies, and the measurement accuracy of TIMC could be approximately split into three groups: 3 meters for distance range of 0.1÷2.5m, 16 meters for 3÷15km and 54 meters for 15÷100km. In comparison with a conventional digital TIMC, where the measurement accuracy varies within the range of 6÷30m it should be noted that for shorter distances the analogue TIMC makes much more accurate measurements. Furthermore, the resulting device is able to work out measurements within distances of 0.1 to 0.2 km, for which the DAQ-2 cannot.



a) b)

c)

Figure 5-22. a) The voltage replies of the evolved TIMC to six arbitrary incoming signals corresponding to 10, 26, 42, 58 74 and 90km. b) The function of the integrated absolute average deviation from the ideal circuit response along 2000 equidistant circuit replies. c) The same as in (b) but the fragment from 0 to 0.2V.

Moreover, during the solution of the generalization problem, it was noticed that there was a tendency for the accuracy of the measurements to directly depend on the number of input cases during evolution. Thus, it is logical to conclude that reaching the same accuracy (30m) for longer distances and even exceeding it directly depends on the computing time.

5.6.5.2 Evolution

The general view of the one-run experiment, consisting of seven evolving sub-systems with the details of each migration, is shown on Figure 5-23. In total, the migrant operator activated 33 times during evolution, happening only once during the first sub-stage. The first subtask is significantly easier than the second one: only one sub-system turned to a migrant operation while six others have finished the task on their own. During the second sub-stage, no one evolution came directly to a global solution. Due to the complexity of the problem, all of them got into the local optimum and have required "assistance" from others on a number of occasions (Table 5-6).

From Table 5-6 and Figure 5-23 it can be noted that the most influential during the experiment was sub-system No.1, the SR of which (0.2%) dominated the others and whose genotype has spread to every sub-system. Surely, it cannot be the case that one could declare the principle dominance of lower SRs over the higher ones - more statistics would be needed for that. However, two facts display a general tendency of the SR. First are statistics as to how many times systems with a particular SR become a winner. Systems with the minimum SR of 0.2 became winner seven times, systems with a SR of 0.8 did so three times, while there were no winners with any other SR. Secondly, it took 77 generations for the SR of 0.2 to occupy every system.

What is interesting is that during the process each sub-system - powerful ones as well as weak - has been a "winner" several times. This fact ensures that the WDWC strategy uniformly distributed the probability of finding successful solutions among all the processors.

The case of tough competition is expressed at the moment in between the sub-stages when only one chromosome is allowed to be bred by every sub-system except for a "winner". This extreme migrating act is implemented instead of the release of a system to gradually transfer to the next sub-stage, due to the proven fact of the superiority of the first approach over the second.

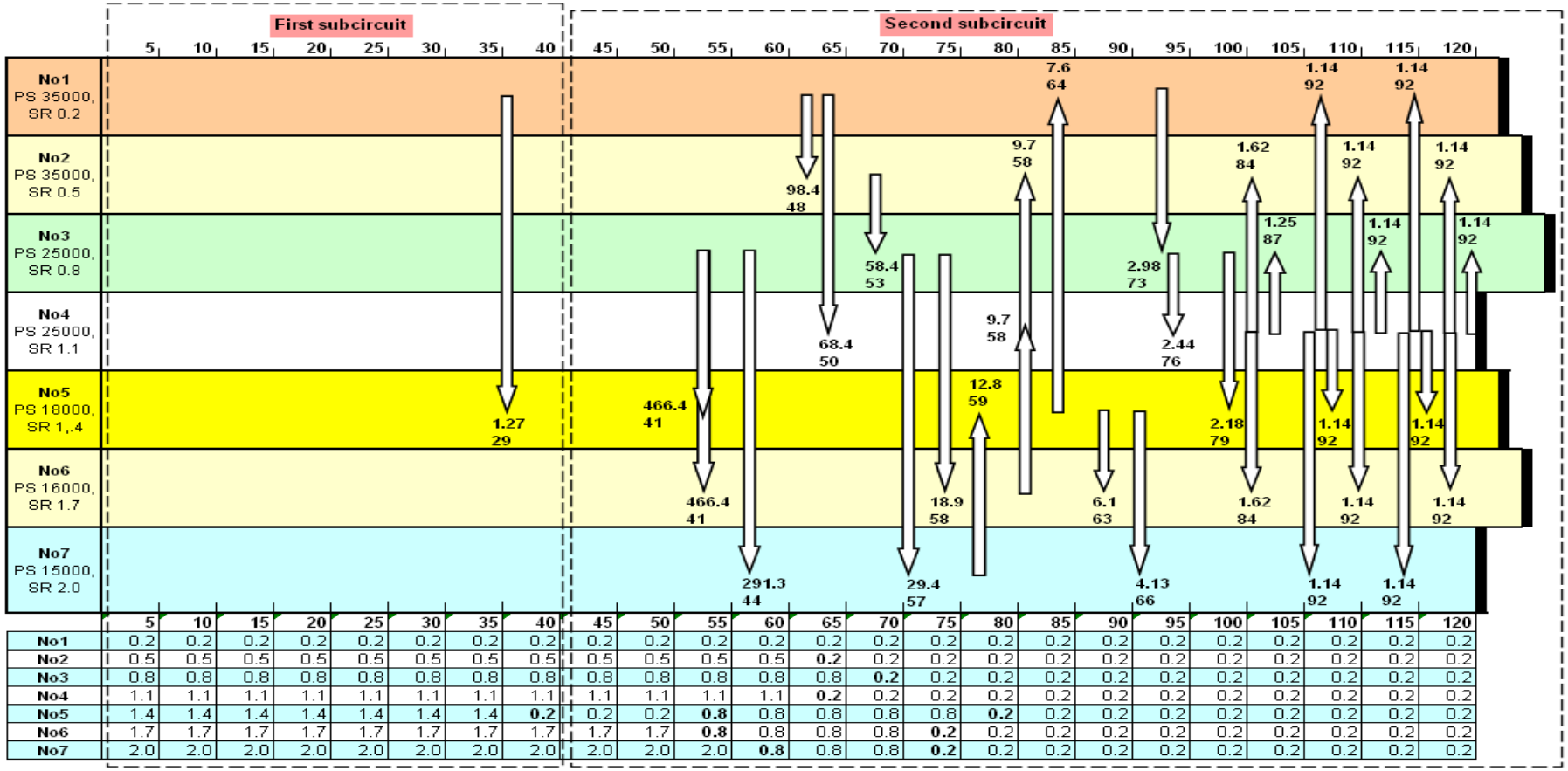
Table 5-6. Initial vs. final parameters and migrant import/export numbers against the properties of 7 parallel sub-systems, where # of individuals and # of migrants is the number of individuals and migrants

No.	**PC description**	**Initial population size, # of individuals**		**Initial SR, %**		**# of Migrants**	
		Initially	Finally	Initially	Finally	Import	Export
1	Intel Core2Quad, 2.4Ghz, 4GB	35000	35000	0.2	0.2	3	3
2	Intel Core2Quad, 2.4Ghz, 4GB	35000	36800	0.5	0.2	5	1
3	Intel Core2Duo, 2.2Ghz, 2GB	25000	19400	0.8	0.2	5	7
4	Intel Core2Duo, 2.2Ghz, 2GB	25000	19900	1.1	0.2	3	15
5	Pentium4, 2.8Ghz, 2.0GB	18000	11800	1.4	0.2	5	3
6	Pentium4, 2.5Ghz, 1.0GB	16000	6600	1.7	0.2	6	2
7	Pentium4, 2.8Ghz, 0.5MB	15000	7200	2.0	0.2	5	1

From generations 91 to 105 the 4th sub-system dominates the rest, spreading out its best genotype to every PC. But after generation 106 no one sub-system can improve and after 15 generations the experiment has stopped.

In Figure 5-24 is shown the results of the population size adaptation. It should be noted from there is a tendency according to which the difference in the computing properties of the sub-systems brings a more divergent population size with a longer than average length of chromosome. On average, the chromosome productivity of PCs No.6 and No.7 has reduced twice of their initial populations.

The best chromosomes with a length of five and six genes from sub-systems No.1 and No.4 and their fragments were stored in and used as substructures.



	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120
No1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No2	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No3	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No4	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	1.1	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No5	1.4	1.4	1.4	1.4	1.4	1.4	1.4	**0.2**	0.2	0.2	**0.8**	0.8	0.8	0.8	0.8	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No6	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	1.7	**0.8**	0.8	0.8	0.8	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2
No7	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	**0.8**	0.8	0.8	**0.2**	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2

Figure 5-23. The migrant schedule. The diagram shows when and how the migrant takes place along a horizontal axis representing generation numbers. Seven numbered sub-systems with an initial population size and SR evolve in parallel from left to right. The arrow indicates which sub-system is a receiver, from where and at which generation. In total, 33 migrants are shown where only one occurred during the first sub-stage. Each migrant is described by the fitness value of the migrant individual and the length of its genes. The table below carries additional information on the process of how SRs migrate along the same axis. The rates just imported are in bold. Since generation 77, the selection rate of 0.2 has dominated the last fifth of the evolution.

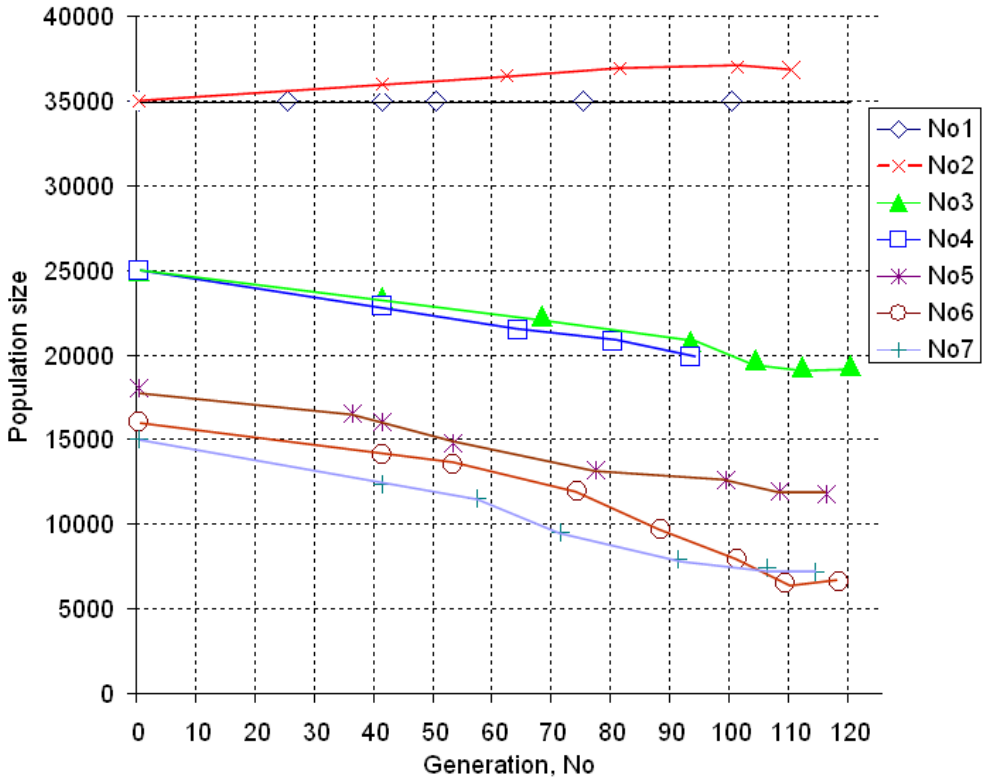


Figure 5-24. Adaptation schedule of population size. The seven curves correspond to seven populations along the horizontal axis representing the generation number. Curve No.1 is a primitive for which all others must synchronize their generation cycles by varying the number of individuals during each migrant operation.

Figure 5-25 demonstrates seven fitness cases (the fitness of the best individuals) during the experiment and in between the sub-stages. The high complexity of the second sub-circuit makes the fitness value scale for longer than three decades. There are distinctly visible “waves” of migrations when straight vertical lines connect one function with another.

Another very important notion concerns an *aggregated selection rate* (ASR), which is a correlation between a total number of selected individuals of the whole system at each generation and a current total number of individuals inside all populations. It is introduced in order to differentiate local SRs and the global one of the system. At the initial conditions when SRs are fixed from 0.2% to 2.0% and the total number of individuals is 169,000, the ASR is equal to 0.914%. However, during the

experiment there are two events that lead to a change in the ASR:

1) When some sub-system activates a migrant operator. According to our methodology, only one chromosome migrates to the stuck sub-system; in other words, one is selected, and the ASR and the SR of the sub-system fall down. Furthermore, together with an individual a new SR comes from a "winner." Since sub-system No.1 has been dominating the others, the ASR has finally converged to a rate of 0.2. Migrations bring spikes in the ASR more frequently and for longer towards the end of the experiment (Figure 5-25) leading to an ASR of 0.124%;

2) When evolution is incrementing to the next sub-stage. At this point, only one chromosome among all the populations (except for the "winner" population) is selected for the next generations and the ASR falls down drastically. In Figure 5-26, this occurs at generation 41 where the ASR reaches 0.048%.

It is useful to watch the adaptive behaviour of the ASR because one can notice the rule according to which the evolution moves forward: as it becomes harder for the evolution to continue the lower the average ASR becomes. In other words, during the successful periods the system tries to expand its gene pool, but at "crisis" periods it sharply reduces a gene pool, focusing on breeding the populations from fewer individuals. One can view this methodology as similar to natural evolution, whereby such kinds of crisis like natural disasters, pandemics and wars force a few survivors to regenerate the rest of the kind.

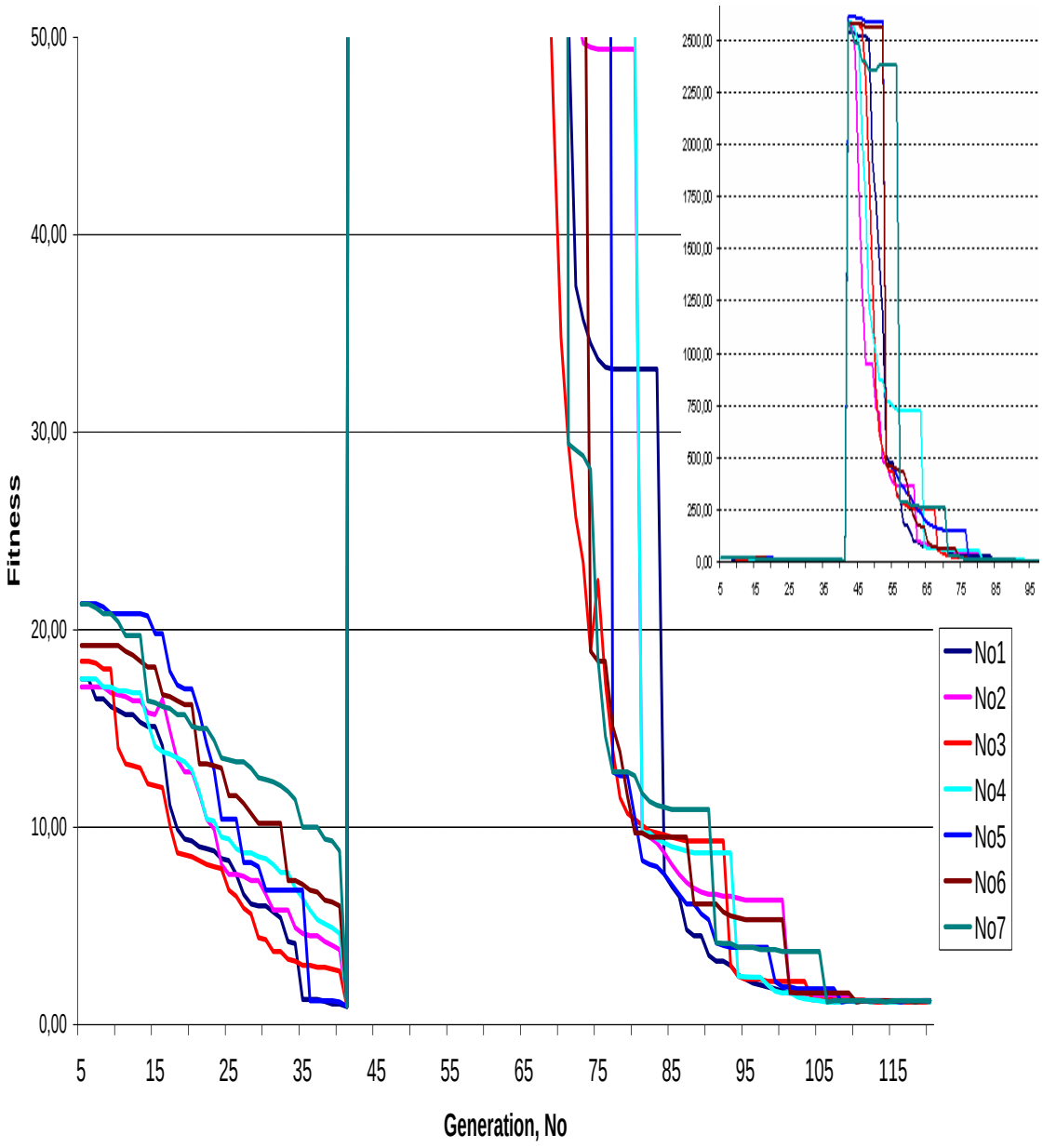


Figure 5-25. Seven fitness cases (the fitness of the best individuals) during both incremental sub-stages. The general view is in the upper right corner. It shows how different they are when scaled to each other due to unlike levels of complexity. The central picture focuses on a fragment between the stages. At generation 41 there is a transition to a second sub-stage. The frequent migrating "waves" are distinctly visible at the end of the second sub-stage (lower right corner).

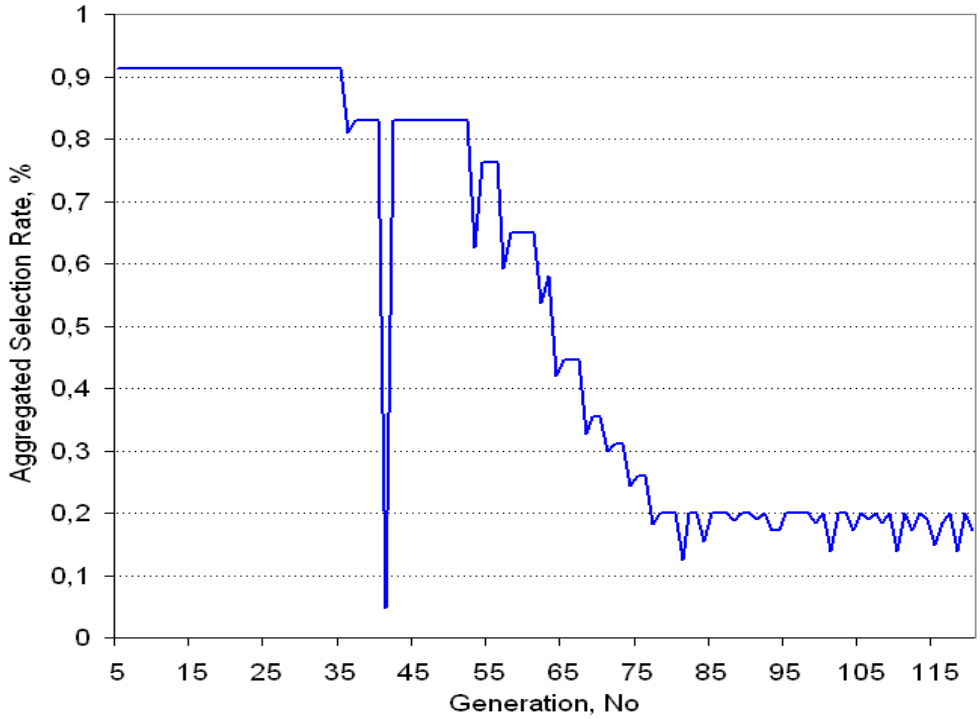


Figure 5-26. The ASR along the generation number. The ASR is converging on a 0.2 selection-rate initially set on PC No.1. The spikes represent the moments of migrations. The ASR contractions are visible when the largest is reaching 0.048% during the first generation of the second sub-stage.

5.6.6 Comparison

The methodology presented so far was discovered after a number of failed attempts to evolve a TIMC by means of standard non-parallel ES-based evolution with more conventional SRs. In this section, some of these failed attempts are presented so as to enable a comparison.

Being applied towards less complicated circuits ([33], [65], [115]) our previous approach could successfully solve all of the tasks set, including low-pass filters, computational circuits and 4-output voltage distributor with both superior functional and physical features. If earlier experiments [33] are utilized at SR=10%, the later ones [65] and [115] are applied at SR=1-10%.

During this work - and being inspired by the previous 1%-rate approach's success - a run of three independent evolutions with a SR 1% was undertaken. Next, we have tried to apply SRs of 5.0%, 3% and 0.5% in running another three evolutions. Here we present six independent non-parallel evolutions with SRs of 5%, 3%, 1% and 0.5% applied towards a TIMC (Table 5-7). All of them have been applied at the second incremental sub-stage of the experiment, since the first one is too easy to play the role of a challenging task. All of the evolutionary parameters and operators have been the same as described in this section except for the SRs and the absence of communication among the sub-systems.

Table 5-7. The population size and initial conditions of the 3 non-parallel PCs

No.	PC description	Initial population size, individ.	SR, %	Best fitness achieved
1s	**Intel Core2Quad, 2.4Ghz, 4GB**	35,000	1.0	2410,8
2s	**Intel Core2Quad, 2.4Ghz, 4GB**	35,000	1.0	873,4
3s	**Intel Core2Duo, 2.2Ghz, 2GB***	25,000	1.0	985,6
4s	**Intel Core2Quad, 2.4Ghz, 4GB***	35,000	5.0	2303,5
5s	**Intel Core2Duo, 2.2Ghz, 2GB***	25,000	3.0	2389,5
6s	**Intel Core2Quad, 2.4Ghz, 4GB***	35,000	0.5	507,4

* The population sizes are set in a way that the duration of the first generation at each sub-system is approximately equal. For the purpose of comparison, different SRs have been set for identical PC configurations.

In Figure 5-27 there are six fitness cases of evolutions that are superposed with seven cases of VNFEs. It should be noted that there is a very high fitness barrier, at about 40-43 chromosome lengths for both evolutions. This represents the specific feature that is inherent to the functioning of a TIMC. Once an evolution gets over this barrier, the fitness improves from about 2200 until 1000. If in the case of WDWC three sub-systems that got stuck at this barrier continued evolving after the migrant operation, three sub-systems of the standard evolution left the experiment after 15 futile generations. The second barrier appears at length 48-50 and a fitness of about 100. However, the conventional evolution met it much earlier at a fitness of higher than 500. It can be

supposed that this happens due to the relatively high SRs. That is, since the DM is applied to both cases, it requires the same large amount of clones per individual for manifesting itself, for which evolutions with higher SRs make even worse provision. As such, the rest of the conventional evolutions were stopped in there.

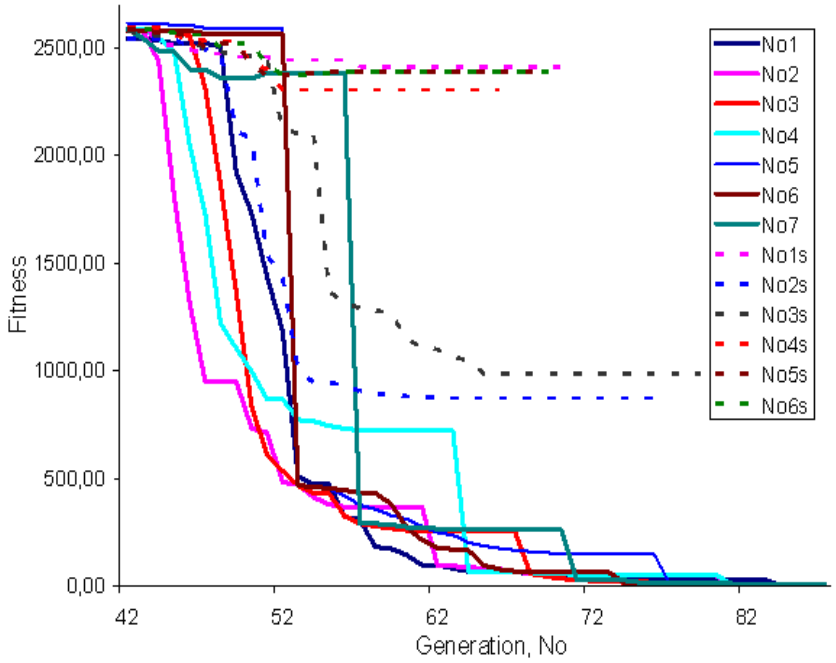


Figure 5-27. By the dotted lines there are six fitness cases, No.1s-No.6s along the generation number corresponding to the SR from 0.5% to 5.0%, and 7 cases, No.1-No.7 of VNFEs with selection rates from 0.2 to 2.0%. All six "conventional" evolutions have stuck far away from the targeted fitness.

5.6.7 Discussion

In this section is described a novel methodology of parallel island-model sub-systems with adaptive parameters. The methodology is called *Very Narrow Focused Evolution* (VNFE) due to its possession of very small SRs. It has been described as to why authors are obliged to apply narrow SRs and enable the sub-systems to destroy the genotypes of competitors during the evolution, called *Winner-Dominates-Winner-Cooperates* (WDWC). The novel *Differentiated Mutation* (DM) strategy that is built up is based on

four concepts: *virtual mutation,* the *mutation type,* the *mutation way* and the *mutation strategy*. Together with such the operating tools as the *diversification of mutation history*, *mutation pressure* and *radical mutation,* the DM strategy represents quite an aggressive operator which increases the mutation rate each time a chromosome does not bring an improvement. These two relatively extreme approaches - DM and WDWC - work in balance with each other during both "crisis" and "wealth" periods: whenever the mutation adapts so as to operate more aggressively and without success, the more actively WDWC works in bringing the aggregated selection rate (ASR) to lower values. Conversely, if the evolutions gradually move forward together, both operators let the system work alone within the frames of standard parameters. Thus, the proposed system is able to apply two adaptive modes; one is a "standard evolution" within the frames of conventional *reference mutation rate* (4%) and selection (0.2-2.0%) parameters; the other - VNFE - is triggered whenever the first regime meets any problem and involves DM (up to 80%) and WDWC (down to 0.124%).

Indeed, this doubled technique has expressed itself in full during an experiment. At the first sub-stage, WDWC brought little assistance in solving an initial subtask with only one migration. Conversely, since generation 53 of the second sub-stage during the remaining 67 generations, there are 33 migrations, which meant that VNFE played an important role.

It is obvious that if the migrant operator activation term were to be adaptive to the complexity level of the subtasks, the first sub-stage would be involved in the genotype exchange more actively and probably would converge in much quicker and better results, in terms of component economy and functionality. The question that arises here as to the effectiveness of VNFE towards such easy problems is the trade-off between "standard evolution" and VNFE, namely what is better: seven solutions to choose among evolved at N independent stations or one solution made in a team by N with the use of VNFE? The answer to the last question is possible only with the running of tests for both approaches on relatively easy tasks, which should be done in future.

The evolved TIMC is a core part of a class of commercialized devices that are known as "time to amplitude converters" (TAC) [161]. Human designers with substantial

practical experience in the design of analogue and digital circuits have attempted to design TIMCs with the obtained technical features. Considering the task, the designer draws the conclusion that the problem is not in the design of a TIMC itself, but rather in reaching such performance features as “usable distance/time range” and “measurement accuracy.” Meanwhile, is using purely analogue components there is no visible methodology for making it, while the digital approach to this task has been well-known for many years [118].

5.7 Summary of Chapter 5

The main contribution of Chapter 5 to the thesis is by the development a novel methodology based on *incremental parallel island-model evolution with adaptive parameters*.

The chapter begins with the introduction of types of incremental evolution by the degree of involvement of sub-solutions in the mutation procedure. Another helpful idea is the defining of different types of incremental coding. Both ideas have enriched the conventional concept of incremental evolution and are described from a practical point of view and with practical recommendations. The developed system was then applied towards the most interesting circuit regarded thus far, the 8-output VDC. The results of Experiment 14 have persuaded us to make further improvements to the system in constructing the parallel island-model system. The new system had an unconventional feature: only a single migrant is allowed to migrate from a winner sub-system to every other by destroying all the previous genotypes. This feature causes the *aggregated SR* of the sub-systems to drastically fall down. However, due to the DM technique - which requires a lower SR - the new approach succeeded at Experiment 15.

During Experiment 15 another disadvantage became visible, namely the idle running and idle waiting of the sub-systems. The tackling of these problems led to a brand new methodology called the WDWC parallel evolution strategy. This last strategy is characterized by an even lower *aggregated SR,* which led to the naming of the whole approach as *Very Narrow Focused Evolution* (VNFE). During VNFE - in contrast to the

previous approach - the migrations were regarded as the being near to *radical mutation* in their treatment against the *stalling effect*. That is, migrations are enabled if radical mutations are helpless. At the same time, the migration procedure happens under the same terms: a single migrant and the destruction of all the population genotypes at "loser" sub-systems.

As it is described in Section 5.6.7, the VNFE operates very small SRs due to a combination of two unconventional approaches - DM and WDWC - which work in balance with each other during "crisis" and "wealth" periods. During Experiment 16, this doubled technique was applied towards the unconventional target in the area of automatic analogue circuit synthesis, the TIMC which originally consists of several digital circuits. While being uninvolved in the first sub-stage due to its easiness, the VNFE has very actively expressed itself during the second sub-stage and it successfully evolved the targeted circuit.

Chapter 6. Conclusions

6.1 Summary

The work presented in a step-by-step fashion and followed the construction of an automatic circuit synthesis tool. The author's target during the writing of the thesis was to present in the most feasible way a method for creating such a tool that would be able to synthesise complex analogue circuits from scratch.

Thus, this thesis consists of the number of versions of the proposed system and narration focuses on the construction and probation of each system stage step-by-step. Since the test results were not foreseen in advance, the scope was widened each time the system failed to achieve the objectives while the initial overall target was reached. Thus, the work is segmented into three sections corresponding to three main versions of the system against the three objectives set in Section 1.3. Each section in its turn describes several subversions of the evolutionary system, so in total eleven system versions are described and tested in the thesis. Each version has been built upon the previous one in such a way that every subroutine of the previous system has been utilized in the next version. The only exception is the subroutine that checks the validity of the circuits to be evaluated. The last one is to provide the constrained evolution.

First of all, there is creation of the framework system from scratch in Chapter 3. Here, the motivating features of the proposed system - like unconstrained evolution, fined-grained evolution, etc. - have been established, along with such basic system parts as representation, mutation, evaluation, etc. (Section 2.2). A novel approach to the mutation procedure was introduced called the *Rule of equal mutation probabilities* (REMP). As a result, the framework system effectively evolved competitive LCR circuits. Six subversions of the system have been tested, which are the intersection of the versions composed of different components, namely LC/LCR, mutation strategy, ILG/OLG and Constrained/Unconstrained versions. Based on the experimental results, such basic techniques as unconstrained evolution, REMP, and the OLG varying strategy

(with DEM), have been approved as basic and essential parts of the system targeted towards the overall aim.

In Chapter 4, and based on such known evolution enhancing techniques as *substructure reuse, individual level mutation,* and *ranking parsimony pressure,* as well as novel developments such as the concept of *virtual mutation* and - based on it – the *differentiated mutation* (DM) approach, the framework system was upgraded to a second version that enabled it to effectively evolve competitive LCRQQ circuits. Two sub-versions of the system successfully evolved two different types of circuits: two-pin and five-pin circuits. It has been shown that although the DM technique has been derived from properties that are inherent - particularly to analogue circuits, it may be extended to other real world instances.

In Chapter 5, three versions of the system evolved two of the most complex circuits in the thesis, which represent the thesis's overall aim: circuits from a prospective application domain that are problematic for conventional design. A novel *incremental approach* for analogue circuits and a *Very Narrow Focused Evolution* (VFNE) based on parallel *Winner-Dominates-Winner-Cooperates* (WDWC) strategy was introduced.

The circuits evolved by the framework and intermediate systems considerably exceeded the rival circuits in terms of performance precision.

In Chapter 5, the importance of the balance between exploitation and exploration during evolution was shown again, demonstrating that the novel VNFE with extremely small selection rates being coupled with quite an "aggressive" DM technique is able to effectively synthesise large scale circuits. All of these novel techniques have been discovered and explored as *ad hoc* problems.

Besides the primary aims and objectives already achieved, this work plainly stated through a series of experiments that *the scalability of an analogue circuit's evolution can be enhanced by exploiting together unconstrained evolution, incremental evolution, parallel evolution and the adaptation technique within one system.* Namely, it has been shown that:

- In the first part of the thesis unconstrained evolution can discover analogue circuits that may exceed the functionality, component economy and computing effort utilized by the circuits evolved by constrained evolution;

- In the second part of the thesis, unconstrained evolution, incremental evolution, the parallel strategy and the adaptation technique in one system can synthesis analogue circuits of a larger scale and with better functionality, component economy and computing effort utilized than the circuits evolved by other evolutionary methods.

6.2 Critical Evaluation of the Work

This thesis does not pretend to tackle any other problem outside of its objectives and - as with any other work - it has its own weaknesses. If the contents up until now narrated the research's achievements, in this section the weak sides of the work from the author's point of view are presented. There are two main classes of critical points that should be mentioned. The first one concerns the general approach to the research. The second refers to the details of how and why some techniques were developed and applied.

6.2.1 Critique of the approach

There are three main points relating to the general approach that could be criticized.

1. Not every design decision in the thesis is fully justified. Mainly there are two kinds of such decisions. The first are the technical decisions when designing certain experimental setups, such as embryo circuits, fitness functions, etc. - mostly in Chapters 3 and 4. This is done for the sake of reaching the first and the second objectives of the thesis, i.e. providing similar start ups to those published in comparable works so as to make the comparison of results easier.

2. The second kind of "poor decisions" is more strategic, such as the application of some techniques without empirical proofs of their effectiveness. The same applies to the experimental data which has not been explored and analyzed deeply enough before starting another one, for example, the application of such techniques as the range-dependant ranking procedure, *R-support* components and the *Rule of equal mutation probabilities* (REMP). These are made based two criteria: simplicity and intuition. The first one is admittedly advantageous in enabling one to focus on other issues. As an example of the second case is the assumption from [38] that unconstrained evolution stimulates unconventional designs and encourages the development and application of *R-support* components and REMP so as to enhance the unconstraining of analogue circuits, without empirical exploration of these techniques.

3. Another criticism of the technique should relate to the lack of empirical data as to how the resulting system works on relatively easy tasks. It really would be helpful in getting the data on how the resulted system became powerful, if to evolve the targets of Chapters 3 and 4 (filters and CCs).

Despite the justifications given to the "poor decisions", the last ones are always about the trade-off between time and reaching final targets, and thus are amenable to critics.

6.2.2 Critic on techniques

Concerning the technical details, there are four remarks that should be made.

1. On one side, the work has been motivated and inspired by potential industrial applications, but on the other side the work looks purely theoretical and stands far behind real world applications. This arises from the need to perform a multi-objective evolution. Multi-objective fitness measures are typically associated with industrial strength problems enabling, first of all, a count of more targets that concern the circuit itself, for example, high/low input/output impedance, the

level of noise, etc. More than a dozen of them are utilized in [58] and [53]. Secondly, it may help to reach robustness by tackling the problem of internal and external variations.

2. There should be more effort made towards defining the optimum mutation strategy, namely, the combination relating the three mutation operators, the diversification of mutation history, the mutation pressure and radical mutation. This problem refers to the more general, where little exploration has been made in relation to the dependence between the highly explorative DM operator and exploitative VNFE. This would be good to due to the "extreme" nature of both and, surely, for the sake of setting out the effective relationship between them.

3. As a further subject of criticism we can be look to the lack of problem-specific knowledge during all of the experiments. Indeed, only the minimum problem-specific knowledge has been used for setting up the experiments: at the FF setting and the embryo setting (no inputs/ outputs), some knowledge was required for the decomposition of the tasks. Probably, problem-specific knowledge would significantly help in reaching faster and better results. For example, based on such knowledge, the number of component parameter ranges to choose among for evolution could be significantly reduced. However, this was not done because it would be somewhat against the thesis' strategy according to which the system should create circuits about which little design knowledge is available.

4. The *pressure-constant* - the meaning of which is a predicted number of genes in the target (the smaller number the higher the pressure is applied) – can, with high level of confidence, be criticized for a lack of flexibility. According to the classification set by Bentley [150], the proposed ranking belongs to the range-dependent methods, while such a highly sensitive parameter should be more adaptable (range-independent).

There are probably other types of criticisms applicable to the work, since this section has not targeted to draw out all the weaknesses but rather to indicate the most

meaningful. Some of the drawbacks are set out for subsequent work in *Section 6.4 Future Work.*

6.3 Perspectives

It should be noticed, first of all, that the suggested approach requires significant computational resources, and the first perspective holds that the speed of commercially available single computers increases quickly in accordance with "Moore's law"[12] [160], which promises faster and better results.

More than two decades have passed since the appearance of EHW and a decade has passed since the first hope for a "killer application" of EHW [3] - quite a long period of time from a commercial point of view. With time, many developed systems are required to make a contribution to the real world problems. This pressure makes some researchers turn to methodologies that combine an evolutionary approach with other optimization engines, like simulated annealing, ant colony algorithms, etc. creating so-called hybrid EA. These last ones, together with closed-ended constrained evolution and knowledge-based substructure databases, produce trustworthy circuits in a very short time [53], [104].

However, the initial idea behind EHW was not only to use it as a CAD system, but also for significantly broader kinds of applications, including adaptive systems, evolutionary robotics, etc. [111]. The last one requires faster rates of interaction with the environment and higher computational power in order to evolve the enormous number of configurations for every interacted situation. These kinds of applications still stand in the wings, waiting for when the "Moor's Law" [160] - which is not expected to stop until 2015 or 2020 or later - will bring the suitable HW conditions.

In this situation, the proposed work suggests open-ended unconstrained EHW perspectives in the design of such analogue circuits that lie beyond the conventional boundaries of applications. This may become a reality soon due to the following reasons:

[12] Moore's law describes a long-term trend in the history of computing hardware, namely that the number of transistors that can be placed inexpensively on an integrated circuit doubles approximately every two years.

1) the proposed system will benefit from the real nature of EA, whose power is to find unconventional solutions for conventional tasks as well as to find solutions for unconventional tasks; 2) the proposed system does not need such extensive processing capacity as is required by adaptive systems. Namely, the main perspective of the approach suggested from the point of view of future applications is the system's *ability to design the analogue system-on-a-chip* (SOC), where all parts of the proposed electronic system are integrated into a single integrated circuit (IC) chip. Conventionally, SOC "may contain digital, analogue, mixed-signal, and often radio-frequency functions – all on a single chip substrate" [162]. The difference with the current concept of SOC is - while conventional SOCs are supposed to contain digital and mixed analogue-digital signals - the proposed SOC is suggested to contain only analogue components that process purely analogue signals. The advantages that this technology may bring to potential users in comparison with conventional SOC are:

1. The compactness of such the systems being located in one crystal. This advantage is based on economizing on synchronization circuits, and other redundant circuits that are required to support digital logic inside ICs. Moreover, as is presented in this thesis, unconstrained evolution suggests considerable economy in terms of components in comparison with human-designed analogue circuits.

2. Decreased power consumption. This feature is caused by prior advantage as well as by the inherent nature of analogue electronics. Furthermore, the evolutionary approach is able to enable a designer to set the preferable power supply as one of an evolution's objective. This feature is especially pertinent given trends in global energy efficiency.

3. As a cause of the previous advantage, the larger IC size is due to decreased heat extraction. This, and other features like faster input-to-output signal processing, is a well-known fact when comparing analogue and digital circuits.

In this sense, the CCs evolved in this thesis, the 4/8-output VDCs and TIMC - may be regarded as analogue SOCs. The functions that these systems perform are quite simple from the point of view of the digital designer, but they are hard issues for the specialist in analogue. The last two targets do not exist in analogue circuitry, but in digital circuitry they comprise the bulk of digital circuits. There are a lot of applications that are potentially waiting for the technique described in this thesis. Some of them are those that utilize sensors and require tiny sizes and low power consumption. For example, *wearable electronics* [163] and *embedded systems* [164]. The concept of the prospective application of the proposed system is shown on Figure 6-1.

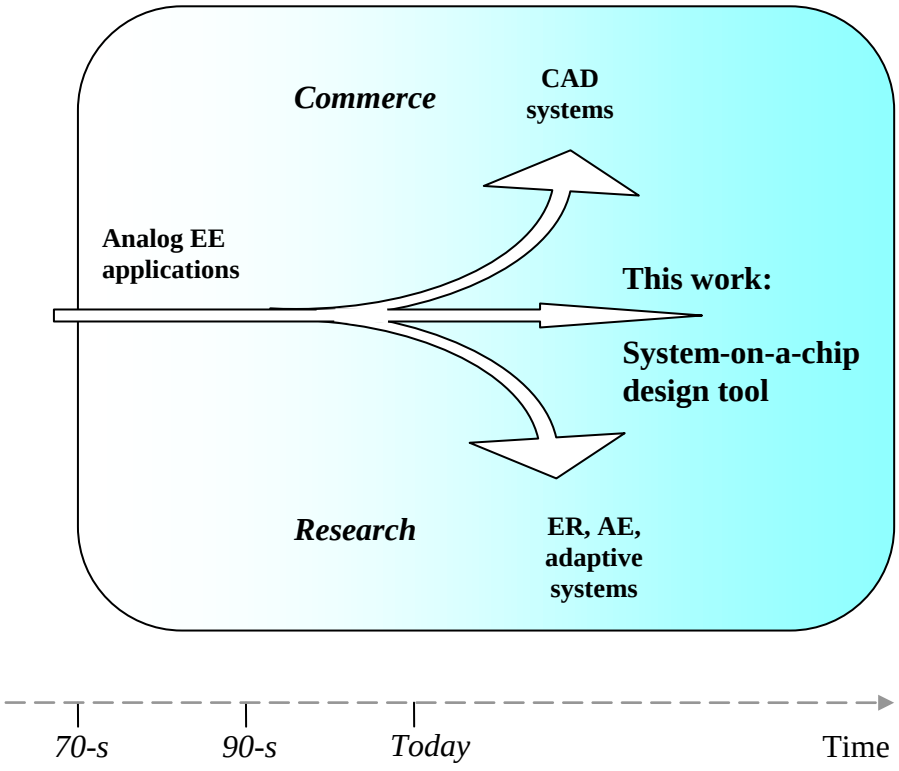


Figure 6-1. The perspective positioning of the work presented inside an application domain where CAD is Circuit Automated Design, ER is Evolutionary Robotics, AE is Artificial Embryology and EE is Evolutionary Electronics.

This idea looks alluring from the following point of view. In past - and with the arrival of the digital era - analogue circuits have been excluded by digital ones, leaving them important mostly as components for digital logic. However, with a novel approach

analogue circuits have a chance of coming back, replacing digital circuits in some conventionally adopted applications. In this sense, the contribution of this work to the future perspective is not the design of circuits that might rival those designed in the conventional manner in the analogue domain, but rather is bringing forward the day when the novel approach might evolve industry-feasible analogue SOCs that are rival to digital circuits.

During the evolution of circuits for unconventional applications, it has been noticed that the character of the evolutionary search differs from evolving conventional analogue circuits. While in the second case the steps towards fitness improvement over recent generations lay inside quite a narrow range relative to the whole fitness range, in the first case this range is significantly wider due to one to three occasional “leaps” of improvements. The first “leaps” always has a crucial meaning and one of the largest. As can be seen by Figures 5-7 and 5-20, evolution was unable to make it in those cases.

This exciting and intuitive hypothesis has arisen to explain such behaviour. It looks as though evolution needs some time for accumulating genotype material inside chromosomes in order to search for some kind of helpful heuristics. According to the DM strategy, it is possible for chromosomes to accumulate eight extra genes beyond the best individual size. As such, it looks like evolution tries to build up usable intermediate substructures to provide the current required functional fragment to the overall singular output function. If for the less complex tasks (CCs, 4-output VDC) the smaller sizes of such the substructures were enough for the progress, for the complicated ones (4-output VDC, TIMC) they tend to be larger.

In this regard, it would be quite interesting in future experiments to enable evolution to accumulate such substructures of larger sizes and to target the performance of more complex functions, for example analogue memory. The interesting question is whether evolution will be able to organize the structure of, say 30 analogue components, so as to memorize a fragment of an input voltage signal limited in time? If in digital logic the simple capacitor ability of keeping a charge as a logic unit is exploited, will evolution exploit it in the same way or will it settle in some way the continuous nature of analogue components for memory construction?

6.4 Future work

The main problem of the technique described by this thesis is that the circuits evolved with unconstrained open-ended evolution are not robust. The problem is aggravated by the fact that evolution is extrinsic, which makes the resulting circuits even further from real-life application ICs. However the last disadvantage is compensated in some way by the fine-grained flexibility that the simulation SW seems to suggest, because all of the available HW tools have fixed structural cells with limited connections among them. In this sense, SW may even help in discovering optimum reconfigurable evolution-oriented HW [165] which suits it to particular kinds of tasks.

To make a circuit robust, its work should be simulated in SW against such variations as processing, both internal and external [6]. The last two are of special importance. Internal includes variations in component parameters, xSPICE model parameters, voltage/current sources, noise, etc. External variation, first of all, concerns temperature. There is no common understanding as to how to make an analogue circuit robust against all of the variations - it is an inalienable part of the whole design process "that is characterized by a combination of experience and intuition and requires a thorough knowledge of the process characteristics and the detailed specifications of the actual product" [140]. However, by being armed with an evolutionary tool, this task is approachable.

As it was mentioned in Section 2.3.4, there are two ways to design robust circuits: to evolve designs and robustness at once [53], [54], [57] or else to make it in two evolutions [28], [55], [56], [58], [59]. The last method will allow the preservation of the results obtained and features of the technique developed previously without significant adjustments.

Thus, the resulting circuits obtained by the present technique should be downloaded into a newly developed system targeted in particular towards increasing their robustness. There are different ways of implementing such the system. One known way is to use multi-objective evolution [58] where different types of variations are used as objectives. Another way is an iterative approach [99], where variations are added as objectives one

by one. Utilizing parallel evolution, it is possible to evolve each sub-system which is struggling per variation. As such, the resulted genotypes are united in one evolution toward the final solution.

Another factor that should be counted for future work are the different performance requirements of the targeted circuit. When it concerns commercial circuits, the data sheets used to specify them typically contain a dozen or more different performance requirements [58]. Therefore, multi-objective evolution should be implemented by the current technique.

To enhance the evolution, it is decided that the use of multi-cluster computers should accelerate whole the process. For example, if it were possible to use a home-built low-cost Beowulf-style parallel cluster computer system consisting of 66 processing nodes - similar to that one used in [52] - the evolution time for the 8-output VDC would be 5-times less, that is about 2-3 days instead of 2 weeks.

As to the next short-term targets, it would be interesting to test the novel approach on evolution of multi-input circuits such as convergent neuron. From an industry-feasible point of view, it would be interesting to design a temperature-sensing circuit.

Appendix A – PSPICE Model Parameters

A1 - Bipolar transistor model parameters

Model parameters	Description	Units	Default
AF	flicker noise exponent		1.0
BF	ideal maximum forward beta		100.0
BR	ideal maximum reverse beta		1.0
CJC	base-collector zero-bias p-n capacitance	farad	0.0
CJE	base-emitter zero-bias p-n capacitance	farad	0.0
CJS (CCS)	substrate zero-bias p-n capacitance	farad	0.0
CN	quasi-saturation temperature coefficient for hole mobility		2.42 NPN 2.20 PNP
D	quasi-saturation temperature coefficient for scattering-limited hole carrier velocity		0.87 NPN 0.52 PNP
EG	bandgap voltage (barrier height)	eV	1.11
FC	forward-bias depletion capacitor coefficient		0.5
GAMMA	epitaxial region doping factor		1E-11
IKF (IK)	corner for forward-beta high-current roll-off	amp	infinite
IKR	corner for reverse-beta high-current roll-off	amp	infinite
IRB	current at which Rb falls halfway to	amp	infinite

Model parameters	Description	Units	Default
IS	transport saturation current	amp	1E-16
ISC (C4)	base-collector leakage saturation current	amp	0.0
ISE (C2)	base-emitter leakage saturation current	amp	0.0
ISS	substrate p-n saturation current	amp	0.0
ITF	transit time dependency on Ic	amp	0.0
KF	flicker noise coefficient		0.0
MJC (MC)	base-collector p-n grading factor		0.33
MJE (ME)	base-emitter p-n grading factor		0.33
MJS (MS)	substrate p-n grading factor		0.0
NC	base-collector leakage emission coefficient		2.0
NE	base-emitter leakage emission coefficient		1.5
NF	forward current emission coefficient		1.0
NK	high-current roll-off coefficient		0.5
NR	reverse current emission coefficient		1.0
NS	substrate p-n emission coefficient		1.0
PTF	excess phase @ 1/(2π·TF)Hz	degree	0.0
QCO	epitaxial region charge factor	coulomb	0.0

Model parameters	Description	Units	Default
QUASIMOD	quasi-saturation model flag for temperature dependence if QUASIMOD = 0, then no GAMMA, RCO, VO temperature dependence if QUASIMOD = 1, then include GAMMA, RCO, VO temperature dependence		0
RB	zero-bias (maximum) base resistance	ohm	0.0
RBM	minimum base resistance	ohm	RB
RC	collector ohmic resistance	ohm	0.0
RCO	epitaxial region resistance	ohm	0.0
RE	emitter ohmic resistance	ohm	0.0
TF	ideal forward transit time	sec	0.0
TR	ideal reverse transit time	sec	0.0
TRB1	RB temperature coefficient (linear)	$^{\circ}C^{-1}$	0.0
TRB2	RB temperature coefficient (quadratic)	$^{\circ}C^{-2}$	0.0
TRC1	RC temperature coefficient (linear)	$^{\circ}C^{-1}$	0.0
TRC2	RC temperature coefficient (quadratic)	$^{\circ}C^{-2}$	0.0
TRE1	RE temperature coefficient (linear)	$^{\circ}C^{-1}$	0.0
TRE2	RE temperature coefficient (quadratic)	$^{\circ}C^{-2}$	0.0
TRM1	RBM temperature coefficient (linear)	$^{\circ}C^{-1}$	0.0
TRM2	RBM temperature coefficient (quadratic)	$^{\circ}C^{-2}$	0.0

Model parameters	Description	Units	Default
T_ABS	absolute temperature	°C	
T_MEASURED	measured temperature	°C	
T_REL_GLOBAL	relative to current temperature	°C	
T_REL_LOCAL	relative to AKO model temperature	°C	
VAF (VA)	forward Early voltage	volt	infinite
VAR (VB)	reverse Early voltage	volt	infinite
VG	quasi-saturation extrapolated bandgap voltage at 0° K	V	1.206
VJC (PC)	base-collector built-in potential	volt	0.75
VJE (PE)	base-emitter built-in potential	volt	0.75
VJS (PS)	substrate p-n built-in potential	volt	0.75
VO	carrier mobility knee voltage	volt	10.0
VTF	transit time dependency on Vbc	volt	infinite
XCJC	fraction of CJC connected internally to Rb		1.0
XCJC2	fraction of CJC connected internally to Rb		1.0
XCJS	fraction of CJS connected internally to Rc		
XTB	forward and reverse beta temperature coefficient		0.0
XTF	transit time bias dependence coefficient		0.0
XTI (PT)	IS temperature effect exponent		3.0

A2 - Capacitor model parameters

Model parameters	Description	Units	Default
C	capacitance multiplier		1.0
TC1	linear temperature coefficient	$°C^{-1}$	0.0
TC2	quadratic temperature coefficient	$°C^{-2}$	0.0
T_ABS	absolute temperature	°C	
T_MEASURED	measured temperature	°C	
T_REL_GLOBAL	relative to current temperature	°C	
T_REL_LOCAL	relative to AKO model temperature	°C	
VC1	linear voltage coefficient	$volt^{-1}$	0.0
VC2	quadratic voltage coefficient	$volt^{-2}$	0.0

A3 - Inductor model parameters

Model parameters	Description	Units	Default
L	Inductance multiplier		1.0
IL1	Linear current coefficient	amp^{-1}	0.0
IL2	Quadratic current coefficient	amp^{-2}	0.0
TC1	Linear temperature coefficient	$°C^{-1}$	0.0
TC2	Quadratic temperature coefficient	$°C^{-2}$	0.0
T_ABS	Absolute temperature	°C	
T_MEASURED	Measured temperature	°C	

A4 - Resistor model parameters

Model parameters	Description	Units	Default
R	resistance multiplier		1.0
TC1	linear temperature coefficient	$^{\circ}C^{-1}$	0.0
TC2	quadratic temperature coefficient	$^{\circ}C^{-2}$	0.0
TCE	exponential temperature coefficient	%/°C	0.0

Appendix B – PSPICE Distribution Values

Options	Description	Units	Default
GMIN	minimum conductance used for any branch	ohm^{-1}	1.0E-12
ITL1	DC and bias point blind repeating limit		150.0
ITL2	DC and bias point educated guess repeating limit		20.0
ITL4	the limit at any repeating point in transient analysis		10.0
ITL5	total repeating limit for all points for transient analysis (ITL5=0 means ITL5=infinity)		0.0
LIMPTS	maximum points allowed for any print table or plot (LIMPTS=0 means LIMPTS=infinity)		0.0
NUMDGT	number of digits output in print tables (maximum of 8 useful digits)		4.0
PIVREL	relative magnitude required for pivot in matrix solution		1.0E-3
PIVTOL	absolute magnitude required for pivot in matrix solution		1.0E-13
RELTOL	relative accuracy of V and I		0.001
SOLVER	performance package solution algorithm (Solver = 0 selects the original solution algorithm; Solver = 1 selects the advanced solution algorithm)		0 (In PSpice A/D Basics) 1 (In all other PSpice products)
TNOM	default nominal temperature (also the temperature at which model parameters are assumed to have been measured)	°C	27.0
VNTOL	best accuracy of voltages	volt	1.0 uV
WIDTH	same as the .WIDTH OUT= statement (can be set to either 80 or 132)		80.0

Options	Description	Units	Default
GMIN[1]	minimum conductance used for any branch	ohm^{-1}	1.0E-12
ITL1	DC and bias point blind repeating limit		150.0
ITL2	DC and bias point educated guess repeating limit		20.0
ITL4[1]	the limit at any repeating point in transient analysis		10.0
ITL5[2]	total repeating limit for all points for transient analysis (ITL5=0 means ITL5=infinity)		0.0[3]
LIMPTS[2]	maximum points allowed for any print table or plot (LIMPTS=0 means LIMPTS=infinity)		0.0[3]
NUMDGT	number of digits output in print tables (maximum of 8 useful digits)		4.0
PIVREL[2]	relative magnitude required for pivot in matrix solution		1.0E-3
PIVTOL[2]	absolute magnitude required for pivot in matrix solution		1.0E-13
RELTOL[1]	relative accuracy of V and I		0.001
SOLVER[5]	performance package solution algorithm (Solver = 0 selects the original solution algorithm; Solver = 1 selects the advanced solution algorithm)		0 (In PSpice A/D Basics) 1 (In all other PSpice products)
TNOM	default nominal temperature (also the temperature at which model parameters are assumed to have been measured)	°C	27.0
VNTOL[1]	best accuracy of voltages	volt	1.0 uV
WIDTH	same as the .WIDTH OUT= statement (can be set to either 80 or 132)		80.0

Appendix C – PSPICE Decks of the Thesis

C1 - Low-Pass Filter Circuit [33] from Chapter 3.8.5, Experiment 1, Figure 3-17

```
**Low-Pass Filter, Constr.Evol, ILG, LC

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PRINT AC V(2,0)
.lib "nom.lib"

Vs 999 0 AC 2
Rload 999   1     1000
Rsource     0     2     1000

L_59   11    3     1.2e-1
L_57   4     3     2.2e-1
L_58   2     4     1.5e-1
C_86   0     12    2.2e-7
C_87   3     6     1.8e-7
C_88   0     2     6.6e-8
C_89   5     2     3.4e-8
C_90   1     0     3.9e-8
L_60   4     5     2.2e-3
C_94   0     9     1.2e-8
C_95   13    5     1.0e-9
C_96   5     4     3.3e-6
L_66   0     8     1.2e-2
L_62   10    7     3.9e-3
L_63   6     8     2.2e-3
L_64   7     9     2.7e-1
C_98   9     7     1.0e-8
L_65   1     10    1.8e-1
C_100  1     10    1.0e-6
L_67   1     11    1.0e-2
C_101  8     10    3.3e-9
C_102  7     14    4.7e-6
C_103  7     11    3.63e-7
L_68   4     12    4.7e-5
L_69   2     13    6.8e
L_70   1     14    2.7e-2

.END
`
```

C2 - Low-Pass Filter Circuit [33] from Chapter 3.8.5, Experiment 2, Figure 3-18

```
**Low-Pass Filter, Unconst.Evol.,ILG,LC

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PRINT AC V(2,0)
.lib "nom.lib"

Vs 999 0 AC 2
Rload 999   1     1000
Rsource     0     2     1000

C_29   0     3     2.2e-7
L_24   3     12    2.2e-1
L_25   15    6     1.6e-3
C_4    5     10    1.0e-7
Rc_2   5     10    1E+9
L_7    2     4     1.8e-1
C_6    3     8     2.2e-8
C_30   0     9     3.3e-7
C_17   7     0     2.2e-7
C_9    0     11    6.8e-8
Rc_6   0     11    1E+9
C_12   11    6     4.7e-5
L_15   4     7     8.2e-3
C_24   15    8     3.3e-6
C_25   14    10    3.9e-7
L_26   1     8     1.0e-5
C_23   11    8     1.5e-9
L_18   4     12    6.8e-3
L_19   10    14    1.0e-4
L_20   8     6     3.257e-3
L_16   1     3     1.547e-1
L_23   2     14    1.5e-4
C_28   10    2     3.3e-5
L_27   1     2014  4.7
Rl_15  2014  6     1E-7
C_31   12    4     1.5e-7
L_12   9     5     7.775e-2

.END
```

C3 - Low-Pass Filter Circuit [96] from Chapter 3.8.6, Experiment 3, Figure 3-19

```
**Low-pass Filter, Uncons.Evol., ILG, LCR

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PRINT AC V(2,0)
.lib "nom.lib"

Vs 999 0 AC 2
Rload 999   1     1000
Rsource     0     2     1000


L_12   1     6     9.9e-005
Rsprt  1     6     1E-7
C_2    3     6     2.7e-008
L_5    6     3     1.5e-001
C_6    6     0     6.5e-008
L_3    3     4     2.2e-001
C_17   0     3     2.2e-007
C_7    4     0     1.83e-007
L_6    2     4     1.0e-001
C_13   4     2     1.5e-008
C_10   8     0     2.2e-008
L_13   2     8     2.7e-001

.END
```

C4 - Low-Pass Filter Circuit [103] from Chapter 3.10.2.1, Experiment 4, Figure 3-23

```
**Low-Pass Filter, Costr.Evol,LCR, OLG

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PLOT AC V(2,0)
.lib "nom.lib"
.PROBE

Vs 999 0 AC 2
Rsource     999    1     1K
Rload 0     2      1K

C_0    4      6      1.5e-7
L_4    3      12     4.7e-2
L_0    3      6      2.2e-1
L_1    4      0      3.9e-2
C_2    3      14     2.2e-7
L_2    5      7      1.2e-1
L_3    2      6      6.8e-2
C_14   1      3      1.2e-8
C_4    15     0      6.8e-8
C_5    8      9      3.3e-9
L_5    1      7      3.9e-3
C_6    1      7      9.0e-7
L_6    5      8      1.5e+0
C_8    11     12     2.2e-9
L_7    9      0      1.0e-4
C_10   9      10     2.2e-9
L_8    2      10     2.2e+0
C_11   3      9      1.5e-9
L_10   11     0      3.3e-1
L_11   12     13     1.8e-3
C_12   5      12     1.5e-6
L_12   5      13     4.7e-5
C_13   5      13     1.5e-5
C_15   9      0      2.2e-5
L_14   9      14     4.7e-5
C_16   5      14     1.5e-9
L_15   1      15     1.0e-3

.END
```

C5 - Low-Pass Filter Circuit [103] from Chapter 3.10.2.2, Experiment 5, Figure 3-24

```
**Low-Pass Filter, Uncon.Evol, LCR, OLG

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PLOT AC V(2,0)
.lib "nom.lib"
.PROBE

Vs 999 0 AC 2
Rsource     999   1     1K
Rload 0     2     1K

L_18   4     9     2.2e-1
C_22   3     0     2.2e-7
C_10   4     0     1.83e-7
L_2    1     5     6.77e-2
C_27   1     7     1.5e-8
L_8    2     4     1.0e-1
C_25   5     0     6.9e-8
L_17   3     9     3.9e-3
C_28   8     0     1.0e-5
Rc_6   8     0     1E+9
C_17   3     7     8.2e-7
C_20   2     8     2.2e-8
L_19   7     9     3.9e-3
C_21   8     9     4.7e-9
C_29   2     4     1.2e-8
L_10   1     3     1.735e-1


.END
```

C6 - Low-Pass Filter Circuit [106] from Chapter 3.11.1, Experiment 6, Figure 3-25

```
**Low-Pass Filter, Unconst.Evol.,OLG,LCR

.OPTIONS NOOUTMSG NOBIAS
.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PRINT AC V(2,0)
.lib "nom.lib"

Vs 999 0 AC 2
Rload 999   1     1000
Rsource     0     2     1000

C_273  10    8     2.7e-006
C_270  1     6     6.2e-009
C_271  6     0     1.0e-007
L_164  6     10    5.6e-003
L_157  1     5     1.0e-001
L_177  10    12    1.5e-001
L_180  8     12    3.3e+000
L_173  5     8     2.7e-002
L_168  7     5     1.2e-001
L_176  9     1     1.1371
C_249  8     9     2.2e-009
L_174  11    1     6.8e-003
C_253  7     4     1.76e-007
L_167  3     7     2.2e-001
C_272  0     11    5.6e-008
C_276  12    11    1.5e-009
L_169  13    0     1.5e-002
C_261  3     0     2.2e-007
C_269  2     3     3.9e-008
L_171  3     2     1.5e-001
C_258  0     2     6.8e-008

.END
```

C7 - Low-Pass Filter Circuit [106] from Chapter 3.11.2, Experiment 7, Figure 3-26

```
** Close-to-Ideal Low-Pass Filter

.IC V(1)=0 V(2)=0 V(999)=0
.AC DEC 19 1 100000
.PRINT AC V(2,0)
.lib "nom.lib"

Vs 999 0 AC 2
Rsource 999 1 1K
Rload 0 2 1K

L_10  1     8     1.5e-1
C_13  1     5     4.7e-8
C_19  1     0     6.8e-8
C_32  7     1     4.7e-9
C_0   5     0     2.2e-7
L_25  4     5     2.7e-1
C_18  4     5     3.3e-9
L_18  3     20    4.7e-5
Rl_1 20     5     1E-7
L_26  5     8     1.0e-2
L_16  5     10    8.2e-1
C_22  8     0     2.2e-8
L_19  6     8     6.8e-1
L_23  3     8     2.7e-2
C_33  3     8     3.9e-7
L_14  6     7     3.3e-1
C_31  3     7     1.5e-8
C_28  4     0     2.32e-7
L_3   2     4     1.2e-1
C_26  2     4     1.0e-7
L_15  3     9     1.0e-5
C_29  10    6     6.8e-5
C_25  0     2     4.58e-8
L_24  6     21    1.5e-3
Rl_0  21    9     1E-7
R_7   3     10    5.6e+2
C_34  3     9     1.2e-8
C_36  3     6     4.7e-4
C_37  6     0     4.7e-9

.END
```

C8 - Cube Root Circuits [109] from Chapter 3.12.3, Experiment 8 at generations 3 and 15, Figures 3-27 and 3-28

```
**GENERATION No 3
**CHROMOSOME No 24999
**Fitness 65.9108
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.DC LIN Vs -.25 .25 .025
.PRINT DC V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0
Rsource     999   1     1K
Rload 0     2     1K
Qp0    995   1     5     PBJT
R0     5     996   1.38e+4
Qp1    5     2     1     PBJT
.END

**GENERATION No 15
**CHROMOSOME No 23882
**Fitness 5.9834
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.DC LIN Vs -.25 .25 .025
.PRINT DC V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0
Rsource     999   1     1K
Rload 0     2     1K

Qn0    3     3     4     NBJT
Qp0    3     6     5     PBJT
Qn1    6     1     4     NBJT
R0     2     4     5.63e+2
R1     5     2     7.1e+1
R2     1     3     1.73e+4
Qn2    7     5     7     NBJT
Qp1    5     6     2     PBJT
Qn3    2     7     4     NBJT
R3     7     6     8.11e+4
Qn4    6     2     2     NBJT
Qp4    2     7     3     PBJT
.END
```

C9 - Cube Root Circuit [109] from Chapter 3.12.3, Experiment 8, at generation 133, Figure 3-29

```
**GENERATION No 133
**CHROMOSOME No 34318
**Fitness 2.6839
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.DC LIN Vs -.25 .25 .025
.PRINT DC V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0
Rsource     999   1     1K
Rload 0     2     1K
Qn1    4     5     6     NBJT
Qp1    5     1     3     PBJT
R0     0     1     1.191e+3
Qp2    2     5     3     PBJT
Qn2    3     6     4     NBJT
R1     2     6     3.9e+2
R2     0     5     6.8e+2
Qn3    4     3     4     NBJT
Qn7    6     5     2     NBJT
Qn9    7     9     1     NBJT
Qp9    0     6     1     PBJT
Qn12   7     3     3     NBJT
Qn15   3     7     6     NBJT
Qn16   3     4     2     NBJT
Qp16   5     9     3     PBJT
Qp18   10    7     3     PBJT
R21    2     7     1.8e+5
Qn18   3     7     6     NBJT
Qn19   5     5     0     NBJT
Qn21   9     7     1     NBJT
Qn22   3     4     9     NBJT
R26    3     7     3.9e+5
Qn23   3     7     8     NBJT
Qn26   7     9     2     NBJT
Qn27   10    7     3     NBJT
Qp23   7     2     1     PBJT
Qn28   3     6     7     NBJT
R29    4     8     1.49e+3
R30    4     1     2.706e+5
Qn29   3     4     9     NBJT
R31    9     3     1.487e+6
R32    10    9     2.7e+3
Qp25   6     9     10    PBJT
Qp26   10    2     1     PBJT
R33    4     1     2.7e+5
Qn30   3     4     10    NBJT
.END
```

C10 - Square Root Circuit [115] from Chapter 4.4.2.1, Experiment 9, Figure 4-7

```
**GENERATION No 123

**CHROMOSOME No 678
**Fitness 0.1942 c=32
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.PROBE V(2)
.lib "nom.lib"

.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  996   0     DC    +15.0V
Vneg  995   0     DC    -15.0V
Rpos1  996  3     1e-8
Rneg1  995        4     1e-8
Vs 999 0 PULSE .0 .5 0 .2
Rsource     999   1     1K
Rload 0     2     1K

R0     6     1     1.2e+4
Qn0    5     1     2     NBJT
R1     0     1     1.5e+4
R2     4     2     2.2e+4
Qp2    3     5     6     PBJT
R4     1     8     1.5e+6
R5     7     6     4.7e+3
Qp5    4     6     7     PBJT
R6     7     3     5.6e+3
R8     0     11    5.6e+5
Qn3    3     8     4     NBJT
Qp10   21    22    4     PBJT
R13    6     11    6.8e+5
Qp20   4     17    14    PBJT
Qn13   3     17    3     NBJT
Qp23   3     14    14    PBJT
Qp24   7     14    17    PBJT
R31    0     17    1.0e+4
Qp25   16    16    19    PBJT
Qn15   21    19    15    NBJT
Qp27   3     21    15    PBJT
R36    22    0     4.3e+4
R37    5     22    3.3e+6

.END
```

C11 - Squaring Circuit [115] from Chapter 4.4.2.1, Experiment 10, Figure 4-8

```
**SQUIRING FUNCTION
**GENERATION No 92 CHROMOSOME No 288
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0 PULSE -.25 .25 0 .2
Rsource     999   1     1K
Rload 0     2     1K
Qn0    2     1     3     NBJT
R0     3     1     2.7e+4
Qp0    2     1     6     PBJT
R3     3     6     5.6e+4
R4     5     3     2.16e+6
R5     6     10    8.2e+6
R6     1     31    1.0e+5
R7     7     6     1.5e+3
Qp1    1     9     2     PBJT
Qn2    0     7     19    NBJT
R9     1     10    2.7e+5
Qp2    32    14    3     PBJT
Qn3    18    5     7     NBJT
R10    2     9     2.2e+3
Qp3    4     10    6     PBJT
Qn4    3     25    12    NBJT
Qp5    1     9     12    PBJT
Qp6    24    10    19    PBJT
Qp7    2     31    9     PBJT
Qp9    33    27    27    PBJT
Qp10   10    22    18    PBJT
R14    19    31    6.8e+5
Qn5    17    12    9     NBJT
Qp11   4     12    22    PBJT
Qn7    3     22    8     NBJT
Qn8    2     23    18    NBJT
R16    17    22    1.8e+6
Qp14   10    23    24    PBJT
Qn10   30    4     10    NBJT
Qn11   8     28    12    NBJT
Qn12   27    25    28    NBJT
Qp16   1     14    30    PBJT
Qn13   28    12    12    NBJT
Qn14   4     33    34    NBJT
Qn15   32    32    34    NBJT
.END
```

C12 - Cube Root Circuit [115] from Chapter 4.4.2.2, Experiment 11, Figure 4-9

```
**GENERATION No 152

**CHROMOSOME No 11
**Fitness 0.2507888
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP

Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0 PULSE -.25 .25 0 .2
Rsource     999   1     1K
Rload 0     2     1K

Qp0    4     5     2     PBJT
R0     1     4     1.111e+4
Qp1    8     6     0     PBJT
R1     3     2     1.8e+4
Qn1    5     6     1     NBJT
R3     7     6     4.7e+5
Qp3    22    25    4     PBJT
Qp4    21    1     7     PBJT
Qn2    15    12    20    NBJT
Qp5    2     10    8     PBJT
Qn3    24    13    6     NBJT
Qn4    3     10    6     NBJT
Qp8    4     11    0     PBJT
Qp9    0     11    10    PBJT
Qn5    12    10    5     NBJT
Qn6    3     18    17    NBJT
Qp15   4     6     11    PBJT
Qn9    19    17    12    NBJT
Qn10   8     22    10    NBJT
Qp16   4     15    8     PBJT
Qp17   27    18    19    PBJT
Qp19   20    15    4     PBJT
Qn13   21    20    8     NBJT
R26    19    25    6.5e+3
Qn15   0     20    25    NBJT
R27    29    13    2.2e+4
Qp26   4     27    28    PBJT
```

```
Qn15    0     20    25    NBJT
R27     29    13    2.2e+4
Qp26    4     27    28    PBJT
Qn18    28    5     24    NBJT
R34     24    21    3.3
Qn19    29    13    21    NBJT
Qp30    17    25    25    PBJT
R37     25    17    1.2e+4
Qn20    13    31    17    NBJT
R38     34    24    2.2
Qp31    33    31    34    PBJT
R39     27    31    1.2e+4
Qp32    29    27    34    PBJT
R40     27    29    1.2e+4
Qn21    28    33    29    NBJT

.END
```

C13 - Cubing Circuit [115] from Chapter 4.4.2.2, Experiment 12, Figure 4-10

```
**GENERATION No 78
**CHROMOSOME No 2091
**Fitness 0.0061444
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.lib "nom.lib"
.PROBE V(2)

.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0 PULSE -.25 .25 0 .2
Rsource     999   1     1K
Rload 0     2     1K

R1     1     17    2.2e+4
Qn0    6     5     4     NBJT
Qp0    9     1     5     PBJT
R3     0     16    0.976e+5
Qp1    2     6     1     PBJT
Qn1    24    10    11    NBJT
Qn2    0     10    36    NBJT
R4     8     9     3.27e+4
R5     10    3     4.7e+2
R6     11    8     1.8e+3
Qn3    35    25    12    NBJT
Qp2    12    21    13    PBJT
Qn4    14    5     31    NBJT
Qp3    4     15    18    PBJT
Qn5    15    16    4     NBJT
Qp4    6     18    15    PBJT
Qn6    0     20    32    NBJT
Qp5    40    4     0     PBJT
Qn7    26    13    37    NBJT
Qn8    18    22    21    NBJT
R10    20    13    3.9e+6
Qn9    14    2     22    NBJT
R11    14    9     2.2e+4
R12    13    10    8.16e+6
Qp6    17    24    3     PBJT
Qn10   22    22    21    NBJT
Qn11   20    18    4     NBJT
```

```
R13     17      19      6.81e+4
R14     25      0       4.7e+2
R15     16      26      2.2e+3
Qn13    27      25      22      NBJT
Qn14    3       28      29      NBJT
Qn15    34      27      21      NBJT
Qn16    41      33      32      NBJT
Qp8     34      33      21      PBJT
Qp9     35      38      19      PBJT
Qp10    34      28      36      PBJT
Qn17    37      38      11      NBJT
R19     28      31      9.4e+5
Qn18    38      27      21      NBJT
Qp12    40      41      3       PBJT
Qp13    38      34      19      PBJT
Qn19    29      4       4       NBJT
Qn20    3       13      8       NBJT

.END
```

C14 - 4-output Voltage Distributor Circuit [116] from Chapter 4.6.3, Experiment 13, Figure 4-14

```
**CHROMOSOME No 15110 of Gen 120

**Random No=532
**Best fitness 0.370650

.OPTIONS NOREUSE NOMOD NOECHO
+NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN 62.5ms 5s
.PROBE
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  6     0     DC    15.0V
Vneg  7     0     1.5V
Vs 999 0 PWL (0,0) (3.5,5) (5,0)
Rs    999   1     30

.PRINT TRAN V(2) V(3) V(4) V(5)
R12   0     2     10K
R13   0     3     10K
R14   0     4     10K
R15   0     5     10K

R0     9     6     2.700e+004
R1     6     8     8.200e+002
R2     12    13    3.900e+003
R3     11    17    8.200e+002
R4     26    16    1.500e+003
R5     43    19    2.700e+003
R6     1     20    6.800e+003
R7     22    6     4.700e+004
R8     21    14    1.500e+006
Qn0    9     11    0     NBJT
Qn1    8     9     2     NBJT
Qn2    7     10    2     NBJT
Qn3    13    20    17    NBJT
Qn4    18    15    3     NBJT
Qn5    6     1     12    NBJT
Qp0    9     1     25    PBJT
Qp1    10    8     6     PBJT
Qp2    16    14    6     PBJT
```

```
Qp3     15      19      16      PBJT
Qp4     3       1       15      PBJT
Qp5     3       18      6       PBJT
Qp7     22      1       24      PBJT
C0      12      14      2.200e-005
Rsprt_1         12      14      1E+9
C1      11      25      8.200e-009
Rsprt_2         11      25      1E+9
R9      19      26      1.000e+004
R10     41      35      2.200e+001
Qn6     22      23      0       NBJT
Qn7     28      22      4       NBJT
Qn8     6       17      23      NBJT
Qn9     26      37      37      NBJT
Qn10    33      24      6       NBJT
Qp8     4       21      6       PBJT
Qp9     7       28      21      PBJT
Qp10    27      27      0       PBJT
Qp11    23      14      6       PBJT
R11     31      6       1.500e+005
Qn11    31      33      0       NBJT
Qn12    30      31      5       NBJT
Qn13    6       32      5       NBJT
Qn24    35      40      27      NBJT
Qp13    32      30      6       PBJT
Qp16    31      1       0       PBJT
R18     33      26      1.200e+004
C9      11      34      1.000e-008
Rsprt_3         11      34      1E+9
R19     38      24      3.900e+005
Qn23    39      37      35      NBJT
Qn25    6       38      23      NBJT
R21     39      27      4.700e+002
C12     40      9       2.700e-005
Rsprt_4         40      9       1E+9
Qn26    11      24      41      NBJT
R23     34      27      4.700e+002
R24     39      41      4.700e+001
C13     43      38      5.600e-007
Rsprt_5         43      38      1E+9

.END
```

C15 - 8-output Voltage Distributor Circuit [65], [116] from Chapter 5.4.2, Experiment 15, Figure 5-12

```
**CHROMOSOME No 178 of Gen 629
**Random No=806
.OPTIONS NOREUSE NOMOD NOECHO NOOUTMSG
+NOBIAS NOPRBMSG NOPAGE
.TRAN 62.5ms 5s
.MODEL NBJT NPN
.MODEL PBJT PNP
.PROBE
Vpos  3     0     DC    15.0V
Vneg  4     0     1.5V
Vs 999 0 PWL (0,0) (3.5,5) (5,0)
Rs1   999   101   30
Rs2   999   201   30
Rs3   999   301   30
Rs5   999   501   30
Rs4   999   401   30
Rs6   999   601   30
Rs7   999   701   30
Rs8   999   801   30
.PRINT TRAN V(2)
R11   0     102   10
R12   0     202   10
R13   0     302   10
R14   0     402   10
R15   0     502   10
R16   0     602   10
R17   0     702   10
R18   0     802   10

*Qp-39 Qn-46 R-38 C-8 L-7,
*10+22+16+23+14+23+22+8=138,
*Total fitness = 1.757

**first pin 10el  Best fitness 0.09535
Qp0    107   101   105   PBJT
Qp1    3     106   105   PBJT
R0     107   106   2.7e+5
Qn0    102   105   106   NBJT
Qn1    0     107   106   NBJT
Qn2    0     101   105   NBJT
R1     4     101   1.8e+3
Qn3    101   108   105   NBJT
Qn4    4     4     108   NBJT
R2     0     108   9.4e+3
```

```
**second pin 22el Best fitness 0.028614
Qn5     205    201    207    NBJT
Qp2     211    212    206    PBJT
Qp3     202    207    4      PBJT
R3      3      201    1.8e+3
C0      207    4      3.3e-8
Qp4     207    206    4      PBJT
Qn6     205    202    206    NBJT
Qn7     0      205    206    NBJT
Qn8     205    206    208    NBJT
Qp5     4      210    209    PBJT
Qn9     205    210    0      NBJT
R4      210    202    3.9e+4
Qp6     208    210    212    PBJT
R5      3      210    1e+6
Qp7     211    213    4      PBJT
R6      3      0      1E-7
C1      213    202    2.7e-5
R18     213    0      2.7E+7
R7      3      209    1.5e+5
R8      211    213    3.9e+2
Qp8     208    207    214    PBJT
Qp9     208    214    207    PBJT

**third pin 16el Best fitness 0.1744
Qp10    305    306    3      PBJT
Qp11    302    310    3      PBJT
Qp12    307    3      4      PBJT
Qp13    312    3      3      PBJT
Qp14    313    306    3      PBJT
Qn10    310    308    4      NBJT
Qn11    0      307    308    NBJT
Qn12    306    301    307    NBJT
Qn13    307    302    311    NBJT
Qn14    301    309    309    NBJT
R9      308    305    4.7e+1
R10     306    3      1.2e+5
C2      4      308    3.9e-9
C3      312    305    3.9e-8
R37     309    311    1E+5
R38     311    313    1E+5

**fourth pin 23el Best fitness 0.3237
Qn45    406    401    4      NBJT
R39     401    420    1.800e+003
Qp37    409    4      412    PBJT
Qp38    405    408    3      PBJT
Qn50    3      407    402    NBJT
Qp39    407    409    410    PBJT
Qp40    410    406    405    PBJT
Qn46    425    413    427    NBJT
R40     408    0      3.300e+003
R41     418    410    1.200e+004
```

```
C9      416    412    2.700e-003
Rsprt_1         416    412    1E+9
Qn51    414    417    0      NBJT
R42     416    4      5.600e+004
Qn47    417    4      4      NBJT
Qp41    414    417    411    PBJT
Qp42    429    416    405    PBJT
R43     414    418    1.200e+004
R44     412    425    8.200e+004
L9      418    413    1.000e-003
Qn48    412    418    411    NBJT
L10     420    429    2.700e-003
Qn49    409    402    427    NBJT

**fifth pin 14el Best fitness 0.049285
R20     508    513    6.8e+4
R21     508    505    1.8e+3
R22     506    3      1.2e+3
R23     507    4      4.7e+2

L4      511    502    6.8e-1
L5      513    507    5.6e-2

Qn21    510    508    4      NBJT
Qn22    4      507    508    NBJT
Qn23    506    501    507    NBJT
Qn24    507    502    509    NBJT

Qp20    505    506    3      PBJT
Qp21    502    510    3      PBJT
Qn25    512    509    511    NBJT
Qp22    511    505    512    PBJT

**sixth pin 23el Best fitness 0.2
R24     605    607    1.8e+3
C6      3      602    1.8e-4
Qn26    606    605    4      NBJT
Qn27    4      609    605    NBJT
Qp23    602    606    3      PBJT
Qn28    613    616    3      NBJT
Qn29    608    601    607    NBJT
Qp24    608    609    612    PBJT
R25     609    614    6.8e+1
Qp25    608    610    612    PBJT
R26     606    3      1.8e+2
Qp26    609    611    4      PBJT
R27     607    617    1.2e+4
Qp27    611    612    602    PBJT
Qp28    615    616    614    PBJT
Qn30    612    0      607    NBJT
Qn31    3      612    606    NBJT
Qn32    0      613    0      NBJT
C7      602    611    2.7e-8
Qp29    611    615    618    PBJT
L6      602    617    6.8e-4
R28     0      618    1.8e+5
```

```
R29     610    4      1.8e+4


**seventh pin 22el Best fitness 0.7969
Qn33    3      706    709    NBJT
R30     707    709    8.2e+2
Qp30    3      705    708    PBJT
Qn34    708    707    4      NBJT
Qn35    705    701    706    NBJT
Qp31    711    4      712    PBJT
Qp32    710    708    3      PBJT
R31     713    705    2.2e+3
Qn36    3      709    4      NBJT
Qn37    709    711    705    NBJT
C8      719    714    8.2e-8
Qp33    702    713    710    PBJT
R32     718    709    4.7e+3
Qp34    0      711    713    PBJT
R33     702    718    2.2e+3
Qn38    709    719    4      NBJT
Qp35    719    706    702    PBJT
Qn39    717    4      717    NBJT
R34     716    714    1.5e+3
Qn40    716    716    3      NBJT
R35     712    715    1E+9
Qn41    715    717    718    NBJT

**eightth pin 8el Best fitness 0.089
R36     805    3      1.8e+1
Qn42    808    807    4      NBJT
Qn43    3      806    807    NBJT
Qn44    805    801    806    NBJT
Qp36    802    805    3      PBJT
Qp19    802    806    809    PBJT
L7      806    809    6.8e-3
L8      808    4      3.3e-1

.END
```

Appendix D – PSPICE Decks Netlisted From Other Works

D1 - The Netlisted Low-Pass Filter Circuit from [2]

```
**Low-Pass Filter

.OPTIONS NOOUTMSG NOBIAS RELTOL=0.05
.AC DEC 19 1 100000
.PLOT AC V(2)
.lib "nom.lib"
.PROBE

Vs              5       999     AC 2
Rload           999   0       1000
Rsource         0     2     1011
C0              0       7       19.277nF
C1              7       8       1.0816uF
R00             0       7       1e+10
R0              8       6       155.86k
R1              6       5       0.000001
R2              6       5       23.7677k
L0              5       71      282.327mH
R1              71      70      1e-7
C3              6       70      0.218046uF
C4              70      0       37.161pF
C5              70      3       84.0648nF
C6              3       0       1.299uF
R3              3       5       16.5447M

C7              5       2       0.13149nF
L2              2       70      477.403mH
L3              6       2       367.835mH
C8              6       2       23.419nF
C9              3       2       12.8176uF
L1              70      5       7152.78mH

.END
```

D2 - The Netlisted Ladder Low-Pass Filter Circuit from [12]

```
**LADDER Low-pass filter

.OPTIONS NOOUTMSG NOBIAS RELTOL=0.05
.AC DEC 19 1 100000
.PLOT AC V(2,0)
.PROBE

.lib "nom.lib"

Vs 999 0 AC 2
Rload   999     1       1000
Rsource 0       2       1000

L_5         1       3    9.68uH
L_10        3      4     182000uH
L_22        4      5     209000uH
L_28        5      6     209000uH
L_31        6      7     209000uH
L_25        7      8     209000uH
L_13        8      2     182000uH
C_12        3      0     86.1nF
C_24        4      0     202nF
C_30        5      0     202nF
C_3         6      0     202nF
C_33        7      0     202nF
C_27        8      0     202nF
C_15        2      0     86.1nF

.END
```

D3 - The Netlisted Bridge Low-Pass Filter Circuit from [12]

```
**Bridge-T Low-pass filter

.OPTIONS NOOUTMSG NOBIAS RELTOL=0.05
.AC DEC 19 1 100000
.PLOT AC V(2)
.lib "nom.lib"
.PROBE

Vs          0     999    AC 2
Rsour      999    1      1000
Rload       2     0      1000
L5          1     3      22400uH
C27         3     0      127nF
L25         3     4      229000uH
C18         4     0      127nF
C33         4     0      127nf
L31         4     5      229000uH
C24         5     0      127nF
C21         5     0      127nF
C12         5     0      0.338nF
L28         5     6      229000uH
C30         6     0      127nF
L14         6     2      214000uH
C3          6     7      118nF
L11         7     0      0.796uH
C15         7     2      118nF

.END
```

D4 - The Elliptic Low-Pass Filter Circuit from [12]

```
**Elliptic Low-Pass filter

.OPTIONS NOOUTMSG NOBIAS RELTOL=0.05
.AC DEC 14 1 100000
.PLOT AC V(1,0)
.PROBE
.lib "nom.lib"
Vs 999 0 AC 2
Rload   999     1       1000
Rsource 0       2       1000

L_0         1     3     51800uH
L_1         3     4     51800uH
L_2         4     6     51800uH
L_4         4     5     51800uH
C_3         5     0     136nF
L_5         6     7     51800uH
L_6         7     8     51800uH
L_7         8     9     51800uH
L_77        8     99    51800uH
C_4         99    0     136nF
L_8         9     10    51800uH
L_9         10    11    51800uH
L10         11    12    51800uH
L78         11    98    51800uH
C_44        98    0     136nF
L11         12    13    51800uH
L12         13    14    51800uH
L13         14    15    51800uH
L79         14    97    51800uH
C_5         97    0     136nF
L14         15    16    51800uH
L15         16    17    51800uH
L16         17    2     51800uH
L17         17    96    51800uH
C_444       96    0     136nF

.END
```

D5 - The Netlisted Squaring Circuit from [12]

```
**SQUIRING FUNCTION
.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.DC LIN Vs -.25 .25 .025
.PRINT DC V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0
PULSE -.25 .25 0 .2
Rsource     999   1     1K
Rload 0     2     1K
R30   1     3     2.9K
Qp15  3     3     4     PBJT
Qn20  5     6     3     NBJT
Qn37  7     3     0     NBJT
Qn21  4     4     8     NBJT
Qn68  9     10    4     NBJT
Qp34  0     4     11    PBJT
Qp19  12    5     996   PBJT
Qn12  996   6     13    NBJT
Qn13  6     6     0     NBJT
Qp29  14    6     0     PBJT
Qp88  6     15    0     PBJT
Qn26  7     16    2     NBJT
Qp23  8     2     17    PBJT
Qp66  0     9     18    PBJT
Qn81  996   10    19    NBJT
Qn110 996   10    19    NBJT
Qn45  996   11    17    NBJT
Qp54  11    20    996   PBJT
Qn10  996   12    21    NBJT
Qn11  16    21    13    NBJT
Qn41  996   14    0     NBJT
Qn40  996   15    14    NBJT
Qn104 17    22    22    NBJT
Qn71  18    23    24    NBJT
Qn59  25    19    0     NBJT
Qp35  19    19    0     PBJT
Qp55  20    26    27    PBJT
Qn77  26    23    22    NBJT
Qn101 996   23    28    NBJT
Qn115 30    29    23    NBJT
```

```
Qp84  23    31    0     PBJT
Qn80  996   996   24    NBJT
Qn49  32    32    24    NBJT
Qn65  33    25    34    NBJT
Qn96  996   32    26    NBJT
Qn107 34    33    27    NBJT
Qn119 996   28    29    NBJT
Qn70  996   30    31    NBJT
.END
```

D6 - The Netlisted Cubing Circuit from [12]

```
**CUBING FUNCTION

.OPTIONS SOLVER=0 STEPGMIN NOREUSE NOMOD
+NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0 PULSE -.25 .25 0 .2
Rsource     999   1     1K
Rload 0     2     1K

Qp94  3     4     1     PBJT
R77   1     5     29.7K
Qp41  1     6     7     PBJT
R33   3     7     1.39K
R92   3     8     32.2K
Qp95  3     9     0     PBJT
Qn101 4     10    995   NBJT
Qp103 4     11    0     PBJT
Qp76  5     7     0     PBJT
Qp30  12    5     0     PBJT
R14   5     13    291K
R39   6     14    44.3K
R91   6     15    1.39K
R10   7     2     4.84K
R23   7     16    74.2K
R107  8     9     5.29K
Qn108 8     8     0     NBJT
R117  10    17    11.5K
R121  11    17    30.2K
R93   11    18    30.2K
Qp49  19    12    0     PBJT
Qp63  11    996   996   PBJT
R29   12    19    13.6K
R68   12    20    13.6K
Qn36  12    21    995   NBJT
R15   13    16    30.2K
Qp22  13    13    22    PBJT
Qp9   13    13    22    PBJT
```

```
R12    13     22     36.7K
Qn17   14     23     995    NBJT
Qn87   15     24     995    NBJT
Qn24   16     16     0      NBJT
R119   17     18     0.241K
Qp114  18     996    996    PBJT
Qp88   19     26     25     PBJT
Qp70   27     20     996    PBJT
R74    20     21     0.204K
R59    21     28     0.000152K
Qn80   21     21     995    NBJT
Qn28   22     22     29     NBJT
Qp44   23     29     0      PBJT
Qp2    23     29     0      PBJT
Qp56   24     996    996    PBJT
Qp106  25     27     30     PBJT
R130   26     31     1.39K
Qn112  0      26     32     NBJT
Qp110  26     32     0      PBJT
Qn140  33     28     34     NBJT
Qp54   35     29     29     PBJT
Qn105  31     30     995    NBJT
Qn137  36     31     995    NBJT
Qn136  37     34     33     NBJT
R146   34     36     1.39K
Qp35   35     19     0      PBJT
Qn153  37     37     38     NBJT
Qn149  0      38     0      NBJT
.END
```

D7 - The Netlisted Square Root Circuit from [12]

```
**SQUIRE ROOT
.OPTIONS SOLVER=0 STEPGMIN NOREUSE
 NOMOD NOECHO NOOUTMSG NOBIAS NOPRBMSG NOPAGE
.TRAN .01 .20
.PRINT TRAN V(2)
.DC LIN Vs .0 .5 .025
.PRINT DC V(2)
.lib "nom.lib"
.MODEL NBJT NPN
.MODEL PBJT PNP
Vpos  3     0     DC    +15.0V
Vneg  4     0     DC    -15.0V
Vs 999 0 PULSE .0 .5 0 .2
Rsource     999   1     1K
Rload 0     2     1K

Qp9   995   1     3     PBJT
R13   3     996   0.696K
Qp38  4     5     3     PBJT
Qn20  2     4     6     NBJT
R102  5     996   39.4
R10   5     0     1.02K
Qp79  5     5     0     PBJT
Qn100 27    5     5     NBJT
Qp169 9     8     5     PBJT
Qp101 995   9     5     PBJT
Qn140 10    10    5     NBJT
Qn118 11    5     12    NBJT
Qp39  995   6     7     PBJT
Qn25  996   13    7     NBJT
Qn19  996   7     7     NBJT
R40   7     996   0.747K
Qn17  996   7     0     NBJT
Qn68  996   7     0     NBJT
Qn106 996   13    7     NBJT
R167  8     9     1.31K
Qp108 9     9     0     PBJT
Qp138 9     9     14    PBJT
Qp148 9     9     14    PBJT
R214  9     996   15.3K
Qn195 996   9     9     NBJT
R201  9     15    59.9K
Qn203 15    0     9     NBJT
Qp219 995   14    9     PBJT
Qn157 10    16    17    NBJT
Qp184 16    10    16    PBJT
Qp156 995   18    10    PBJT
Qn189 11    11    14    NBJT
Qn116 11    11    14    NBJT
Qp158 995   12    19    PBJT
Qp81  995   996   13    PBJT
```

```
R82    13     996    0.696K
R121   996    14     0.51K
Qp120  995    14     18     PBJT
Qp245  14     14     0      PBJT
Qn256  14     20     995    NBJT
R247   14     0      17.6K
Qp207  995    14     14     PBJT
Qp208  995    21     14     PBJT
Qp155  995    18     17     PBJT
Qp228  18     996    996    PBJT
R159   19     996    4.25K
Qn64   996    19     22     NBJT
R209   20     21     9.19K
Qn144  996    21     0      NBJT
Qn163  24     22     23     NBJT
Qp206  23     25     25     PBJT
Qn123  996    24     996    NBJT
Qp122  995    996    25     PBJT
R49    996    0      0.0631K
R36    27     996    1.32K
Qn261  996    0      0      NBJT
Qp263  28     29     996    PBJT
R235   29     0      72.9K
R264   28     0      72.9K
R153   0      995    1.03K
Qp258  995    995    0      PBJT
Qp211  995    0      0      PBJT
R75    31     0      59.9K
Qn77   31     0      0      NBJT

.END
```

D8 - The Netlisted Low-Pass Filter Circuit from [13]

```
**Low-pass filter

.AC DEC 14 1 100000
.PLOT AC V(1,0)
.lib
.PROBE

Vs 999 0 AC 2
Rload   999     1       1000
Rsource 0       2       1000

L_9      1       8       3.300694e-001
L_5      1       777     2.113217e-001
R_6      777     7       1.000000e-006
L_4      1       6       2.008357e-001
C_6      6       7       1.261844e-004
C_1      1       800     1.083701e-004
C_2      800     3       3.547581e-005
L_3      3       4       7.713100e-001
C_4      4       6       2.519509e-005
R_2      6       7       3.648645e+004
L_8      7       5       0.659257
R_5      5       8       1.000000e-006
L_11     8       9       0.25536662
L_13     9       10      0.22130646
R_3      10      2       1.000000e-006
C_3      800     3       9.862769e-005
R_1      800     3       1.042709e+004
L_2      800     3       5.285780e-003
L_1      1       3       2.147189e-001
L_6      6       7       1.017696e-003
C_5      6       0       1.048100e-007
C_7      8       0       2.449500e-007
C_8      9       0       2.506300e-007
C_9      10      0       1.256400e-007

.END
```

D9 - The Netlisted Low-Pass Filter Circuit from [17]

```
**Low-Pass filter

.OPTIONS NOOUTMSG NOBIAS
.AC DEC 19 1 100000
.PLOT AC V(2,0)
.lib "nom.lib"
.PROBE

Vs 999 0 AC 2
Rload   999     1       1000
Rsource 0       2       1000

C_12      1      0       5.80000e-008
L_1       1      3        0.082
L_3       3      4       0.0058
L_2       4      5       0.015
C_1       1      5       6.7e-008
C_10      5      0       1.8e-007
C_8       5      2       1.0e-008
L_4       5      6       0.088
L_5       6      2       0.082
C_9       6      0       1.0e-009
C_15      2      0       1.2e-008
C_16      2      0       1.0e-007

.END
```

D10 - The Netlisted Low-Pass Filter Circuit from [28]

```
**Low-Pass Filter

.OPTIONS NOOUTMSG NOBIAS RELTOL=0.05
.AC DEC 14 1 100000
.PLOT AC V(1,0)
.PROBE

.lib "nom.lib"
**.IC V(1)=0 V(2)=0 V(999)=2

Vs 999 0 AC 2
Lload   999     1       1H
Csource 0       2       10u
L1      1       3       4.7m
C1      1       3       0.56u
R1      999     3       22
L2      1       2       47u
R2      3       2       6800
C2      3       0       3.9u
L3      3       4       22m
C3      4       0       0.82u
.END
```

D11 - The Netlisted Low-Pass Filter Circuit from [102]

```
**Chebyshev Low-Pass Filter

.OPTIONS NOOUTMSG NOBIAS
.AC DEC 19 1 100000
.PLOT AC V(2,0)
.lib "nom.lib"


Vs 999 0 AC 2
Rload 999   1     1000
Rsource     0     2     1000

L1    1     4     0.130443
C1    4     0     2.28690e-007
L2    4     5     0.289535
C2    5     0     2.7551300e-007
L3    5     6     0.308156
C3    6     0     2.7995400e-007
L4    6     7     0.30327
C4    7     0     2.6303500e-007
L5    7     2     0.251735
C5    2     0     1.1850700e-007

.END
```

Bibliography

[1] J. Williams, *Analog Circuit Design — Art, Science and Personalities*, Butterworth–Heinemann, MA, 1991.

[2] H. Shibata, "Computer aided design of analog circuits based on genetic algorithms," *Ph.D. dissertation*, Tokyo, Tokyo Inst. Technol., Japan, 2001.

[3] Hugo de Garis, "Review of Proceedings of the First NASA/DoD Workshop on Evolvable Hardware, IEEE Transactions on Evolutionary Computation (IEEE-TEC), Nov 1999, vol. 3, no. 4.

[4] G. Van der Plas, G. Debyser, F. Leyn, K. Lampaert, J. Vandenbussche, G. E. Gielen,W. Sansen, P. Vaselinovic and D. Leenaerts, "AMGIE—A synthesis environment for CMOS analog integrated circuits," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 20, no. 9, Sep. 2001, pp. 1037-1058.

[5] P. Veselinovic, D. Leenaerts, W. Bokhoven, F. Leyn, F. Proesmans, G. Gielen and W. Sansen, "A flexible topology selection program as part of an analog synthesis system," in *Proc. Eur. Design Test Conf.*, Mar. 1995, pp. 119-123.

[6] F. El-Turkey and E. E. Perry, "BLADES: An artificial intelligence approach to analog circuit design," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 8, no. 6, Jun. 1989, pp. 680-692.

[7] R. Harjani, R. A. Rutenbar and L. R. Carley, "OASYS: A framework for analog circuit synthesis," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 8, no. 12, Dec. 1989, pp. 1247-1266.

[8] E. S. Ochotta, R. A. Rutenbar and L. R. Carley, "Synthesis of high-performance analog circuits in ASTRX/OBLX," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 15, no. 3, Mar.1996, pp. 273-294.

[9] G. G. E. Gielen, H. C. C. Walscharts and W. M. C. Sansen, "Analog circuit design optimization based on symbolic simulation and simulated annealing," in *IEEE J. Solid-State Circuits*, vol. 25, no. 3, Jun. 1990, pp. 707-713.

[10] R. Phelps, M. Krasnicki, R. A. Rutenbar, L. R. Carley and J. R. Hellums, "Anaconda: Simulation based synthesis of analog circuits via stochastic pattern search," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 19, no. 6, Jun. 2000, pp. 703-717.

[11] M. Krasnicki, R. Phelps, R. A. Rutenbar and L. R. Carley, "MAELSTROM: An efficient simulation based synthesis for custom analog cells," in *Proc. 36th Design Autom. Conf.*, 1999, pp. 945-950.

[12] J. Koza, F. Bennett III, D. Andre, A. Keane and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," in *IEEE Trans. Evol. Comput.*, vol. 1, no. 2, Jul. 1997, pp. 109-128.

[13] J. Lohn and S. Colombano, "A circuit representation technique for automated circuit design," in *IEEE Trans. Evol. Comput.*, vol. 3, no. 3, Sep. 1999, pp. 205-219.

[14] M. Davis, L. Liu and J. G. Elias, "VLSI circuit synthesis using a parallel genetic algorithm," in *Proc. 1st IEEE Conf. Evol. Comput.*, vol. 1, 1994, pp. 104-109.

[15] T. Sripramong and C. Tomazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," in *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 21, no. 11, Nov. 2002, pp. 1237-1252.

[16] G. Alpaydin, S. Balkir and G. Dundar, "An evolutionary approach to automatic synthesis of high-performance analog integrated circuits," in *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, Jun. 2003, pp. 240-252.

[17] C. Goh and Y. Li, "GA automated design and synthesis of analog circuits with practical constraints," in *Proc. Congr. Evol. Comput.*, vol. 1, 2001, pp. 170-177.

[18] J. Grimbleby, "Automatic analogue circuit synthesis using genetic algorithms," in *Proc. Inst. Elect. Eng. Circuits, Devices, Syst.*, vol. 147, 2000, pp. 319-323.

[19] J. H. Holland, *Adaption in Natural and Artificial Systems*, The University of Michigan Press, Ann Arbor, 1975.

[20] Churchill F. B., "William Johannsen and the genotype concept," in *Journal of the History of Biology*, 7, 1974, pp. 5-30.

[21] J. R. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, The MIT Press, Cambridge, MA, 1992.

[22] I. Rechenberg, Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution, *PhD dissertation*, Technical University of Berlin, 1970.

[23] J. Hu, "Sustainable Evolutionary Algorithms and Scalable Evolutionary Synthesis of Dynamic Systems," *PhD. Dissertation*, East Lansing: Michigan State University, 2004.

[24] J. Koza, "Automated synthesis of computational circuits using genetic programming," in 1997 *IEEE Conference on Evolutionary Computation*, 1997, pp. 447-452.

[25] H. de Garis, "Genetic Programming: GenNets, Artificial Nervous Systems, Artificial Embryos," *PhD. thesis*, Brussels University, 1992.

[26] X. Yao, "A review of evolutionary artificial neural networks", in *International Journal of Intelligent Systems*, vol. 8 no. 4, (1993a), pp. 539-567.

[27] P. K. Lehre and M. Hartmann, "Development and complexity-based fitness function modifiers," in *Workshop on Regeneration and Learning in Developmental Systems*, at GECCO'04, June 2004.

[28] R. S. Zebulum, M.A. Pacheco and M. Vellasco, "Comparison of different evolutionary methodologies applied to electronic filter design," in *IEEE Trans Conf. on Evolutionary Computation*, Piscataway, NJ: IEEE Press, 1998, pp. 434-439.

[29] I. Harvey, "The Artificial Evolution of Adaptive Behaviour," *D.Phil. thesis*, School of Cognitive and Computing Science, University of Sussex, Brighton, Sussex, England, 1993.

[30] M. Ridley, *Evolution*, Blackwell Science, Cambridge, Massachusetts, 1993.

[31] E. Guerra-Gomez, Tlelo-Cuautle, L. G. de la Fraga, T. McConaghy and G. G. E. Gielen, "Sizing Mixed-Mode Circuits by Multi-Objective Evolutionary Algorithms," in *Proc. IEEE Midwest Symposium on Circuits and Systems* (MWSCAS), 2010.

[32] E. Zitzler, M. Laumanns and S. Bleuler, "A Tutorial on Evolutionary Multi-Objective Optimization," in *Metaheuristics for Multi-Objective Optimisation*, Springer, 2004, pp. 3-37.

[33] Y. Sapargaliyev and T. Kalganova, "On Comparison of Constrained and Unconstrained Evolutions in Analogue Electronics on the Example of "LC" Low-Pass Filters," in *IEICE transactions on Electronics*, vol. E89-C no. 12, pp. 1920-1927

[34] T. Bäck, "The interaction of mutation rate, selection, and adaptation within genetic algorithm," in *Parallel Problem Solving from Nature 2*, R. Männer and B. Manderick (eds.), Amsterdam, The Netherlands: Elsevier, 1992, pp. 85-94.

[35] E. Stomeo, T. Kalganova and C. Lambert, "Mutation Rate for Evolvable Hardware," in *International Conference on Computational Intelligence -ICCI 2005*, Prague, Czech Republic, Aug. 2005, pp. 117-124.

[36] I. Harvey, "Species Adaptation Genetic Algorithms: A basis for a continuing SAGA," in *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, F. J. Varela and P. Bourgine (eds.), MIT Press/Bradford Books, Cambridge, MA, 1992, pp. 346-354.

[37] T. Kuo and S-H. Hwang, "Using disruptive selection to maintain diversity in genetic algorithms," *Appl. Intel. 7*, 1997, pp. 257-267.

[38] A. Thompson, "Hardware Evolution: Automatic design of electronic circuits in reconfigurable hardware by artificial evolution," *Distinguished dissertation series*, Springer-Verlag, 1996.

[39] D. Keymeulen, G. Klimeck, R. Zebulum, A. Stoica and C. Salazar-Lazaro, "EHWPack: A Parallel Software/Hardware Environment for Evolvable Hardware," in Whitley Darrell (ed.), *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO- 2000), July 8-12, 2000, Las Vegas, Nevada USA.

[40] A. Stoica, R. Zebulum and D. Keymeulen, "Mixtrinsic evolution," in *Int. Conf. Evolvable Systems*, Edinburgh, UK, Apr. 2000, pp. 208-217.

[41] T. G.W. Gordon and P. J. Bentley, "On Evolvable Hardware," in *Soft Computing in Industrial Electronics*, Physica-Verlag, Heidelberg, Germany, pp. 279-323.

[42] A. B. Williams and F. J. Taylor, *Electronic Filter Design Handbook*, Third Edition, New York, NY: McGraw-Hill, 1995.

[43] C. Mattiussi and D. Floreano, "Analog Genetic Encoding for the Evolution of Circuits and Networks," in *IEEE Trans. on Evolutionary Computation*, vol. 11, 2007, pp. 596-607.

[44] T. McConaghy, G. G. E. Gielen, "Globally reliable variation-aware sizing of analog integrated circuits via response surfaces and structural homotopy," in *IEEE Transactions on Computer-Aided Design*, vol. 28, no. 11, Nov. 2009, pp. 1627-1640.

[45] J. Hu, X. Zhong and E. Goodman, "Open-ended Robust Design of Analog Filters Using Genetic Programming," in *Genetic & Evolutionary Computation Conference*, ACM Press, vol. 2, Washington, DC, 2005, pp. 1619-1626.

[46] J. Koza, "Genetic Programming II: Automatic Discovery of Reusable Programs," MIT Press, Cambridge, MA, 1994.

[47] F. Wang, L. Yuanxiang, L. Kangshun and L. Zhiyi, "A New Circuit Representation Method for Analog Circuit Design Automation," in *IEEE World Congress on Computational Intelligence*, IEEE Press, Hong Kong, 2008.

[48] T. Kalganova, "Bidirectional incremental evolution in evolvable hardware," in *Proc. 2nd NASA/DoD Workshop Evolvable Hardware*, Jul. 2000, pp. 65-74.

[49] J. Walker, K. Völk, S. Smith and J. F. Miller, "Parallel evolution using multi-chromosome Cartesian genetic programming," in *Genetic Programming and Evolvable Machines*, vol. 10, no. 4, 2009.

[50] J. F. Miller and K. Downing, "Evolution in materio: Looking beyond the silicon box," in *NASA/DOD Conference on Evolvable Hardware*, IEEE Comp. Soc. Press, 2002, pp. 167-176.

[51] A. Thompson, "An evolved circuit, intrinsic in silicon, entwined with physics," in T. Higuch, M. Iwata and W. Liu, (eds.), *Proc. of The 1st Int. Conf. on Evolvable Systems: From Biology to Hardware*, LNCS, vol. 1259, Springer-Verlag, 1997, pp. 390-40.

[52] F. Bennett III, J. Koza, M. Keane, J. Yu, W. Mydlowec and O. Stiffelman, "Evolution by Means of Genetic Programming of Analog Circuits That Perform Digital Functions," in *GECCO-99: Proceedings of the Genetic and Evolutionary Computation Conference*, July 13–17, 1999, pp. 1477-1483

[53] T. McConaghy, P. Palmers, G. Gielen and M. Steyaert, "Variation-aware structural synthesis of analog circuits via hierarchical building blocks and structural homotopy," in *IEEE Transactions on Computer-Aided Design,* vol. 28, no. 9, Sep. 2009, pp. 1281-1294.

[54] K. Kim, A. Wong and H. Lipson, "Automated Synthesis of Resilient and Tamper-Evident Analog Circuits without a Single Point of Failure," *Genetic Programming and Evolvable Machines (online)*, 2009.

[55] A. Thompson and P. Layzell, "Evolution of Robustness in an Electronics Design," in *International Conference on Intelligent Engineering Systems*, Springer, Berlin/Heidelberg, Germany, 2000, pp. 218-228.

[56] A. Stoica, D. Keymeulen, T. Arslan, V. Duong, R. Zebulum, I. Ferguson and X. Guo, "Circuit Self-Recovery Experiments in Extreme Environments," *Proceedings of the 2004 NASA/DoD Conference on Evolvable Hardware*, Seattle, USA, June, 2004, pp. 142-145.

[57] J. Walker, J. Hilder and A. Tyrrell, "Towards Evolving Industry-feasible Intrinsic Variability Tolerant CMOS Designs," *11th IEEE Congress on Evolutionary Computation,* Trondheim, Norway, May, 2009, pp. 1591-1598.

[58] J. Koza, L. Jones, M. Keane, M. Streeter and S. Al-Sakran, "Toward automated design of industrial-strength analog circuits by means of genetic programming," in *Genetic Programming Theory and Practice II*, Chapter 8, Boston: Kluwer Academic Publishers, 2004, pp. 121-142.

[59] J. D. Lohn, G. L. Haith, S. P. Colombano and D. Stassinopoulos, "A Comparison of Dynamic Fitness Schedules for Evolutionary Design of Amplifiers," in *Proc. of the NASA/DoD Workshop on Evolvable Hardware*, IEEE Computer Society, CA, USA, 19-21 July, 1999, pp. 87-92.

[60] S. G. Ficici, R. A. Watson and J. B. Pollack, "Embodied Evolution: A Response to Challenges in Evolutionary Robotics," in *Proc. of the 8th European Workshop on Learning Robots*, 1999, pp. 14-22.

[61] A. E. Eiben, Z. Michalewicz, M. Schoenauer and J. E. Smith, "Parameter Control in Evolutionary Algorithms," in F. G. Lobo, C. F. Lima and Z. Michalewicz, (eds.), *Parameter Setting in Evolutionary Algorithms*, Chapter 2, Springer Verlag, 2007, pp. 19-46.

[62] F. Herrera and M. Lozano, "Adaptive genetic operators based on co-evolution with fuzzy behaviors," in *IEEE Trans. Evolutionary Computation 5*, vol. 2, 2001, pp. 149-165.

[63] J. Koza and D. Andre, "Evolution of both the architecture and the sequence of work-performing steps of a computer program using genetic programming with architecture-altering operations," in P. Angeline and K. Kinnear, (eds.), *Advances in Genetic Programming*, vol. 2, 1996.

[64] H.-G. Beyer and H.-P. Schwefel, "Evolution Strategies: A Comprehensive Introduction," in *Journal Natural Computing*, Vol. 1, no. 1, 2002, pp. 3-52.

[65] Y. Sapargaliyev and T. Kalganova, "Automated Synthesis of 8-Output Voltage Distributor using Incremental Evolution," in *Proc. of 2010 NASA/ESA Conference on Adaptive Hardware and Systems*, IEEE, Jun. 15-18, 2010, pp. 186-193.

[66] W. B. Langdon. and R. Poli, *Foundations of Genetic Programming*, Springer-Verlag, ISBN 3-540-42451-2, 2002.

[67] J. Hu, E. D. Goodman, K. Seo and M. Pei, "Adaptive Hierarchical Fair Competition (AHFC) Model for Parallel Evolutionary Algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO), New York, 2002, pp. 772-779.

[68] R. Lohmann, "Application of Evolution Strategy in Parallel Populations," in *Parallel Problem Solving from Nature - Proceedings of 1st Workshop*, (PPSN 1), vol. 496 of Lecture Notes in Computer Science, pp. 198-208, edited by: Schwefel HP, Mäanner R. Berlin, Germany, Springer-Verlag; 1991.

[69] A. Ngom, "Parallel evolution strategy on grids for the protein threading problem," *Journal of Parallel and Distributed Computing*, vol. 66, no. 12, 2006, pp. 1489-1502.

[70] L. Jostins, "A Comparison of Parallel Global Optimisation Algorithms for Reverse Engineering Gene Networks," *MPhil thesis*, University of Cambridge, August 21, 2008.

[71] J. Lohn, W. Kraus and G. Haith, "Comparing a co-evolutionary genetic algorithm for multi-objective optimization," in *Proceedings of the Evolutionary Computation*, CEC'02. Proceedings of the 2002 Congress, May 12-17, 2002, pp. 1157-1162.

[72] W. Johannsen, "The Genotype Conception of Heredity," in *American Naturalist*, vol. 45, no. 531, 1911, pp. 129-159.

[73] J. Lohn, A. Stoica, D. Keyeulen and S. Colombano, *the Second NASA/DoD Workshop on Evolvable Hardware, IEEE Transactions on Evolutionary Computation*, June 2001, 5(3), pp. 298-302.

[74] Z. Michalewicz, R. Hinterding and M. Michaelewicz, *Evolutionary algorithms, Fuzzy evolutionary computation*, Kluwer Academic Publishers, Norwell, MA, 1997.

[75] J. Torresen, "A Divide-and-Conquer Approach to Evolvable Hardware," in M. Sipper, D. Mange and A. Pérez-Uribe, (eds.), *ICES 1998*, LNCS, vol. 1478, Springer, Heidelberg, 1998, pp. 57-65.

[76] J. Wang, K. Je, Y. Lee and H. Chong, "Using Reconfigurable Architecture-Based Intrinsic Incremental Evolution to Evolve a Character Classification System," in Y. Hao, J. Liu, Y-P. Wang, Y-M. Cheung, H. Yin, L. Jiao, J. Ma and Y-C Jiao, (eds.), CIS 2005. LNCS (LNAI), vol. 3801, Springer, Heidelberg, 2005, pp. 216-223.

[77] R. Zebulum, M. S. Vellasco and M. Pacheco, "Variable Length Representation in Evolutionary Electronics," in *Evolutionary Computation*, vol. 8 no. 1, March 2000, pp. 93-120.

[78] S. Ando and H. Iba, "Analogue Circuit Design with a Variable Length Chromosome," in *Congress on Evolutionary Computation*, IEEE Press, 2000, pp. 994-100.

[79] J. B. Grimbleby, "Hybrid genetic algorithms for analogue network synthesis," in *Congress on Evolutionary Computation*, Washington USA, 1999, pp. 1781-1787.

[80] Z. Gan, Z. Yang, G. Li and M. Jiang, "Automatic Synthesis of Practical Passive Filters Using Clonal Selection Principle-Based Gene Expression Programming," in *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, ISSN 0302-9743, vol. 4684, 2007, pp. 1611-3349 (Online).

[81] S. Chang, H. Hou and Y. Su, "Automated Passive Filter Synthesis Using a Novel Tree Representation and Genetic Programming," in *IEEE Trans. on Evolutionary Computation*, vol. 10, 2006, pp. 93-100.

[82] A. Thompson, "Artificial evolution in the physical world," in Gomi, (ed.), *Evolutionary Robotics*, AAI Books, 1996.

[83] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 11, 2002, pp. 1237-1252.

[84] H. Shibata and N. Fujii, "Automated design of analog computational circuits using cell-based structure," in *Circuits and Systems, ISCAS 2002. IEEE International Symposium*, vol. 2, May 26-29, 2002, pp.53-56.

[85] X. Xia, Y. Li, W. Ying and L. Chen, "Automated design approach for analog circuit using genetic algorithm," in *Lect. Notes Comput. Sci.*, 2007, pp. 1124-1130.

[86] T. Dastidar, P. Chakrabarti and P. Ray, "A synthesis system for analog circuits based on evolutionary search and topological reuse," in *IEEE Trans. on Evolutionary Computation*, vol. 9, no. 2, 2005, pp. 211-224.

[87] W. Kruiskamp and D. Leenaerts, "Darwin: CMOS opamp synthesis by means of a genetic algorithm," in *Proc. 36th Design Autom. Conf.*, 1995.

[88] T. Menzies, "21st century AI: proud, not smug," *IEEE Intelligent Systems, Special Issue on AI Pride*, 2003, Available from http://menzies.us/pdf/03aipride.pdf.

[89] A. Das and R. Vemuri, "An Automated Passive Analog Circuit Synthesis Framework using Genetic Algorithms," in *IEEE Computer Society Annual Symposium on VLSI,* 2007, pp. 145-152.

[90] W. Feng, L. Yuanxiang, L. Kangshun and L. Zhiyi, "A New Circuit Representation Method for Analog Circuit Design Automation," in *IEEE World Congress on Computational Intelligence*, IEEE Press, Hong Kong, 2008.

[91] PSpice A/D Reference Guide, includes PSpice A/D, PSpice A/D Basics, and Pspice. Product Version 15.7, Cadence, July 2006.

[92] M. J. Streeter, M. A. Keane and J. R. Koza, "Use of genetic programming for automatic synthesis of post-2000 patented analog electrical circuits and patentable controllers," in S. Hernandez., C. A. Brebbia and M. E. M. El-Sayed, (eds.), *Computer Aided Optimum Design of Structures VIII*, Southampton, UK: WIT Press, 2003, pp. 35-44.

[93] P. Layzell, "Hardware Evolution: On the Nature of Artificially Evolved Electronic Circuits," *PhD thesis*, School of Cognitive and Computing Science, University of Sussex, Sussex, 2001.

[94] J. Lohn and S. Colombano, "Automated Analog Circuit Synthesis using a Linear Representation," in *Proceedings of the 2nd Int'l Conference on Evolvable Systems: From Biology to Hardware*, Springer-Verlag, Berlin, 1998, pp. 125-133.

[95] J. Shapiro, A. Prügel-Bennett and M. Rattray, "A statistical mechanical formulation of the dynamics of genetic algorithms," in *Lecture Notes in Computer Science*, vol. 865, pp. 17 - 27, 1994.

[96] Y. Sapargaliyev and T. Kalganova, "Absolutely free extrinsic evolution of passive low-pass filter," in *IEEE Canadian Conference on Electrical and Computer Engineering*, May 7-10, 2006, Ottawa, Canada.

[97] T. Kuo and S-H. Hwang, "Using disruptive selection to maintain diversity in genetic algorithms," *Appl. Intel.* 7, 1997, pp. 257-267.

[98] W. Mydlowec and J. Koza, "Use of Time-domain Simulations in Automatic Synthesis of Computational Circuits Using Genetic Programming," in *Genetic and Evolutionary Computation Conference*, Late Breaking Papers, Las Vegas, Nevada, 2000, pp. 187-197.

[99] M. Streeter, M. Keane and J. Koza, "Iterative Refinement of Computational Circuits Using Genetic Programming," in *Proc. of Genetic and Evolutionary Computation Conference*, San Francisco, CA: Morgan Kaufmann, 2002, pp. 877-884.

[100] M. Brameier, "On Linear Genetic Programming", *PhD thesis*, University of Dortmund, Dortmund, Germany, February 2004.

[101] K. Vesselin and J. Miller, "The advantages of landscape neutrality in digital circuit evolution," in *Proc. of 3rd International Conference on Evolvable Sytems (ICESO3)*, Lecture Notes in Computer Science, Springer, 2000, pp. 252-263.

[102] LC filter design, http://www-users.cs.york.ac.uk/~fisher/lcfilter/, accessed March 31, 2011.

[103] Y. Sapargaliyev and T. Kalganova, "Constrained and Unconstrained evolution of "LCR" low-pass filters with oscillating length representation," in *IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, July 16-21, 2006, pp. 1529-1536.

[104] T. Dastidar, P. Chakrabarti and P. Ray, "A synthesis system for analog circuits based on evolutionary search and topological reuse," in *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, April 2005, pp. 211-224

[105] W. Kruiskamp and D. Leenaerts, "DARWIN: Analogue circuit synthesis based on genetic algorithms," in *Int. J. Circ. Theor. Appl*., vol. 23, 1996, pp. 285-296.

[106] Y. Sapargaliyev and T. Kalganova, "Unconstrained Evolution of Close-to-ideal "LCR" Low-pass Filter," in *INES 2006, 10th International Conference on Intelligent Engineering Systems*, June 26-28, 2006, London Metropolitan University, London, UK, pp.145-150.

[107] V. K. Vassilev and J. F. Miller, "Scalability problems of digital circuit evolution evolvability and efficient designs," *Proceedings of the Second Proceedings NASA/DoD Workshop on Evolvable Hardware, 2000*, IEEE Computer Society, July 13-15, 2000, pp. 55-64.

[108] S. Cipriani and A. Takeshian, "Compact Cubic Function Generator," US patent 6,160,427. Filed September 4, 1998, Issued December 12, 2000.

[109] Y. Sapargaliyev and T. Kalganova, "Unconstrained evolution of analogue computational "QR" circuit with oscillating length representation," *The 8-th International Conference on Evolvable Systems: from Biology to Hardware, ICES 2008*, September 21-24, Prague, Czech Republic, pp. 1-10.

[110] H, Hemmi, J. Mizoguchi and K. Shimohara, "Development and evolution of hardware behaviours",in Sanchez, E., & Tomassini, M. (Eds.), *Towards Evolvable Hardware: The evolutionary engineering approach*, vol. 1062 of *LNCS*, 1996, pp. 250–265. Springer-Verlag.

[111] X. Yao and T. Higuchi, "Promises and challenges of evolvable hardware," in *IEEE Trans. on Systems, Man, and Cybernetics*, Part C 29, no. 1, 1999, pp. 87-97.

[112] R. L. Graham, M. Groetschel and L. Lovász, (eds.), *Handbook of Combinatorics*, Volumes 1 and 2. Elsevier (North-Holland), Amsterdam, and MIT Press, Cambridge, Mass, 1996.

[113] F. E. Bloom, Edited by Bloom F.E., Kupfer D.J. New York, "Introduction to preclinical neuropsychopharmocology," *Psychopharmocology: The Fourth Generation of Progress*, Raven, 1995, pp. 1-7.

[114] D. Hubel, "Eye, Brain, and Vision," in *Scientific American Library*, New York, 1988. Available: http://hubel.med.harvard.edu/book/bcontex.htm.

[115] Y. Sapargaliyev and T. Kalganova, "Challenging the Evolutionary Strategy to Synthesis Analogue Computational Circuits," in *Journal of Software Engineering and Applications*, vol. 3, no. 11, , Scientific Research Publishing, USA, November 2010, 2010, pp.1032-1039.

[116] Y. Sapargaliyev and T. Kalganova, "Open-Ended Evolution to Discover Analog Circuits for Beyond Conventional Applications," Accepted for the *Springer journal of Genetic Programming and Evolvable Machines*, Manuscript submitted on November 22, 2011.

[117] J. B. Grimbleby, "Automatic analogue network synthesis using genetic algorithms," in *Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems*, GALESIA 95, Sheffield, 12-14 September 1995, IEE, pp. 53-58.

[118] Feodosia state Optical Factory, The Artillery Quantum rangefinder, http://fkoz.feodosia.com.ua/main3.phtml?link=23, accessed March 23, 2011.

[119] E. Stomeo and T. Kalganova, "Improving EHW performance introducing a new decomposition strategy," in *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, Singapore, December 2004.

[120] T. Kuyucu, M. A. Trefzer, J. F. Miller and A. M. Tyrrell, "Task Decomposition and Evolvability in Intrinsic Evolvable Hardware," in *Proceedings of the IEEE Congress on Evolutionary Computation*, CEC 2009, Trondheim, Norway, May 2009.

[121] M. Trefzer, J. Langeheine, J. Schemmel and K. Meier, "Operational Amplifiers: An Example for Multi-Objective Optimization on an Analog Evolvable Hardware Platform," in J. M. Moreno, J. Madrenas and J. Cosp, (eds.), *Evolvable Systems: From Biology to Hardware*, Sixth International Conference, ICES 2005, no. 3637 in LNCS, Sitges, Spain, September 2005. Springer-Verlag, pp. 86-97.

[122] Mark Owen, *Practical signal processing*, Cambridge University Press, ISBN 9780521854788, 2007.

[123] T. G. Gordon and P. J. Bentley, "Development brings scalability to hardware evolution," in *Proceedings of the 2005 NASA/DoD Conference on Evolvable Hardware*, 2005, pp. 272-279.

[124] H. Liu, J. Miller and A.M. Tyrell, "A biological development model for the design of robust multiplier," in F. Rothlaf (Ed.), *Evo Workshops 2005, Lecture Notes in Computer Science*, vol. 3449, Springer-Verlag, pp. 195-204.

[125] E. Schlessinger, P. J. Bentley and R. Beau Lotto, "Investigating the Emergence of Multicellularity Using a Population of Neural Network Agents," in *Conference: Parallel Problem Solving from Nature*, pp. 711-720.

[126] T. Soule and J. A. Foster, "Effects of code growth and parsimony pressure on populations in genetic programming," in *Evolutionary Computation*, Vo. 6, no. 4, 1998, pp. 293-309.

[127] W.B. Langdon, T. Soule, R. Poli and J.A. Foster, ed. by L. Spector et al., "The evolution of size and shape," in *Advances in Genetic Programming 3*, MIT Press, Cambridge, MA, 1999, pp. 163-190.

[128] S. Luke, "Code growth is not caused by introns," in *Late Breaking Papers at GECCO-2000*, 2000, pp. 228–235.

[129] J. Hesser and R. Manner, "Towards an optimal mutation probability for genetic algorithms," in *PPSN, ser. Lecture Notes in Computer Science*, H.-P. Schwefel and R. Manner, Eds., vol. 496, Springer, 1990, pp. 23-32.

[130] J. Cervantesand C. R. Stephens, "Optimal mutation rates for genetic search," in *Proceedings of the 2006 Genetic and Evolutionary Computation Conference*, Washington, USA, July 2006, pp. 1313 - 1320.

[131] J. N. Richter, "On Mutation and Crossover in the Theory of Evolutionary Algorithms, *PhD dissertation*, Montana State University, 2010.

[132] A. Eiben and J. Smith, "Introduction to Evolutionary Computing," Natural Computing Series, Springer, 2010.

[133] Davis, L. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[134] S. Manos, L. Poladian, P. J. Bentley and M. Large, "Photonic device design using Multi-objective Evolutionary Algorithms," in *Proc. of Third International Conference on Evolutionary Multi-Criterion Optimization (EMO'05)*, 2005.

[135] P. J. Bentley, and J. P. Wakefield, "Finding Acceptable Pareto-Optimal Solutions using Multi-objective Genetic Algorithms." Submitted to *Soft Computing*, Springer Verlag Ltd. Research Report RN/98/66, 1998.

[136] K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, "A Fast Elitist Nondominated Sorting Genetic Algorithm for Multi-objecive Optimisation: NSGA-II," in *Parallel Problem Solving from Nature VI Conference*, 849-858, Springer, 2000.

[137] P. Adamidis, "Parallel evolutionary algorithms: a review," in *Proc. of the 4th Hellenic-European Conference on Computer Mathematics and its Applications*, 1998.

[138] A. Jain and D. B. Fogel, "Case studies in applying fitness distributions in evolutionary algorithms. II. Comparing the improvements from crossover and Gaussian mutation on simple neural networks," in X. Yao and D. B. Fogel, eds., *Proc. Of the 2000 IEEE Symposium on Combinations of Evolutionary Computation and Neural Networks*, IEEE Press, 2000, pp. 91-97.

[139] J. Torresen, "A divide-and-conquer approach to evolvable hardware," in *Evolvable Systems: From Biology to Hardware. Second International Conference*, ICES 98, LNCS 1478, Springer-Verlag, 1998, pp. 57-65.

[140] O. Aaserud, I. Nielsen and I. Ring, "Trends in current analog design: A panel debate," in *Analog Integrated Circuits and Signal Processing*, vol. 7, no. 1, 1995, pp. 5-9.

[141] C. G. Langton, "Computation at the edge of Chaos: Phase-Transitions and Emergent Computation," *PhD Thesis*, University of Michigan, 1990.

[142] T. Froese, "Sociality and the Life-Mind Continuity Thesis: A Study in Evolutionary Robotics," *PhD thesis*, Department of Informatics, University of Sussex, UK, June 2009.

[143] J. F. Miller, P. Tomson, "Cartesian genetic programming," in *Proc. of the European Conference on Genetic Programming*, 2000, pp. 121-132.

[144] D. Pescovitz, "1972: The release of SPICE, still the industry standard tool for integrated circuit design," *Lab Notes: Research from the Berkeley College of Engineering*, http://www.coe.berkeley.edu/labnotes/0502/history.html, 2002-05-02, Retrieved 2011-20-10

[145] J. N. Babanezhadand G. C. Temes, "Analog MOS Computational Circuits," in *Proceedings of the IEEE Circuits and System International Symposium*, Piscataway, NJ: IEEE Press, 1986, pp. 1156-1160.

[146] Y. Sapargaliyev and T. Kalganova, "Synthesis of Time-to-Amplitude Converter by Mean Coevolution with Adaptive Parameters," in *Journal of Software Engineering and Applications*, vol. 4, no. 8, pp. 447-464 Scientific Research Publishing, USA, August, 2011.

[147] J. Miller, "What bloat? Cartesian genetic programming on Boolean problems", In E. D. Goodman *Genetic and Evolutionary Computation Conference Late Breaking Papers*, San Francisco, California, USA, July 9-11, 2001, pp. 295-302.

[148] M. A. Trefzer, "Evolution of Transistor Circuits," *Dissertation for the degree of Doctor of Natural Sciences*, Ruperto-Carola-University of Heidelberg, Germany, December 2006.

[149] R. S. Zebulum, M. A. Pacheco and M. Vellasco. "A multi-objective optimization methodology applied to the synthesis of low-power operational amplifiers." in I. J. Cheuri and C. A. dos Reis Filho, eds., *Proceedings of the XIII International Conference in Microelectronics and Packaging,* vol. 1, Curitiba, Brazil, August 1998, pp. 264-271.

[150] P. J. Bentley and J. P. Wake, "Finding Acceptable Solutions in the Pareto-Optimal Range using Multi-objective Genetic Algorithms," in P. K. Chawdhry, R. Roy and R. K. Pant, eds., *Soft Computing in Engineering Design and Manufacturing*, Part 5, London, 1997. Springer Verlag London Limited, pp. 231-240.

[151] T. Soule and R. Heckerdorn, "An analysis of the causes of code growth in genetic programming," in *Genetic Programming and Evolvable Machines*, vol. 3, 2002, pp. 283-309.

[152] M. Toussaint and C. Igel, "Neutrality: A Necessity for Self-Adaptation." in *IEEE Congress on Evolutionary Computation* 2002 (CEC 2002), IEEE Press, pp. 1354-1359.

[153] C.R. Bonham and I. C. Parmee, "An investigation of exploration and exploitation within cluster oriented genetic algorithms (COGAs)," in *Proceedings of the Genetic and Evolutionary Computation Conference*, vol. 2,Morgan Kaufmann, San Francisco, 1999, pp. 1491-1497.

[154] Huayang Xie and Mengjie Zhang, "Balancing Parent and Offspring Selection in Genetic Programming," in A. E. Nicholson and X. Li, eds., in *Proceedings of the 22nd Australasian Joint Conference on Artificial Intelligence* (AI'09), vol. 5866, Melbourne, Australia, 2009, pp. 454-464.

[155] D. E. Goldberg, D. Thierens and K. Deb, "Toward a better understanding of mixing in genetic algorithms," in *Journal of the Society of Instrument and Control Engineers*, vol. 32, no. 1, 1993, pp. 10-16.

[156] N. Noman, J. Vatanutanon and H. Iba, "Tuning selection pressure in differential evolution using local selection," in *Second World Congress on Nature & Biologically Inspired Computing*, Kitakyushu, Japan, IEEE, 2010, pp. 66-71.

[157] P. Smith and K. Harries, "Code growth, explicitly defined introns, and alternative selection schemes," in *Evolutionary Computation*, vol. 6, no. 4, 1998, pp. 339–360.

[158] L. Altenberg, "The Evolution of evolvability in genetic programming," in K. J. Kinnear, (ed.), *Advances in Genetic Programming*, MIT Press, Cambridge, Mass, 1994, pp. 47-74.

[159] T. Soule and J. A. Foster, "Code Size and Depth Flows in Genetic Programming," in J. R. Koza et al., (eds.), *Genetic Programming 1997: Proceedings of the Second International Conference*, Morgan Kaufmann, 1997, pp. 313-320.

[160] G. E. Moore, "Cramming more components onto integrated circuits," in *Electronics*, vol. 38, no. 8, Apr. 1965.

[161] "Model 2145 Time to Amplitude Converter/Single channel Analyzer", Datasheet, Canberra Industries, Inc., Available: www.canberra.com/pdf/Products/Model-2145-SS-0226.pdf.

[162] B. Al-Hashimi (ed.), *System On Chip: Next Generation Electronics*, IEE Circuits Devices and Systems Book Series, 2006.

[163] A. Lymberis and D. De Rossi, *Wearable ehealth systems for personalised health management — state of the art and future challenges*, IOS Press, 2004.

[164] L. Sekanina and V. Drábek, "Theory and applications of evolvable embedded systems," in *Proc. of the 11th IEEE Int. Conference and Workshop on the Engineering of Computer-Based Systems*, IEEE CS Press, 2004, pp. 186-193.

[165] A. Stoica, R. Zebulum and D. Keymeulen, "On Polymorphic Circuits and Their Design Using Evolutionary Algorithms," in *Proceedings of IASTED International Conference on Applied Informatics* (AI2002), Innsbruck, Australia (2002).

[166] B. Vigoda, "Analog Logic: Continuous-Time Analog Circuits for Statistical Signal Processing," *PhD dissertation*, Massachusetts Institute of Technology, Sep., 2003.

[167] I. Harvey, P. Husbands and D. Cliff, "Seeing the light: artificial evolution; real vision," in D. Cliff, P. Husbands, J. A. Meyer and S. Wilson, (eds.), *From Animals to Animals 3, Proceedings of the third international conference on Simulation of Adaptive Behavior*, MIT Press/Bradford Books, 1994.

[168] F. Gomez and R. Miikkulainen, "Incremental evolution of complex general behavior," in *Adaptive Behavior 5*, 1997, pp. 317-342.

[169] S. Nolfi and D. Parisi, "Evolving non-Trivial Behaviors on Real Robots: An Autonomous Robot that Picks up Objects," in *Lecture notes in computer science*, 1995, pp. 243–243.

[170] J. He; M. Liu and Y. Chen, "A Novel Real-Coded Scheme for Evolutionary Analog Circuit Synthesis, Ligent Systems and Applications," in *International Workshop*, May 23-24 2009, pp. 1-4.

[171] K. Ohe, M. Konishi and J. Imai, "Design support classifier of filter circuit structure," in *SICE Annual Conference*, Aug. 20-22 2008, pp. 2695-2699.

[172] S. L. Sabat, K. S. Kumar and S. K. Udgata, "Differential Evolution and Swarm Intelligence Techniques for Analog Circuit Synthesis," in *World Congress on Nature & Biologically Inspired Computing*, Dec. 9-11 2009, pp. 469-474.

[173] H. Yuan and J. He, "Evolutionary design of operational amplifier using variable-length differential evolution algorithm, Computer Application and System Modeling (ICCASM)," in *2010 International Conference* on, pp. V4-610 - V4-614, Oct. 2010

[174] A. Das and R. Vemuri, "Fuzzy logic based guidance to graph grammar framework for automated analog circuit design," *in Proc. of International Conference on VLSI Design* (VLSID), Jan. 2009, pp. 445-450.

[175] P. Palmers, T. McConaghy, M. Steyaert and G. Gielen, "Massively multi-topology sizing of analog integrated circuits," in *Proceedings of Design, Automation & Test in Europe Conference & Exhibition*, Nice, 2009, pp. 706-711.

[176] A. Somani, P. P. Chakrabarti and A. Patra, "An Evolutionary Algorithm-Based Approach to Automated Design of Analog and RF Circuits Using Adaptive Normalized Cost Functions," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, June 2007, pp. 336-353.

[177] P. Conca, G. Nicosia, G. Stracquadanio and J. Timmis, "Nominal-Yield-Area Tradeoff in Automatic Synthesis of Analog Circuits: A Genetic Programming Approach Using Immune-Inspired Operators," NASA/ESA Conference, Adaptive Hardware and Systems, AHS 2009, in Francisco, CA, July 29 2009-Aug. 1 2009, pp. 399-406.

[178] H. Xu and Y. Ding, "Optimizing Method for Analog Circuit Design Using Adaptive Immune Genetic Algorithm," *Fourth International Conference Frontier of Computer Science and Technology, 2009*, FCST '09. , in Shanghai, 17-19 Dec. 2009, pp. 359-363.

[179] T. McConaghy, P. Palmers, M. Steyaert, and G. G. E. Gielen, "Trustworthy genetic programming-based synthesis of analog circuit topologies using hierarchical domain-specific building blocks," in *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, Aug. 2011, pp. 557-570.

[180] A. Thompson, "Notes on design through artificial evolution: Opportunities and algorithms," in I. C. Parmee, (ed.), *Adaptive computing in design and manufacture V*, Springer-Verlag, 2002, pp. 17-26..

[181] J. von Neumann, *Theory of Self-Reproducing Automata*, University of Illinois Press, Urbana, IL, 1966. Edited and completed by A.W. Burks.