

基于 Memetic 算法的电路演化设计研究

莫宏伟¹, 徐立芳²

(1. 哈尔滨工程大学自动化学院, 黑龙江哈尔滨 150001; 2. 哈尔滨工程大学工程训练中心, 黑龙江哈尔滨 150001)

摘 要: 针对传统演化算法在设计数字逻辑电路时存在的演化速度缓慢和容易陷入局部最优解等问题, 设计了一种 Cartesian 进化编程编码的电路演化 Memetic 算法, 采用遗传算法作为全局搜索方法, 并设计了适合电路演化的基本门种类局部搜索策略. 通过一位全加器电路证明所提出的 Memetic 硬件演化算法的搜索能力, 实验证明所提出的算法能够有效地完成进化任务, 具有较强的全局快速和局部搜索能力.

关键词: 演化硬件; 演化算法; Memetic 算法; 局部搜索策略

中图分类号: TP181 **文献标识码:** A **文章编号:** 0372-2112 (2013)05-1036-05

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.05.034

Research on Evolvable Hardware Design Based on Memetic Algorithm

MO Hong-wei¹, XU Li-fang²

(1. Automation College, Harbin Engineering University, Harbin, Heilongjiang 150001, China;

2. Engineering Training Center, Harbin Engineering University, Harbin, Heilongjiang 150001, China)

Abstract: In order to overcome the problems of traditional evolution algorithms in designing digital logic circuits, such as slow evolution speed and premature convergence, a memetic algorithm (MA) for evolving digital circuits is presented based on CGP encoding. It adopts genetic algorithm as global search method. A local search strategy based on gate type is proposed. The search ability of the proposed method is tested by evolving one-bit full adder. Experimental results show that it is efficient in solving the problems of hardware evolution and has the ability of fast global and local convergence.

Key words: evolvable hardware; evolutionary algorithm; memetic algorithm; local search strategy

1 引言

演化硬件(Evolvable Hardware, EHW), 是一种将进化算法与可重构器件有机结合的硬件系统^[1]. 它以演化算法作为全局搜索的主要工具, 以现场可重构器件作为评估手段和实现载体, 寻求在不依赖先验知识和人工干预的情况下, 通过演化来获得满足给定要求的电路和系统结构, 进而使系统自动、实时地调整其内部结构, 以适应内部条件(如局部故障)和外部环境的变化. 目前, 研究设计演化算法进行电路演化设计成为演化硬件研究领域的热点之一^[2~6]. 演化算法是演化硬件的搜索工具, 演化算法的搜索效率直接影响着演化收敛的速度. 日本, 欧美等在 EHW 基本原理以及 EHW 应用拓展等方面取得了很大的成就, 并且一直引领此领域的研究方向. 而国内在这一方面的研究才刚刚起步. 目前仅有中国科学院自动化研究所、武汉大学、中国科技大学、西安电子科技大学、中国地质大学等少量机构对 EHW 的基本原理和实现技术进行了初步的探讨研究.

2 相关工作

武汉大学的康立山教授为代表的一些人率先在国内开展了 EHW 的研究, 他们总结了当时国际上 EHW 的研究现状与关键技术, 并用简单 PLD 作为硬件基础尝试对简单的电路结构进行演化, 后期又提出以函数型(树型结构函数)可编程器件作为进行 EHW 研究的基本结构, 演化实现了四位比较器电路^[7].

中国科学技术大学的王熙法教授及其团队将 EHW 技术用于设计冗余容错电路, 运用遗传算法演化生成多个功能相同而结构却不同的电路, 接着选择性集成所演化生成的多个电路, 进而提高系统容错性能^[8].

西安电子科技大学的赵曙光教授等人对电路演化设计方法进行了大量的研究, 他们提出一种基于最小项表达式的编码方式, 设计了多目标自适应遗传算法对多个电路进行多目标演化设计, 算法演化实现了偶校验器、数字乘法器、四位可级联比较器、交通灯控制器等电路^[9].

南京航空航天大学王友仁教授带领的 EHW 研究团队研究了 EHW 的基本原理以及关键技术,他们基于 FPTA 模型借助离线演化方式演化了运算放大器等电路,后期又致力于组建数字系统演化基础平台,数字电路与系统在线演化方法及容错方法,胚胎型 EHW 等的研究^[10].

深圳大学朱明程教授主要对 EHW 结构和相应的算法进行了研究,他在研究 EHW 算法的基础上在单一的电路体系结构下演化实现了拥有多种不同算术功能的演化电路细胞结构^[11].汪鹏君等采用量子遗传算法搜索多输出 Reed-Muller 逻辑电路最佳极性^[12],另外,还有一些科研机构也针对 EHW 各个研究方向进行了研究.EHW 作为一个新兴的研究领域,目前的投入产出比较小,距离产业化还有一定的距离.

目前用于数字电路演化设计的演化算法主要为遗传算法 GA^[13]ES^[14]和 PSO^[15],其中 GA 主要是采用简单 GA、自适应 GA 变长染色体编码 GA 和对遗传操作算子改进后的 GA.GA 具有较好的鲁棒性和内在的并行性,在数字演化硬件研究中应用广泛,但是其维持多样性和局部搜索能力较弱,易陷入局部最优而导致早熟,尤其是当搜索空间增大时,算法的收敛性较差,收敛速度较慢.

针对上述问题,本文引入人工设计电路的经验和规则,采用 Cartesian 进化编程编码方法^[16],设计了一种适合数字电路进化的 MA.设计了基本门种类局部搜索策略,并结合遗传算法的全局搜索能力.它不仅具有很强的全局寻优能力,同时每次交叉变异后均进行局部搜索,通过优化种群分布及早剔除不良种群,进而减少迭代次数,加快算法的求解速度,保证解的质量.通过一位全加器演化实验证明该方法在一定程度上可提高演化电路的规模及其设计效率,未来发展可用于滤波器等电路设计^[17].

3 电路演化 MA

3.1 MA 基本思想

MA 是基于人类文化演化思想的群体智能优化算法^[18].该算法通过个体信息的选择、信息的加工和改造等文化传播行为产生群体智能,并为工程优化问题提供有效的解决方法.MA 首先初始化种群,生成一组空间分布比较合理的染色体(随机解),然后通过迭代搜索最优解.在每一次迭代中,染色体通过交叉、变异和局部搜索进行更新.

MA 提供了一种求解复杂优化问题的通用方法,对不同的问题有不同的解决方式,可以广泛应用于函数优化、组合优化、车间生产调度等许多领域^[19].MA 的基本过程如下.

MA 步骤:

- 1 初始化群体
- 2 计算群体中每个个体适应度
- 3 复制所选择的个体形成一个新群体
- 4 群体进化操作:交叉和变异
- 5 执行局部搜索策略
- 6 如果停止条件满足,算法结束,否则返回 2.

MA 与传统 GA 的主要区别在于局部搜索策略,因此针对不同问题设计不同的局部搜索策略成为近年来 MA 研究热点^[20].

针对不同的函数优化问题或组合优化问题采用不同的适用策略,例如爬山法、贪婪算法、内点法等.

3.2 编码设计

运用 EHW 技术来进行数字逻辑电路设计,首先要考虑的问题就是如何将电路表示成便于演化算法设计的形式,也就是要进行编码方法设计.目前主要的编码方式有矩阵编码法、CGP(Cartesian Genetic Programming)编码法,本文就是采用 CGP 的编码方式.CGP 编码法是一种基于 FPGA^[13]结构的编码模型,其主体由一个 $r \times c$ 的节点单元阵列组成,其中 r 代表节点单元阵列的行数, c 代表节点单元阵列的列数.每个节点单元 C_{ij} 都有两个输入和一个输出,完成特定功能.为了对数字逻辑电路进行编码设计,每一种基本电路单元功能都应该有唯一的编码与其对应.表 1 给出了 CGP 编码规则.其中一种节点单元完成功能对应的编码,每个节点单元之间的连接方式也是可以编程的,每个节点单元的输入来自另外节点单元的输出或者整个电路的原始输入,为了避免反馈循环,规定电路只能够由左至右连接,即列 i 上节点单元的输入仅可以是列 $j(j < i)$ 上节点单元的输出.整个节点单元阵列的输入和输出数由待演化的电路所决定;整个阵列的大小的选择应该基于待演化电路的功能复杂度;整个阵列的输出则直接取自阵列中某些节点单元的输出.

由上述描述可知,一个染色体的 CGP 编码方式如图 1 所示:每个节点有 3 位字符组成,In1、In2 表示一个节点单元的两个输入来源,Gate 则表示该节点单元完成的功能.按照表 1 所示的节点功能对照进行编码,然

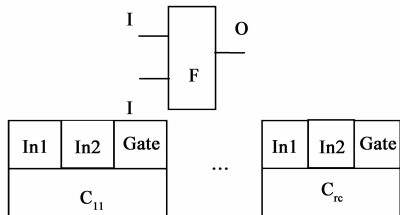


图1 染色体编码方式示意图

后按一定的顺序组合每个节点单元的编码。

表 1 节点功能对照表

功能编号	功能	符号表示
0	and	.
1	or	+
2	not	-
3	xor	⊕

3.3 算法的评价函数设计

在电路演化算法设计过程中,染色体适应度的评估方法是否能够准确反映电路的优劣程度至关重要,它直接影响到演化操作,并导致算法能否得到最优解.对于数字逻辑电路来说,采用测试集仿真演化电路功能是染色体适应度评估的基本方式,针对 EHW 技术所设计数字逻辑电路的评价方法主要是通过列举所有可能的输入组合,验证其电路的输出值.把得到的输出值一一与期望电路真值表进行比较,将比较的结果作为演化电路个体好坏的评价标准.除此之外,在某些时候,还希望所设计的电路完成特定功能的同时能够拥有较小的电路规模,即将演化电路所用的门电路资源作为另一个评价标准,这样就必须进行多目标算法设计.为了验证适应度评估函数对演化算法性能的影响,本文中个体电路的评价函数 F 分为 f_1 和 f_2 两个部分,分别用来评价个体电路的功能和结构:

$$f_s = 2^{n_i} \times n_o \tag{1}$$

$$\begin{cases} f_1 = f_1 + 1, & \text{If } \text{bit } i \text{ of } \mathbf{Y} = \{\text{bit } i \text{ of } \mathbf{T}\} \\ f_2 = f_2 + 1, & \text{If } \{\text{gatetype} = \text{wire}\} \end{cases} \tag{2}$$

$$F = \begin{cases} f_1, & F < f_{i0} \\ f_1 + f_2, & F \geq f_{i0} \end{cases} \tag{3}$$

其中 n_i 和 n_o 分别代表电路的输入和输出个数, $\mathbf{Y}$ 代表个体对应电路的输出向量, $\mathbf{T}$ 代表期望电路输出向量. Gatetype 为门类型, wire 为连线.

3.4 算法演化策略

3.4.1 全局搜索策略

全局搜索策略选择:本文全局搜索策略采用 GA^[9].

①交叉策略:由于 CGP 编码要求每一个节点的输入来自前一系列节点,为了保持染色体的有效性和整体性,算法将编码中的每一个节点的编码看成一个整体进行单点交叉.图 2 给出演化过程中交叉实现过程.

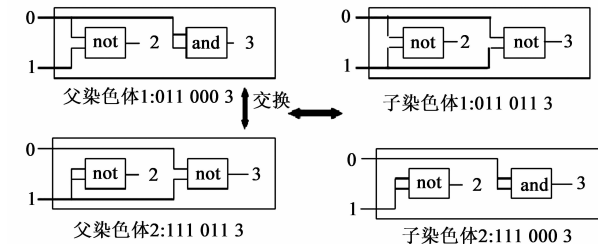


图 2 电路交叉

②变异策略:针对每个节点的两个输入和最终电路阵列的输出进行等概率变异.变异的一个例子如图 3 所示.

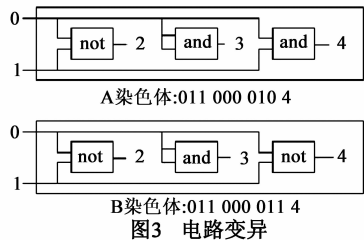


图 3 电路变异

③选择策略:当群体规模很大时,对于基于适应值比例的选择与基于排名的选择方式,其额外计算量较大.而基于局部竞争机制的锦标赛(tournament)选择策略则能够一定程度避免这个问题.在选择的时候,在群体中随机地选择 n 个染色体进行适应度比较,将其中适应值最好的个体选择进入下一代,参数 n 称为竞赛规模,常取 $n = 2$.

3.4.2 局部搜索策略

本文的 MA 中,设计了一种适合电路演化设计的局部搜索策略——演化基本门种类局部搜索(Gate Type Local Search, GTLS). 在本文中,MA 第 5 步采用 GTLS 策略,如下:

GTLS 局部搜索算法

- 1 对群体中所有染色体
- 2 对每个染色体的门基因位,利用邻近门类型替换基因位中门类型
- 3 如果新的解适应度更高
- 4 则新解取代旧解
- 5 对所有染色体执行结束
- 6 对所有个体执行结束

4 实验设计及结果分析

为了验证本文所设计的 MA 硬件电路进化性能,本文将 MA、GA 和 PSO 演化设计了一位全加器.一位全加器真值表如表 2 所示.表中有三个输入 $\{A, B, C_{in}\}$ 和两个输出 $\{S, C_{out}\}$,其中 A, B 是加数, C_{in} 是低位向本位的进位, S 是本位的和, C_{out} 是本位从高位来的进位.

表 2 一位全加器真值表

A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1
1	1	1	1	1

表 3 给出所进化电路涉及的门类型.

表 3 门种类设置

门种类	逻辑门
6	{AND, OR, XOR, NOT, NAND, NOR, WIRE}
4	{AND, OR, XOR, NOT, WIRE}
3	{AND, OR, XOR, WIRE}

实验采用不同的门种类设置如表 3 所示.演化矩阵 $r \times c = 3 \times 3$, 评价函数如前文所述. 因为一位全加器和

一位全减器的输入个数 $n_i = 3$, 输出个数 $n_o = 2$, 所以 $f_{i0} = 16$. 针对不同的门的种类本文做了两组实验:第一组实验中仅函数 f_1 评价演化结果, 第二组实验采用 f_1 和 f_2 评价演化结果. 每组实验 30 次. 图 4(a) ~ (f) 给出了 MA、GA、PSO 演化一位全加器(1-bit full adder, FA1)的收敛代数分布图.

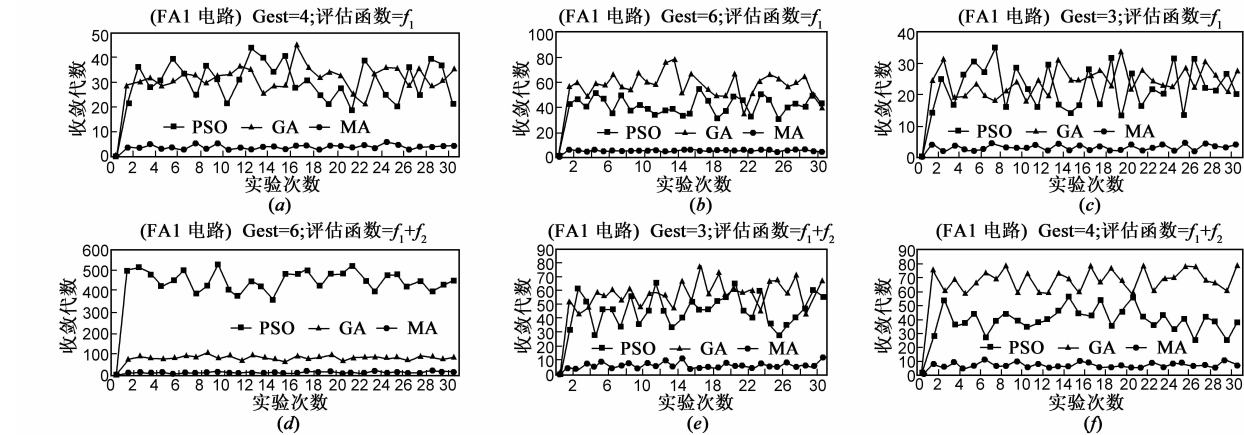


图4 一位全加器(FA1)电路种群收敛代数分布图

表 4 FA1 实验统计结果

算法	门种类	评估函数 f_1		评估函数 $f_1 + f_2$	
		A_V	S_V	A_V	S_V
PSO	Gset6	42.2	45.8	449.0	918.4
	Gset4	30.6	37.2	40.7	87.9
	Gset3	21.4	40.0	46.0	89.1
GA	Gset6	58.4	12.6	83.5	61.4
	Gset4	32.5	11.3	68.6	96.3
	Gset3	24.7	7.0	57.6	19.9
MA	Gset6	4.1	1.6	9.6	2.9
	Gset4	3.8	0.8	6.5	0.9
	Gset3	2.8	0.8	5.1	0.5

表 4 中 A_V 、 S_V 分别为各算法演化结果平均值、演化代数方差. 从以上统计结果可以看出, 在设计一位全加器电路时, MA 所用的平均演化代数比 GA、PSO 都少很多, 说明 MA 算法能够快速地找到最优值, 即所设计的局部搜索策略 GTLS 确实加快了算法的收敛速度. 从演化代数的方差来看, MA 算法演化的方差较小, 演化生成电路的演化代数较为稳定, 这为用演化算法演化更复杂的电路提供了新思路. 另外, 还可以看出在电路演化设置过程中采用不同的门种类设置对演化结果影响较大. 对于 PSO 和 GA 来说, 表现最好和最坏的分别是 Gset3 和 Gset6, 而对 MA 来说却是 Gset4 和 Gset6, 很明显演化算法的演化效率会受到门种类设置的制约. 从结果中还可以看出, 以 $f_1 + f_2$ 为评价函数的算法与以 f_1 为评价函数的算法做对比, 前者的收敛性都较慢, 即以 $f_1 + f_2$ 为评价函数的算法要付出更大的演化代价, 说明

评价函数对演化过程有较大影响.

5 结论

本文将 MA 应用到了电路演化设计当中, 采用遗传算法作为全局搜索方法, 设计了基于门种类的局部搜索, 与 GA、PSO 等其他智能优化算法相比较, 实验证明该算法有很好的收敛速度, 为电路演化算法设计提供了新思路. 演化硬件最终要付诸工程应用, 必须考虑功耗、面积、速度等进行多目标设计, 如何将这些目标融合在适应度函数中进行评估, 并保证算法效率是我们下一步研究应该考虑的问题.

参考文献

[1] S Lukas. Evolvable Components: From Theory to Hardware Implementations[M]. Berlin: Springer, 2004.

[2] Z G Bao, T Watanabe. A novel genetic algorithm with cell crossover for circuit design optimization[A]. IEEE International Symposium on Circuits and Systems[C]. Taipei, Taiwan: IEEE Press, 2009. 2982 – 2985.

[3] A P Shanthi, R Parthasarathi. Practical and scalable evolution of digital circuits[J]. Applied Soft Computing, 2008, 9(2): 1 – 7.

[4] S M Cheang, K H Lee, KS Leung. Applying genetic parallel programming to synthesize combinational logic circuits[J]. IEEE transactions on evolutionary computation, 2007, 11(4): 503 – 520.

[5] 何国良, 李元香, 史忠植. 基于精英池演化算法的数字电路在片演化方法[J]. 计算机学报, 2010, 33(2): 365 – 372.

- HE Guo-liang, LI Yuan-xiang, SHI Zhong-zhi. Elitist pool evolutionary algorithm for on-line evolution of digital circuits[J]. Chinese Journal of Computers, 2010, 33(2): 365 – 372. (in Chinese)
- [6] 梁厚军. 电路进化设计算法研究[D]. 合肥: 中国科学技术大学, 2009.
- Liang Hou-jun. Research on Evolutionary Design Algorithms for Circuits[D]. Hefei, Anhui: University of Science and Technology of China, 2009. (in Chinese)
- [7] 陈毓屏, 康立山, 潘正君. 一个新的研究领域 – 演化硬件[J]. 航空计算技术, 1998, 28(1): 1 – 8.
- [8] 林勇, 罗文坚, 王煦法. 硬件电路的选择性进化冗余[J]. 中国科学技术大学学报, 2006, 36(5): 523 – 529.
- LIN Yong, LUO Wen-jian, WANG Xu-fa. Selective redundancy of evolved circuits[J]. Journal of University of Science and Technology of China, 2006, 36(5): 523 – 529. (in Chinese)
- [9] 赵曙光, 王宇平, 杨万海, 等. 基于多目标自适应遗传算法的逻辑门级进化方法[J]. 计算机辅助设计与图形学学报, 2004, 16(4): 402 – 406.
- Zhao Shuguang, Wang Yuping, Yang Wanhai, et al. Multi-objective adaptive genetic algorithm for gate-level evolution of logic circuits[J]. Journal of Computer-Aided Design & Computer Graphics, 2004, 16(4): 402–406. (in Chinese)
- [10] 姚睿, 于盛林, 王友仁, 等. 采用主流 FPGA 的数字电路在线生长进化设计方法[J]. 南京航空航天大学学报, 2007, 26(5): 582 – 587.
- YAO Rui, YU Sheng-lin, WANG You-ren, et al. Online growing evolution and evaluation approach based on mainstream FPGA[J]. Journal of Nanjing University of Aeronautics and Astronautics, 2007, 26(5): 582 – 587. (in Chinese)
- [11] 徐渊, 杨波, 朱明程, 等. 模拟退火与遗传算法结合的数字 FIR 滤波器硬件进化算法[J]. 计算机辅助设计与图形学学报, 2006, 18(5): 674 – 679.
- Xu Yuan, Yang bo, Zhu Mingcheng. A new genetic algorithm involving mechanism of simulated annealing for digital FIR evolving hardware[J]. Journal of Computer-Aided Design & Computer Graphics, 2006, 18(5): 647 – 679. (in Chinese)
- [12] 汪鹏君, 李辉, 吴文晋, 王伶俐, 张小颖, 戴静. 量子遗传算法在多输出 Reed-Muller 逻辑电路最佳极性搜索中的应用[J]. 电子学报, 2010, 38(5): 1058 – 1063.
- WANG Peng-jun, WU Wen-jin, WANG Ling-li, ZHANG Xiao-ying, DAI Jing. Application of quantum genetic algorithm in searching for best polarity of multi-output reed-muller logic circuits[J]. Acta Electronica Sinica, 2010, 38(5): 1058 – 1063. (in Chinese)
- [13] 吴志波, 张忠萍, 陈菊平. 基于 FPGA 的高重复率距离门控电路实现[J]. 电子学报, 2010, 38(4): 919 – 0922.
- WU Zhi-bo, ZHANG Zhong-ping, CHEN Ju-ping. The implementation of range-gate control circuit with high repetition rate based on FPGA[J]. Acta Electronica Sinica, 2010, 38(4): 919 – 0922. (in Chinese)
- [14] C Reis, J A Tenreiro Machado, J Boaventura Cunha. Evolutionary design of combinational logic circuits[J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2004, 8(5): 507 – 513.
- [15] H G Beyer. The Theory of Evolution Strategies[M]. Berlin, Heidelberg: Springer, 2001.
- [16] P W Moore, G K Venayagamoorthy. Evolving digital circuits using hybrid particle swarm optimization and differential evolution[J]. International Journal of Neural Systems, 2006, 16(3): 163 – 177.
- [17] 田海燕, 曹鹏, 王明飞, 许鹏鹏, 孟凡俊. 宽带中频采样抗混迭滤波器设计与实现[J]. 电子学报, 2010, 38(2A): 60 – 64.
- TIAN Hai-yan, CAO Peng, WANG Ming-fei, XU Peng-peng, MENG Fan-jun. The design and implementation of the anti-aliasing filter for wideband IF signals sampling[J]. Acta Electronica Sinica, 2010, 38(2A): 60 – 64. (in Chinese)
- [18] J F Miller, P Thomson. Cartesian Genetic Programming[M]. Edinburgh, UK: Springer Berlin, 2000.
- [19] X H Chen, YS Ong, M H Lim, K C Tan. A multi-facet survey on memetic Computation[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(5): 591 – 608.
- [20] H shibuchi, Y Hitotsuyanagi, N Tsukamoto, Y Nojima. How to choose solutions for local search in multi-objective memetic algorithms[A]. Proceedings of the 2010 International Workshop on Nature Inspired Computation and Applications[C]. Hefei: USTC, 2010. 516 – 525.

作者简介



莫宏伟 男, 1973 年 5 月出生, 黑龙江省佳木斯人. 教授、博士生导师, IEEE 会员. 1996 年、2003 年和 2005 年在哈尔滨工程大学分别获工学学士、工学硕士和工学博士学位. 现为哈尔滨工程大学模式识别与智能系统研究所副所长, 主要从事计算智能、人工智能、智能系统等方面的研究工作.

E-mail: honwei2004@126.com



徐立芳 女, 1973 年 12 月出生, 黑龙江省富锦人. 1996 年、2005 年和 2008 年在哈尔滨工程大学分别获工学学士、工学硕士和工学博士学位. 2005 年进入哈尔滨工程大学工程训练中心, 现为讲师, 从事智能控制、计算智能方面的研究.

E-mail: mxlfang@163.com