



**BRNO UNIVERSITY OF TECHNOLOGY**

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

**FACULTY OF INFORMATION TECHNOLOGY**

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

**DEPARTMENT OF COMPUTER SYSTEMS**

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

**AUTOMATED MULTI-OBJECTIVE PARALLEL EVOLUTIONARY CIRCUIT DESIGN AND APPROXIMATION**

AUTOMATICKÝ MULTIKRITERIÁLNÍ PARALELNÍ EVOLUČNÍ NÁVRH A APROXIMACE OBVODŮ

**EXTENDED ABSTRACT OF A PHD THESIS**

ROZŘÍŘENÝ ABSTRAKT DIZERTAČNÍ PRÁCE

**AUTHOR**

AUTOR PRÁCE

**Ing. RADEK HRBÁČEK**

**SUPERVISOR**

VEDOUCÍ PRÁCE

**Prof. Ing. LUKÁŠ SEKANINA, Ph.D.**

BRNO 2017

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Open Problems . . . . .	4
1.3	Research Objectives . . . . .	4
1.4	Thesis Outline . . . . .	4
<b>2</b>	<b>Survey of the State of the Art</b>	<b>6</b>
2.1	Approximate Computing . . . . .	6
2.1.1	Application Error Resilience . . . . .	6
2.1.2	Approximate Circuits . . . . .	7
2.1.3	Design Objectives . . . . .	8
2.1.4	Overview of Circuit Approximation Methods . . . . .	10
2.2	Evolutionary Design . . . . .	12
2.2.1	Cartesian Genetic Programming . . . . .	12
2.2.2	Evolutionary Design of Digital Circuits . . . . .	14
2.2.3	Multi-Objective EAs . . . . .	14
2.2.4	Design Acceleration . . . . .	17
<b>3</b>	<b>Research Summary</b>	<b>21</b>
3.1	Overview . . . . .	21
3.2	Papers included in this thesis . . . . .	22
3.2.1	Paper I . . . . .	22
3.2.2	Paper II . . . . .	22
3.2.3	Paper III . . . . .	23
3.2.4	Paper IV . . . . .	23
3.2.5	Paper V . . . . .	24
3.2.6	Paper VI . . . . .	25
3.2.7	Paper VII . . . . .	26
3.3	List of Other Publications . . . . .	26
3.4	Research Projects and Grants . . . . .	28
3.4.1	Czech Science Foundation . . . . .	28
3.4.2	Anselm & Salomon Supercomputer Allocations . . . . .	28
3.5	Awards . . . . .	28
<b>4</b>	<b>Discussion and Conclusions</b>	<b>29</b>
4.1	The Approach . . . . .	29
4.2	Software Outcomes . . . . .	29
4.3	Contributions . . . . .	31

4.4	Future Work . . . . .	31
<b>A</b>	<b>Curriculum Vitae</b>	<b>39</b>
A.1	Education . . . . .	39
A.2	Conferences, Summer Schools, Institutions . . . . .	39
A.3	Awards, Courses & Certifications . . . . .	40
A.4	Projects . . . . .	40
A.5	Teaching . . . . .	41
A.6	Work Experience . . . . .	41
A.7	Languages . . . . .	41

# Chapter 1

## Introduction

This chapter gives an introduction to the thesis. It starts with the motivation for the whole research, then the open problems and research objectives of the thesis are formulated. At the end of the chapter, an outline of the thesis is given.

### 1.1 Motivation

Computers and computer based systems play a crucial role in people's everyday lives. Embedded systems can be found almost everywhere. Power efficiency is becoming increasingly important property of computing platforms, especially because of limited power supply capacity of embedded devices and high costs associated with operating growing data centers and cloud infrastructure. At the same time, in an increasing number of applications users are able to tolerate inaccurate or incorrect computations to a certain extent due to imperfections of human senses, statistical nature of data processing, noisy input data etc.

*Approximate computing*, an emerging paradigm in computer engineering, takes advantage of relaxed functionality requirements to make computer systems more efficient in terms of energy consumption, computing speed or complexity [33]. Error resilient applications can achieve significant savings while still serving their purpose with the same or a slightly degraded quality.

The complexity of computer systems is permanently growing and thus, automated design tools have to deal with more complex problems specified on higher level of abstraction than before. The same holds true for approximate computing. Even though new methods are emerging, there is a lack of methods for automated approximate HW/SW design offering a rich set of compromise solutions. Moreover, conventional synthesis algorithms often produce solutions that are far from an optimum [8].

*Evolutionary algorithms* (EAs) have been confirmed to bring innovative solutions to complex design and optimization problems. Recently, complex digital circuits have been optimized by means of EAs while the scalability of the method has been improved substantially [20, 55].

Every year, a special competition *Humies* is held at the Genetic and Evolutionary Computation Conference to award scientific results that utilize an evolutionary computation technique and are *human-competitive* [24]. In years 2004-2013, there were 42 Humie winners and 10 of them published results that were patented or would qualify as a patentable new invention [24]. The same trend can be observed for years 2014-2016.

## 1.2 Open Problems

The main issue of approximate computing at the moment is the lack of available automated methods capable of providing approximations for arbitrary combinational circuits under different error metrics and with respect to multiple objectives. The solutions provided by conventional circuit synthesis methods are often far from optimum.

On the other hand, the evolutionary based design and approximation methods suffer from several problems, mainly the scalability of the methods (i.e. the scalability of the fitness function and the representation of candidate circuits) is not sufficient. A high number of fitness evaluations needed to evolve competitive results implies simplifications in circuit parameters estimation and thus leads to reduced accuracy of the estimations. Although complex digital circuits have been optimized using single-objective EAs, the same cannot be said about the multi-objective methods.

## 1.3 Research Objectives

The first main research objective for this thesis is to

*develop an automated scalable design method based on evolutionary algorithms, capable of multi-objective design and approximation of digital circuits.*

As indicated, such a method has to meet several requirements. It has to be able to design circuits of a sufficient complexity. It has to take into account multiple design criteria. Moreover, the estimation of the circuit parameters has to be accurate enough. Finally, the implementation should be parallelized and should efficiently utilize computational resources.

The second main objective is to

*show on several real-world problems that the method provides human-competitive results.*

These objectives can be translated into the following partial goals:

1. To develop an optimized parallel evolutionary algorithm for digital circuits design.
2. To extend the evolutionary design method with multi-objective design capability.
3. To identify objectives relevant for approximate circuits and transform them to fitness functions.
4. To carry out experiments on different real world applications to show the performance of the method.
5. To validate the achieved results by means of professional simulation tools.

## 1.4 Thesis Outline

The thesis is composed as a collection of papers. The research contribution of this thesis is comprised of seven peer-reviewed research papers in their original publication format. The thesis is organized as follows: Chapter 1 gives an introduction to the thesis. Chapter 2 surveys the state of the art and presents relevant background information for the research.

Chapter 3 summarizes the research process and gives an overview over the papers constituting the research contribution. Finally, Chapter 4 presents conclusions and proposes future research directions.

## Chapter 2

# Survey of the State of the Art

This chapter gives relevant background information needed for a proper understanding of the work presented in the thesis. It primarily addresses the areas of approximate computing and evolutionary circuit design.

### 2.1 Approximate Computing

Recently, power efficiency has become one of the most important parameters of almost every computing platform. At the same time, a wide range of applications in which we are willing to tolerate imperfections in computations has spread out. As a consequence, a new research field – *approximate computing* – has emerged to investigate how computer system can be made more efficient in terms of energy consumption, computing speed or complexity assuming that some errors are acceptable. It has been believed, that significant savings can be achieved by relaxing the requirement of perfect functionality thanks to the *error resilience* of some applications. Therefore, the *accuracy* (error) of the system can be used as a design metric and inaccurate solutions can be accepted if an improvement in other parameters occurs.

The approximation can be introduced at various levels including the entire computer system architecture [30], particular components (e.g. ALU) [13], operating system, algorithm or even programming language [3]. As the complexity of today’s computer systems grows, manual approximation is not an efficient design method. Hence, several automated approximate design methods have been introduced.

#### 2.1.1 Application Error Resilience

Inherent application *error resilience* is the property of an application to produce acceptable outputs even if some underlying computations are approximate or incorrect [6]. Whether an output is acceptable or not is given by an *output quality* metric if the concept of approximate computing is considered. Applications are designed to produce outputs of acceptable quality rather than a unique correct output.

The sources that contribute to the application resilience can be classified into following the categories [6]:

- *Inputs*: Applications that process noisy or redundant data can be inherently resilient.

- *Outputs*: If the specification does not define a unique golden output or the outputs are consumed by human senses, minor output variations that are often indistinguishable are acceptable.
- *Computation patterns*: Statistical computations can result in attenuation or cancellation of error. Applications employing statistical computations are thus resilient.
- *Iterative processing*: Many applications feature iterative processing and undergo successive refinement or aggregation to obtain converged results. The quality gain tends to attenuate as the iterations continue [65].

The level, at which the system is approximated, influences the resilience to approximations. For example, introducing approximations at the software level can lead to a very different conclusion about the error resilience than at the hardware level [6].

The overall output quality is given by individual responses to different system inputs, which makes the output quality a statistic. In general, the most frequently used quality metrics are the *error probability* (error rate), *error magnitude* (mean error) and *error predictability* (error variance) [6]. These metrics form a three-dimensional space and all acceptable qualities form a subspace in this space. This subspace is highly application dependent, but in general, a wide range of applications accept outputs with low error rate or low error magnitude [6].

The computation patterns (e.g. in statistical processing) present in particular applications affect the error resilience significantly. Thanks to that, there are applications that accept output errors with small variance present in all computations (which correspond to very high error rate) while the error magnitude can be very large. In addition to computation patterns, the context in which the application is used significantly impacts the resilience. For example, the error resilience of a k-means clustering algorithm used in an image segmentation application depends on chosen quality metric. The application is able to tolerate more aggressive approximations if mean centroid distance is used as the quality metric in comparison with the percentage of mis-clustered points [6].

Approximations can be done at multiple levels by applying several approximate computing techniques at the same time. The resilience is again application dependent in such situation. Generally, different approximation techniques can be applied in a synergistic manner, but there can be cases for which the combination of particular techniques leads to unacceptable results [6].

### 2.1.2 Approximate Circuits

While automatic design of digital circuits has been well established in the past, the correct functionality has always been an essential requirement put on the circuits. The other parameters, like the area, delay or power consumption, have been considered as secondary and have not been optimized as long as a fully working solution has been found.

The *approximate circuits* are designed in such a way that the functionality specification (assuming a perfect operation) is not fully met in exchange for savings in terms of area, delay, power consumption etc. Although the circuit is not working properly, it can still be suitable for applications in which certain level of error is not recognizable (e.g. human perception in the context of multimedia applications). Moreover, in some cases (e.g. low battery), users could knowingly tolerate even more inaccuracy in order to extend the time of operation.



### 2.1.3 Design Objectives

When designing an approximate computer system, the functionality requirement (*accuracy*) is traded off for improvements in other design objectives. These objectives are application dependent, but they usually include *size*, *power consumption* and *performance*. In the case of hardware solutions, one has to deal with the *reliability and dependability* of the system. In order to correctly determine parameters (i.e. particular values for the objectives) of computer systems, careful benchmarking has to be performed. The acceptable level of inaccuracy is application dependent.

#### Accuracy/Error

The accuracy (error) of a computer system is the main objective tracked when doing approximations. In each application, different requirements on the accuracy metric can be formulated resulting in a wide range of different accuracy metrics being used. Usually, an opposite metric, the *error*, is used instead of accuracy.

For combinational circuits, one can use the number of incorrect outputs (i.e. the *Hamming distance*):

$$e_{\text{hamm}} = \sum_{\forall i, \forall j, O_{\text{approx}}^{(i,j)} \neq O_{\text{orig}}^{(i,j)}} 1, \quad (2.1)$$

where  $O_{\text{approx}}^{(i,j)}$  is the  $j$ -th bit of the  $i$ -th circuit output and  $O_{\text{orig}}^{(i,j)}$  is the correct one. The other option is to calculate the error probability [54]:

$$e_{\text{prob}} = \frac{\sum_{\forall i, O_{\text{approx}}^{(i)} \neq O_{\text{orig}}^{(i)}} 1}{2^{n_i}}, \quad (2.2)$$

where  $n_i$  is the number of circuit inputs,  $O_{\text{approx}}^{(i)}$  is the  $i$ -th circuit output (all bits) and  $O_{\text{orig}}^{(i)}$  is the correct one.

The significance of the aforementioned two metrics is very low for arithmetic circuits or digital signal processing systems in general, for which more suitable metrics based on the arithmetic distance between the actual and correct values exist. One can use the worst case error  $e_{\text{wce}}$ , the mean absolute error  $e_{\text{mae}}$ , the mean squared error  $e_{\text{mse}}$  or their relative versions  $e_{\text{wcre}}$  and  $e_{\text{mre}}$  as follows [54]:

$$e_{\text{wce}} = \max_{\forall i} \left| O_{\text{approx}}^{(i)} - O_{\text{orig}}^{(i)} \right|, \quad (2.3)$$

$$e_{\text{mae}} = \frac{\sum_{\forall i} \left| O_{\text{approx}}^{(i)} - O_{\text{orig}}^{(i)} \right|}{2^{n_i}}, \quad (2.4)$$

$$e_{\text{mse}} = \frac{\sum_{\forall i} \left| O_{\text{approx}}^{(i)} - O_{\text{orig}}^{(i)} \right|^2}{2^{n_i}}, \quad (2.5)$$

$$e_{\text{wcre}} = \max_{\forall i} \frac{\left| O_{\text{approx}}^{(i)} - O_{\text{orig}}^{(i)} \right|}{\max(1, O_{\text{orig}}^{(i)})}, \quad (2.6)$$

$$e_{\text{mre}} = \frac{\sum_{\forall i} \frac{\left| O_{\text{approx}}^{(i)} - O_{\text{orig}}^{(i)} \right|}{\max(1, O_{\text{orig}}^{(i)})}}{2^{n_i}}, \quad (2.7)$$

where  $n_o$  is the number of circuit outputs.

## Size

The size corresponds to the amount of resources statically used by the system. Computer programs vary in the amount of program memory occupied or data memory operating on, digital circuits occupy an area on die or the number of used reconfigurable cells on Field Programmable Gate Arrays (FPGAs).

## Power Consumption

The power consumption is the most important objective being shrunk in approximate computing. It can be measured as energy needed for performing an operation or as average power consumption of a permanently running system.

In the case of digital circuits (CMOS technology), there are three major sources of power dissipation [5]:

$$\begin{aligned} P_{\text{avg}} &= P_{\text{switching}} + P_{\text{short.circuit}} + P_{\text{leakage}} \\ &= \alpha \cdot C_L \cdot V_{\text{dd}}^2 \cdot f_{\text{clk}} + I_{\text{sc}} \cdot V_{\text{dd}} + I_{\text{leakage}} \cdot V_{\text{dd}}. \end{aligned} \quad (2.8)$$

The switching power  $P_{\text{switching}}$  depends on the switching activity  $\alpha$  (probability of switching a gate's output), the operating frequency  $f_{\text{clk}}$ , load capacitance  $C_L$  and power supply voltage  $V_{\text{dd}}$ . The second term, the short circuit power  $P_{\text{short.circuit}}$ , is caused by the short circuit current  $I_{\text{sc}}$  which arises when both complementary transistors are active at the same time, i.e. conducting current directly from supply to ground. These two terms together represent the dynamic power, while the switching power is usually 10 times greater than the short circuit power. The last term is the leakage (static) power  $P_{\text{leakage}}$ , which is primarily determined by fabrication technology considerations [5].

The introduction of CMOS technology led to significant reductions in static power consumption. For a long time, the dynamic power dissipation was the dominant component. However, with the decreasing size of the semiconductor technology process, the static dissipation is increasing due to rising leakage currents and is becoming the major component of the power consumption [49].

## Performance

The next commonly used objective is the performance (speed) of computation. In the case of computer programs, it usually applies to the execution time. The speed of digital circuits can be determined by the maximum operating frequency or the latency, i.e. the interval between the stimulation of the inputs and the response on the outputs. Another way of measuring the speed is to compute the throughput of the system.

## Reliability/Dependability

Reducing the probability of failure and increasing the reliability of digital circuits is an important design objective. Many applications (e.g. automotive, aerospace, operating in remote environments) are safety-critical and need to be built using the principles of fault-tolerant system design. As the complexity of computer systems increases, more complex mechanisms must be introduced to preserve the reliability of the systems [49].

### 2.1.4 Overview of Circuit Approximation Methods

Automated approximate computing techniques are being developed to speed up the design process and to find the best trade-off solutions between the resources being shrunk and the inaccuracy of the computation. The design of approximate circuits is typically based on modifying fully functional circuits [47]. Most of the methods deal with combinational circuit design, however, there are methods for sequential circuit approximation as well.

#### Voltage Over-Scaling

As can be seen from Equation 2.8, the power consumption  $P_{\text{avg}}$  is highly dependent on the supply voltage  $V_{\text{dd}}$ . Present computer systems often utilize voltage scaling together with frequency scaling in order to lower the power consumption when full performance is not needed. Voltage over-scaling extends this concept beyond the critical voltage value at which the critical path delay is just met. This leads to significant energy savings for the price of possible incorrect computations [25].

The supply voltage can be controlled adaptively with respect to the occurrence of errors in the circuit. For example, the adaptive voltage over-scaling strategy presented in [28] monitors several locations in the circuit where errors can be detected. The signals are sampled with a delayed clock and compared to the value sampled with the main clock. If they differ, an error is detected. The supply voltage is then controlled according to current error rate.

The drawback of the voltage over-scaling approach is the difficulty of controlling the error. Since the behavior of the circuit after voltage over-scaling depends on many factors (each logic gate behaves differently according to its type, input timing, output load, etc.), accurate timing analysis has to be performed so as to measure the output quality. For example, the Modeling and Analysis of Circuits for Approximate Computing (MACACO) methodology is based on the construction of an equivalent circuit that represents the behavior of the approximate circuit at a given voltage and clock frequency [61].

#### Manual Methods

In the first approximation methods, the design of approximate circuits was typically based on manual modifications of fully functional circuits. These first results include arithmetic circuits, such as combinational adders [13] or multipliers [29]. In general, only small components have been approximated manually, e.g. 2-bit multiplier occupying nearly half area and working almost correctly except for a single output value ( $3 \cdot 3 = 7$ ) [29]. By using this simple component as a building block, one can design larger circuits, however, the method clearly does not exploit the whole potential of approximate computing.

#### SALSA

The Systematic methodology for Automatic Logic Synthesis (SALSA) uses a quality function which decides whether a predefined quality constraint is met or not. The algorithm is allowed to modify the circuit as long as the quality constraint is not exceeded. SALSA has been applied to a number of problems, e.g. 32-bit adders, 8-bit multipliers, FIR filters, DCT blocks and others [59].

## SASIMI

Another approach, Substitute-and-Simplify (SASIMI), looks for signal pairs having similar values with a high probability. By substituting one signal for the other, a part of the circuit can be removed resulting in area and power savings at the cost of an error introduced to the circuit. Moreover, SASIMI further extends the approach to synthesize quality configurable circuits, where at runtime, processing of selected input vectors is given an additional cycle to correct errors due to approximations [60].

## ABACUS

Unlike the aforementioned methods, ABACUS (Automated Behavioral Approximate Circuit Synthesis) operates directly on the behavioral descriptions of circuits. ABACUS automatically generates approximate circuits from input behavioral descriptions by performing global transformations on an abstract synthesis tree (AST) created from the behavioral description. The outcome approximate circuits are still expressed in behavioral code and can be synthesized by means of standard synthesis tools. Complementary approximate computing methods, e.g. voltage over-scaling or manually created approximate components, may be still used [36]. The latest version of the algorithm supports multi-objective design based on the principles of the NSGA-II algorithm [37].

## ASLAN

Although most of the design methods deal with combinational circuits, there are methods capable of approximating sequential circuits. As an example, the Automatic Methodology for Sequential Logic Approximation (ASLAN) creates an approximate version of a sequential circuit that consumes lower energy, while meeting a specified quality constraint. ASLAN identifies combinational blocks in the sequential circuit that are amenable to approximation and iteratively approximates the entire sequential circuit using a gradient-descent approach [43].

## EA-based Methods

Several evolutionary algorithm based methods have been used in approximate computing recently. Most of the methods are single-objective and the optimization of a secondary objective is achieved either by restricting the circuit resources (by constraining the genotype size) [53] or using a multi-phase approach [54]. A multi-objective evolutionary algorithm was used to design approximate multiple constant multipliers [39]. However, the method operated on functional unit level and the complexity of the circuits was relatively small.

## Summary

Despite numerous attempts, almost all papers dealing with the design of approximate circuits show some of the following features that are undesirable [35]:

- The approximation method is described, but a corresponding software implementation is not available.
- An implementation of the original (accurate) circuit is not available.

- The quality of approximation and other parameters of approximate circuits are expressed relatively to parameters of the original circuits.
- Implementations of the resulting approximate circuits are not available.
- Only a few approximate versions created from the original circuit are reported, forming thus a sparsely occupied Pareto front.
- It is unclear if a given number of test vectors used to evaluate approximate circuits is sufficient for obtaining a trustworthy error quantification if the error is determined using simulation.
- A given approximation method is only rarely compared against competitive approximation methods.

## 2.2 Evolutionary Design

Evolutionary Algorithms (EAs), generic population-based metaheuristic optimization algorithms, use mechanisms inspired by biological evolution, such as reproduction, recombination, mutation or selection for purposes of optimization and design. Population of individuals represents a set of candidate solutions to a specified problem. Each individual is assigned a *fitness value* depending on the ability of the individual to solve the problem. In each generation, a subset of the population is selected according to the fitness value to create offspring population by means of recombination and mutation [1].

While EAs were originally used to solve optimization problems, they are able to bring innovative solutions to design problems as well. Evolutionary design of hardware is a growing research area since the beginning of the 1990s. In particular, it includes evolutionary design of digital and analog circuits, antennas, optical systems and microelectromechanical systems (MEMS) [46].

EAs have been applied to a number of real problems, however, their computational complexity can be enormous. The scalability of the fitness function is often a prohibiting factor and thus, one has to deal with the acceleration of the fitness function or fitness approximation. Besides the scalability of the fitness evaluation, another problems that limit the application of EAs are known, such as the scalability of the representation (complex problems are represented by long chromosomes which implies large search space), the non-deterministic nature of EAs or slow convergence. Potential solutions to the problems have been recently summarized in [49].

In the following section, we will introduce Cartesian Genetic Programming (CGP) since it has been routinely used in the area of evolutionary based digital circuit design and optimization.

### 2.2.1 Cartesian Genetic Programming

Cartesian genetic programming has been introduced by Miller [31] as a branch of genetic programming. Unlike GP which uses tree representation, an individual in CGP is represented by a directed acyclic graph which enables the candidate solution to have multiple outputs and automatically reuse intermediate results. This makes CGP very suitable for the design of various kinds of digital circuits (such as arithmetic and logic circuits, digital filters, etc.) and computer programs [31].

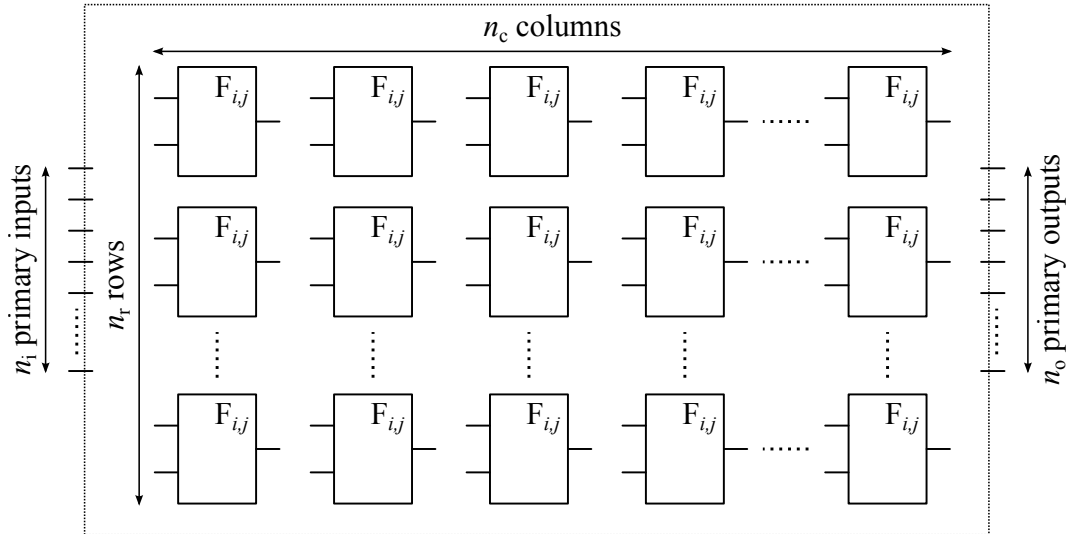


Figure 2.1: Cartesian genetic programming scheme.

CGP uses a cartesian grid of  $n_r \times n_c$  programmable elements (nodes) interconnected by a feed-forward network (Figure 2.1). Each node's input (each node has a fixed number of inputs, e.g.  $n_{ni} = 2$ ) can be connected either to one of  $n_i$  primary inputs or to a node output in the preceding  $l$  columns. By setting the  $l$ -back parameter and the grid size, one can control the area and delay of the circuit. Each node can be programmed to perform one of  $n_{ni}$ -input functions defined in the set  $\Gamma$  (let  $n_f = |\Gamma|$ ). The  $n_o$  primary circuit outputs are connected either to the primary inputs or nodes. The output connectivity can be optionally restricted by the  $o$ -back parameter.

Since all the CGP parameters are fixed, each chromosome is encoded using a fixed-size string of  $n_r \cdot n_c \cdot (n_{ni} + 1) + n_o$  integers. Each primary input is assigned a number from  $\{0, \dots, n_i - 1\}$  and the nodes are assigned numbers from  $\{n_i, \dots, n_i + n_r \cdot n_c - 1\}$ . The genotype is of fixed length, whereas the phenotype is of variable length depending on the number of inactive nodes, i.e. nodes whose output is not used by any other node or primary output. Hence, the genotype-phenotype mapping is not injective. The existence of genotypes with the same fitness is usually referred to as neutrality. The role of neutrality has been intensively studied [66] and it was shown that for certain problems the neutrality significantly reduces the computational effort and helps to find more innovative solutions [32].

In CGP, a simple mutation based  $(1 + \lambda)$  evolutionary strategy is used as a search mechanism. The population size  $1 + \lambda$  is usually very small, typically,  $\lambda$  is between 1 and 15. The initial population is constructed either randomly (then we speak about evolutionary design) or by mapping of a known solution to the CGP chromosome (evolutionary optimization) [55]. In each generation, a randomly selected individual with the best fitness value (if there are more of them, an individual genotypically distinct from the parent) is passed to the next generation unmodified and its  $\lambda$  offspring individuals are created by means of point mutation operator which modifies  $m$  randomly selected genes of the chromosome. The mutation rate is usually set to modify up to 5% of the total number of genes. For some problem classes (e.g. symbolic regression problem), special crossover operators have been investigated [7], however, none of them has been confirmed to significantly improve the search process.

Recently, several modifications to CGP have been published. Embedded CGP is an extension of the CGP which is capable of automatically acquiring, evolving and re-using partial solutions in the form of modules [62]. By introducing multiple chromosomes, each connected to a single output, large problems with multiple outputs can be broken down into many smaller problems leading to significant performance increase for particular problems [63]. Self Modifying CGP enables self-modifications of CGP individuals by introducing operations into the CGP chromosome [14]. Multi Expression CGP modifies CGP individual evaluation in such a way that multiple nodes are compared to the desired output instead of just a single node [4]. Recurrent CGP allows recurrent connections [52]. All these modifications share a common objective – to increase the scalability of CGP and speed up the evolutionary process.

### 2.2.2 Evolutionary Design of Digital Circuits

In the case of combinational circuit evolution, the fitness function corresponds to the quality of the candidate circuit measured as the number of correct output bits compared to a specified truth table (see Equation 2.1). In order to obtain a fully working circuit, all combinations of input values have to be evaluated. For a circuit with  $n_i$  inputs and  $n_o$  outputs,  $2^{n_i}$  test vectors need to be fetched to the primary inputs and  $n_o \cdot 2^{n_i}$  output bits have to be verified so as to compute the fitness value. In this thesis, we assume this scenario.

Recently, complex digital circuits have been successfully optimized by means of CGP [57]. However, designing complex circuits from scratch (from a randomly generated initial population) has been shown to be much more difficult [20].

Besides using the Hamming distance as the fitness function for digital circuits design, there are other possibilities for particular applications. For example, digital image filters can be designed by means of CGP. In this case, the functional specification is not complete, the quality of the candidate circuits is evaluated on a limited training data set [46].

Other applications of CGP include the design and optimization of digital circuits at the transistor level [34], evolutionary design of polymorphic circuits [38] or transistor-level design and optimization of FPGA architectures with respect to production variability [49].

### 2.2.3 Multi-Objective EAs

Unlike the single-objective optimization, which enables to compare any two candidate solutions and decide which one is better, the multi-objective optimization leads to the existence of a set of solutions showing different trade-offs, if the objectives are conflicting.

A multi-objective evolutionary optimization problem can be defined as

$$\begin{aligned} & \text{minimize/maximize} && f_m(p), && m = 1, 2, \dots, M, \\ & \text{subject to} && g_j(p) \geq 0, && j = 1, 2, \dots, J, \\ & && h_k(p) = 0, && k = 1, 2, \dots, K, \end{aligned} \tag{2.9}$$

where  $f_m$  are the optimized objectives,  $p$  is an individual. The solutions must fulfill the inequity constraints  $g_j$  and equity constraints  $h_k$  to be acceptable [10].

Many multi-objective evolutionary algorithms have been proposed. Most of them are based on the idea of *Pareto dominance*. The solution  $p$  *dominates* the solution  $q$  ( $p \prec q$ ) if  $p$  is no worse than  $q$  in all objectives and  $p$  is strictly better than  $q$  in at least one objective. The principle can be seen in Figure 2.2, where the Pareto optimal solutions are not dominated by any other solutions and form the so called *Pareto front*.

## Strength Pareto Evolutionary Algorithm 2

Strength Pareto Evolutionary Algorithm 2 (SPEA2), a multi-objective EA introduced by Zitzler et al. [67], maintains two sets of individuals: an archive with non-dominated solutions and a breeding population. In each generation, the fitness of all individuals from both sets is evaluated and the non-dominated solutions are found. The archive is then updated with the non-dominated solutions; a nearest neighbor density estimation algorithm is applied if the archive size is exceeded. The fitness of an individual is computed based on the number of individuals it dominates, the number of individuals that are dominated and the density estimate. The offspring population is created using recombination and mutation of individuals selected using a binary tournament selection [27].

## TSPEA2

TSPEA2 is a branch of SPEA2 introduced by Kaufmann and Platzner [26]. The only difference between SPEA2 and TSPEA2 is that TSPEA2 favours one (main) objective over several others. In the binary tournament, the main objective is first checked and if one of the individuals is better, it is preferred regardless of other objectives. TSPEA2 was motivated by an earlier algorithm MO-Turtle GA introduced by Trefzer et al. [50].

## $\mu$ GA and $\mu$ GAI

The  $\mu$ GA and  $\mu$ GAI algorithms use three populations: an external population for non-dominated individuals of high diversity, a working (breeding) population and an immutable population containing randomized solutions [48]. In each generation, a small set of individuals is selected randomly from the breeding and the immutable population and a standard GA is applied on them. After reaching nominal convergence (the situation when all individuals have similar chromosomes), the best individuals are copied to the breeding and the external population. After several generations, a subset of the breeding population is replaced by non-dominated individuals from the external population [27].

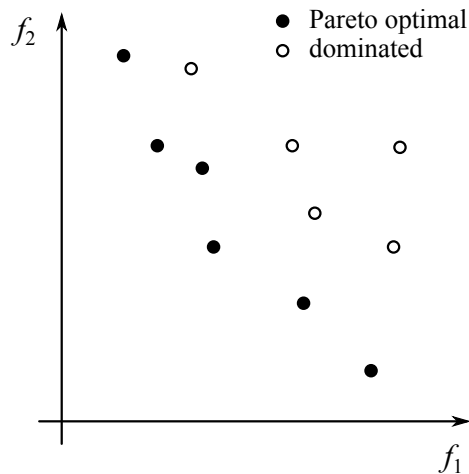


Figure 2.2: Pareto optimal and dominated solutions (when  $f_1$  and  $f_2$  have to be minimized).



## Non-dominated Sorting Genetic Algorithm II

One of the most popular multi-objective evolutionary algorithms is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [9]. It is based on sorting individuals from population  $P$  according to the dominance relation into multiple fronts. The first front  $F_0$  contains all non-dominated solutions. Each subsequent front  $F_i$  is constructed by removing all the preceding fronts from the population and finding a new Pareto front. Each solution is assigned a *rank* according to the front it belongs to, the solutions from the front  $F_i$  have the rank equal to  $i$ . The NSGA-II fast non-dominated sort (see Algorithm 1) is very efficient, the overall complexity is  $\mathcal{O}(MN^2)$ , where  $N$  is the population size and  $M$  is the number of objectives. The set  $S_p$  contains all individuals from the population that are dominated by  $p$ . The number of individuals that dominate  $p$  is denoted by  $n_p$ . The rank of an individual  $p$  (the order of the frontier it belongs to) is denoted by  $p_{\text{rank}}$ .

```

fast-non-dominated-sort( $P$ )
 $F_0 = \emptyset$ 
foreach  $p \in P$  do
     $S_p = \emptyset$ 
     $n_p = 0$ 
    foreach  $q \in P$  do
        if  $p \prec q$  then
             $S_p = S_p \cup \{q\}$ 
        end
        else if  $q \prec p$  then
             $n_p = n_p + 1$ 
        end
    end
    if  $n_p = 0$  then
         $p_{\text{rank}} = 0$ 
         $F_0 = F_0 \cup \{p\}$ 
    end
end
 $i = 0$ 
while  $F_i \neq \emptyset$  do
     $Q = \emptyset$ 
    foreach  $p \in F_i$  do
        foreach  $q \in S_p$  do
             $n_q = n_q + 1$ 
            if  $n_q = 0$  then
                 $q_{\text{rank}} = i + 1$ 
                 $Q = Q \cup \{q\}$ 
            end
        end
    end
     $i = i + 1$ 
     $F_i = Q$ 
end
 $F = (F_0, F_1, \dots)$ 
return  $F$ 

```

**Algorithm 1:** Non-dominated sort [19].

```

crowding-distance-assignment( $F_i$ )
 $l = |F_i|$ 
foreach  $p \in P$  do
     $p_{\text{dist}} = 0$ 
end
foreach objective  $m$  do
     $I = \text{sort}(F_i, m)$ 
     $I[0]_{\text{dist}} = \infty$ 
     $I[l-1]_{\text{dist}} = \infty$ 
    for  $i$  in 1 to  $l-2$  do
         $I[i]_{\text{dist}} = I[i]_{\text{dist}} + \frac{I[i+1]_m - I[i-1]_m}{f_m^{\text{max}} - f_m^{\text{min}}}$ 
    end
end
end

```

**Algorithm 2:** Crowding distance assignment [19].

```

constraint-violation-assignment( $P$ )
foreach  $p \in P$  do
     $p_{\text{constr.viol}} = 0$ 
    foreach objective  $m$  do
        if  $p_m < c_m^{\text{min}}$  then
             $p_{\text{constr.viol}} = p_{\text{constr.viol}} + \frac{c_m^{\text{min}} - p_m}{f_m^{\text{max}}}$ 
        end
        if  $p_m > c_m^{\text{max}}$  then
             $p_{\text{constr.viol}} = p_{\text{constr.viol}} + \frac{p_m - c_m^{\text{max}}}{f_m^{\text{max}}}$ 
        end
    end
end
end

```

**Algorithm 3:** Constraint violation assignment [19].

The solutions within the individual fronts are then sorted according to the *crowding distance metric*. This metric helps to preserve the diversity of the population along the fronts [9]. It is computed as the average distance of two solutions on either side along each of the objectives. Solutions on the boundaries are assigned an infinite crowding distance, which ensures that these solutions will always dominate the other solutions (see Algorithm 2). Any

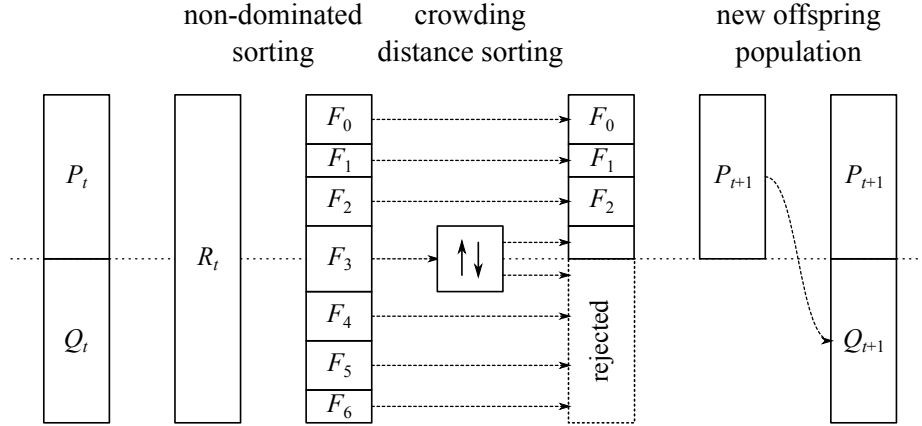


Figure 2.3: NSGA-II algorithm scheme.

solution from the front  $F_i$  always dominate any solution from  $F_j$ ,  $j > i$ . Within the fronts, solutions with higher crowding distance are preferred [19].

Many real world applications require constraining the solutions on particular objectives. NSGA-II offers a simple way to handle the constraints, whilst the low algorithm complexity is preserved. Each solution can be either feasible or infeasible. The infeasible solutions are assigned a constraint violation according to the Algorithm 3. The constraints on the objective  $m$  are denoted by  $\langle c_m^{\min}, c_m^{\max} \rangle$ . When comparing two solutions, a feasible solution is always preferred. If both solutions are infeasible, the solution with smaller constraint violation is better. In the opposite case, if both solutions are feasible, the dominance depends on the rank and the crowding distance metric [19].

The overall algorithm works as follows. In each generation  $t$ , the parental population  $P_t$  and the offspring population  $Q_t$  (both of the same size) form an unified population  $R_t$ . The individuals in  $R_t$  are assigned the equivalence rank and the crowding distance. Then, the Pareto fronts are identified and the new parental population  $P_{t+1}$  is filled with the individuals from the first fronts as long as  $P_{t+1}$  is not overcrowded. The individuals from the last used Pareto front are sorted using the crowding distance and a fraction of them is selected just to fill the population  $P_{t+1}$  (see Figure 2.3) [19].

The first attempts to use NSGA-II with CGP used the GA representation of the individuals [27]. Knieper et al. compared the performance of four multi-objective EAs (SPEA2, TSPEA2, NSGA-II and  $\mu$ GA) with standard GA in the task of combinational 2- and 3-bit adders and multipliers and 6- and 7-parity circuits. Hilder et al. [15] used NSGA-II with CGP to evolve 2- and 3-bit combinational adders and multipliers and a Hex to 7-Segment display driver. Unfortunately, the complexity of the circuits used for the evaluation is not comparable to real world applications in both cases.

Petrlík [39] evolved approximate multiple constant multipliers with respect to multiple objectives by means of NSGA-II and CGP on functional level.

## 2.2.4 Design Acceleration

The evolutionary design is a very computationally demanding approach. In order to reduce the design time, one has to deal with the acceleration of the fitness function or search algorithm modifications. We will briefly survey relevant approaches in the context of CGP.

## Spatially Structured EAs

Spatially structured evolutionary algorithms have been intensively studied in the past and a variety of approaches differing in the used evolutionary algorithm or communication topology has emerged [2]. By introducing multiple populations evolving in parallel, one can increase the population diversity and thus make the EA more explorative leading to a higher probability of finding the global optimum for particular problems.

As the combinational circuit design is a very complex problem, the search space is generally rugged containing lots of local optima and thus the potential of exploiting parallel EA is high. Unfortunately, the absence of a crossover operator in CGP is a very limiting factor since most parallel models take advantage of combining genotypes from different isolated populations. Nevertheless, the model of isolated islands with migration of the best individuals in each population can be applied to CGP [20].

## Coevolutionary Algorithms

Often, the fitness in CGP is calculated over a set of *fitness cases* (e.g. when designing digital image filters). A fitness case corresponds to a representative situation in which the ability of a program to solve a problem can be evaluated. Each fitness case consists of potential program inputs and target values expected from a perfect solution as a response for these program inputs.

A set of fitness cases can be either a complete specification or just a small sample of the entire domain space. The choice of how many fitness cases (and which ones) to use is often crucial since whether or not an evolved solution will generalize over the entire domain depends on this choice. However, in the case of digital circuit evolution, it is necessary to verify whether a candidate  $n$ -input circuit generates correct responses for all possible fitness cases (input combinations, i.e.  $2^n$  assignments). It was shown that testing just a subset of  $2^n$  fitness cases does not lead to correctly working circuits [22].

Hillis [16] introduced an approach that can automatically evolve subsets of fitness cases concurrently with problem solution. He used a two-population coevolutionary algorithm (CoEA) in the task of minimal sorting network design. Subsets of test cases used to evaluate sorting networks evolved simultaneously with the sorting networks. Evolved sorting networks were used to evaluate the test cases subsets. The fitness of each sorting network was measured by its ability to correctly solve fitness cases while the fitness of the fitness cases subsets was better for those that could not be solved well by currently evolved sorting networks. This approach was recently used to evolve digital image filters [21].

Other CoEA approaches and techniques include compositional coevolution [12], indirectly encoded fitness predictors [11] or plastic fitness predictors [64].

Coevolutionary algorithms are traditionally used to evolve interactive behavior which is difficult to evolve with an absolute fitness function. The state of the art of coevolutionary algorithms has recently been summarized in [42].

## Parallelization

When designing combinational circuits, the CGP implementation usually must process all the  $2^{n_i}$  test vectors on the whole phenotype for the entire population of individuals and compare all the  $n_o$  outputs to the desired ones. In order to take advantage of modern superscalar out-of-order processors, the parallelism at various levels has to be employed and special attention to memory access policy has to be paid.

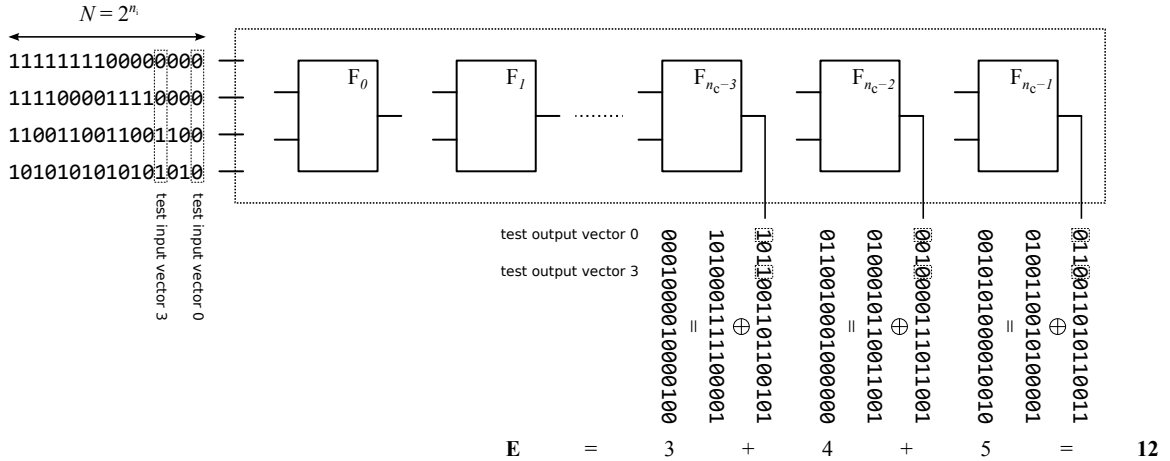


Figure 2.4: Parallel evaluation of a CGP individual – multiple test vectors can be evaluated in parallel using bitwise or vector operations. The hamming distance can be efficiently calculated by XORing the output value with the desired one and counting the number of ones.

- *Vectorization*: The most fundamental optimization we can apply is the bit-level parallelism. Instead of separate test vector processing, up to 64 test vectors can be processed in parallel on 64-bit processors thanks to bitwise operations (see Figure 2.4). Furthermore, by introducing the data-level parallelism using SIMD instructions, 128, 256 or even 512 test vectors can fit into the SSE, AVX or AVX-512 registers respectively.

Significant speed-up can be achieved by introducing the so called native implementation [20]. Instead of traversing the chromosome and computing the node outputs directly, the chromosome is compiled at first. The compiled program is then executed on each test vector for each individual in the population.

- *Thread Parallelism*: The most straightforward way of dividing the computations into multiple threads is to assign each thread a subset of the population and compute the fitness values in parallel. However, CGP uses a very small population, often much smaller than the number of physical cores present in today’s processors. Nevertheless, one can parallelize the fitness function in a different way, such as assigning each thread a portion of the test vectors [20].
- *General Purpose GPUs*: Recent advances in scientific computing have made it possible to use general purpose GPUs (GPGPUs) for parallel EAs. GPGPUs are low-cost, massively parallel, many-core processors. Although the parallelism of EAs is well suited for the single-program multiple-data based GPGPUs, there are many issues to be resolved such as the thread divergence caused by the randomness of EAs. The state of the art of EAs on GPGPUs has been recently summarized in [51].
- *Coprocessors*: Coprocessors have been mostly used to accelerate specific tasks, e.g. audio or video encoding/decoding, cryptography etc. Recently, Intel introduced a general purpose Many Integrated Core Architecture (Intel MIC). Intel Xeon Phi coprocessor is an example of this approach, it has been designed for applications that can exploit vector instructions and are scalable enough to efficiently run in a huge number

of threads [23]. Unlike GPGPUs, the user can exploit standard programming model and thus reuse a lot of code optimized for CPUs. However, to reach the maximum performance, one has to seriously deal with manual code optimizations [18].

- *Computer Clusters*: Spatially structured EAs are inherently suitable for running on computer clusters. The distributive nature of spatially structured EAs in combination with other complementary parallelization techniques enables to fully utilize multiple computing nodes. Communication is usually not a bottleneck, since the populations are evolving on individual nodes independently and exchange data occasionally [20].

## Hardware Accelerators

Reconfigurable hardware (i.e. FPGAs or hybrid platforms, such as Xilinx Zynq) offers a great possibility for accelerating computationally intensive applications. Recently, CGP has been accelerated by means of so called Virtual Reconfigurable Circuit (VRC) [45] or Dynamic Partial Reconfiguration (DPR) [44]. Multiple VRCs have been used to even increase the performance [21].

## Formal Methods

Computing the fitness function for complex digital circuits (i.e. circuits with more than 20 inputs) is not efficient. In the case of evolutionary optimization, the exact fitness value is often not needed, because the evolution starts with a fully working circuit and every destructive mutation is unwanted. Therefore, checking the output equivalence of the original and the candidate circuit is sufficient to perform in this case.

Recently, the fitness calculation has been sped up by introducing formal methods, e.g. based on the Boolean Satisfiability (SAT) problem [55] or the Binary Decision Diagrams (BDD) [56]. Although the fitness function is mostly based on Hamming distance [56], the latest published results suggest that formal methods can be used to calculate even more complex error metrics (e.g. the worst case error) [17].

- *SAT Solvers*: The problem of output equivalence can be easily transformed to the Boolean satisfiability problem, which can be then solved by means of standard tools (SAT solvers) [55].
- *Binary Decision Diagrams*: A BDD is a directed acyclic graph with one root and two terminal nodes that are referred to as '0' and '1'. The other (non-terminal) nodes are associated with a primary input variable and have exactly two outgoing edges corresponding to assigning the variable *true* or *false* truth value. Every path in a BDD is unique; if we find a path from the root node to the terminal node '1', then we have found a value assignment to the variables for which the function is evaluated to 1. A CGP individual can be represented by a BDD. When properly used, various error metrics can be computed [17].

## Chapter 3

# Research Summary

This chapter summarizes the research presented in the thesis. After a brief overview of the research process, the motivation and abstracts for each included paper are presented. Finally, a complete list of publications, research projects, grants and awards are listed.

### 3.1 Overview

The research presented in this thesis extends the previous research in several ways. The scalability of the evolutionary design method is improved by introducing a highly optimized parallel implementation of CGP. To address all demands placed by the hardware community, the method is extended to be multi-objective and the estimation accuracy of various circuit parameters is substantially improved. The thesis primarily deals with approximate circuits design, the performance of the method is demonstrated on several real-world problems.

The research started with a detailed analysis of the state of the art methods. It was shown in Chapter 2 that the evolutionary design methods suffer from low scalability and thus, a highly optimized CGP implementation was proposed and various acceleration techniques were analyzed in Paper I. The scalability of the implementation was then evaluated on several problems – design of combinational adders and multipliers (Paper I) and bent Boolean functions (Paper II). The design of the bent Boolean functions was a very complex problem with a high potential of parallelization and thus the Xeon Phi Coprocessor was utilized to further accelerate the design process (Paper III).

The work was then directed to the multi-objective design approach. The first version of multi-objective CGP was published in Paper IV. In Paper V, the method was improved by replacing the randomly generated initial population by a set of conventional circuits. The accuracy of the estimation of circuit parameters was enhanced by more accurate modeling of a real technology process library. The results were compared to a state of the art method and published as the EvoApprox8b library in Paper VII.

In Paper VI, the method was used to generate approximate circuits to be used in a TMR schema. Experimental results demonstrated that the evolutionary approach produced better solutions than the probabilistic approach developed by our colleagues from University Carlos III de Madrid.

## 3.2 Papers included in this thesis

### 3.2.1 Paper I

Radek Hrbáček and Lukáš Sekanina. **Towards Highly Optimized Cartesian Genetic Programming: From Sequential via SIMD and Thread to Massive Parallel Implementation.** In *GECCO '14 Proceedings of the 2014 conference on Genetic and evolutionary computation*. New York: Association for Computing Machinery, 2014, pp. 1015-1022. ISBN 978-1-4503-2662-9.

Author participation: 80%.  
Conference Rank: A1 (Qualis).

#### Abstract

Most implementations of Cartesian genetic programming (CGP) which can be found in the literature are sequential. However, solving complex design problems by means of genetic programming requires parallel implementations of search methods and fitness functions. This paper deals with the design of highly optimized implementations of CGP and their detailed evaluation in the task of evolutionary circuit design. Several sequential implementations of CGP have been analyzed and the effect of various additional optimizations has been investigated. Furthermore, the parallelism at the instruction, data, thread and process level has been applied in order to take advantage of modern processor architectures and computer clusters. Combinational adders and multipliers have been chosen to give a performance comparison with state of the art methods.

#### Contribution

As a highly optimized implementation of the evolutionary design method based on CGP was one of the first goals of the research, a deep analysis of possible optimization techniques was desirable. Such an analysis is covered within this paper. Although the results presented in this paper suggest that the most efficient approach, at least for complex circuits, is the native implementation (see Chapter 2), subsequent research revealed a weakness of this method – low flexibility in terms of function set modification. Therefore, the standard interpreted approach was preferred in further research.

*This work resulted in a very efficient CGP implementation capable of running on a wide range of computers – from single-core processors to supercomputers.*

### 3.2.2 Paper II

Radek Hrbáček and Václav Dvořák. **Bent Function Synthesis by Means of Cartesian Genetic Programming.** In *Parallel Problem Solving from Nature - PPSN XIII*. Heidelberg: Springer Verlag, 2014, LNCS vol. 8672, pp. 414-423. ISBN 978-3-319-10761-5.

Author participation: 80%.  
Conference Rank: A2 (Qualis).

#### Abstract

In this paper, a new approach to synthesize bent Boolean functions by means of Cartesian Genetic Programming (CGP) is proposed. Bent functions have important applications in

cryptography due to their high nonlinearity. However, they are very rare and their discovery using conventional brute force methods is not efficient enough. We show that by using CGP we can routinely design bent functions of up to 16 variables. The evolutionary approach exploits parallelism in both the fitness calculation and the search algorithm.

### Contribution

The proposed efficient CGP implementation presented in the previous papers can be applied to a wide range of applications. In this paper, the method is used to find Boolean functions with the highest nonlinearity, which are very rare, but very important for networking and cryptography.

*This work resulted in a new efficient approach for finding Boolean functions with high nonlinearity. It was the first time CGP was successfully used for this purpose and the paper gave an impulse for other researchers to further develop this approach [41, 40]. The results were awarded by the bronze medal at the Humies competition (2014).*

### 3.2.3 Paper III

Radek Hrbáček. **Bent Functions Synthesis on Xeon Phi Coprocessor.** In *Mathematical and Engineering Methods in Computer Science*. Heidelberg: Springer Verlag, 2014, LNCS vol. 8934, pp. 88-99. ISBN 978-3-319-14895-3.

Author participation: 100%.

### Abstract

A new approach to synthesize bent Boolean functions by means of Cartesian Genetic Programming (CGP) has been proposed recently. Bent functions have important applications in cryptography due to their high nonlinearity. However, they are very rare and their discovery using conventional brute force methods is not efficient enough. In this paper, a new parallel implementation is proposed and the performance is evaluated on the Intel Xeon Phi Coprocessor.

### Contribution

The computational demands of the method proposed in Paper II are very high. The fitness (nonlinearity) evaluation time grows exponentially with the number of variables. However, there is a great potential of parallelization, even higher than in the case of the fitness function based on Hamming distance. This paper deals with the implementation and optimization of the method for running on the Intel Xeon Phi Coprocessor. The implementation is highly parallel and allows to utilize all 60 cores of the coprocessor by running 240 threads.

*This work resulted in a significant speedup and an increase in complexity of the bent functions designed using the proposed evolutionary design method – up to 18 variable bent functions were found.*

### 3.2.4 Paper IV

Radek Hrbáček. **Parallel Multi-Objective Evolutionary Design of Approximate Circuits.** In *GECCO '15 Proceedings of the 2015 conference on Genetic and evolutionary*



*computation*. New York: Association for Computing Machinery, 2015, pp. 687-694. ISBN 978-1-4503-3472-3.

Author participation: 100%.  
Conference Rank: A1 (Qualis).

## Abstract

Evolutionary design of digital circuits has been well established in recent years. Besides correct functionality, the demands placed on current circuits include the area of the circuit and its power consumption. By relaxing the functionality requirement, one can obtain more efficient circuits in terms of the area or power consumption at the cost of an error introduced to the output of the circuit. As a result, a variety of trade-offs between error and efficiency can be found. In this paper, a multi-objective evolutionary algorithm for the design of approximate digital circuits is proposed. The scalability of the evolutionary design has been recently improved using parallel implementation of the fitness function and by employing spatially structured evolutionary algorithms. The proposed multi-objective approach uses Cartesian Genetic Programming for the circuit representation and a modified NSGA-II algorithm. Multiple isolated islands are evolving in parallel and the populations are periodically merged and new populations are distributed across the islands. The method is evaluated in the task of approximate arithmetical circuits design.

## Contribution

Since the most important goal of the thesis was to develop an automated design method capable of multi-objective evolutionary design, an extension to the standard CGP was needed. This paper introduces such an extension. The approach is based on the NSGA-II algorithm, but several modifications were required to adapt the algorithm for CGP. The implementation preserves all benefits of the single-objective parallel CGP implementation – even a new multi-objective island model was introduced to utilize computer clusters.

*As a result of this work, the CGP implementation was extended with the multi-objective approach. The method was used to design approximate arithmetical circuits from scratch.*

### 3.2.5 Paper V

Radek Hrbáček, Vojtěch Mrázek and Zdeněk Vašíček. **Automatic Design of Approximate Circuits by Means of Multi-Objective Evolutionary Algorithms**. In *Proceedings of the 11th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*. Istanbul: Istanbul Sehir University, 2016, pp. 239-244. ISBN 978-1-5090-0335-8.

Author participation: 50%.

## Abstract

Recently, power efficiency has become the most important parameter of many real circuits. At the same time, a wide range of applications capable of tolerating imperfections has spread out especially in multimedia. Approximate computing, an emerging paradigm, takes advantage of relaxed functional requirements to make computer systems more efficient in terms of energy consumption, speed or complexity. As a result, a variety of trade-offs

between error and efficiency can be found. In this paper, a design method based on a multi-objective evolutionary algorithm is proposed. For a given circuit, the method is able to produce a set of Pareto optimal solutions in terms of the error, power consumption and delay. The proposed design method uses Cartesian Genetic Programming for the circuit representation and a modified NSGA-II algorithm for design space exploration. The method is used to design Pareto optimal approximate versions of arithmetic circuits such as multipliers and adders.

### Contribution

In Paper IV, a new multi-objective CGP implementation was proposed. However, the estimation of circuit parameters was simplified and the complexity of evolved circuits was relatively low (4-bit combinational adders and multipliers). In this paper, we proposed to use a set of conventional arithmetic circuits of various architectures as the initial population instead of randomly generated initial population. Moreover, we implemented more accurate estimation of the power consumption, propagation delay and area of the circuits based on an existing technology process library.

*As a result of this work, hundreds of approximate 8-bit adders and multipliers were designed with respect to three objectives – mean relative error, power consumption and delay.*

### 3.2.6 Paper VI

Antonio José Sánchez-Clemente, Luis Entrena, Radek Hrbáček and Lukáš Sekanina. **Error Mitigation using Approximate Logic Circuits: A Comparison of Probabilistic and Evolutionary Approaches.** *IEEE Transactions on Reliability*. 2016, vol. 65, no. 4, pp. 1871-1883. ISSN 0018-9529.

Author participation: 25 %.  
Impact factor: 2.287.

### Abstract

Technology scaling poses an increasing challenge to the reliability of digital circuits. Hardware redundancy solutions, such as Triple Modular Redundancy, produce very high area overhead, so partial redundancy is often used to reduce the overheads. Approximate logic circuits provide a general framework for optimized mitigation of errors arising from a broad class of failure mechanisms, including transient, intermittent and permanent failures. However, generating an optimal redundant logic circuit that is able to mask the faults with the highest probability while minimizing the area overheads is a challenging problem. In this work we propose and compare two new approaches to generate approximate logic circuits to be used in a TMR schema. The probabilistic approach approximates a circuit in a greedy manner based on a probabilistic estimation of the error. The evolutionary approach can provide radically different solutions that are hard to reach by other methods. By combining these two approaches, the solution space can be explored in depth. Experimental results demonstrate that the evolutionary approach can produce better solutions, but the probabilistic approach is close. On the other hand, these approaches provide much better scalability than other existing partial redundancy techniques.

## Contribution

In this work, we provided the CGP based method for the design of approximate logic circuits to be used in partially protected circuits in a TMR schema. The advantage of the proposed technique was the ability to generate good trade-off solutions between the reliability and the area overhead of the circuit.

*This work resulted in a new method of designing approximate logic circuits to be used in a TMR schema. In comparison with other existing partial redundancy techniques, the proposed method provided much better scalability.*

### 3.2.7 Paper VII

Vojtěch Mrázek, Radek Hrbáček, Zdeněk Vašíček and Lukáš Sekanina. **EvoApprox8b: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods.** In *Proc. of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Lausanne: European Design and Automation Association, 2017, pp. 258-261. ISBN 978-3-9815370-9-3.

Author participation: 25%.  
Conference Rank: A1 (Qualis).

## Abstract

Approximate circuits and approximate circuit design methodologies attracted a significant attention of researchers as well as industry in recent years. In order to accelerate the approximate circuit and system design process and to support a fair benchmarking of circuit approximation methods, we propose a library of approximate adders and multipliers called EvoApprox8b. This library contains 430 nondominated 8-bit approximate adders created from 13 conventional adders and 471 non-dominated 8-bit approximate multipliers created from 6 conventional multipliers. These implementations were evolved by a multi-objective Cartesian genetic programming. The EvoApprox8b library provides Verilog, Matlab and C models of all approximate circuits. In addition to standard circuit parameters, the error is given for seven different error metrics. The EvoApprox8b library is available at: [www.fit.vutbr.cz/research/groups/ehw/approxlib](http://www.fit.vutbr.cz/research/groups/ehw/approxlib).

## Contribution

This paper compares the results of Paper V to a conventional design method and presents the evolved circuits as a library of circuits (synthesized using Synopsys Design Compiler for 180nm and 45nm technologies).

*As a result of this paper, a free library of approximate arithmetic circuits was proposed. The library provides several representations of the circuits along with their properties. One can filter the circuits by the properties. The work was awarded by the Best IP Award at the Design, Automation and Test in Europe (DATE) conference 2017.*

## 3.3 List of Other Publications

- Radek Hrbáček. Simulation Based Neural Motion Planner Learning. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume 1*. Brno: NOVAPRESS s.r.o.,

2011. pp. 189-191. ISBN: 978-80-214-4271-9.  
Author participation: 100 %.
- Radek Hrbáček. Introduction to Compressive Sampling. In *Proceedings of the 17th Conference STUDENT EEICT 2011 Volume 1*. Brno: NOVAPRESS s.r.o., 2011. pp. 45-47. ISBN: 978-80-214-4271-9.  
Author participation: 100 %.
  - Jan Hrbáček, Radek Hrbáček and Stanislav Věchet. Modular Control System Architecture for a Mobile Robot. In *Proceedings of the 17th international conference Engineering Mechanics 2011*. 1. Prague: Institute of Thermomechanics, Academy of Sciences of the Czech Republic, 2011. pp. 211-214. ISBN: 978-80-87012-33-8.  
Author participation: 33 %.
  - Radek Hrbáček, Pavel Rajmic, Vítězslav Veselý and Jan Špiřík. Introduction to sparse signal representations. *Elektrorevue – internet journal (<http://www.elektrorevue.cz>)*, 2011, vol. 2011, no. 50, pp. 1-10. ISSN: 1213-1539.  
Author participation: 40 %.
  - Radek Hrbáček, Pavel Rajmic, Vítězslav Veselý and Jan Špiřík. Sparse signal representations: compressed sensing. *Elektrorevue – internet journal (<http://www.elektrorevue.cz>)*, 2011, vol. 2011, no. 67, pp. 1-6. ISSN: 1213-1539.  
Author participation: 44 %.
  - Jiri Krejsa, Stanislav Věchet, Jan Hrbáček, Tomáš Ripel, Vítězslav Ondroušek, Radek Hrbáček and Petr Schreiber. Presentation robot Advée. *Engineering Mechanics*, 2012, vol. 18, no. 5/6, pp. 307-322. ISSN: 1802-1484.  
Author participation: 5 %.
  - Radek Hrbáček. Hardware Platform for Coevolutionary Design. In *Proceedings of the 19th Conference STUDENT EEICT 2013 Volume 2*. Brno: LITERA, 2013. pp. 279-281. ISBN: 978-80-214-4694-6.  
Author participation: 100 %.
  - Radek Hrbáček and Michaela Šikulová. Coevolutionary Cartesian Genetic Programming in FPGA. In *Advances in Artificial Life, ECAL 2013, Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems*. Cambridge, US: MIT, 2013. pp. 431-438. ISBN 978-0-262-31709-2.  
Author participation: 60 %  
Conference Rank: B1 (Qualis).
  - Jiří Toman and Radek Hrbáček. Redundant Control Algorithm for a Brushless DC Motor. In *Electrical Drives and Power Electronics*. Košice: Technical University of Košice, 2015, pp. 117-124. ISBN: 978-80-553-2208-7.  
Author participation: 50 %.
  - Filip Vaverka, Radek Hrbáček and Lukáš Sekanina. Evolving Component Library for Approximate High Level Synthesis. In *2016 IEEE Symposium Series on Computational Intelligence*. Athens: IEEE Computational Intelligence Society, 2016, pp. 1-8. ISBN 978-1-5090-4240-1.  
Author participation: 25 %  
Conference Rank: B5 (Qualis).

## 3.4 Research Projects and Grants

### 3.4.1 Czech Science Foundation

- GA16-17538S *Relaxed equivalence checking for approximate computing*.  
Co-investigator.
- GA14-04197S *Advanced Methods for Evolutionary Design of Complex Digital Circuits*.  
Co-investigator.

### 3.4.2 Anselm & Salomon Supercomputer Allocations

- OPEN-8-4 *Multi-objective Approximate Circuit Design on Computer Cluster*.  
500 000 core hours.  
Primary investigator.
- IT4T-10-4 *Evolutionary Design of Cryptographic Boolean Functions*.  
300 000 core hours.  
Investigator.
- IT4T-9-2 *Approximate circuit design on computer cluster*.  
300 000 core hours.  
Investigator.
- IT4I-7-6 *Evolutionary design on computer cluster*.  
80 000 core hours.  
Investigator.
- IT4I-5-9 *Evolvable hardware on computer cluster*.  
75 000 core hours.  
Investigator.

## 3.5 Awards

- Special Prize IT4Innovations (Joseph Fourier Prize 2017).
- Best Interactive Presentation (DATE 2017).
- Bronze medal in Humies competition (GECCO 2014).
- BUT rector's award for excellent master study and scientific research results.
- 1. prize in the competition *ICT Master thesis of the year 2013* awarded for the master thesis *Coevolutionary Algorithm in FPGA*.
- FIT BUT dean's award for the master thesis *Coevolutionary Algorithm in FPGA*.
- 1. prize in the *Student EEICT 2013* competition awarded for the paper *Hardware Platform for Coevolutionary Design*.

## Chapter 4

# Discussion and Conclusions

This chapter discusses and summarizes the results presented in the thesis and gives conclusions and possible directions for future work.

### 4.1 The Approach

An analysis of the state of the art evolutionary design methods and a study of possible acceleration techniques have been performed within the research presented in this thesis. In order to accomplish the given research objectives, a new highly optimized implementation of Cartesian Genetic Programming was proposed. The next step was to extend the CGP to support the design with respect to multiple objectives – a modified NSGA-II algorithm was used for this purpose. The performance of the implementation was evaluated in multiple different applications, in particular (approximate) combinational arithmetic circuits design, bent Boolean functions discovery, approximate logic circuits for TMR schema and others. The experiments were conducted on computers operated by two organizations – MetaCentrum VO and IT4Innovations (supercomputers Anselm, Salomon). All particular research steps are presented in the description of relevant scientific papers mentioned in Chapter 3.

### 4.2 Software Outcomes

During the research, a new CGP implementation was developed and actively used. This implementation was improved and extended with new features step by step to maintain the universality of the tool as great as possible, except for a few particular cases (e.g. Intel Xeon Phi coprocessor implementation).

The CGP tool is implemented using C/C++. Thread level parallelism is based on the OpenMP library and the island model utilizes MPI message passing communication approach. The tool is a command line utility, where all parameters are passed to the tool using command line arguments. The desired circuit functionality can be specified either as a truth table or by importing a PLA (Programmable Logic Array) file. The evolutionary design can start either from scratch (randomly generated initial population) or a set of CGP chromosomes can be imported. The results of the evolutionary design can be exported to various output formats:

- chromosome file – CGP chromosome representation,
- VHDL, Verilog and BENCH – hardware description languages,

- C/C++ – logic and arithmetic representation.

The CGP implementation is able to model various technology processes as listed in Table 4.1. The simplest one implements general logic *gates* (BUF, NOT, AND, OR, XOR, NAND, NOR, XNOR, 1, 0) and the estimation of the area, power consumption and delay is very rough.

A much more realistic technology library *osu180* is based on a real technology process library. It is available in four variants; from a set of up to 4-input gates to a subset containing just 2-input gates.

The last technology supported by the tool is the *look-up table* (LUT) technology with 3-6 node inputs. The LUT-level implementation was optimized by the tool itself (on gate-level) by generating highly optimized implementations of all 3-input and 4-input Boolean functions. These optimized functions were exported to C/C++ and compiled together into a new LUT-level implementation. This implementation is intended for the design and approximation of digital circuits running on FPGAs. The LUT-based approach is currently prepared for publication.

Technology	Node inputs $n_{ni}$	Node outputs $n_{no}$	Node functions $n_f$
<i>gate</i>	2	1	10
<i>osu180</i>	4	2	17
	3	2	15
	2	2	9
	2	1	8
<i>look-up table</i>	3	1	256
	4	1	65536
	5	1	4294967296
	6	1	$2^{64}$

Table 4.1: List of supported CGP circuit primitives.

All the technologies can be used with the same algorithms. There are basically two different approaches supported by the tool. The *single-objective* algorithm is based on standard CGP. The fitness function can be composed of multiple objectives; when reaching the maximum of the first objective, the next one is included into the fitness value calculation and so on. The *multi-objective* algorithm is an extension of CGP with a modified NSGA-II algorithm. All objectives are optimized simultaneously. Both algorithms support constraining the individual objectives by entering a minimum or/and a maximum value. The supported fitness functions are as follows:

- Hamming distance, on/off-set Hamming distance
- mean absolute error (MAE), mean squared error (MSE), mean relative error (MRE)
- worst case error (WCE), worst case relative error (WCRE)
- variance of absolute error (VAE)
- error probability (EP)
- area, delay, power, power-delay product (PDP)

Both, single-objective and multi-objective algorithms can be executed on multiple isolated islands. On each island, a separate population of individuals is evolving independently of the others until a predefined number of generations or a time interval is reached. In the single-objective case, the best individual of all islands is determined and the evolutionary process is restarted on all islands with the best individual as a seed. In the multi-objective case, the populations of all islands are merged, a global Pareto front is formed and the individual populations are seeded with individuals from the front.

### 4.3 Contributions

The main contributions of this thesis can be summarized as follows:

- Development of a new highly optimized parallel implementation of CGP that is much more scalable than the state of the art implementations and thus applicable for complex problems.
- Extension of CGP with new function sets based on real technology processes (180nm CMOS library, LUTs).
- Extension of CGP with multi-objective design capability.
- Evolved comprehensive library of 8-bit approximate combinational adders and multipliers that can be used for benchmarking of approximate design methods or as a component library for high-level approximate circuit design.
- Evolved bent Boolean functions of up to 18 variables that are more difficult instances than previous solutions.
- Evolved approximate combinational circuits for TMR schema that show better properties than the circuits obtained by a conventional probabilistic method.

The aforementioned contributions are important for approximate computing as well as for other research areas dealing with digital circuits or related technologies. The work has been awarded by both EAs (Bronze medal in Humies competition) and hardware (DATE Best IP Award) community.

### 4.4 Future Work

The scalability of the CGP method was significantly improved, as presented in this thesis. However, a great potential in further increasing the complexity of problems solved by CGP resides in using formal verification methods in the fitness function. Until recently, these methods could only be used to check if two circuits are equivalent in terms of their output responses. The latest results suggest that some formal methods (e.g. binary decision diagrams) could be used to directly determine an arithmetic distance or to check if a circuit satisfies an arithmetic constraint [17]. Integrating these methods to the proposed CGP implementation is straightforward – only the fitness function has to be re-designed.

The EvoApprox8b library presented in this thesis can be directly used as a component library for a high level approximate circuit synthesis algorithm. For example, a multi-objective evolutionary algorithm has been developed for this purpose [58]. This way, the scalability of the method can be further improved.



The LUT-level CGP implementation mentioned in this chapter promises to further extend the application area of the proposed method to FPGA devices. This will need further experiments and it will be definitely an interesting and challenging continuation of the research started in this thesis.

Another issue that should also be addressed is high computation time of the method in comparison with conventional synthesis methods. Nevertheless, the evolutionary design method is still valuable considering the fact that better results can be obtained in comparison with the conventional methods. Besides, the performance and parallelism of computers is growing and the EA based methods can take advantage of that, as shown in this thesis.

# Bibliography

- [1] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford, UK: Oxford University Press, 1996. ISBN: 0-19-509971-0.
- [2] E. Cantu-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 2000. ISBN: 0792372212.
- [3] M. Carbin, S. Misailovic, and M. C. Rinard. Verifying Quantitative Reliability for Programs That Execute on Unreliable Hardware. In: *Proceedings of the 2013 ACM SIGPLAN International Conference on Object Oriented Programming Systems Languages & Applications*. OOPSLA '13. Indianapolis, Indiana, USA: ACM, 2013, pp. 33–52. ISBN: 978-1-4503-2374-1.
- [4] P. Cattani and C. Johnson. ME-CGP: Multi Expression Cartesian Genetic Programming. In: *Evolutionary Computation (CEC), 2010 IEEE Congress on*. 2010, pp. 1–6.
- [5] A. Chandrakasan and R. Brodersen. Minimizing power consumption in digital CMOS circuits. In: *Proceedings of the IEEE* 83.4 (1995), pp. 498–523. ISSN: 0018-9219.
- [6] V. Chippa et al. Analysis and characterization of inherent application resilience for approximate computing. In: *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*. 2013, pp. 1–9.
- [7] J. Clegg, J. A. Walker, and J. F. Miller. A new crossover technique for Cartesian genetic programming. In: *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. Vol. 2. London: ACM Press, 2007, pp. 1580–1587.
- [8] J. Cong and K. Minkovich. Optimality Study of Logic Synthesis for LUT-based FPGAs. In: *Proceedings of the 2006 ACM/SIGDA 14th International Symposium on Field Programmable Gate Arrays*. FPGA '06. Monterey, California, USA: ACM, 2006, pp. 33–40. ISBN: 1-59593-292-5.
- [9] K. Deb et al. *A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II*. Piscataway, NJ, USA, Apr. 2002.
- [10] K. Deb and D. Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN: 047187339X.
- [11] M. Drahošová, J. Hulva, and L. Sekanina. Indirectly Encoded Fitness Predictors Co-evolved with Cartesian Programs. In: *Genetic Programming*. LNCS 9025. Berlin, DE: Springer International Publishing, 2015, pp. 113–125. ISBN: 978-3-319-16500-4.

- [12] M. Drahošová, G. Komjáthy, and L. Sekanina. Towards Compositional Coevolution in Evolutionary Circuit Design. In: *2014 IEEE International Conference on Evolvable Systems Proceedings*. Piscataway, US: Institute of Electrical and Electronics Engineers, 2014, pp. 157–164. ISBN: 978-1-4799-4479-8.
- [13] V. Gupta et al. Low-Power Digital Signal Processing Using Approximate Adders. In: *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 32.1 (2013), pp. 124–137. ISSN: 0278-0070.
- [14] S. L. Harding, J. F. Miller, and W. Banzhaf. Self-modifying cartesian genetic programming. In: *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*. Vol. 1. London: ACM Press, 2007, pp. 1021–1028.
- [15] J. Hilder, J. Walker, and A. Tyrrell. Use of a multi-objective fitness function to improve cartesian genetic programming circuits. In: *Adaptive Hardware and Systems (AHS), 2010 NASA/ESA Conference on*. 2010, pp. 179–185.
- [16] W. D. Hillis. Co-evolving parasites improve simulated evolution as an optimization procedure. In: *Physica D* 42.1 (1990), pp. 228–234.
- [17] L. Holík et al. Towards Formal Relaxed Equivalence Checking in Approximate Computing Methodology. In: *2nd Workshop on Approximate Computing (WAPCO 2016)*. Prague, 2016, pp. 1–6.
- [18] R. Hrbacek. Bent Functions Synthesis on Xeon Phi Coprocessor. In: *Mathematical and Engineering Methods in Computer Science*. LNCS 8934. Heidelberg, DE: Springer Verlag, 2014, pp. 88–99. ISBN: 978-3-319-14895-3.
- [19] R. Hrbacek. Parallel Multi-Objective Evolutionary Design of Approximate Circuits. In: *GECCO '15 Proceedings of the 2015 conference on Genetic and evolutionary computation*. Under review. New York, US: Association for Computing Machinery, 2015, p. 8.
- [20] R. Hrbacek and L. Sekanina. Towards Highly Optimized Cartesian Genetic Programming: From Sequential via SIMD and Thread to Massive Parallel Implementation. In: *GECCO '14 Proceedings of the 2014 conference on Genetic and evolutionary computation*. New York, US: Association for Computing Machinery, 2014, pp. 1015–1022. ISBN: 978-1-4503-2662-9.
- [21] R. Hrbacek and M. Sikulova. Coevolutionary Cartesian Genetic Programming in FPGA. In: *Advances in Artificial Life, ECAL 2013, Proceedings of the Twelfth European Conference on the Synthesis and Simulation of Living Systems*. MIT Press, 2013, pp. 431–438.
- [22] K. Imamura, J. A. Foster, and A. W. Krings. The Test Vector Problem and Limitations to Evolving Digital Circuits. In: *Proc. of the Second NASA/DoD Workshop on Evolvable Hardware*. IEEE Computer Society, 2000, pp. 75–79.
- [23] J. Jeffers and J. Reinders. *Intel® Xeon Phi coprocessor high-performance programming*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013. ISBN: 978-0-12-410414-3.
- [24] K. Kannappan et al. Analyzing a Decade of Human-Competitive (“HUMIE”) Winners: What Can We Learn? In: *Genetic Programming Theory and Practice XII*. Ed. by R. Riolo, W. P. Worzel, and M. Kotanchek. Cham: Springer International Publishing, 2015, pp. 149–166. ISBN: 978-3-319-16030-6.

- [25] G. Karakonstantis and K. Roy. Voltage over-scaling: A cross-layer design perspective for energy efficient systems. In: *Circuit Theory and Design (ECCTD), 2011 20th European Conference on*. 2011, pp. 548–551.
- [26] P. Kaufmann and M. Platzner. Toward Self-adaptive Embedded Systems: Multi-objective Hardware Evolution. In: *Proceedings of the 20th International Conference on Architecture of Computing Systems*. ARCS'07. Zurich, Switzerland: Springer-Verlag, 2007, pp. 199–208. ISBN: 978-3-540-71267-1.
- [27] T. Knieper et al. On Robust Evolution of Digital Hardware. In: *Biologically-Inspired Collaborative Computing: IFIP 20th World Computer Congress, Second IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing, September 8–9, 2008, Milano, Italy*. Ed. by M. Hinchey et al. Boston, MA: Springer US, 2008, pp. 213–222. ISBN: 978-0-387-09655-1.
- [28] P. Krause and I. Polian. Adaptive voltage over-scaling for resilient applications. In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*. 2011, pp. 1–6.
- [29] P. Kulkarni, P. Gupta, and M. Ercegovac. Trading Accuracy for Power with an Underdesigned Multiplier Architecture. In: *VLSI Design (VLSI Design), 2011 24th International Conference on*. 2011, pp. 346–351.
- [30] S.-L. Lu. Speeding up processing with approximation circuits. In: *Computer* 37.3 (2004), pp. 67–73. ISSN: 0018-9162.
- [31] J. F. Miller, ed. *Cartesian Genetic Programming*. Natural Computing Series. Berlin, DE: Springer Verlag, 2011, p. 334. ISBN: 978-3-642-17309-7.
- [32] J. F. Miller and S. L. Smith. Redundancy and computational efficiency in Cartesian genetic programming. In: *Evolutionary Computation, IEEE Transactions on* 10.2 (2006), pp. 167–174. ISSN: 1089-778X.
- [33] S. Mittal. A Survey of Techniques for Approximate Computing. In: *ACM Comput. Surv.* 48.4 (Mar. 2016), 62:1–62:33. ISSN: 0360-0300.
- [34] V. Mrázek and Z. Vašíček. Evolutionary Design of Transistor Level Digital Circuits using Discrete Simulation. In: *Genetic Programming, 18th European Conference, EuroGP 2015*. LCNS 9025. Berlin, DE: Springer International Publishing, 2015, pp. 66–77. ISBN: 978-3-319-16500-4.
- [35] V. Mrázek et al. EvoApprox8b: Library of Approximate Adders and Multipliers for Circuit Design and Benchmarking of Approximation Methods. In: *Proc. of the 2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. Lausanne, CH: European Design and Automation Association, 2017, pp. 258–261. ISBN: 978-3-9815370-9-3.
- [36] K. Nepal et al. ABACUS: A technique for automated behavioral synthesis of approximate computing circuits. In: *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. 2014, pp. 1–6.
- [37] K. Nepal et al. Automated High-Level Generation of Low-Power Approximate Computing Circuits. In: *IEEE Transactions on Emerging Topics in Computing* PP.99 (2017), pp. 1–1. ISSN: 2168-6750.

- [38] J. Nevoral, R. Růžička, and V. Mrázek. Evolutionary Design of Polymorphic Gates Using Ambipolar Transistors. In: *2016 IEEE Symposium Series on Computational Intelligence*. Athens, GR: Institute of Electrical and Electronics Engineers, 2016, pp. 1–8. ISBN: 978-1-5090-4240-1.
- [39] J. Petrlik and L. Sekanina. Multiobjective evolution of approximate multiple constant multipliers. In: *IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems 2013*. Brno, CZ: IEEE Computer Society, 2013, pp. 116–119. ISBN: 978-1-4673-6133-0.
- [40] S. Picek et al. Cryptographic Boolean functions: One output, many design criteria. In: *Applied Soft Computing Journal* 40 (2016). cited By 6, pp. 635–653.
- [41] S. Picek et al. Evolutionary algorithms for boolean functions in diverse domains of cryptography. In: *Evolutionary Computation* 24.4 (2016). cited By 0, pp. 667–694.
- [42] E. Popovici et al. Coevolutionary Principles. In: *Handbook of Natural Computing*. Springer Berlin Heidelberg, 2012, pp. 987–1033. ISBN: 978-3-540-92909-3.
- [43] A. Ranjan et al. ASLAN: Synthesis of approximate sequential circuits. In: *Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014*. 2014, pp. 1–6.
- [44] R. Salvador et al. Implementation Techniques for Evolvable HW Systems: Virtual vs. Dynamic Reconfiguration. In: *Proc. of the 22nd International Conference on Field Programmable Logic and Applications (FPL)*. Oslo, NO: IEEE Computer Society, 2012, pp. 547–550. ISBN: 978-1-4673-2257-7.
- [45] L. Sekanina. *Evolvable Components - From Theory to Hardware Implementations*. Natural Computing Series. Berlin, DE: Springer Verlag, 2003, p. 194. ISBN: 3-540-40377-9.
- [46] L. Sekanina. Evolvable hardware. In: *Handbook of Natural Computing*. Berlin, DE: Springer Verlag, 2012, pp. 1657–1705. ISBN: 978-3-540-92909-3.
- [47] L. Sekanina and Z. Vasicek. Evolutionary Computing in Approximate Circuit Design and Optimization. In: *1st Workshop on Approximate Computing (WAPCO 2015)*. Amsterdam, 2015, pp. 1–6.
- [48] G. Toscano Pulido and C. A. Coello Coello. The Micro Genetic Algorithm 2: Towards Online Adaptation in Evolutionary Multiobjective Optimization. In: *Evolutionary Multi-Criterion Optimization: Second International Conference, EMO 2003, Faro, Portugal, April 8–11, 2003. Proceedings*. Ed. by C. M. Fonseca et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 252–266. ISBN: 978-3-540-36970-7.
- [49] M. Trefzer and A. Tyrrell. *Evolvable Hardware: From Practice to Application*. Natural Computing Series. Springer Berlin Heidelberg, 2015. ISBN: 9783662446164.
- [50] M. Trefzer et al. Operational Amplifiers: An Example for Multi-objective Optimization on an Analog Evolvable Hardware Platform. In: *Proceedings of the 6th International Conference on Evolvable Systems: From Biology to Hardware*. ICES’05. Sitges, Spain: Springer-Verlag, 2005, pp. 86–97. ISBN: 3-540-28736-1, 978-3-540-28736-0.
- [51] S. Tsutsui and P. Collet. *Massively Parallel Evolutionary Computation on GPGPUs*. Natural Computing Series. Springer Berlin Heidelberg, 2013. ISBN: 9783642379581.

- [52] A. J. Turner and J. F. Miller. Recurrent Cartesian Genetic Programming. English. In: *Parallel Problem Solving from Nature – PPSN XIII*. Vol. 8672. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 476–486. ISBN: 978-3-319-10761-5.
- [53] Z. Vasicek and L. Sekanina. Evolutionary Approach to Approximate Digital Circuits Design. In: *IEEE Transactions on Evolutionary Computation* 99.99 (2015), pp. 1–13. ISSN: 1089-778X.
- [54] Z. Vasicek and L. Sekanina. Evolutionary Design of Approximate Multipliers Under Different Error Metrics. In: *17th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*. Warsaw, PL: IEEE Computer Society, 2014, pp. 135–140. ISBN: 978-1-4799-4558-0.
- [55] Z. Vasicek and L. Sekanina. Formal Verification of Candidate Solutions for Post-Synthesis Evolutionary Optimization in Evolvable Hardware. In: *Genetic Programming and Evolvable Machines* 12.3 (2011), pp. 305–327.
- [56] Z. Vasicek and L. Sekanina. How to Evolve Complex Combinational Circuits From Scratch? In: *2014 IEEE International Conference on Evolvable Systems Proceedings*. Piscataway, US: Institute of Electrical and Electronics Engineers, 2014, pp. 133–140. ISBN: 978-1-4799-4480-4.
- [57] Z. Vasicek and L. Sekanina. On Area Minimization of Complex Combinational Circuits Using Cartesian Genetic Programming. In: *2012 IEEE World Congress on Computational Intelligence*. CA, US: Institute of Electrical and Electronics Engineers, 2012, pp. 2379–2386. ISBN: 978-1-4673-1508-1.
- [58] F. Vaverka, R. Hrbáček, and L. Sekanina. Evolving Component Library for Approximate High Level Synthesis. In: *2016 IEEE Symposium Series on Computational Intelligence*. Athens, GR: IEEE Computational Intelligence Society, 2016, pp. 1–8. ISBN: 978-1-5090-4240-1.
- [59] S. Venkataramani et al. SALSA: Systematic logic synthesis of approximate circuits. In: *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*. 2012, pp. 796–801.
- [60] S. Venkataramani, K. Roy, and A. Raghunathan. Substitute-and-simplify: A unified design paradigm for approximate and quality configurable circuits. In: *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*. 2013, pp. 1367–1372.
- [61] R. Venkatesan et al. MACACO: Modeling and analysis of circuits for approximate computing. In: *Computer-Aided Design (ICCAD), 2011 IEEE/ACM International Conference on*. 2011, pp. 667–673.
- [62] J. A. Walker and J. F. Miller. Embedded Cartesian Genetic Programming and the Lawnmower and Hierarchical-if-and-only-if Problems. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. GECCO '06. Seattle, Washington, USA: ACM, 2006, pp. 911–918. ISBN: 1-59593-186-4.
- [63] J. A. Walker, J. F. Miller, and R. Cavill. A Multi-chromosome Approach to Standard and Embedded Cartesian Genetic Programming. In: *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. GECCO '06. Seattle, Washington, USA: ACM, 2006, pp. 903–910. ISBN: 1-59593-186-4.

- [64] M. Wiglasz and M. Drahošová. Plastic Fitness Predictors Coevolved with Cartesian Programs. In: *19th European Conference on Genetic programming*. LNCS 9594. Berlin, DE: Springer International Publishing, 2016, pp. 164–179. ISBN: 978-3-319-30667-4.
- [65] H. Yu, Y. Ha, and J. Wang. Quality Optimization of Resilient Applications Under Temperature Constraints. In: *Proceedings of the Computing Frontiers Conference*. CF'17. Siena, Italy: ACM, 2017, pp. 9–16. ISBN: 978-1-4503-4487-6.
- [66] T. Yu and J. Miller. Finding Needles in Haystacks Is Not Hard with Neutrality. English. In: *Genetic Programming*. Vol. 2278. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, pp. 13–25. ISBN: 978-3-540-43378-1.
- [67] E. Zitzler, M. Laumanns, and L. Thiele. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*. Tech. rep. 2001.

# Appendix A

## Curriculum Vitae

### A.1 Education

- 2013–Present **Ph.D. of Computer Science and Engineering**  
Faculty of Information Technology, Brno University of Technology  
Supervisor: prof. Ing. Lukáš Sekanina, Ph.D.
- 2011–2013 **Master of Computer and Embedded Systems**  
Faculty of Information Technology, Brno University of Technology  
*Coevolutionary Algorithm in FPGA*  
Supervisor: Ing. Michaela Šikulová  
Passed with honour, acquired title Ing.
- 2011–2013 **Master of Communications and Informatics**  
Faculty of Electrical Engineering and Communications, Brno University of Technology  
*Sparse Signal Representation Application in Nuclear Magnetic Resonance*  
Supervisor: Mgr. Pavel Rajmic, Ph.D.  
Passed with honour, acquired title Ing.
- 2008–2011 **Bachelor of Information Technology**  
Faculty of Information Technology, Brno University of Technology  
*On-Chip Debugger Generator*  
Supervisor: prof. Ing. Tomáš Hruška, CSc.  
Passed with honour, acquired title Bc.
- 2008–2011 **Bachelor of Teleinformatics**  
Faculty of Information Technology, Brno University of Technology  
*Compressive sampling and one-pixel camera simulation*  
Supervisor: Mgr. Pavel Rajmic, Ph.D.  
Passed with honour, acquired title Bc.
- 2000–2008 **Grammar school**  
Gymnázium, Brno-Řečkovice

### A.2 Conferences, Summer Schools, Institutions

- 2016 ICES 2016 International Conference on Evolvable Systems From Biology to Hardware, Athens GR
- 2015 HPCSE 2015 High Performance Computing in Science and Engineering, Soláň CZ



- 2015 GECCO 2015 Genetic and Evolutionary Computation Conference, Madrid ES
- 2014 MEMICS 2014 Doctoral Workshop on Mathematical and Engineering Methods in Computer Science, Telč CZ
- 2014 PPSN 2014 13th International Conference on Parallel Problem Solving from Nature, Ljubljana SI
- 2014 GECCO 2014 Genetic and Evolutionary Computation Conference, Vancouver CA
- 2014 PRACE Spring School 2014, Hagenberg AT
- 2013 LTA 2013 3rd Student Conference *Language Theory with Applications*, Brno CZ
- 2013 ECAL 2013 *12th European Conference on Artificial Life*, Taormina IT
- 2013 EEICT 2013 Conference and Competition *STUDENT EEICT 2013*, Brno CZ
- 2012 ESI12 *Modern Methods of Time-Frequency Analysis II*, Vienna AT
- 2012 *2nd SPLab Workshop 2012*, Brno CZ
- 2012 MIA'12 *Mathematics and Image Analysis*, Paris FR
- 2011 *1st SPLab Workshop 2011*, Brno CZ
- 2011 EEICT 2011 Conference and Competition *STUDENT EEICT 2011*, Brno CZ
- 2011 MACHA 11 *1st Workshop: A computational approach to harmonic analysis*, Marburg DE

### A.3 Awards, Courses & Certifications

- 2017 Special Prize IT4Innovations (Joseph Fourier Prize 2017)
- 2017 Best Interactive Presentation (DATE 2017)
- 2014 Bronze medal in Humies competition (GECCO 2014)
- 2014 *Intel Xeon Phi* coprocessor programming course
- 2013 BUT rector's award for excellent master study and scientific research results
- 2013 1. prize in the competition *ICT Diplomová práce roku 2013* awarded for the master thesis *Coevolutionary Algorithm in FPGA*
- 2013 FIT BUT dean's award for the master thesis *Coevolutionary Algorithm in FPGA*
- 2013 FEEC BUT dean's award for the master thesis *Sparse Signal Representation Application in Nuclear Magnetic Resonance*
- 2013 1. prize in the *Student EEICT 2013* competition awarded for the paper *Hardware Platform for Coevolutionary Design*
- 2011 FIT BUT dean's award for the bachelor thesis *On-Chip Debugger Generator*
- 2011 FEEC BUT dean's award for the bachelor thesis *Compressive sampling and one-pixel camera simulation*
- 2011 2. prize in the *Student EEICT 2011* competition awarded for the paper *Simulation Based Neural Motion Planner Learning*
- 2010 Microsoft Certified Technology Specialist (MCTS): Windows 7, Configuration
- 2010 Cisco Certified Network Associate (CCNA) 1-4
- 2007–2011 Scholarships of the South Moravian Center for International Mobility for talented students

### A.4 Projects

- 2016–Present GA16-17538S *Relaxed equivalence checking for approximate computing*  
Multi-objective evolutionary algorithms for approximate computing
- 2014–Present FIT-S-14-2297 *Architecture of parallel and embedded computer systems*

- Parallel and distributed algorithms for evolvable hardware
- 2014–2016 GA14-04197S *Advanced Methods for Evolutionary Design of Complex Digital Circuits*  
Evolutionary design of combinational circuits
- 2010–2013 EE.2.3.20.0094 *Support for incorporating. R&D teams in international cooperation in the area of image and audio signal processing*  
Sparse signal representation in nuclear magnetic resonance, compressed sensing for spectroscopy imaging
- 2010–2011 FR-TI1/038 *System for programming and realization of embedded systems*  
Lissom project – test-bench generator and on-chip debugger generator development for architecture description language ISAC

## A.5 Teaching

- 2015–2017 Bio-Inspired Computers (laboratory exercises)
- 2015–2016 Parallel System Architecture and Programming (projects)
- 2014–2015 Processor Architecture (projects)

## A.6 Work Experience

- 2009–Present **Embedded Systems Architect**, BENDER ROBOTICS, Brno, Czech Republic  
Embedded and mechatronic systems
- 2014–2015 **Software Developer**, CASPIA TECH, Brno, Czech Republic  
Home automation system development
- 2012–2014 **Embedded Systems Developer**, UNIS, Brno, Czech Republic  
Development of control unit and tester for fuel metering pump, code verification
- 2012–2013 **Research Worker**, INSTITUTE OF SCIENTIFIC INSTRUMENTS, ACADEMY OF SCIENCES OF THE CZECH REPUBLIC, Brno, Czech Republic  
Sparse signal representation in nuclear magnetic resonance

## A.7 Languages

- Czech Mother tongue
- English Advanced
- German Mittelstufe Deutsch (B2)