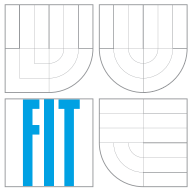
BRNO UNIVERSITY OF TECHNOLOGY
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ



FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

ACCELERATION METHODS FOR EVOLUTIONARY DESIGN OF DIGITAL CIRCUITS

METODY AKCELERACE EVOLUČNÍHO NÁVRHU ČÍSLICOVÝCH OBVODŮ

EXTENDED THESIS ABSTRACT
TEZE DISERTAČNÍ PRÁCE

AUTHOR — Ing. ZDENĚK VAŠÍČEK
AUTOR PRÁCE

SUPERVISOR — Prof. Ing. LUKÁŠ SEKANINA, Ph.D.
VEDOUCÍ PRÁCE

BRNO 2012

Contents

1 Introduction

The electronic manufacturing industry, especially electronic circuit production, is an area that has gone through a substantial development in the recent fifty years. In the second half of the 20th century, innovations in electronic computer systems made the personal computer a reality. Each new generation of computers was cheaper to purchase, more powerful and easier to operate. Comparing the current requirements to the requirements formulated a few years ago, significantly more complex circuits and behaviors are demanded today. This demand is caused by the relentless improvements of the available technologies. While the current advance is driven mainly by the necessity to minimize the overall power consumption of the produced systems, in the 1990s the goal was a relative simple – doubling of the performance of the computer systems and keeping up with the Moore's law. The current situation is much complicated and requires discovering and applying new approaches as the power consumption requirements are generally in contrast with the performance requirements.

One of the main bottlenecks that has been identified by scientific community is a low efficiency of circuit design [9]. Traditional circuit design methodologies rely on rules and design techniques that have been developed over many decades. However, the need for human input to the increasingly complex design process means that the circuit design has to be simplified by imposing greater and greater abstraction to the design space.

In the beginning of nineties, a new field applying evolutionary techniques to hardware design and synthesis has been established. This field is referred to as Evolvable Hardware [13]. The evolvable hardware draws inspiration from three main fields – biology, computer science and electronic engineering. The aim is to provide (1) electronic systems exhibiting a degree of self-adaptive and self-repair behavior and/or (2) a robust design approach that could even replace a human designer in some cases. Typical application domains include design of digital circuits, analog circuits, antennas, optical systems and MEMS [24, 15, 16].

In the context of the circuit design, the evolvable hardware is very attractive approach as it provides another option to the traditional design methodology – to use evolution to design circuits for us. Moreover, the key strength of the evolvable hardware approach is that it can be applied for designing of the circuits that cannot be fully specified a priori, but where the desired behavior is known. In fact, the search-based approaches seem to be the only viable option in this case. Another often emphasized advantage of this approach is that the circuits can be adopted for a particular environment.

During the last two decades, the evolvable hardware community has demonstrated that very efficient (and sometimes also patentable) implementations of physical designs can be obtained using evolutionary computation. For example, John Koza, the pioneer of the field, dealing primarily with the evolutionary design of analog circuits, has reported tens of human-competitive results in various areas of science and technology. The results were obtained automatically using evolutionary techniques, in particular using genetic programming [22] that has mainly been adopted for analog circuit design. In case of digital logic synthesis, the evolutionary synthesis has also led to several innovative designs (e.g. [30, 38]).

However the obtained results belong to the category of relatively small circuits.

Although the evolutionary design has been shown to be a promising and general-purpose design method, there exist several problems that make the evolutionary approach problematic in some applications [14]. The scalability problem has been identified as one of the most difficult problems the researchers are faced with in the evolvable hardware field. The scalability problem means such situation in which the evolutionary algorithm is able to provide a solution to a small problem instance; however, only unsatisfactory or even none solutions can be obtained for larger problem instances in a reasonable time. Another problem related to this issue is enormous computational power which evolutionary algorithms usually need for obtaining innovative results for some applications.

The scalability problem can primarily be seen from two perspectives: scalability of representation and scalability of fitness evaluation. From the viewpoint of the scalability of representation, the problem is that long chromosomes (a set of genes which defines a candidate solution) which are usually required to represent complex solutions imply large search spaces that are typically difficult to search. Another issue is the scalability of fitness evaluation, i.e. the problem that complex candidate solutions might require a lot of time to be evaluated. For example, in the case of the evolutionary design of combinational circuits, the evaluation time of a candidate circuit grows exponentially with the increasing number of inputs (assuming that 2^n test vectors are generated for n-input circuit). This represents the main weakness of the evolutionary approach. It also causes that real-world applications of evolutionary circuit design are not able to compete with conventional design.

1.1 Goals of the Research

It was argued in the PhD thesis that the fitness scalability issue can be eliminated by seeking for new sophisticated evaluation methods. We solely dealt with evolvable hardware as the method for automated design, i.e. the scenario in which the evolutionary algorithm is used only in the design phase of a product. The thesis postulated two main objectives.

The first goal was to propose problem-specific methods that allow designers to reduce the scalability problem in the area of digital system design. As the scalability problem represents a general problem, we considered only a very narrow but important subarea – the scalability of evaluation of a candidate digital circuit.

The second goal was to evaluate the impact of the proposed methods and show that by means of the proposed methods it is possible to evolve innovative solutions in various problem domains. In the context of evolutionary circuit design, we mean by the term innovative that a solution exhibits better features with respect to existing designs of the same category.

2005 EEICT Student Conference, Brno, 1st place
2005 International Honeywell EMI Student Competition and Conference, 4th place
2006 EEICT Student Conference, Brno, 1st place
2006 Josef Hlávka Award for Diploma Thesis
2007 Jan Hlavička Best Paper Award, Computer architecture and diagnostics Seminar
2008 Jan Hlavička Award for Outstanding Results in PhD Studies
2008 Brno University of Technology Rector's Award
2009 Best paper award for Efficient Hardware Accelerator for Symbolic Regression Problems (MEMICS, Znojmo)
2010 Jan Hlavička Award for Outstanding Results in PhD Studies
2011 Silver Medal at Humies 2011 (GECCO, Dublin) for evolutionary design of digital circuits
2011 Joseph Fourier Award in computer science

Fellowships

11/2009 University of Oslo, department of Informatics, prof. Jim Torresen
9/2009 ARTIST Summer School in Europe (embeded systems)

2 Evolutionary Algorithms and Evolvable Hardware

2.1 Evolutionary Algorithms

Several decades ago, researchers started to explore how some ideas taken from nature could be employed for solving hard computing problems. Evolutionary algorithms (EAs) [3] inspired by biological evolution represent one of the most successful examples. The evolutionary algorithms are stochastic search algorithms inspired by Darwin's theory of evolution. In contrast with common search algorithms, such as random search or hill climbing, the EAs are population-based algorithms. It means that they work with more candidate solutions (i.e. individuals) in the same time. By a candidate solution we mean a point in the search space, the space that contains all possible considered solutions to a given problem.

Every new population is formed using genetic operators such crossover and mutation and through a selection pressure. The selection pressure together with the *fitness function*, sometime referred to as objective function, is responsible for guiding the evolution towards better areas of the search space. The guidance is received from the fitness function that assigns so called fitness value to each candidate solution. The fitness value indicates how well a candidate solution fulfills the problem objective; in other words, it indicates how a particular candidate solution meets the specification. A better fitness value implies a greater chance that a candidate solution will remain for a longer while and produce offspring, which inherit parental genetic information. A well-designed evolutionary algorithm should converge to a population containing desired solutions.

The evolutionary algorithms are traditionally divided into four distinct branches: *genetic algorithm* [11], *genetic programming* [20], *evolutionary strategies* [35] and *evolutionary programming* [10]. The algorithms mainly differ in the mechanism of the candidate solutions encoding, implementation of the evolutionary operators applied to the candidate solutions and finally, utilized search strategy that guides the EA through the search space. In the following sections, the algrithms related to this thesis are briefly introduced.

Genetic Programming

Genetic Programming (GP) was introduced by John Koza in late eighties as an extension to genetic algorithms in order to enrich the chromosome representation [20]. GP evolves pieces of code written over a specified alphabet consisting of a set of functions and a set of terminals. Genetic programming allows automatic programming and program induction (i.e. automatically developing of computer programs). Unlike genetic algorithms, genetic programming does not distinguish between phenotype and genotype. As genetic programming is able to effectively evolve symbolic expressions, the problem of symbolic regression became the most popular application of GP.

The evolved programs are usually represented either as tree structures or in a linear form using a list of machine-language instructions. Similarly to the GA, crossover is considered as a major genetic operator.

The GP representation has its own pitfalls. An evolved program may contain segments which do not alter the result of the program execution when they are removed from it. Another well-known issue of GP is that the program size can grow uncontrollably until it reaches the tree-depth maximum, while the fitness remains unchanged. These pitfalls and their relations are discussed, for example, in [2, 4].

Cartesian Genetic Programming

Cartesian genetic programming (CGP) is a variant of genetic programming where the genotype is represented as a list of integers that are mapped to directed oriented graphs rather than trees [32]. The motivation for this representation came from the previous analysis covering the effectiveness of this approach in learning Boolean functions where the CGP has been proved to be considerably more efficient than any other variant of GP.

Cartesian genetic programming encodes a candidate solution (typically a circuit or a program) using an array consisting of $n_{cols} \times n_{rows}$ programmable nodes. Each programmable node has the fixed number of inputs and outputs. Each node input can be connected either to the output of a node placed in the previous l columns or to one of the program inputs. Because of the complicated evaluation, feedback is not allowed in the standard version of CGP. Each node can be programmed to perform one of n_{ei}-input functions defined in the set Γ.

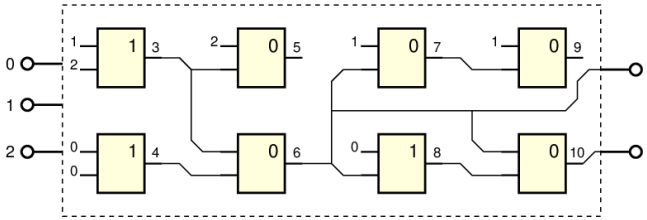


Figure 1: Example of a candidate circuit encoded using CGP with the following parameters: $n_{cols} = 4$, $n_{rows} = 2$, $l = 2$, $n_{ei} = 2$, $\Gamma = \{$AND (0), OR(1) $\}$. Chromosome: 1, 2, 1, 0, 0, 1, 2, 3, 0, 3, 4, 0, 1, 6, 0, 0, 6, 1, 1,3, 0, 6, 8, 0, 6, 10. Functions of elements are typed in bold. The first 24 integers encode the interconnection of the CGP elements and function of each element. The last two integers indicate the output of the circuit. Elements 5, 7 and 9 are not utilized.

Figure 1 shows a digital circuit encoded using CGP representation. The figure also demonstrates the main feature of CGP encoding – while the genotype (i.e. chromosome) is of fixed length, the phenotype is of variable length depending on the number of unexpressed genes. In this example, three nodes do not contribute to the phenotype. The utilized representation also significantly reduces the bloat which is inevitable in GP [29]. This fact has been confirmed by Miller in [31].

Zdeněk Vašíček

Personal

Born on	5.6.1982
Nationality	Czech
Home Address	Rybničky 224, 679 38 Cetkovice, Czech Republic
Affiliation	UPSY FIT VUT, Božetěchova 2, 612 66 Brno, Czech Republic
Contact	vasicek@fit.vutbr.cz

Education

1997-2001	Secondary School of Electrical Engineering, Kounicova 16, Brno
	Specialization: Electronics computer systems; graduated with honors
2001-2006	Brno University of Technology, Faculty of Information Technology, Božetěchova 2, Brno
	Master study programme: Information Technology
	Title of thesis: Hardware Implementation of Evolvable Systems; graduated with honors
2006-2012	Brno University of Technology, Faculty of Information Technology, Božetěchova 2, Brno
	Doctoral study programme: Computer Science and Engineering
	Research topic: Evolvable Hardware and Its Application

Employment

2002–2009	Computer and Information Services Centre, BUT Brno, Antonínská 548/1, Brno
	Delphi / Oracle programmer, development of university information system
2006–2009	PD-Refractories CZ, a.s., Velké Opatovice
	Heterogenous computer network administrator
2007–present	Department of Computer Systems, Faculty of Information Technology, BUT Brno
	Member of Evolvable Hardware Research Group
	specialization: evolvable hardware in FPGAs, evolutionary logic synthesis and optimization
2010–present	Department of Computer Systems, Faculty of Information Technology, BUT Brno
	Assistant lecturer, researcher

Participation in research projects

2006-2008	Methods of polymorphic digital circuit design, GA102/06/0599, team member
2007-2009	Design and hardware implementation of a patent-invention machine, GA102/07/0850
2007	Stabilisation of Image Signal from Camera System, JC-8500/HS1470031, team member
2007-2013	Security-Oriented Research in Information Technology, MSM0021630528, researcher
2007	Technical Evaluation of Biometric Systems, ST20072007006, researcher
2009-2012	Mathematical and Engineering Approaches to Developing Reliable and Secure Concurrent and Distributed Computer Systems, GD102/09/H042, researcher
2010-2013	Natural Computing on Unconventional Platforms, GP103/10/1517, researcher
2011-2013	Advanced secured, reliable and adaptive IT, FIT-S-11-1, team member

Honors, Awards

[69] M. Bidlo, Z. Vašíček, and K. Slaný. Sorting network development using cellular automata. In *Evolvable Systems: From Biology to Hardware*, volume 6274 of *Lecture Notes in Computer Science*, pages 85–96. Springer Berlin / Heidelberg, 2010.

[70] T. Dulík, Z. Křivka, J. Kadlec, M. Bližňák, V. Budíková, O. Jirák, N. Olšarová, J. Trbušek, and Z. Vašíček. *Virtuální laboratoř pro vývoj aplikací s mikroprocesory a FPGA*. CERM, Brno, 2011.

[71] O. Jirák, Z. Křivka, N. Olšarová, and Z. Vašíček. Odvozování propojení komponent pro podporu návrhu pro malé fpga čipy. In *Proc. of the DATAKON 2010*, pages 81–90. VŠB TU, 2010.

[72] O. Jirák, Z. Křivka, and Z. Vašíček. Component interconnection inference tool supporting the design of small fpga-based embedded systems. In *Proc. of the IADIS International Conference Applied Computing 2010*, pages 230–234. IADIS Press, 2010.

[73] O. Jirák, Z. Křivka, and Z. Vašíček. Integrated development environment for virtual laboratory. In *International Technology, Education and Development Conference*, page 10. IATED, 2011.

[74] Z. Křivka and Z. Vašíček. The virtualization of development boards in the virtual laboratory of microprocessor technology. In *12th International Carpathian Control Conference*, pages 424–428. VŠB TU, 2011.

[75] L. Sekanina, R. Růžicka, Z. Vašíček, R. Prokop, and L. Fujčík. Repomo32 – new reconfigurable polymorphic integrated circuit for adaptive hardware. In *2009 IEEE Workshop on Evolvable and Adaptive Hardware*, pages 39–46. IEEE Computational Intelligence Society, 2009.

[76] Z. Vašíček. Evolutionary synthesis of gate-level digital circuits. In *Proceedings of 11th Conference and Competition HONEYWELL EMI 2005*. FEKT VUT, 2005.

[77] Z. Vašíček. Implementation of high-performance reconfigurable systems on a chip. In *Proceedings of 12th Conference and Competition STUDENT EEICT 2006 Volume 2*, pages 232–234. FEKT VUT, 2006.

[78] Z. Vašíček. Reálné aplikace evolučního návrhu. In *Počítačové architektury a diagnostika 2007. Česko-slovenský seminář pro studenty doktorandského studia*, pages 137–142. University of West Bohemia in Pilsen, 2007.

[79] Z. Vašíček. Towards automatic design of competitive image filters in fpgas. In *Proceedings of Junior Scientist Conference*. TU-Wien, 2008.

[80] Z. Vašíček, L. Čapka, and L. Sekanina. Analysis of reconfiguration options for a reconfigurable polymorphic circuit. In *Proceedings of the 2008 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 3–10, Washington, DC, USA, 2008. IEEE Computer Society.

[81] L. Čapka and Z. Vašíček. Investigating the influence of mutation operators in cartesian genetic programming. In *13th International Conference on Soft Computing*, pages 43–47. Faculty of Mechanical Engineering BUT, 2007.

In CGP, the search is performed using a mutation-based evolutionary strategy $(1 + \lambda)$-ES that does not utilize crossover. The influence of the crossover operator has been intensively studied in literature, however, it has been confirmed that crossover does not improve the search [43]. CGP operates with the population of $1 + \lambda$ individuals, where λ is typically from 1 to 15. In case of the evolutionary design, the initial population is usually generated randomly whereas in case of the evolutionary optimization the initial population can be constructed by means of mapping of a known conventional solution to the CGP representation.

The CGP became the routinely used approach in the area of evolutionary-based digital circuit synthesis and optimization. The main advantage of CGP is that it generates very compact solutions, i.e. it can effectively reduce the total number of gates in the case of circuit evolution [30].

2.2 Evolvable Hardware

A massive application of evolutionary principles to hardware design and self-configuration has led to a new concept called Evolvable Hardware (EHW). The growth of this interest has been caused by emerging a new class of programmable devices, in particular FPGAs. The main idea is to accomplish the whole process of circuit design by evolutionary algorithms. Evolvable hardware refers to a hardware that a) has been created using EA or b) embeds a variant of EA in order to either adapt the system to changing environments, or repair the system autonomously during its lifetime. While the first scenario is usually called evolutionary design, the second approach is referred to as evolvable hardware. The evolutionary design uses evolutionary algorithms to evolve a system that meets a predefined specification. EA is employed only in the design phase. In contrast with this approach, the adaptive systems reconfigure a part or whole existing design to repair the faults or adapt to a changed operational environment. Thus the evolutionary algorithm is an integral part of the adaptive system.

EHW typically utilizes a reconfigurable hardware, particularly programmable logic devices such as FPGAs. The programmable logic devices allow the candidate solutions to be tested in situ which is well suited to embedded applications such as adaptive image filters or adaptive controllers.

Using evolution to design electronics brings a number of benefits. Some of the most important areas where evolutionary electronics can successfully be applied include: automatic design of low cost hardware, automatic design of hardware systems for poorly specified problems, innovation in poorly understood design spaces, design of adaptive systems, or design of fault tolerant systems. The ability to generate solutions to poorly specified problems can be considered as a form of creativity which is one of the features of evolutionary processes. In case of the adaptive and fault tolerant system, the evolvable hardware is usually used due to its potential of autonomous adaption to changes in its environment (e.g. noise level in case of adaptive image filters, or presence of faults in case of fault-tolerant adaptive systems). The advantageous feature of the evolutionary approach is that it can not be necessary constrained to the well-known topologies that usually prevent from achieving novel solutions.

On the contrary, evolutionary design approach has some drawbacks. Evolutionary meth-

ods are sometimes criticized that they do not produce robust and trustworthy designs. The evolved circuits are usually different from the well-known and proven structures which complicate their analysis and verification. Another discussed problem is enormous computational power which is usually needed for obtaining a satisfactory result. In some real-time applications (e.g. adaptive and fault tolerant systems), slow convergency or even stuck in a local extreme may represent an issue.

2.3 Current Problems of Evolutionary Design

From the view of the design automation and ability to produce novel designs, the evolutionary design represents a promising and general-purpose design method. However, there are known problems that limit the application of evolutionary approach in some domains. During the last decade, a number of researchers have been addressing the so called scalability problem. Unfortunately, this issue has not been yet successfully solved [14].

In terms of the scalability of representation, the problem is that long chromosomes which are usually required to represent complex solutions imply large search spaces that are typically difficult to search. In order to evolve large designs and simultaneously keep the size of chromosome small, various techniques have been proposed. Among others variable length representation [45], function-level evolution [33], incremental evolution [41], developmental genetic programming [23] and modularization [21] represent the most applied approaches.

Even if an encoding is chosen such that it allows the candidate circuits to be represented effectively, there is another scalability problem having substantial impact on the evolutionary design of digital circuits. In case of the combinational circuit evolution, the evaluation time of a candidate circuit grows exponentially with the increasing number of inputs (assuming that all possible input combinations are tested in the fitness function). Hence, the evaluation time becomes the main bottleneck of the evolutionary approach when complex circuits with many inputs are evolved. In case of popular benchmark circuits, the limit when the evolution provides some solution is 17 inputs for parity circuits [39], 8 inputs for combinational multipliers when a novel solution has been obtained [42] and 16 inputs for combinational multipliers when only functionality has been evolved [38]. The 28-input sorting networks evolved using FPGA accelerator are probably the largest circuits that were successfully evolved using the fitness function which evaluates all possible assignments to the inputs, however, only functionality has been evolved [19].

Evolutionary circuit design is considered as a very challenging research area. The first difficulty comes from the complexity of the search space. Another difficuilty is caused by the presence of very good and robust conventional design tools that have been extensively developed for many years to produce compact and efficient solutions.

One of the goals of the early pioneers of the evolvable hardware was to evolve complex circuits, push the limits of traditional design and find ways how to exploit the vast computational resources available on today's computation platforms. However, the scalability issue for evolvable hardware continues to be out of reach [14]. To reduce this problem, a kind of domain knowledge is usually employed in focusing the search algorithm on promising areas of the search space and reducing the computation overhead.

[55] Z. Vašíček and L. Sekanina. An area-efficient alternative to adaptive median filtering in FPGAs. In *Proc. of 2007 Conf. on Field Programmable Logic and Applications*, pages 216–221. IEEE Computer Society, 2007.

[56] Z. Vašíček and L. Sekanina. Evaluation of a new platform for image filter evolution. In *Proc. of the 2007 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 577–584. IEEE Computer Society, 2007.

[57] Z. Vašíček and L. Sekanina. Reducing the area on a chip using a bank of evolved filters. In *Evolvable Systems: From Biology to Hardware*, volume 4684 of *Lecture Notes in Computer Science*, pages 222–232. Springer Verlag, 2007.

[58] Z. Vašíček and L. Sekanina. Hardware accelerators for cartesian genetic programming. In *European Conference on Genetic Programming*, volume 4971 of *Lecture Notes in Computer Science*, pages 230–241. Springer Verlag, 2008.

[59] Z. Vašíček and L. Sekanina. Novel hardware implementation of adaptive median filters. In *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, pages 110–115. IEEE Computer Society, 2008.

[60] Z. Vašíček and L. Sekanina. A global postsynthesis optimization method for combinational circuits. In *Proc. of the Design, Automation and Test in Europe, DATE*, pages 1525–1528. IEEE Computer Society, 2011.

[61] Z. Vašíček, L. Sekanina, and M. Bidlo. A method for design of impulse bursts noise filters optimized for FPGA implementations. In *DATE 2010: Design, Automation and Test in Europe*, pages 1731–1736. European Design and Automation Association, 2010.

[62] Z. Vašíček and K. Slaný. Efficient phenotype evaluation in cartesian genetic programming. In *15th European Conference on Genetic Programming*, volume 7244 of *Lecture Notes in Computer Science*, pages 265–276. Springer Verlag, 2012.

[63] Z. Vašíček, L. Čapka, and L. Sekanina. Analysis of reconfiguration options for a reconfigurable polymorphic circuit. In *Proceedings of the 2008 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 3–10. IEEE Computer Society, 2008.

[64] Z. Vašíček, M. Žádník, L. Sekanina, and J. Tobola. On evolutionary synthesis of linear transforms in FPGA. In *Proc. of the 8th Int. Conference on Evolvable Systems: From Biology to Hardware*, volume 5216 of *LNCS*, pages 141–152, Berlin, 2008. Springer Verlag.

Books

[65] L. Sekanina, Z. Vašíček, R. Růžička, M. Bidlo, J. Jaroš, and P. Švenda. *Evoluční hardware: Od automatického generování patentovatelných invencí k sebemodifikujícím se strojům.* Academia, Praha, 2009.

Other publications

[66] M. Bidlo and Z. Vašíček. Cellular automata-based development of combinational and polymorphic circuits: A comparative study. In *Evolvable Systems: From Biology to Hardware*, volume 6274 of *Lecture Notes in Computer Science*, pages 106–117. Springer Verlag, 2008.

[67] M. Bidlo and Z. Vašíček. Comparison of the uniform and non-uniform cellular automata-based approach to the development of combinational circuits. In *Proc. of NASA/ESA Conference on Adaptive Hardware and Systems*, pages 423 – 430. IEEE Computer Society, 2009.

[68] M. Bidlo and Z. Vašíček. Investigating gate-level evolutionary development of combinational multipliers using enhanced cellular automata-based model. In *IEEE Congress on Evolutionary Computation*, pages 2241–2248. IEEE Computer Society, 2009.

Recognition Letters, 16:341–347, 1994.

[41] J. Torresen. A Divide-and-Conquer Approach to Evolvable Hardware. In *Proc. of the 2nd International Conference on Evolvable Systems: From Biology to Hardware ICES'98*, volume 1478 of *LNCS*, pages 57–65, Lausanne, Switzerland, 1998. Springer.

[42] V. K. Vassilev, D. Job, and J. F. Miller. Towards the Automatic Design of More Efficient Digital Circuits. In *Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware*, pages 151–160, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[43] V. K. Vassilev, J. F. Miller, and T. C. Fogarty. On the nature of two-bit multiplier landscapes. In *EH 1999: Proceedings of the 1st NASA/DOD workshop on Evolvable Hardware*, page 36, Washington, DC, USA, 1999. IEEE Computer Society.

[44] Y. Voronenko and M. Püschel. Multiplierless multiple constant multiplication. *ACM Transactions on Algorithms*, 3(2), 2007.

[45] R. S. Zebulum, M. S. Vellasco, and M. A. Pacheco. Variable length representation in evolutionary electronics. *Evol. Comput.*, 8(1):93–120, 2000.

Author's publications

Journal papers

[46] Z. Vašíček and L. Sekanina. Evoluční návrh kombinačních obvodů. *Elektrorevue*, 2004(43):6, 2004.

[47] Z. Vašíček and L. Sekanina. An evolvable hardware system in Xilinx Virtex II Pro FPGA. *International Journal of Innovative Computing and Applications*, 1(1):63–73, 2007.

[48] Z. Vašíček and L. Sekanina. Hardware accelerator of cartesian genetic programming with multiple fitness units. *Computing and Informatics*, 29(6+):1359–1371, 2010.

[49] Z. Vašíček and L. Sekanina. Formal verification of candidate solutions for post-synthesis evolutionary optimization in evolvable hardware. *Genetic Programming and Evolvable Machines*, 12(3):305–327, 2011.

Conference papers

[50] P. Fišer, J. Schmidt, Z. Vašíček, and L. Sekanina. On logic synthesis of conventionally hard to synthesize circuits using genetic programming. In *Proc. of the 13th Int. IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems*, pages 346–351. IEEE Computer Society, 2010.

[51] L. Sekanina and Z. Vašíček. On the practical limits of the evolutionary digital filter design at the gate level. In *Applications of Evolutionary Computing*, volume 3907 of *Lecture Notes in Computer Science*, pages 344–355. Springer Berlin / Heidelberg, 2006.

[52] Z. Vašíček and M. Bidlo. Evolutionary design of robust noise-specific image filters. In *IEEE Congress on Evolutionary Computation*, pages 269–276. IEEE Computer Society, 2011.

[53] Z. Vašíček, M. Bidlo, L. Sekanina, and K. Glette. Evolutionary design of efficient and robust switching image filters. In *Proc. of the 2011 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 192–199. IEEE Press, 2011.

[54] Z. Vašíček, M. Bidlo, L. Sekanina, J. Torresen, K. Glette, and M. Furuholmen. Evolution of impulse bursts noise filters. In *Proc. of the 2009 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 27–34. IEEE Press, 2009.

3 Evolutionary Synthesis of Linear Transforms

The linear transforms represent a key concept that is in some way employed in every digital signal processing application. Many numerically intensive applications have computations that are based on linear transforms such as convolutions, the discrete Fourier transform, the discrete cosine transforms, etc. Mathematically, they consist exclusively of additions and multiplications by constants and usually involve a large number of multiplications of one variable with several constants. A proper optimization of this part of the computation, referred to as multiple constant multiplication problem (MCM), often results in a significant improvement not only of the performance but also the power consumption.

In this section, we will introduce an evolutionary method based on Cartesian Genetic Programming that can synthesize complex instances of the MCM problem. The goal of this research is to show that the evolutionary algorithm is able to generate not only complex but also close to optimal structures. In order to eliminate the scalability problem of a candidate MCM evaluation, the linearity of MCM problem is exploited.

3.1 Theoretical Background

In order to implement a linear transform in hardware, two basic building blocks are used – additions/subtractions and multiplications by constants. When multiplying by constants in hardware, costly combinational multipliers may be avoided by replacing them with structure consisting of additions, subtractions and shifts. In most cases, the shifts can be effectively implemented using the wires. The principles of multiplierless multiplier design will be briefly introduced in the following paragraphs.

The multiplication $y = kx$ of variable x by a known constant k can be decomposed to into additions, subtractions and shifts. The problem of finding the optimal decomposition is known as the single constant multiplication problem (SCM) [44]. Although there exist several straightforward methods for decomposing the multiplication into the additions and shifts based on binary representation, signed digit system (SD) or a canonic signed digit system (CSD), it is known that CSD in general does not yield the minimum cost solution. It can be sometimes more efficient to firstly factor the coefficient into several factors and implement the individual factors in an optimal CSD sense. In order to find an optimal solution that minimizes the cost, a kind of reusing of intermediate results has to be introduced. However, finding the optimal addition chain for a given constant k is known to be NP-complete problem.

The single constant multiplication problem can be extended to the problem of multiplying a variable x with several constants $k_1, \ldots, k_N$ in parallel. The resulting structure referred to as multiplier block can be used to implement digital FIR filters, linear signal transforms such as the discrete Fourier transform or discrete cosine transform, and so on [44]. For example, discrete Fourier and trigonometric transform algorithms involve rota-

tions, which require simultaneous multiplication by two constants. The problem of finding the decomposition with fewest operations is known as the multiple constant multiplication (MCM) design problem [44].

Comparing to the previous problem, the design of an optimal multiplierless MCM structure is more complicated since intermediate results of the SCM decompositions may be shared. In addition to that, the optimal decomposition can not be obtained as a simple combination of optimal SCM multipliers obtained for each constant independently.

Even if the MCM problem is NP-complete, several efficient heuristics have been proposed. A good survey of the proposed methods can be found in [44].

3.2 Proposed Method

The goal is to synthesize a multiple constant multiplier block which generates N output values $y_i = c_i x$ where $1 \leq i \leq N$, c_i are given constants and x is the input variable. The circuit is composed of high-level linear components such as additions, subtractions and logic shifts. The evolution is conducted at function level. In order to design a multiplier block having the minimal cost, two-stage fitness strategy is employed. At the beginning of the search, the objective of the evolutionary algorithm is to evolve a fully functional multiplier only. Once the first fully functional solution appears, an optimization phase rewarding the solutions with lower cost is conducted. During this stage, the number of components is optimized.

The problem is approached using evolutionary algorithm in which the problem representation is borrowed from the CGP. A candidate multiplier block is represented as an array of programmable nodes. The number of columns defines the maximum MCM delay. The number of inputs, n_i, and outputs, n_o, is fixed and chosen as follows: $n_i = 1$, $n_o = N$. Feedback is not allowed. Each node is programmed to perform one of functions defined in the set Γ which includes addition, subtraction, various shifts and identity function. These functions as well as all connections are defined over b bits.

EA operates with the population of λ individuals where $\lambda = 5$. The initial population is randomly generated. In the fitness calculation, the goal is to minimize the difference between circuit outputs and required products. When a functionally perfect solution is obtained, the fitness function is switched to a new fitness function in which the number of components is optimized. In order to measure the similarity of a candidate solution and the required response, sum of absolute differences (SAD) is used.

The evolution is stopped when the best fitness value stagnates or a predefined number of generations is exhausted. In theory, it is sufficient to evaluate a candidate solution using one test-case, e.g. $x = 1$. Nevertheless, especially in case when Γ contains right shifts and subtractions, the limited number of bits may introduce a kind of nonlinearity. In order to avoid this behavior more test cases should be used (e.g. the powers of two over b bits).

3.3 Results

In order to evaluate the proposed method, we have chosen to evolve multipliers with 3, 5, 10, 20 and 54 coefficients (given in Table 1). The coefficients were encoded at 16 bits. All the multipliers were evaluated using single training vector $x = 1$. The evolved multipliers were

[21] J. R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs*. MIT Press, Cambridge, MA, 1994.

[22] J. R. Koza. Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3–4):251–284, 2010.

[23] J. R. Koza, F. H. Bennett III, D. Andre, and M. A. Keane. *Genetic Programming III: Darwinian Invention and Problem Solving*. Morgan Kaufmann Publishers, San Francisco, CA, 1999.

[24] J. R. Koza, M. A. Keane, M. J. Streeter, W. Mydlowec, J. Yu, and G. Lanza. *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. Kluwer Academic Publishers, 2003.

[25] T. Larrabee. Test pattern generation using boolean satisfiability. *IEEE Transactions on Computer-Aided Design*, 11:4–15, 1992.

[26] C.-C. Lee, J.-H. R. Jiang, C.-Y. R. Huang, and A. Mishchenko. Scalable exploration of functional dependency by interpolation and incremental SAT solving. In *Proceedings of the 2007 IEEE/ACM international conference on Computer-aided design*, ICCAD '07, pages 227–233, Piscataway, NJ, USA, 2007. IEEE Press.

[27] S. Marshall. New direct design method for weighted order statistic filters. *VISP*, 151(1):1–8, February 2004.

[28] K. L. McMillan. Interpolation and SAT-based model checking. *Computer Aided Verification*, pages 1–13, 2003.

[29] J. F. Miller. What bloat? Cartesian Genetic Programming on Boolean problems. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2001*, pages 295–302. Morgan Kaufmann Publishers, 2001.

[30] J. F. Miller, D. Job, and V. K. Vassilev. Principles in the Evolutionary Design of Digital Circuits – Part I. *Genetic Programming and Evolvable Machines*, 1(1):8–35, 2000.

[31] J. F. Miller and S. L. Smith. Redundancy and Computational Efficiency in Cartesian Genetic Programming. *IEEE Transactions on Evolutionary Computation*, 10(2):167–174, 2006.

[32] J. F. Miller and P. Thomson. Cartesian Genetic Programming. In *Proc. of the 3rd European Conference on Genetic Programming EuroGP2000*, volume 1802 of *LNCS*, pages 121–132. Springer, 2000.

[33] M. Murakawa et al. Evolvable hardware at function level. In *In Proc. of the Parallel Problem Solving from Nature IV*, volume 1141 of *LNCS*, pages 62–71. Springer Berlin / Heidelberg New York, 1996.

[34] S. Safarpour, A. Veneris, G. Baeckler, and R. Yuan. Efficient SAT-based boolean matching for FPGA technology mapping. In *Proceedings of DAC 2006*, 2006.

[35] H. P. Schwefel. *Evolution and Optimum Seeking*. John Wiley, New York, 1995.

[36] L. Sekanina. Image Filter Design with Evolvable Hardware. In *Applications of Evolutionary Computing – Proc. of the 4th Workshop on Evolutionary Computation in Image Analysis and Signal Processing EvoIASP'02*, volume 2279 of *LNCS*, pages 255–266, Kinsale, Ireland, 2002. Springer Verlag.

[37] L. Sekanina. *Evolvable components: From Theory to Hardware Implementations*. Natural Computing. Springer-Verlag Berlin, 2004.

[38] A. P. Shanthi and R. Parthasarathi. Practical and scalable evolution of digital circuits. *Applied Soft Computing*, 9(2):618–624, 2009.

[39] E. Stomeo, T. Kalganova, and C. Lambert. Generalized disjunction decomposition for evolvable hardware. *IEEE Transaction Systems, Man and Cybernetics, Part B*, 36(5):1024–1043, 2006.

[40] T. Sun and Y. Neuvo. Detail-preserving median based filters in image processing. *Pattern*

Bibliography

[1] M. O. Ahmad and D. Sundararajan. A fast algorithm for two-dimensional median filtering. *IEEE Transactions on Circuits and Systems*, 34:1364–1374, 1987.

[2] P. J. Angeline. Subtree crossover causes bloat. In *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 745–752. Morgan Kaufmann, 1998.

[3] T. Bäck. *Evolutionary Algorithms in Theory and Practice.* Oxford University Press, 1996.

[4] W. Banzhaf and W. B. Langdon. Some considerations on the reason for bloat. *Genetic Programming and Evolvable Machines*, 3(1):81–91, 2002.

[5] D. R. K. Brownrigg. The weighted median filter. *Commun. ACM*, 27(8):807–818, 1984.

[6] V. Crnojevic, V. Senk, and Z. Trpovski. Advanced impulse detection based on pixel-wise MAD. *SPLetters*, 11(7):589–592, July 2004.

[7] Y. Dong and S. Xu. A new directional weighted median filter for removal of random-valued impulse noise. *Signal Processing Letters*, 14(3):193–196, 2007.

[8] E. R. Dougherty and J. T. Astola, editors. *Nonlinear Filters for Image Processing.* SPIE/IEEE Series on Imaging Science & Engineering. SPIE/IEEE, 1999.

[9] M. Flynn and P. Hung. Microprocessor design issues: thoughts on the road ahead. *Micro, IEEE*, 25(3):16 – 31, 2005.

[10] D. B. Fogel. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence.* IEEE Press, 1995.

[11] E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[12] E. Goldberg, M. Prasad, and R. Brayton. Using SAT for combinational equivalence checking. In *DATE '01: Proceedings of the conference on Design, automation and test in Europe*, pages 114–121, Piscataway, NJ, USA, 2001. IEEE Press.

[13] T. G. W. Gordon and P. J. Bentley. On evolvable hardware. In *Soft Computing in Industrial Electronics*, pages 279–323, London, UK, 2002. Physica-Verlag.

[14] P. C. Haddow and A. Tyrrell. Challenges of evolvable hardware: past, present and the path to a promising future. *Genetic Programming and Evolvable Machines*, 12:183–215, 2011.

[15] T. Higuchi, M. Iwata, D. Keymeulen, H. Sakanashi, M. Murakawa, I. Kajitani, E. Takahashi, K. Toda, M. Salami, N. Kajihara, and N. Otsu. Real-World Applications of Analog and Digital Evolvable Hardware. *IEEE Transactions on Evolutionary Computation*, 3(3):220–235, 1999.

[16] G. Hornby, A. Globus, D. S. Linden, and J. Lohn. Automated Antenna Design with Evolutionary Algorithms. In *Proc. 2006 AIAA Space Conference*, page 8, San Jose, CA, 2006. AIAA.

[17] H. Hwang and R. Haddad. Adaptive median filters: new algorithms and results. *IP*, 4(4):499–502, April 1995.

[18] S. Ko and Y. Lee. Center weighted median filters and their applications to image enhancement. *IEEE Transactions on Circuits and Systems*, 15:984–993, 1991.

[19] J. Korenek and L. Sekanina. Intrinsic evolution of sorting networks: A novel complete hardware implementation for FPGAs. In *Evolvable Systems: From Biology to Hardware*, volume 3637 of *LNCS*, pages 46–55. Springer Verlag, 2005.

[20] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, 1992.

verified at symbolic level. All experiments were repeated 200 times with the population of eight individuals and five genes mutated in the chromosome. $\Gamma = \{a,\ a+b,\ a-b,\ 2a,\ 4a,\ 8a,\ \ldots,\ 8192a\ \}$. Table 1 gives other parameters of the experiments, average results (the number of generations and used adders/subtractors), the success rate and parameters of the best evolved solutions.

Table 1: Results of evolutionary design of MCMs with different coefficients. Population size is 8. Averages are calculated from 200 independent runs.

Settings		Average Results			The Best MCM			
cols × rows	maxgen	geners.	#add/sub	succ. rate	delay	add/sub	shifts	operations
3 constants: 2925, 23111, 13781								
Heuristics [44]					8	8	8	16
5×6	20M	1M62	14	68.5	5	9	8	17
6×6	20M	1M27	14	86.5	6	8	8	16
7×4	40M	2M15	13	99.0	7	8	6	14
5 constants: 83, 221, 71, 387, 13								
Heuristics [44]					5	6	6	12
4×6	20M	461k	10	99.5	4	7	6	13
5×6	20M	207k	11	99.5	5	6	6	12
6×6	20M	114k	11	100.0	6	6	5	11
10 constants: 117, 1123, 743, 221, 1069, 7605, 987, 16689, 3033, 29								
Heuristics [44]					8	14	13	27
10×4	40M	4M8	23	99.0	7	15	12	27
7×6	20M	4M7	23	95.5	6	17	11	28
9×4	40M	9M5	22	91.0	9	17	9	26
20 constants: 1, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71								
Heuristics [44]					4	19	8	27
4×10	40M	457k	23	100	4	19	4	23
5×10	40M	347k	23	100	4	19	4	23
6×5	40M	772k	21	100	5	19	3	22
54 constants: 1, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251								
Heuristics [44]					6	53	53	106
5×20	40M	12M9	66	98	5	56	17	73
6×14	40M	19M7	63	90	6	56	12	68
6×16	40M	9M8	65	98.5	6	55	19	74

Results are compared with the best known heuristic approach [44] which produces very compact solutions. Table 1 shows that the proposed evolutionary-based approach is able to generate multipliers that are competitive with results obtained using the state of the art heuristic approach. The evolution can reduce the total number of components as well as the delay of the designed MCMs. Even if the goal of this research was to demonstrate and confirm the validity of our hypothesis concerning the linear problems, the proposed method has been able to discover better solutions in some instances. In case of the 3 constant MCM, a solution exhibiting lower delay containing fewer shifts has been evolved. In case of the 20 constant MCM, the number of shift has been reduced by one half.

3.4 Summary

A very time-consuming evaluation of candidate configurations is one of problems which influence the applicability of evolutionary circuit design. In this section, we focused on such

problems in which a candidate solution can be evaluated in a short time if some domain knowledge is employed. Linear transforms in general, and multiple constant multiplier blocks in particular, belong to this class. Although well-optimized heuristics exist for linear transforms design, we confirmed that novel implementations of multiple constant multipliers can be designed using evolutionary algorithm [64]. Using this method, digital circuits with total output width higher than 850 bits have been successfully evolved.

4 Evolutionary Synthesis of Complex Combinational Circuits

Evolvable hardware community has demonstrated that very efficient implementations of digital circuits can be obtained using evolutionary computation, particularly by means of Cartesian Genetic Programming [42]. Unfortunately, the evolutionary circuit design is able to discover innovative designs only for small circuit instances (approx. up to 20 inputs and 100 gates). One of the key problems is a very time consuming fitness calculation which typically grows exponentially with increasing circuit complexity (number of inputs).

The goal of this section is to introduce an approach that enables to significantly reduce the number of gates for complex circuits, too. As it will be demonstrated, very compact implementations can be obtained if the fitness calculation utilizes a formal verification algorithm to check whether a candidate circuit is functionally correct or not. In order to decide the correctness of a candidate solution, we have employed SAT-based equivalence checking.

4.1 Theoretical Background

Boolean satisfiability problem is a well-known decision problem consisting of deciding whether the variables of a propositional formula can be assigned in such a way that the formula evaluates to true. The research area devoted to this problem is today very active as many real-world problems can be effectively solved by transforming them to the SAT problem. In the field of digital system design, the use of SAT has been investigated for more than twenty years and many powerful tools utilizing SAT solvers have been developed. Test pattern generation [25], identification of functional dependencies in Boolean functions [26], technology-mapping [34], combinational equivalence checking [12] or model checking [28] represent successful examples of practical applications of SAT solvers.

Combinational Equivalence Checking

Determining whether two Boolean functions are functionally equivalent represents a fundamental problem in formal verification. A brute-force method to determine combinational equivalence is to expand the combinational functions in minterm form (or in a truth table) and compare them term by term (row by row). This method represents an approach routinely used by EHW community. Unfortunately, this method runs into the problem of exponential size, because the number of minterms (or rows of a corresponding truth table)

Phd Thesis Abstract

Although many examples showing the merits of evolutionary design over conventional design techniques utilized in the field of digital circuits design have been published, the evolutionary approaches are usually hardly applicable in practice due to the various so-called scalability problems. The scalability problem represents a general problem that refers to a situation in which the evolutionary algorithm is able to provide a solution to a small problem instances only. For example, the scalability of evaluation of a candidate digital circuit represents a serious issue because the time needed to evaluate a candidate solution grows exponentially with the increasing number of primary inputs.

In this thesis, the scalability problem of evaluation of a candidate digital circuit is addressed. Three different approaches to overcoming this problem are proposed. Our goal is to demonstrate that the evolutionary design approach can produce interesting and human competitive solutions when the problem of scalability is reduced and thus a sufficient number of generations can be utilized.

In order to increase the performance of the evolutionary design of image filters, a domain specific FPGA-based accelerator has been designed. The evolutionary design of image filters is a kind of regression problem which requires to evaluate a large number of training vectors as well as generations in order to find a satisfactory solution. By means of the proposed FPGA accelerator, very efficient nonlinear image filters have been discovered. One of the discovered implementations of an impulse noise filter consisting of four evolutionary designed filters is protected by the Czech utility model.

A different approach has been introduced in the area of logic synthesis. A method combining formal verification techniques with evolutionary design that allows a significant acceleration of the fitness evaluation procedure was proposed. The proposed system can produce complex and simultaneously innovative designs, overcoming thus the major bottleneck of the evolutionary synthesis at gate level. The proposed method has been evaluated using a set of benchmark circuits and compared with conventional academia as well as commercial synthesis tools. In comparison with the conventional synthesis tools, the average improvement in terms of the number of gates provided by our system is approximately 25%.

Finally, the problem of the multiple constant multiplier design, which belongs to the class of problems where a candidate solution can be perfectly evaluated in a short time, has been investigated. We have demonstrated that there exists a class of circuits that can be evaluated efficiently if a domain knowledge is utilized (in this case the linearity of components).

noise filter consisting of four evolutionary designed filters working with the 3×3 pixel filter window combined in a bank of filters is protected by the Czech utility model (a patent application was submitted in 2009).

To summarize, it has been demonstrated that the evolutionary approach is able to produce innovative solutions if an efficient evaluation procedure is employed. We have presented three different approaches to increase the performance of evolutionary algorithms and showed their applicability.

Future Work

The system employing the formal verification algorithm can be improved in several ways. The SAT-based approach can perform unsatisfactory for some problem instances. There are tens of extensions and algorithms that have been proposed by the SAT community to improve the performance of digital circuit equivalence checking. Some of the extensions can be adopted in order to improve performance of the proposed system. In future research it is also necessary to confirm that the proposed method is able to handle large-scale optimization problems if more advanced version of the SAT solver is utilized.

Since the evolutionary synthesis based on the formal verification algorithm can handle real-world (i.e. complex) circuits, it will be probably necessary to investigate the scalability of CGP representation, efficiency of utilized genetic operators and the utilized search algorithm. As it has been shown, the evolutionary strategy with population containing only two individuals surprisingly provided the best results.

We believe that there are other applications of evolvable hardware where formal verification algorithms are directly or indirectly applicable. Further investigation is needed to identify more complex applications that can benefit from this technique.

of a function can grow exponentially with the increasing number of input variables. In order to decide the functional equivalence problem in reasonable time, we need a representation that is both canonical and compact.

Since the satisfiability solvers were significantly improved during the last few years, the SAT-based equivalence checking becomes a promising alternative to the checking based on binary decision diagrams. The circuits to be checked are transformed into one Boolean formula (CNF) which is satisfiable if and only if the circuits are functionally equivalent [12].

In order to check whether two circuits are functionally equivalent, the following scheme is usually used. Let C_A and C_B be combinational circuits, both with k inputs denoted as $x_1 \dots x_k$ and m outputs denoted as $y_1 \dots y_m$ and $y'_1 \dots y'_m$ respectively. The corresponding primary outputs y_i and y'_i are connected using the XOR-gate. The corresponding primary inputs are connected as well. The goal is to obtain one circuit that has only k primary inputs $x_1 \dots x_k$ and m primary outputs $z_1 \dots z_m$, $z_i = XOR(y_i, y'_i)$. In order to disprove the equivalence, it is necessary to identify at least one XOR gate which evaluates to 1 for an input assignment. This can be done by checking one output after another (i.e. a Boolean formula is created and solved for each XOR gate separately) or by the all outputs approach (all XOR outputs are connected using the m-input OR gate; thus one Boolean formula is created and solved only).

4.2 Proposed Method

The goal of proposed evolutionary circuit synthesis method is to automatically create complex real-world circuits that will contain fewer gates than the circuits routinely designed using conventional synthesis algorithms. The proposed method consists of three steps. In the first step, the target circuit is synthesized using a conventional synthesis tool. The synthesis includes an optimization phase followed by mapping to the standard technology library. In the second step, the synthesized circuit is converted to the CGP representation. Finally, the algorithm based on CGP that employs a formal verification method in fitness evaluation to reduce the number of gates is used.

The initial solution (the seed) is constructed by means of mapping of the circuit obtained from conventional synthesis to the CGP representation. The mapping is straightforward since the CGP representation is in fact a netlist. The initial circuit is also transformed into the conjunctive normal form in order to create a reference solution for the SAT solver.

The following steps are used to calculate the fitness value of a candidate circuit. A new instance of the SAT solver is created and initialized with the reference solution (seed). Then, a candidate solution is transformed to a list of clauses that are submitted into the SAT solver. Then, miter circuit for combinational equivalence checking is created. The XOR gates are applied to each output pair and the outputs of the miter are combined together. Finally, the SAT solver is utilized to determine whether the submitted set of clauses is satisfiable or not. If the CNF is satisfiable, it means that there exists at least one assignment of input variables for which the reference circuits gives different response. Thus the fitness function returns -1 because the candidate circuit and the reference circuit are not equivalent; otherwise the number of utilized gates is returned.

CGP-Specific Performance Improvement Techniques

Although the system can be used directly as it was proposed in the previous section, we have introduces some techniques allowing the SAT solver even to increase the performance. The speed of the SAT-based equivalence checking depends mainly on the number of paths that have to be traversed in order to prove or disprove the satisfiability. In order to simplify the decision problem and increase the performance, CNF reduction based on finding structural similarities were proposed in literature.

In our case we can apply an elegant and simple solution. Since every fitness evaluation is preceded by a mutation, a list of nodes that are different for the parent and its offspring can be calculated. This list can be used to determine the set of outputs that have to be compared with the reference circuit and only these outputs are included into CNF. This can be achieved by omitting the unnecessary outputs during the miter creation phase. According to the experimental results, this simple approach enables to reduce the time needed to evaluate a candidate solution at least three times in average.

Although the SAT-based equivalence checking applied in the fitness function allows to optimize large logic circuits using genetic programming, there exist circuits (e.g. multipliers) for which the runtime of state-of-the-art SAT solvers grows exponentially with the increasing size of the problem instance. In order to shorten the decision time and improve the performance of the evolutionary approach, an enhanced evolutionary method has been proposed. The knowledge of the dissimilarities between the reference circuit and checked (i.e. candidate) circuit is applied to reduce the size of CNF instance. Comparing to the previous approach, this method does not require additional reference circuit since a parental circuit serves simultaneously as a reference. It also means that as the size of the optimized circuit gets smaller, the CNF derived from the reference requires fewer clauses. Thus, the performance has been improved at two different levels – by applying an approach that uses a variable reference and reducing the size of a miter circuit.

Table 2: The mean evaluation time in milliseconds for three fitness functions in the task of k-bit multiplier evolution.

circuit	n_i	n_o	seed (N_g)	t_{cgp}	t_{sat}	t_{imp}	t_{sat}/t_{imp}
mutliplier 7×7	14	14	238	8	1	4	0,3
multiplier 8×8	16	16	416	45	250	8	33,1
multiplier 9×9	18	18	540	183	1 789	17	105,4
multiplier 10×10	20	20	680	901	6 431	44	146,0
multiplier 11×11	22	22	836	n/a	316 333	88	3 607,8

In order to compare the time of evaluation for standard fitness function t_{cgp}, the proposed SAT-based fitness function t_{sat} and the enhanced SAT-based fitness function t_{imp}, the problem of the combinational multiplier optimization has been chosen. Table 2 gives the mean evaluation time. In the last column, we can observe a significant speedup achieved using the improved SAT-based fitness function.

ventional filters especially for lower noise intensity ($\leq 15\%$). Interesting results have also been achieved if the iterative filtering process has been taken into account. Whilst DWMF and PWMAD require from 5 to 10 iterations, the evolutionary designed random-valued impulse noise filter can produce the images of similar quality using two iterations.

We have also demonstrated that the combination of several evolutionary designed filters leads to the significant improvement of the quality of filtering even if the same filter window is utilized.

6 Conclusions

The contribution of the research can be summarized as follows. The thesis dealt with the design of various acceleration techniques that can significantly eliminate the scalability problem of evolutionary design of digital circuits at various levels of abstraction. Because the scalability problems significantly limit the application of evolutionary algorithms, we were primarily focused on the reduction of the fitness evaluation time which represents a serious issue in circuit evolution since the complex candidate solutions require a lot of time to be evaluated. The work has addressed three different design classes.

Firstly, we have shown that there are applications that require a single training vector in order to calculate the fitness value of a candidate solution. This approach is applicable in such cases where the utilized building blocks satisfy the properties of linearity. This method can be used at the gate level as well as function level.

A different approach has been proposed in the area of logic synthesis, where the resulting circuits must perfectly meet the specification. A method based on applying formal verification techniques that allow a significant acceleration of the fitness evaluation procedure was proposed, overcoming thus the major bottleneck of the evolutionary synthesis at gate level. The proposed algorithm can produce complex and simultaneously innovative designs, improving thus the state-of-the art logic synthesis tools.

Finally, a domain-specific single-chip FPGA-based accelerator has been proposed. This accelerator is designed to address the problem of the necessity of huge computation power for designing of digital circuits at the function-level. A typical example is the regression problem which includes e.g. evolutionary design of nonlinear image filters. The common feature of this class of circuits is that small imperfections in circuit behavior are tolerable, e.g. it is acceptable that the error of filtering is not zero but reasonably small value.

The Obtained Results

It was shown that the proposed method employing the formal verification algorithm can handle circuits intractable for common EHW-based approaches utilized so far (one of the benchmark circuits has over 100 inputs and more than 1000 gates) [50, 49, 60]. Apart from the fact that the obtained result indicates the evolutionary approach is able to generate the solutions better than conventional techniques, it also confirms the recent hypothesis that the conventional logic synthesis produces the results that are far from optimum [50].

By means of the proposed FPGA accelerator, very efficient nonlinear image filters have been designed [56, 61, 53, 52, 54]. One of the discovered implementations of an impulse

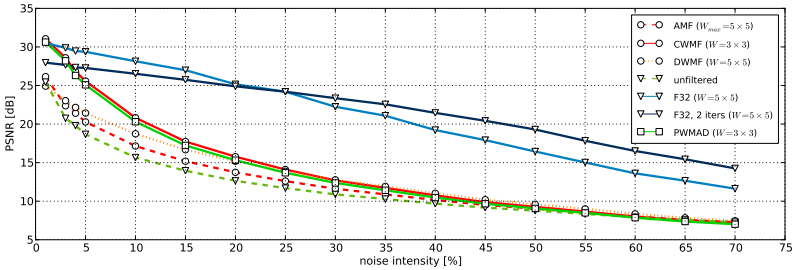


Figure 5: Comparison of various image filters and the evolved impulse burst noise filter using a set of 30 test images corrupted by impulse burst noise of intensity 1-75%.

burst noise affects adjacent horizontal pixels. If the burst noise has been applied in vertical direction, the pixels of filter window utilized by the evolved filters changed.

5.4 Proposed FPGA-based accelerator

In order to reduce the fitness evaluation time, an single-chip FPGA-based accelerator has been proposed [58, 48]. The accelerator consists of two main units – genetic engine and fitness unit. The fitness unit contains multiple instances of so called virtual reconfigurable circuit to evaluate several candidate solutions in parallel. The proposed accelerator running at 100 MHz can provide, using a moderate Xilinx FPGA XC2VP50, approximately 170 times higher performance in comparison with a common PC running at 2.4 GHz [56, 48]. This performance has been achieved by introducing a deeply pipelined architecture which significantly accelerates the evaluation of a candidate solution. In contrast with the CGP implementations based on GPUs or common CPUs, the hardware accelerator provides constant speedup independently on the size of the training set (if a suitable FPGA or external memory is chosen).

5.5 Summary

The experimental results show, that the evolutionary design approach can automatically produce image filters that are competitive with conventional filters not only in terms of filtering quality but also if a the implementation cost is considered. We observed that images filtered by evolved filters preserve more details (and thus provide a higher visual quality) than images filtered by conventional filters (e.g. median-based filters).

It has been also shown, that the designed filters require less FPGA resources than conventional filters. For example, the proposed 3-bank filter provides the same filtering capability as a standard adaptive median filter; however, using four times less slices. The more detailed analysis can be found in [55, 61].

The best results have been achieved when the concept of so called switching filter consisting of the detection and estimation part was applied during the evolutionary process. The switching filters evolved for salt-and-pepper shot noise are able to overcome the con-

4.3 Experimental Evaluation

In order to evaluate the proposed synthesis method, we have used benchmark circuits from LGSynth93 benchmark set. The goal was to optimize the number of gates of complex combinational circuits. Table 3 contains the achieved results as well as the best results obtained using the noncommercial and commercial tools. We have used the standard settings for the tools and technology library with the same set of gates as CGP, i.e. $\Gamma = \{\text{BUF}, \text{NOT}, \text{AND}, \text{OR}, \text{XOR}, \text{NAND}, \text{NOR}\}$. The results from SIS and ABC were obtained by iterative application of the synthesis script (1000 iterations).

Table 3: The min. number of gates obtained using the noncommercial tools (SIS, ABC), commercial tools (C1,C2,C3) and the proposed approach based on CGP.

circuit	SIS	ABC	C1	C2	C3	CGP	impr.
apex1	1394	1862	1439	1272	1368	847	33,4%
apex2	151	225	221	195	299	90	40,4%
apex3	1405	1737	1494	1332	1515	1038	22,1%
apex5	751	768	728	609	921	613	-0,7%
cordic	67	61	67	49	90	32	34,7%
cps	1128	1109	1150	975	967	585	39,5%
duke	406	356	417	366	357	260	27,0%
e64	192	384	183	191	255	129	29,5%
ex4p	488	523	468	467	555	349	25,3%
misex2	111	121	94	89	108	71	20,2%
vg2	95	113	88	83	109	78	6,0%

It can be seen that the commercial synthesis tools provide results that are comparable with the noncommercial synthesis tools such as ABC and SIS. None of the tools has provide better results than CGP (when CGP is seeded using the first result provided by SIS) with the exception of apex5 where the number of gates is very similar.

4.4 Summary

The current methods of evolutionary synthesis are capable of evolving either small and simultaneously innovative circuits or larger circuits that are not interesting from the implementation point of view because of their inherent inefficiency. According to our best knowledge, when a perfect synthesis scenario is considered, the most complex combinational circuit has been successfully evolved by Stomeo in [39]. This circuit, specified by a truth table, consists of tens of gates and has 17 primary inputs and one primary output. Even if this result can be considered by the EHW community as a great success because it was evolved from scratch, it has a marginal significance from the viewpoint of the logic synthesis because commercial synthesis tools are able to handle the combinational circuits having hundreds of inputs and thousands of gates. The main reason that prevents EA from evolving large and real-world competitive circuits is primarily caused by the problem of scalability of the fitness evaluation.

We have shown that it is possible to eliminate the mentioned scalability limits by introducing a different fitness evaluation procedure. The proposed method is based on applying

formal verification techniques that allow a significant acceleration of the fitness evaluation procedure, overcoming thus the major bottleneck of evolutionary design. Proposed algorithm can produce complex and simultaneously innovative designs, quite competitive with the state-of-the art logic synthesis tools.

Comparing to the standard CGP, we have demonstrated that the proposed method is able to evaluate over 40000× more candidate solutions in the same time when the common 32-input parity benchmark problem is considered. In addition, we have introduced some CGP-specific techniques that are able to further improve the performance of a SAT solver. Using the multipliers, known as hard benchmark problems, we have shown that the enhanced version of the proposed method tracking the changes between parent and its offspring is able to provide the additional speedup over 3000 when the 11-bit multiplier is considered. Note that the speedup increases mostly exponentially with the increasing complexity of the solved problem.

The proposed technique has been evaluated using the common LGSynth93 benchmark circuits. It has been shown that this approach enables to optimize large logic circuits having from tens to hundreds of inputs and thousands of logic gates. Using another benchmark set, we have demonstrated that the proposed method can handle the circuits that are known to be hard for common synthesis tools.

We have also demonstrated that despite the fact that various logic synthesis and optimization tools have been proposed in the recent 50 years, the logic synthesis/optimization problem has not been completely solved yet. Using the LGSynth93 benchmark we have shown, that the best-obtained results of conventional synthesis conducted using academia as well as commercial tools can be improved by the proposed method in 20-40%. The experiments with the hard-to-synthesize circuits show that a significant area improvement (33-99%) can be reached using the proposed evolutionary approach. In this case, CGP is able to discover structures, for which conventional synthesis completely fails.

5 Evolutionary Design of Nonlinear Image Filters

As low-cost digital cameras have entered to almost any place, the need for high-quality, high-performance and low-cost image filters is of growing interest. It is a well-known fact that the quality of preprocessing significantly influences the accuracy, reliability, robustness and performance of subsequent image processing steps such as segmentation, classification, recognition etc. In order to perform the required preprocessing (such as image filtering, edge detection etc.) a problem-specific filter has to be created. Traditionally, engineers use a library of predefined filters and operators and manually tune promising variants of these filters for a given application. In case that the filter should be implemented as a digital circuit, the parameters such as area, delay and power consumption play an important role.

Historically, *linear filters* became the most popular filters in image processing. However, there exist many areas in which the *nonlinear filters* provide significantly better results [8]. The advantage of nonlinear filters lies in their ability to preserve edges and suppress the noise without loss of detail. As there is no suitable general theory for the design of non-

searching a robust salt-and-pepper noise filter whose filtering quality can compete with the iterative filters and especially the adaptive median filter even for high noise intensity.

Random-valued impulse noise

The random-valued noise represents a more realistic type of impulse noise in which the corrupted pixels can take an arbitrary value from the entire scale available for the given class of images. The obtained results for the evolutionary design of switching random-valued impulse noise filters are given in Figure 4. The best evolved filter optimized for random-valued impulse noise is denoted as F17.

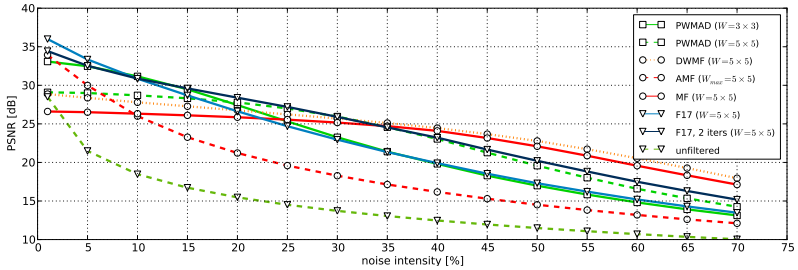


Figure 4: Comparison of various image filters and the evolved random-valued impulse noise filter using a set of 30 test images corrupted by random-valued impulse noise of intensity 1-75%.

Surprisingly, the evolutionary approach succeeded in the search for a robust filter for the noise of this type. As expected, the higher noise intensity requires more iterations of the filter to obtain an acceptable result. For lower noise intensities, very good results can be obtained using two iterations of the evolved filter F17 or conventional PWMAD filter with the 5×5 filter window. Unlike the case of the salt-and-pepper noise, the adaptive median filter fails in filtering random-valued noise even for lower intensity. On the other hand, the conventional DWMF filter exhibits a good quality slightly loosing some detail in comparison with the proposed F17.

Impulse burst noise

Impulse burst noise represents a serious issue because the principle of spatial locality is violated in this case. With the increased noise intensity, more consecutive rows may be affected and subsequent noise filtering becomes difficult as the filtered value need not be determined according to the values of the neighboring pixels.

The results for the evolutionary design of switching impulse burst noise filter are shown in Figure 5. The proposed filter F32 exhibits the best filtering quality in comparison with the commonly used conventional filters, surprisingly, even if we do not apply iterative filtering. We have analyzed the best evolved filter (denoted as F32) and recognized that this filter tries to avoid the calculations based solely on horizontal information [54]. This interesting property shows that the evolutionary approach was able to detect, that the

The extended version of the 3-bank filter, has been registered as Czech Utility Model under No. UV020017/2009.

Evolutionary Design of Switching Filters

The main disadvantage of the common median-based filters is that the filtering transformation is applied on all the pixels of the image regardless if the pixel represents the noise or not. Thus this approach results in loss of the image details and causes the degradation of the image quality especially if a larger filter kernel is used.

In order to address this problem, we have arranged a set of experiments with the aim to experimentally evaluate whether it is possible to design filters that are able to compete with conventionally used non-iterative as well as iterative filters suitable for the impulse noise removal task. In addition to the salt-and-pepper noise, random-valued impulse noise and impulse burst noise was investigated [52, 53]. The objective was to design filters working with 5×5-pixel kernel based on the switching concept. In particular, the evolutionary algorithm had to design a filter system consisting of a noise detector and nonlinear image filter. Both parts are evolved in parallel.

The quality of the evolved filters is compared with several conventional noniterative as well as iterative filters that are known to provide good results in removing of the particular type of impulse noise. In order to show the ability of the evolved solutions to improve the filtered image using the iterative processing, one and two iterations of these filters will be performed.

Salt-and-pepper impulse noise

Figure 3 summarizes the obtained results for the evolutionary design of switching salt-and-pepper noise filter. The evolved filter is denoted as F18. The results show that the evolutionary designed filter exhibits the best results for lower noise intensity (1%–15%) in comparison with the conventional filters. For higher noise intensity (i.e. greater than 20%) the AMF produces the images with the highest values of PSNR. However, the difference between F18 and AMF for these noise intensities is negligible.

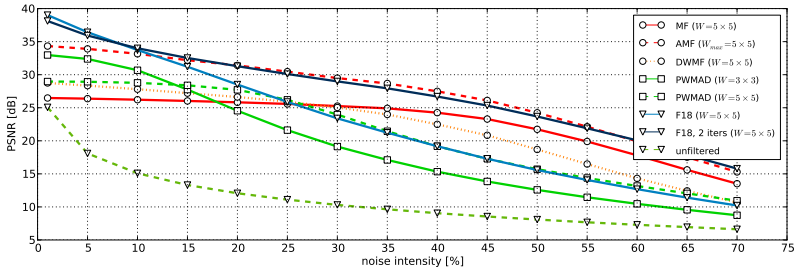


Figure 3: Comparison of various image filters and the evolved salt-and-pepper noise filter using a set of 30 test images corrupted by salt-and-pepper noise of intensity 1-75%.

To summarize the obtained results, the CGP-based evolutionary system succeeded in

linear operators, evolutionary design techniques have been utilized to accomplish this task in the recent years.

The pioneer work in this area has been done by Sekanina who applied Cartesian Genetic Programming in the image filter design task [36]. Sekanina has shown that evolutionary design techniques are able to generate slightly better solutions than the standard filters [37]. Unfortunately, his direct evolutionary design approach which works for low noise intensity does not work for higher noise intensities.

The goal of the research presented in this section is to show that by an innovative combination of evolved designs and conventional designs we are able to propose the systems that exhibit at least comparable quality with respect to the conventionally used approaches and simultaneously significantly reduce the overall implementation cost on a chip in comparison to standard approaches based on sophisticated filtering schemes, such as adaptive median filter.

5.1 Theoretical Background

Due to the imperfections of image sensors, images are often corrupted by a noise. The impulse noise is the most frequently referred type of noise. In most cases, impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or errors in the data transmission. We distinguish two common types of impulse noise; the *salt-and-pepper noise* (commonly referred to as intensity spikes or speckle) and the *random-valued shot noise*. For images corrupted by salt-and-pepper noise, the noisy pixels can take only the maximum or minimum values. In case of the random-valued shot noise, the noisy pixels have an arbitrary value. *Impulse burst noise* represents another type of impulse noise that consists of sudden step-like transitions between two or more discrete values at random and unpredictable times. In fact it is a variant of a random-valued shot noise that is characterized by longer duration. However, the impulse burst noise is a specific kind of noise which is difficult to filter even if a non-linear filter is used.

Traditionally, the impulse noise is removed by a *median filter* (MF) [1] which represents the most popular nonlinear filter even if the quality of the filtered images is poor in comparison with other advanced techniques. The success of the common median filter is mainly based on its simple and efficient software as well as hardware implementation. However, the standard median filter gives a mediocre performance even for images corrupted by impulse noise with lower intensity. The median-based filtering approach has been intensively studied and extended to several approaches such as *center weighted median filter* (CWMF) [18], more general *weighted median filter* (WMF) [5] or *order statistic* and *weighted order statistic filter* [27]. Nevertheless, all these median-based methods tend to smudge the image since applying the median filtering to the entire image would inevitably remove details presented in the image.

In order to overcome the main drawback of the median-based filters, a *switching-based median filtering* concept has been proposed [40]. This concept splits the filtering process into two parts – noise detection and noise replacement. The noise detector determines which pixels are affected by the impulse noise and only these pixels are replaced. However, the common problem of the proposed detection mechanisms is the necessity to predetermine the value of a threshold parameter which significantly influences the filtering quality.

The *adaptive median filter* (AMF) proposed in [17] is a robust approach which tries to identify and replace the affected pixels only. In contrast with the previous approaches, the detection method is based on the statistical ordered filters with gradually increasing kernel size. Compared to the common median-like filters, AMF provides significantly better results even for images corrupted with high intensity impulse noise.

Apart from the non-iterative algorithms, the iterative algorithms such as *pixel-wise median of the absolute deviations from the median* (PWMAD) [6] or *directional weighted median filter* (DWMF) outlined in [7] have been introduced. These approaches provide very good results if the random valued impulse noise is considered; they do not contain any varying parameters and require no previous training or optimization. The main disadvantage is apparent – the iterative approach places higher requirements for the memory resources especially in case of hardware implementation.

5.2 Evolutionary Design of Image Filters using CGP

In order to evolve image filters, we have applied cartesian genetic programming at functional level. A candidate filter is represented using $n_{cols} \times n_{rows}$ nodes arranged in a grid. Each node represents a two-input function which receives two 8-bit values and produces an 8-bit output. Every image filter is considered as a function (a digital circuit) of nine 8-bit inputs and a single 8-bit output. Each node can implement one of 8-bit nonlinear/linear functions such as maximum, minimum, saturated addition, addition, absolute difference, etc.

In order to evolve an image filter capable of removing a given type of noise, the original uncorrupted image is needed to determine the fitness values of candidate filters. The goal of CGP is to minimize the difference between the original image and the filtered image.

5.3 Experimental Results

The evolutionary design of image filters is a time consuming process. There is large amount of training data that have to be evaluated in order to determine the fitness value. In order to speed up the evolutionary design process and give the evolutionary algorithm possibility to explore large portion of search space, we have proposed an FPGA-based accelerator. This accelerator has been utilized in the following experiments.

Evolutionary Design of Salt-and-pepper Noise Filters and Noise-resistant Edge Detectors

In order to evaluate the performance of the proposed FPGA-based evolutionary platform, we have arranged four experiments. The objective was to evolve filter for salt-and-pepper noise of 5%, 10% and 20% intensity and design an edge detector which is able to deal with input images corrupted by the salt-and-pepper noise. Except the 5%-salt-and-pepper noise, the other problems were not approached by means of evolutionary design techniques in literature.

Even if the evolutionary design does not in general guarantee robustness of the evolved filters (i.e. that they exhibit a similar quality independently on the image content), the obtained results shown that the evolved filters exhibit very good performance not only

for the training images but also for the testing images [56]. The results outperform not only the common median filter but also the more advanced adaptive median filter. The most significant improvement has been achieved in the last experiment which combines the edge detector with noise removal. If a conventional edge detector is applied to the image corrupted by the salt-and-pepper noise, the noisy pixels are significantly amplified comparing to the evolutionary designed filter. Unfortunately this direct method is not able to evolve competitive filters for higher noise intensities (e.g. 40%).

Evolutionary Design of Robust Salt-and-pepper Noise Filter

The main feature of evolutionary design of image filters is that each evolutionary run typically produces a solution having different structure as well as properties. This behavior is usually undesirable because the evolutionary design does not guarantee the evolved filters are robust (i.e. that the proposed filters exhibit the constant quality for the whole class of images corrupted with a given type of noise). However, this feature can be exploited to create robust image filters [57, 55].

In order to evolve the filters for the bank, we have used the same approach as in the previous experiment. As a training image we utilized an image which was partially corrupted by 40% salt-and-pepper noise. Evolution was repeated 100 times. In the end, the best five filters were selected.

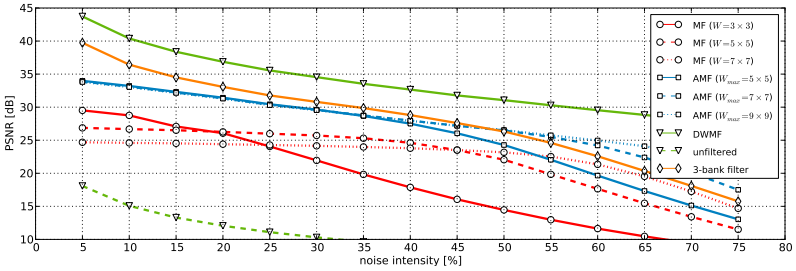


Figure 2: Comparison of various image filters and the extended version of the 3-bank filter using a set of 25 test images corrupted by salt-and-pepper noise of intensity 5-75%.

To evaluate the quality of the proposed bank, we have compared the resulting filter with the adaptive median filter on several test images which contain the salt-and-pepper noise with the intensity from 5% up to 75% corrupted pixels. Figure 2 summarizes the results of filtering properties of the proposed 3-bank filter, adaptive median filters, standard median filters and the DWMF filter which utilizes filtering windows of unlimited size.

Surprisingly, only three filters utilized in the bank are needed to obtain a bank filter which produces images of at least comparable visual quality to the adaptive median filter. We are convinced that this success is caused by the fact that although these filters perform the same task, they operate in a different way. While the median filter gives as its output one of the pixels of the filtering window, evolved filters can sometime produce new pixel values.