PAPER

On Comparison of Constrained and Unconstrained Evolutions in Analogue Electronics on the Example of "LC" Low-Pass Filters

Yerbol SAPARGALIYEV[†a)], *Member* and Tatiana KALGANOVA[†], *Nonmember*

SUMMARY The Evolutionary Electronics refers to the design method of electronic circuits with the help of Evolutionary Algorithms. Over the years huge experience has been accumulated and tremendous results have been achieved in this field. Two obvious tendencies are prevailing in the area over designers to improve the performance of Evolutionary Algorithms. First of all, as with any solution-search-algorithm, the designers try to reduce the potential solution space in order to reach the optimum solution faster, putting some constrains onto search algorithm as well as onto potential solutions. At the same time, the second tendency of unconstraining the Evolutionary Algorithms in its search gives unpredictable breakthroughs in results. Enabling the evolution to optimize with more experimental parameters devoted to drive the evolution and adjusted previously manually, is one of an example where such kind of unconstraining takes place. The evolution with the maximum freedom of search can be addressed as *unconstrained evolution*. The unconstrained evolution has already been applied in the past towards the design of digital circuits, and extraordinary results have been obtained, including generation of circuits with smaller number of electronic components. Recently, the similar method has been introduced by authors of this paper towards the design of analogue circuits. The new algorithm has produced promising results in terms of quality of the circuits evolved and evolutionary resources required. It differed from constrained method by its simplicity and represented one of the first attempts to apply Evolutionary Strategy towards the analogue circuit design. In this paper both conventional constrained evolution and newly developed unconstrained evolution in analogue domain are compared in detail on the example of "LC" low-pass filter design. The unconstrained evolution demonstrates the superior behaviour over the constrained one and exceeds by quality of results the best filter evolved previously by 240%. The experimental results are presented along with detailed analysis. Also, the obtained results are compared in details with low-pass filters designed previously.

key words: *evolutionary electronics, unconstrained evolution, analogue passive low-pass filter*

1. Introduction

Evolvable Hardware (EHW) is one of the most promising areas of today's electronics. Evolutionary Algorithm applied towards reconfigurable hardware enables finding a solution among global solution space. The EHW where the ultimate goal is a circuit design is also referred to as Evolutionary Electronics [4]. The Evolutionary Electronics gives an alluring opportunity for an amateur in the field to reach out the same results as professional one, possessing mostly the knowledge of Darwinian's laws and inspiration. The evolutionary algorithm, navigated by fitness values, provides randomly created and mutated chromosomes. Each chromosome encodes the structure for a circuit and has to be evaluated by a fitness function assigning a fitness value. The fitness value shows how close the current hardware structure is by its behavioural and circuit characteristics to the required one. The circuits evolved may have unconventional designs and less of all depend on personal knowledge of a designer [2]–[11].

For instance, using simulation software, low-pass filters [2]–[7] and amplifiers [3]–[5] are successfully designed with the help of evolutionary algorithm. In [15] the unconstrained evolution, both spatially and temporally, has been applied intrinsically towards the digital FPGA-based reconfigurable hardware. The behaviour of FPGA is defined by allowing any connections among modules, letting the evolution to evolve the granularity of modules as well as the regimes of synchronization. By releasing the full repertoire of FPGA behaviours, the evolution had been able to find out a highly efficient electronic structure, which required 1-2 orders less silicon area to achieve the same performance as conventional design did.

In analogy to this approach, the unconstrained evolution had been applied in [17] towards the originally analogue circuits on the example of "LCR" low-pass filter. In that case as well as for analogue electronics in general, the range of circuit-structure-checking rules at the netlist composition stage, prohibiting the invalid circuit graphs, are regarded as the main constraints for the design methodology. In essence, the unconstrained evolution refers to the use of a circuit modification rule at the netlist composition stage (the R-support) which legitimize invalid circuits, while constrained refers to the conventional checking/pruning procedure.

In this paper, we consider both constrained and unconstrained extrinsic evolution for analogue circuit design on the example of "LC" low-pass filter. The results of experiments will enable us to decide whether the unconstraining the circuit design rules leads to the evolution of filters with the better filter characteristics than that of filters produced by conventional constrained evolution. The paper is organized as follows: the next section gives an overview of previous work in the area. Section 3 introduces both evolutionary techniques, constrained and unconstrained. Section 4 describes the experimental results. Section 5 compares obtained results with the filters evolved previously as well as designed conventionally. And, finally, the last section concludes the paper.

Table 1 Some previous works on the evolution of analogue circuits.

	Koza et al, [2]	Lohn et al, [3]	Goh, et al [5]	Zebulum, et al [4]	Grimbleby [7]	Dastidar, et al [8]	Ando, et al [6]	Sripramong et al [9]	Sapargaliyev et al [17]	Proposed in this paper
Publication year	1997	2000	2000	1998	1999	2005	2003	2002	2006	2006
Type of circuit evolved	LC, LCR, LCRMD	LCR, MR	LCR	LCR, MR	LC	M	LCR	LCRM	LCR	LC
Type of evolutionary algorithm	GP	GA	GA	GP, GA	GA	GA	GP, GA	ES+sim. annealing	ES	ES
Circuit-structure-checking rules	Partially	Yes	Yes	Yes	N/A	Yes	Yes	Yes	No	No
Parameter optimization	No	No	No	No	Yes: numerical	Yes: GA	Yes: GP, GA	Yes: hill-climbing	No	No
Sweeping strategy	ILG	ILG	ILG	ILG,OLG, UDIP	ILG	OLG	N/A	Fixed	ILG	ILG
Circuit growth method*	1	2	1	1	1	1	1	No	1	1

* 1- technique where the place for a new element within a circuit is to be chosen arbitrarily, 2- where the circuit growth is along the way the current/voltage usually goes from input to output. N/A – data is not available. L-inductor, C-capacitor, R-resistor, M-transistor, D-diode. GP-genetic programming, GA-genetic algorithms, ES-evolutionary strategy. ILG-increasing length genotype, OLG-oscillating length genotype, UDIP-uniformly distributed initial population [4].

2. Previous Work on Analogue Filter Evolution

As well as in any evolutionary search, in electronic circuit design *the freedom* of evolutionary search is respected as crucially important for successful results. In [2], [3] and [5] the freedom of search is emphasized, but has not been realized completely.

The considerable results were obtained by Koza et al. in [2]. They used Genetic Programming (GP) circuit-constructing program trees with four kinds of circuit-constructing and automatically defined functions. The last one let them get as results the filters with regular structures within the circuit. They utilized a procedure providing the DC path to ground from each node of a circuit by adding the giga-Ohm resistance, allowing any kind of connections among capacitors. This let them avoid the most of "node floating" errors and reach the amount of invalid circuit graphs up to 2%. Later, they simplified each circuit by removing redundant resistances and replacing all series and parallel compositions. The main drawback of this experience is that the technique required large computing power as well as the methodology was very complex for implementation.

The larger computational efforts in a circuit evolution required by GP were proven by Zebulum et al. [4] and Ando et al. [6], where they have given the comparison between GP and Genetic Algorithm (GA). The first work was made as analogy to biology concept with comparison of performance among different types of sweeping strategies: Increasing Length Genotypes, Oscillating Length Genotypes, and Uniformly Distributed Initial Population. The second work was about evolution of real hardware for robustness purposes.

Recently in [17] the unconstrained evolution has been proposed for analogue circuits. For this purpose the R-support elements were introduced to participate in circuit design process. The evolved LCR low-pass filter was competitive by its characteristics with the best ones evolved before. Moreover, the number of elements inside the circuit and the number of evaluations were lower.

The previous development in evolution of analogue circuits design is summarized in Table 1. The analysis of Table 1 reveals that most of approaches developed in analogue circuit domain previously are based on the circuit-structure–checking rules for avoiding the invalid circuit graphs. In contrast, the experimental results were promising in the unconstrained search space [17]. Furthermore, most of the approaches provided search with GP and GA, and only one described in [17] used purely evolutionary strategy.

3. Constrained and Unconstrained Evolution of "LC" Circuits

3.1 General Structure of a System

Both constrained and unconstrained evolutions are based on one system. The structure of the system contains 4 main steps shown on Fig. 1.

Step1: The Start-block sets the initial conditions for evolutionary process as well as initializes the population. The last one is carried out as follows: generate an embryo-circuit and provide the preliminary grow of this circuit up to 3 elements (except embryo) before the first evaluation. It also sets the target-circuit to evolve ("LC" low-pass filter), seed for random number generator (RNG) and the list of elements' parameters to be used by evolution. The last one is set to E-12[†] series of the totally 51 parameters for each element.

Step 2: The evolutionary strategy part sets the particular parameters for evolutionary strategy, such as: mutation rate, population size, selection and termination criteria. It

---

[†]E-12 standard in electronics is most popular nowadays and sets 12 parameters per decade, e.g., 1.0, 1.2, 1.5, 1.8, 2.2, 2.7, 3.3, 3.9, 4.7, 5.6, 6.8, 8.2.
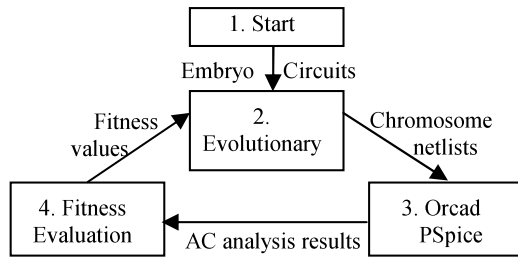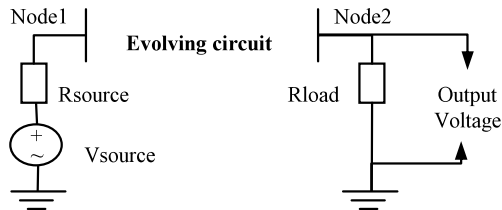
Fig. 1   The flowchart of the experiment.

Fig. 2   Embryo circuits.

modifies the genotype and produces the population of chromosomes in the form of cir-batch-file towards PSpice. At this stage the constrained evolution also includes the circuit-structure-checking rules, prohibiting invalid circuits to be generated. Unconstrained evolution here applies the special design rules that allow the circuit generation to be unconstrained.

Step 3: PSpice has been used in non-interactive batch simulation mode to reduce AC analysis (evaluation time).

Step 4: Fitness evaluation block, as described bellow, fairly assigns the fitness values towards each circuit from AC analysis produced by PSpice in the form of batch out-file and passes the results to evolutionary strategy block.

The PC program written in C programming language describes all 4 parts and unites them in one code.

The embryo-circuit refers to the elements that are definitely known as essential for the target circuit, stay unchangeable during all the evolution and take place in each circuit netlist. In our case of "LC" low pass filter there are 3 such kind of elements: AC voltage source, source resistance and load resistance. We define the embryo circuit similar to the most popular case [2], [3], [5]–[7], [11], [17] where the circuit is driven by an incoming AC voltage source with a 2 V amplitude, has the source resistance R_{source}=1 kΩ and the load resistance R_{load}=1 kΩ (Fig. 2). The output voltage is measured on the pins of R_{load}.

An example of PSpice netlist with the correspondent schematic is shown on Fig. 3 with three embryo elements followed by the netlist.

The circuit growth methodology is very simple and is similar to one reported in [5], [17]: adding one gene to each chromosome at the same time.

The improvement of the circuits is driven by evolutionary strategy (ES) with disruptive selection scheme [12].
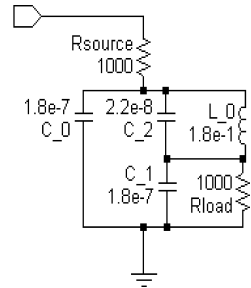

Fig. 3   Chromosome representation at constrained evolution. The typical netlist (left) and the correspondent schematic (right). Three first elements (V_s, R_{source} and R_{load}) compose the embryo-circuit.

3.2   Chromosome Representation

The linear circuit representation is used, similar to one that exploited in [4]. That is every element of a circuit is represented as a particular gene, and each of 4 element's features: name of an element, its parameter, and every its pin is represented by particular loci in a gene. The advantage of this chromosome representation is that it directly maps the description of one element in the PSpice netlist.

The number of genes in chromosome is equal to the number of elements in a circuit. Figure 3 demonstrates the netlist of the circuit and its corresponding schematic. The circuit contains four elements (apart from embryo-elements). Therefore the chromosome representation contains four genes. Each gene is described by four loci. For instance, the first gene in the netlist is "L_0 1 2 1.8e-1," which describes the first inductor with pins N1=1 and N2=2 and with a value of 1.8e-1H. The first locus is reserved for the element's type (L or C), the second and the third loci are reserved for the first and the second pins, and the last loci is for the element's parameter. Similarly the remaining genes in the chromosome are encoded.

3.3   Constrained and Unconstrained Evolutions in "LC" Circuit Design Domain

Constrained evolution is supposed to enable the circuit-structure-checking rules to prevent the evaluation of invalid circuit graphs by simulation software. It was the only way before in analogue circuit domain to reduce the errors during the circuit analysis. However, in [17] these rules were regarded as constraints reducing the diversity of population generated and such kind of evolution was considered as *constrained evolution in analogue circuit domain.* In [2] these constrains were partially released by adding to each node of a circuit the Giga-Ohm resistance, and the best results by circuit characteristics were obtained.

As in [17], we call absolutely *unconstrained evolution* of an analogue circuit the process of a circuit netlist generation during which no any circuit-structure-checking rules applied and all the circuits are counted as valid graphs except ones that have elements with dangling nodes and with
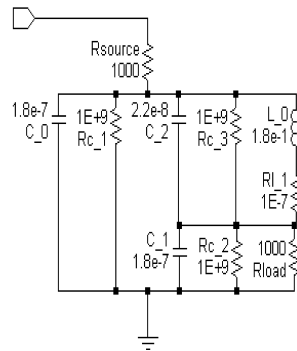
Fig. 4 Chromosome representation at unconstrained evolution. The typical netlist and the correspondent schematic with R-support elements.

Fig. 5 The fragment of the typical experimental data flow: dependency of the best fitness value along the generation number. At generations 29, 38, 48 and 58 the population of chromosomes has gone through ANEM stage, i.e. one element has been added to each circuit.

isolated subcircuits. There are two main kinds of invalidities that are unacceptable for most of simulation software: the nodes that have no DC path to ground (tackled in [2]) and loops that involve inductors and/or voltage source. The most of methodologies in the area (Table 1) just prohibit such kinds of invalidities to appear. In our case the adding to each capacitor the Giga-Ohm resistance in parallel and the adding to each inductor the Micro-Ohm resistance in series, at the stage of PSpice cir-file generation, allows to avoid these invalidities. We call such kind of resistance as *R-support*. Using R-support and avoiding the dangling nodes makes almost any randomly generated circuit as valid, and indeed becomes an absolutely *unconstrained*.

Figure 4 demonstrates how unconstrained evolution generates the circuits with R-support. The circuit depicted on Fig. 3 once being processed by unconstrained evolution will have the view shown on Fig. 4. Each element line describing inductor (L_0 on Fig. 4) is followed by R-support element (Rl_1) in series with infinitesimal parameter; and each element line describing conductor (C_No) is followed by R-support element (Rc_No) in parallel with infinite parameter.

If in unconstrained evolution the circuit-structure-checking rule is applied to this circuit after the fitness assignment as a part of the *simplification procedure*, that is, every R-support element is checked whether it could be removed without damage to the current fitness of a circuit, then the circuit could contain a few R-support elements or even could not contain them at all. For instance, the circuit on Fig. 4 after simplification process becomes the circuit on Fig. 3.

3.4 Mutation

The whole process for both constrained and unconstrained evolutions consists of two types of operations over the circuit to inculcate the simplest ILG sweeping. Every time the best fitness of a generation is not improved, the population falls into "Add new element mutation" (ANEM) procedure, i.e. one randomly generated gene is added to each chromosome except the chromosome with the best fitness value.
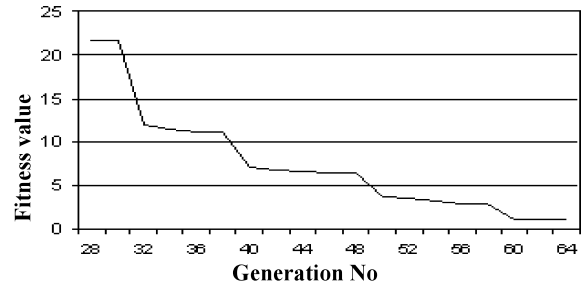
The "circuit structure mutation" (CSM) performs mutation over any of four loci of randomly chosen gene. If the mutation comes to a pin connection, the whole structure of a circuit is changed; however the total amount of elements stays unchangeable, the number of nodes of a circuit could be reduced or increased.

At the start the population comes to ANEM for preliminary circuit growth, and, after evaluation, it falls into CSM. The following switching rule between ANEM and CSM does manage in further: whenever the mutation within CSM does not bring any improvement in the fitness, the algorithm immediately in the next generation switches to ANEM. As the experiment shows the fitness of an individual never improves by infinitesimal values, what makes the switch period between two kinds of mutations finite in time. While CSM searches for the best circuit within the given amount of elements, it unavoidable brings to a fitness improvement stuck. But the adding one element to a circuit during ANEM stage significantly improves fitness reviving the whole process (Fig. 5).

3.5 Fitness Function

We perform AC-analysis along 96 points between 1 Hz and 100 kHz (19 per decade), and measure the absolute deviation voltage between ideal value and value produced by PSpice. We set the fitness evaluation in the analogy with [2], i.e. we distinguish as acceptable a voltage in the passband between 970 mV and 1 V and a voltage in the stop-band between 0 V and 1 mV:

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

where V_{ideal}^i is the voltage in i-th point for ideal filter and $V_{measured}^i$ is the voltage in the i-th point obtained for evolved filter; p is a number of points evaluated in both stop and pass bands equalling 96. We regard voltage at any other location as unacceptable, punishing it as follows:

$$F_2 = 10 \times F_1.$$

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

4. Experimental Results

4.1 Task Setting

Most of the works in the analogue circuit design start from evolving a passive low-pass filter, that is a convenient tool for probation of evolutionary technique and tuning the evolutionary algorithm parameters towards the more sophisticated designs [2], [3]. The behaviour of low-pass filter between frequencies 1Hz and 100 kHz, cut-off frequency 1 kHz and transition band 1 kHz has been actively researched through in [2]–[4]. Thus, the performance of proposed evolutionary technique could be evaluated more precise if the evolution target will have exactly the same filter properties.

The purpose of the experiments below is to compare constrained and unconstrained evolutions on the example of "LC" low-pass filter. For both cases the described in Sect. 3 methodology was absolutely identical except the circuit-structure-checking rules applied in constrained evolution and the R-support technique applied in unconstrained case.

4.2 Initial Settings

For both cases we ran thirty nine experiments with different seeds for RNG: 1-39.

Experimentally it has been established that the disruptive selection scheme [12] is the best: only 9% of the best chromosomes and 1% worst ones are to be chosen for the next generation. Being chosen for the next generation each chromosome contributes 10% of the next population size, that is, totally 10% of selected chromosomes generate 100% population of the next generation. A static mutation rate of 5% was then applied to each chromosome randomly changing with equal probability every loci of a gene. The evolutionary strategy is deserved the name of the simplest evolutionary algorithm, because it doesn't content the recombination stage: all the offspring chromosomes are identical to a correspondent parent.

Population size of 20,000 chromosomes was set. The potential problem of the approach described is that without having enough diversity, the best fitness value could not grow at all within CSM stage and the evolution could just run quickly though out the ANEM stage, exceeding the element limit. There are two ways to tackle this problem: increase the size of the population and, keeping smaller population size, to increase the number of generations (repetitions) that population goes through CSM. For example, technically speaking, a population of 20,000 chromosomes equivalent to a population of 2,000 chromosomes with 10 repetitions or to a population of 50 with 400 repetitions. We used the larger-population-size approach because it gives an advantage in speed: throughout all the three mentioned cases the average chromosome processing speed were 2100, 860 and 250 chromosomes per minute correspondently. It can be explained by the fact that OrCAD PSpice requires

the Web-licensing each time (generation) the software runs, that is, 10–20 seconds are required for each generation to be loaded independently of its size. Therefore in further stages we used only the evolutionary strategy with population size of 20,000[†].

The termination criterion is set the reaching by any of an individual 29 genes, e.g. circuit elements, before simplification (except R-support and embryo elements).

4.3 Experimental Results

All 39 experiment results are shown in Appendix. The best result for constrained evolution has been obtained at chromosome 11,863 of generation 61 ($20,000 \times 60 + 11,863 = 1,211,863$ individuals), with 26 elements (without embryo), with the best fitness value 0.0041[††].

The schematic and the voltage response of the best circuit are shown on Fig. 6. We have got the non-monotonic

Fig. 6　The schematic after simplification and the voltage response of the best low-pass filter evolved with Constrained Evolution. Node1 and Node2 correspond to that ones in Fig. 2.

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

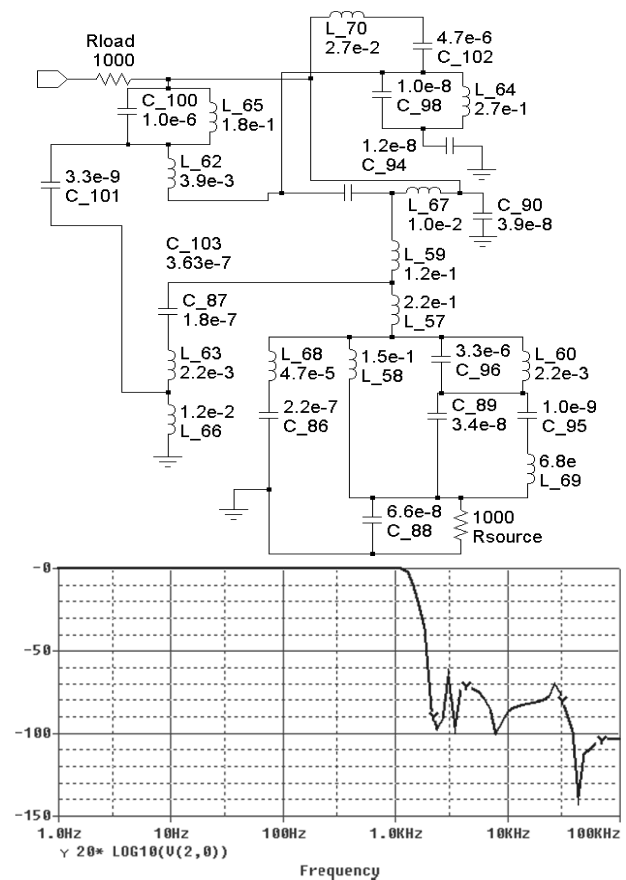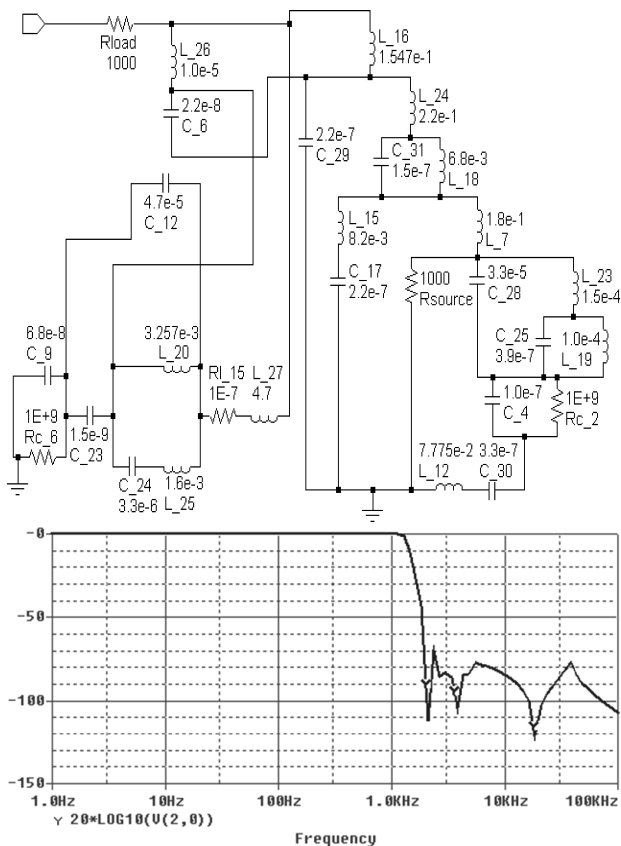[††]Notice that the smaller the fitness value the better filter evolved.

Fig. 7 The schematic after simplification and the voltage response of the best low-pass filter evolved with Unconstrained Evolution. Node1 and Node2 correspond to that ones in Fig. 2.

filter with the following features: the maximum absolute attenuation in the pass band is 0.0015 dB and the maximum attenuation in the stop band is −66 dB.

The best result for unconstrained evolution has been obtained at chromosome 19,993 of generations 60 (1,199,993 individuals), with the best fitness value 0.002855, which is 44% better than the best got in constrained evolution. The schematic after simplification and the voltage response of the best circuit are shown on Fig. 7. As could be seen we have evolved a non-monotonic filter consisting of 27 elements (without embryo) among which 3 are the R-support and with the following features: the maximum absolute attenuation in the pass band is 0.00118 dB and the maximum attenuation in the stop band is −69 dB.

5. The Comparison of Constrained and Unconstrained Evolutions

Experimental data in Appendix show that in average the unconstrained evolution requires 62.7 generations, which is 5% longer than that of constrained one; however the average fitness value (0.0126) is 30% better. The similar situation is with the result producing best fitness: the unconstrained evolution at RNG seed 7 run out 60 generations and reached the fitness value 0.002855, which is 44% better than that of

constrained one (0.0041, reached at generation 61 at RNG seed 24). This result can be explained by two reasons. The analysis of results have shown that exploiting the circuit-structure-checking rules in constrained evolution still allows significant amount of mistaken circuits to be sent to simulation software. For example, in our case, where the rule that checks invalid circuits up to 5 nodes in the chain around new/mutated element whether they are floating, and the rule that checks up to 4 elements around new/mutated element whether they involved in the inductor/voltage source loop, still allowed about up to 15% of error circuits in average to be generated. In contrast, during running unconstrained evolution, the amount of invalid graphs among all randomly generated never exceed 0.03%. That is the effective population size at unconstrained evolution was 15% longer than that of at constrained one.

The second reason for explaining the result is that each generation produced by unconstrained evolution contained more diverse chromosomes. The usage of R-support enabled the unconstrained evolution to create chromosomes that constrained evolution will never allow to appear. For instance, the best circuit reached at RNG seed 7 contains 3 R-support elements, without which the circuit loses the proper functionality and, thus, never could be replicated by constrained evolution.

In order to provide the fair comparison between obtained and previously published results, similar to [17], we validated each result using OrCAD PSpice-10.3, getting filter characteristics for each circuit and its fitness values. For this, we manually converted the published in [2], [3], [17] schematics into netlists and passed them trough OrCAD. The proof of correctness of this operation was verified by perfect match between our fitness value and one published in [2] for ladder low-pass filter. In Table 2 all comparison characteristics are of 2 types: filter and evolution. The first three lines of the table show that compared filters are of the same nature and it is correct to make a comparison among them. The last two lines of filter characteristics, attenuations in the stop and pass bands, are two major features that directly define the fitness value. The number of elements and individuals are not influence on fitness value; however they are presented for comparison. Comparing the characteristics of low-pass filter designed previously and presented in this paper, we conclude that the filter with the best fitness value has been evolved using the proposed technique.

6. Conclusion

The previous process of extrinsic evolutionary design of analogue circuits always been constrained to generation of only the valid circuit graphs. However, recently the introduction of R-support elements in [17] allowed to release these constraints and to apply the absolutely unconstrained evolution towards analogue circuits.

In this paper we compared both constrained and unconstrained evolutions for analogue circuit design on the example of "LC" low-pass filters. Both evolutions were

Table 2   Comparison table of filter and evolution characteristics among works published before and present.

	Ideal filter	10order Chebyshev filter [16]	Koza 1 et al, [2] elliptic	Koza 2 et al, [2] ladder	Koza 3 et al, [2] bridge-T	Lohn et al, [3]	Sapargaliyev, Kalganova [17]	Best of Constrained evolution	Best of Unconstrained evolution
Filter Characteristics									
Pass band, V	1	1	1	1	1	1	1	1	1
Stop band, V	0	0	0	0	0	0	0	0	0
Transition band, KHz	0	1	1	1	1	1	1	1	1
Maximum absolute attenuation in the pass-band, dB	0	0.035	0.179	0.0175	0.137	0.0144	0.012	0.0015	**0.0012**
% of improvement	-	*2866%*	*15069%*	*1383%*	*11510%*	*1120%*	*917%*	*27%*	-
Maximum attenuation in the stop band, dB	$-\infty$	-83	-72	-61	-60	-59	-59	-66	**-69**
% of improvement*	-	*-20%*	*-4%*	*12%*	*13%*	*14%*	*14%*	*4%*	-
Evolution characteristics									
Fitness value		0.0259	0.0805	0.0071	0.0502	0.0134	0.00969	0.0041	**0.0029**
% of improvement*	-	*809%*	*2725%*	*149%*	*1661%*	*369%*	*240%*	*44%*	-
No. Elements		10	25	14	15	24	12	26	27
No. Individuals		N/A	2,048,000	N/A	997,000	829,790	1,211,863	1,199,993	
Circuit simulator[1]		MicroSim				SPICE	OrCAD		

* The value "% of improvement" shows the correlation of difference between the value above in the same column and the correspondent value in the column "Best of Unconstrained evolution".

All the circuit simulator cores are based on the same equations developed in Berkley at University of California describing the behavior of electronic devices, and originally inculcated inside the SPICE. However, the different simulators could have the different interfaces, modes and analysis speeds. Nowadays, MicroSim is no longer supported and OrCAD is one of the most popular commercial simulators nowadays.

based on Evolutionary Strategy in combination with increasing length genotype sweeping strategy. The unconstrained evolution demonstrated the superior behaviour over the constrained one (30%) as well as over the all filters evolved previously. The best fitness value obtained previously has been improved by 240%.

The results received enable us to conclude, that unconstrained evolution as the method inculcated the absolute freedom of evolutionary search, provides the key features for more successful evolutionary design of analogue circuits: the diversity of chromosomes and larger effective population size.

References

[1] A. Thompson, Hardware Evolution: Automatic Design of Electronic Circuits in Reconfigurable Hardware by Artificial Evolution, PhD Thesis, University of Sussex, Brighton, UK, 1996.

[2] J.R. Koza, F.H. Bennett, III, D. Andre, M.A. Keane, and F. Dunlap, "Automated synthesis of analogue electrical circuit by means of genetic programming," IEEE Trans. Evol. Comput., vol.1, no.2, 1997.

[3] J.D. Lohn and S.P. Colombano, "Automated analogue circuit synthesis using a linear representation," Proc. 2nd Int'l Conf. on Evolvable Systems: From Biology to Hardware, pp.125–133, Springer-Verlag, Berlin, 1998.

[4] R.S. Zebulum, M.A. Pacheco, and M. Vellasco, "Comparison of different evolutionary methodologies applied to electronic filter design," Proc. IEEE Int. Conf. on Evolutionary Computation, pp.434–439, IEEE Press, Piscataway, NJ, 1998.

[5] C. Goh and Y. Li, "GA automated design and synthesis of analogue circuits with practical constraints," Proc. 2001 Congress on Evolutionary Computation, vol.1, pp.170–177, 2001.

[6] S. Ando and H. Iba, "Analogue circuit design with a variable length chromosome," Congress on Evolutionary Computation, pp.994–100, IEEE Press, 2000.

[7] J.B. Grimbleby, "Hybrid genetic algorithms for analogue network synthesis," Proc. 1999 Congress on Evolutionary Computing, pp.1781–1787, Washington, D.C., 1999.

[8] T.R. Dastidar, P.P. Chakrabarti, and P. Ray, "A synthesis system for analog circuits based on evolutionary search and topological reuse," IEEE Trans. Evol. Comput., vol.9, no.2, pp.211–224, April 2005.

[9] T. Sripramong and C. Toumazou, "The invention of CMOS amplifiers using genetic programming and current-flow analysis," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.21, no.11, pp.1237–1252, Nov. 2002.

[10] R. Zebulum, A. Stoica, and D. Keymeulen, "Experiments on the evolution of digital to analog converters," Proc. 2001 IEEE Aerospace Conference, Big Sky, MN, March 2001. ISBN: 0-78-3-6600-X. (Published in CD)

[11] J. Hu, X. Zhong, and E. Goodman, "Open ended robust design of analog filters using genetic programming," Proc. Genetic & Evolutionary Computation Conference-2005 (GECCO-2005), vol.2, pp.1619–1626, ACM Press, Washington, D.C., June 2005.

[12] T. Kuo and S.-H. Hwang, "Using disruptive selection to maintain diversity in genetic algorithms," Appl. Intel., vol.7, pp.257–267, 1997.

[13] M. Brameier, On Linear Genetic Programming, PhD Thesis, University of Dortmund, Dortmund, Germany, Feb. 2004.

[14] K. Vesselin and J. Miller, "The advantages of landscape neutrality in digital circuit evolution," Proc. 3rd International Conference on Evolvable Systems (ICESO3), LNCS, pp.252–263, Springer, 2000.

[15] A. Thompson, "Artificial evolution in the physical world," in Evolutionary Robotics, ed. T. Gomi, AAI Books, 1997.

[16] LC Filter Design, Available at: http://www-users.cs.york.ac.uk/~fisher/lcfilter/

[17] Y. Sapargaliyev and T. Kalganova, "Absolutely free extrinsic evolution of passive low-pass filter," accepted for publication at Canadian Conference on Electrical and Computer Engineering (CCECE-CCDEI), May 2006.

Appendix

Table A·1   The experimental results for 39 different seeds of random number generator (RNG). Against each RNG seed there are two values: the best fitness value reached and the generation number when this fitness value has appeared.

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

Yerbol Sapargaliyev   received MSc. degree in physics from Kazakh State University, Kazakhstan, in 1993 and MA in Economics from Kazakhstan Institute of Management Economics and Strategic Research, in 1995. From 1994 to 1999, he worked as a Marketing Manager in Arna-Sprint Data Communications Ltd. From 1999 to 2005 he worked as a Marketing Manager in Huawei Technologies, Co., Ltd. Since 2005 he joined Bio-inspired intelligent system research group, School of Engineering and Design, Brunel University, UK, where he is currently a PhD Research Student.

Tatiana Kalganova   received MSc. degree from Belarusian State University of Informatics and Radioelectronics, Belarus in 1994 and Ph.D. degree from Napier University, UK in 2000. In August 2000 she has joined Electronic and Computer Engineering Department, Brunel University. Her research interests are evolvable hardware, ant colony algorithms, scalability in AI systems. She was awarded a personal grant from the Education Ministry of the Republic of Belarus for distinctive achievements in the field of exact sciences in 1997, and a grant from the International Soros Science Education Program (ISSEP) for distinctive achievements in the field of exact sciences in 1996.