一种基于 FPGA 的自适应遗传算法

房 磊 张焕春 经亚枝
(南京航空航天大学自动化学院 南京 210016)

摘 要: 采用了一种适合硬件实现的自适应遗传算法，利用种群的最大适应度f_{max}、最小适应度f_{min}和适应度平均值f_{ave}这 3 个变量来自适应地控制整个种群的交叉概率 p_c 和变异概率 p_m。选用了适合硬件实现的选择、交叉、变异算子，并将它们设计成流水线结构, 同时，将选择算子与适应度计算并行化，大大提高了算法的运行效率。整个设计采用了XILINX公司的XC2V1000 型号的FPGA芯片。算法利用VHDL语言来描述。实现后的测试表明，这种自适应遗传算法明显改善了算法的搜索性能和全局收敛性，同时利用硬件实现有效减少了运行时间，使其在一些实时性要求较高的场合得到应用成为可能。

关键词：自适应遗传算法, 并行, FPGA
中图分类号: TP18, TN495 **文献标识码**：A **文章编号**：1009-5896(2005)11-1829-05

An FPGA Based Adaptive Genetic Algorithm

Fang Lei Zhang Huan-chun Jing Ya-zhi
(*College. of Automation Eng. , Nanjing University of Aeronautics & Astronautics, Nanjing* 210016, *China*)

Abstract A hardware implement Adaptive Genetic Algorithm (AGA) is proposed in this paper. The adaptive algorithm uses three parameters, i. e. f_{max} , f_{min} and f_{ave} to determine the p_c and p_m of the whole generation adaptively. The selection , crossover and mutation operators which are suitable for hardware implement are selected and they are designed in a pipelining architecture . The parallelism of the selection operator and the computation of the fitness of the individual enhance the efficiency of the algorithm greatly. The hardware GA processor has been implemented in XILINX FPGA(Field Programmable Gate Arrays) XC2V1000. The VHDL language is used to describe the whole algorithm. Experimental results indicate that the adaptive genetic algorithm improves the global convergence and search performance of the algorithm greatly. The hardware implementation of the algorithm reduces the running time efficiently and makes it possible to apply in time-critical systems.

Key words Adaptive genetic algorithm, Parallel, FPGA

1 引言

遗传算法[1](Genetic Algorithms, GA)是模拟生物在自然环境中的遗传和进化过程而形成的一种自适应全局优化概率搜索算法。由美国的Holland于 1975 年提出。它在组合优化[2]，模式识别[3]，图像处理[4]，自适应控制，和人工智能等领域得到了广泛的应用。

人们在遗传算法上的研究主要集中在软件实现的算法上，但这种方法速度慢，大大限制了其在一些实时性要求较高的场合中的应用。而用硬件实现遗传算法，可以充分利用算法内在的并行特性，同时将算法设计成流水线结构，可以在很大程度上提高算法的运行速度。另一方面，近年来 FPGA 芯片技术迅猛发展，规模已达到几百万门甚至上千万门，为实现较为复杂的设计提供了丰富的片内资源。

另一方面，由于简单遗传算法易早熟和陷入局部最优而不能保证算法的收敛，在现有的很多文献中出现了针对基本遗传操作的各种改进算法[5,6]，并取得了一定的效果。本文采用了一种改进的自适应遗传算法，并利用FPGA进行了硬件实现。测试表明，这种基于FPGA的自适应遗传算法有着较好的寻优能力和较高的寻优效率，遗传过程对初始种群性能的依赖性小，同时，硬件算法的速度要比软件算法快 3 个数量级。

2004-05-15 收到，2004-09-20 改回
国家部级基金资助课题

2 算法描述

2.1 遗传算法概述

遗传算法是一个多点并行的迭代过程，在每次迭代过程中都进行如下操作：将一组以一定的基因形式描述的候选解进行交叉和变异操作以及适应度的评价；选取参与后代的候选解。重复此过程，直至满足某种收敛准则而得到全局最优解。典型的遗传算法执行过程可以描述为：(1)确定控制参数；(2)初始化；(3)执行下列操作，直到满足停止条件为止：(a) 选择；(b) 交叉；(c)变异；(d) 计算并统计种群的有关参数。其中，选择、交叉和变异算子是主要的遗传算子。选择算子体现了适者生存的原则，交叉算子是种群进化的主要动力，变异算子则可以维持群体的多样性，防止出现未成熟收敛现象。

2.2 典型的自适应遗传算法

Srinivas等人[7]根据种群适应度集中时，增大交叉概率 p_c 和变异概率 p_m；种群适应度分散时，则减小 p_c 和 p_m 的思想，提出了一种自适应遗传算法(AGA)。其具体定义如下：

$$p_c=\begin{cases}k_1\dfrac{f_{\max}-f'}{f_{\max}-f_{\text{ave}}} & f'\geq f_{\text{ave}}\\ k_3 & f'<f_{\text{ave}}\end{cases}$$

$$p_m=\begin{cases}k_2\dfrac{f_{\max}-f}{f_{\max}-f_{\text{ave}}} & f\geq f_{\text{ave}}\\ k_4 & f<f_{\text{ave}}\end{cases}$$

式中f_{ave}为种群的平均适应度；$f_{\max}$为种群中的最大适应度；f'为交叉双方中适应度较大的个体的适应度；f为个体的适应度；k_1,k_2,k_3,k_4 为 $(0,1]$ 之间的常数。

这种算法根据每代个体适应度的改变自适应地改变 p_c 和 p_m，一方面保护了最优个体，同时也加快了较差个体的淘汰速度。但是该算法针对每个个体来改变 p_c 和 p_m，使其在某些情况下不易跳出局部最优解；同时，由于对每个个体都要分别计算各自的 p_c 和 p_m，大大增加了算法的计算量，降低了算法的运行速度，并且不适合硬件实现。

2.3 本文采用的改进自适应遗传算法

与 Srinivas 的 AGA 相比，文献[8]提出的改进 AGA 根据整个种群的适应度集中程度，自适应地调整整个种群的 p_c 和 p_m，而不是为每个个体计算 p_c 和 p_m，从而有效地减少了算法的计算量，使其更适合于硬件实现。

算法采用种群的最大适应度$f_{\max}$，最小适应度 $f_{\min}$和平均适应度f_{ave} 3 个变量来衡量种群的集中程度。整个种群的集中程度可以通过$f_{\max}$和$f_{\min}$的接近程度来反映，二者越接近，则算法越有可能陷入局部最优，此时就需要增大 p_c 和 p_m；种群内部适应度的分布情况可以通过$f_{\max}$和f_{ave}的接近程度来反映，二者越接近，表明该代个体越集中。本文在文献[8]算法的基础上，增加了控制 p_c 和 p_m 的调整幅度的参数 α,β，其大小由针对特定问题的先验知识决定，这样做的目的是能够有效地利用先验知识，并使其参与到整个进化过程中。 p_c 和 p_m 的计算公式如下：

$$p_c=\begin{cases}p_c\dfrac{1}{1-\alpha\dfrac{f_{\min}}{f_{\max}}}, & \dfrac{f_{\min}}{f_{\max}}>a,\ \dfrac{f_{\text{ave}}}{f_{\max}}>b\\ p_c, & \text{其他}\end{cases}$$

$$p_m=\begin{cases}p_m\dfrac{1}{1-\beta\dfrac{f_{\min}}{f_{\max}}}, & \dfrac{f_{\min}}{f_{\max}}>a,\ \dfrac{f_{\text{ave}}}{f_{\max}}>b\\ p_m, & \text{其他}\end{cases}$$

式中 a，b 为衡量种群是否集中的参数，$0<a<1$，$0.5<b<1$；当满足条件：$\dfrac{f_{\min}}{f_{\max}}>a$，$\dfrac{f_{\text{ave}}}{f_{\max}}>b$ 时，说明种群适应度集中，p_c 和 p_m 根据上式自适应地变化；如果不满足此条件，则说明种群的适应度分散，此时 p_c 和 p_m 保持最初的设定值。

3 硬件结构

硬件包括以下几个部分：选择模块，交叉模块，变异模块，适应度模块，控制模块和存储区。为提高算法的运行效率，将选择模块和适应度模块进行了并行化，并设计了两条进化流水线。图 1 为算法的结构框图。

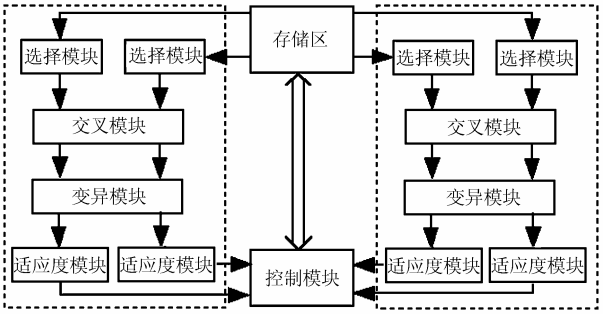

图 1 算法结构框图

3.1 并行化及流水线结构的设计

造成遗传算法运行速度慢的主要原因是计算个体的适应度所需的时间开销，解决这一问题的一个有效方法就是将适应度的计算并行化。这在硬件实现的遗传算法中可以方便地做到，在片内资源允许的情况下，只需增加适应度计算模块，便可以使个体适应度的计算同时进行，从而大大减少了计算的时间开销。在软件实现的遗传算法中，最少需要进行两次选择操作，才能够提供交叉、变异操作所需要的个体，

而用硬件实现时，只需提供两个同样的选择模块 (图 2)，便可使选择操作同时进行，从而将选择所需的时间缩短一半；另外，整条选择，交叉，变异，适应度计算的流水线也可以实现并行化。这些并行化的处理，大大提高了算法的执行效率。

在软件遗传算法中，由于程序只能顺序执行，因而，在进行一个遗传操作时，其他遗传操作总是处于停止状态。而用硬件实现时，可以将遗传操作设计成流水线结构。例如，交叉模块在完成交叉操作后，将产生的两个子染色体送到变异模块，同时向选择模块发送准备好信号，通知选择模块可以送新的染色体。这样，在一个遗传操作进行时，其他的遗传操作也在同时进行，从而大大提高了算法的运行速度。

3.2 遗传算子的硬件化

3.2.1 选择算子　本文选用了一种类似标准遗传算法中轮盘赌的选择方法。具体步骤如下：

(1) 计算种群中所有染色体(V_s)适应度之和

$$F = \sum_{s=1}^{pop\_size} f(V_s) \tag{1}$$

(2) 在[0 , 1]区间内生成一个均匀分布的伪随机数 r，计算 F 和 r 的乘积 Q, 将它作为选择参考概率;

(3) 选择。若$Q \leqslant f(V_1)$，则选择第一个染色体V_1, 否则选择第k个染色体V_k $k(2 \leqslant k \leqslant pop\_size)$，使得

$$\sum_{s=1}^{k-1} f(V_s) < Q \leqslant \sum_{s=1}^{k} f(V_s) \tag{2}$$

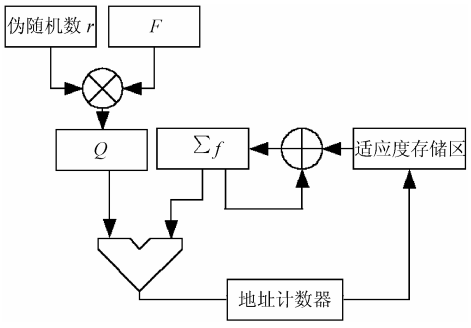

图 2　选择模块结构图

当$\sum_{s=1}^{k} f(V_s) < Q$ 时，地址计数器便会加 1，第 K+1 个个体的适应度就会和前 K 个个体的适应度之和相加，再次与 Q 比较，当 $Q \leqslant \sum_{s=1}^{k} f(V_s)$ 时，地址计数器便停止计数，当前地址对应的个体被选择。

当进行下一次选择时，地址计数器将会被复位，同时，一个新的伪随机数将会替代上一次选择时的伪随机数。

3.2.2 交叉变异算子　由伪随机数发生器提供的伪随机数S_1和交叉概率 p_c 相比较，如果$S_1 < p_c$，执行交叉；如果$S_1 \geqslant p_c$，父代个体不进行交叉而直接进入变异模块(图 3)。

交叉和变异算子均采用适合硬件实现的逻辑算子，这种算子的优点是结构简单，并且在一个时钟周期内就可以完成相应的操作。对交叉而言，可以通过两个父代个体的按位相“与”和按位相“或”来得到两个子代，例如：

父代	子代	算子
1110001101	0110000101	“与”操作
0110010111	1110011111	“或”操作

本文设计了一个移位寄存器，在每次变异操作前，伪随机数发生器将会依次产生 m 个伪随机数，它们分别与变异概率相比较，并将结果保存到移位寄存器中，这个移位寄存器便作为变异模板，它的每一位将决定染色体中对应的基因位是否进行变异。变异模块采用这样的结构(图 3)，使每个基因的变异能够并行进行，整个变异过程能够在一个时钟周期内完成。变异采用的逻辑算子是“异或”和“同或”。

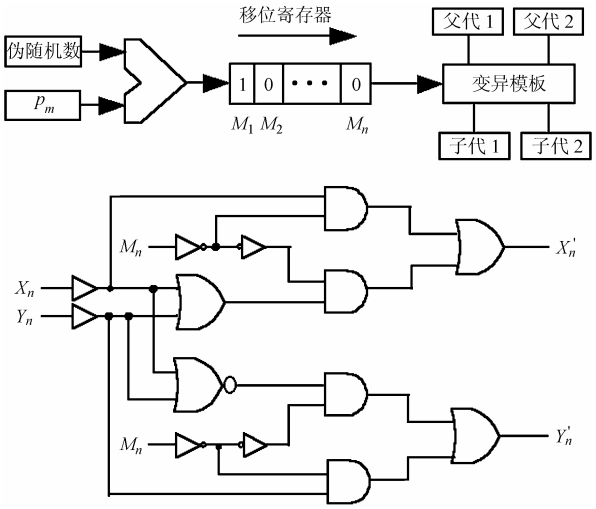

图 3　变异模块结构及硬件电路图

3.3 控制模块

控制模块负责将新个体及其适应度写入存储区，同时它也设有 3 个寄存器，分别用来存储当前最大适应度，最小适应度，以及平均适应度。 p_c 和 p_m 的自适应调整也在控制模块中实现。每进化完一代后，自适应控制器将会根据当前最大适应度，最小适应度，以及平均适应度，确定下一代的 p_c 和 p_m 。自适应控制器如图 4 所示。控制模块还包括 4 个伪随机数发生器，分别供选择模块、交叉模块、变异模块以及种群的初始化应用。另外，它还负责对算法运行的代数进行监控，判断算法终止的条件是否满足，同时还提供一些控制

信号。

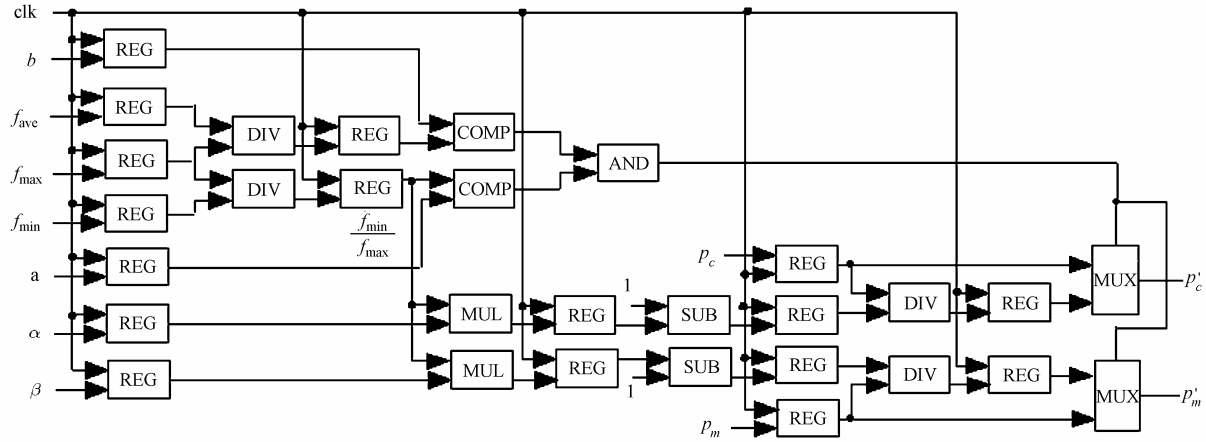

图 4 自适应控制硬件结构图

3.4 存储结构

在本系统中，个体和适应度是分开存储的。存储区分为 4 个部分：父代个体存储区，父代适应度存储区，子代个体存储区，子代适应度存储区。其中，个体和其相应的适应度在存储时用的是同一个地址，只是片选信号不同而已，也就是说，个体和适应度的地址是同步变化的。这样做的好处是根据当前适应度的地址可以直接找到相应的个体。

4 硬件实现和测试

整个设计采用VHDL描述，在Xilinx的XC2V1000(100 万门)的FPGA上进行了实现。为验证系统的正确性同时测试系统的性能，本文用DeJong的 3 个测试函数[9]对系统进行了测试。测试选取的参数如下：

种群规模为 80，初始交叉概率(p_c) 为 0.9 初始变异概率(p_m)为 0.05，基因长度为 32，自适应参数为a＝0.5，b＝0.8，α ＝0.9，β ＝0.85。表 1 为函数f_1的综合报告。

表 1 综合报告

Vendor	XilinX
Family	Virtex2
Device	xc2v1000-fg2256c-5
Slice	2,734 out of 5,120 53%
Maximum frequency	55.972M

在系统时钟为 40MHz，算法在进化到 1000 代时终止的情况下，测试结果如表 2 所示。其中，软件实现的程序用 C 语言编写，运行在主频为 733 的奔腾微机上。

从测试结果可以看出，硬件实现的遗传算法要比软件快 3 个数量级，如果对设计进一步进行优化，则系统时钟还可以提高，从而进一步提高系统运行速度。

表 2 测试结果表

函数	实现方法	平均适应	运行时间
f_1	软件实现	0.042	1.7s
	硬件实现	0.000	5.2ms
f_2	软件实现	0.061	6.1s
	硬件实现	0.002	5.75ms
f_3	软件实现	－27.00	2.6s
	硬件实现	－30.000	5.4ms

5 结束语

本文提出了一种基于 FPGA 的自适应遗传算法，并且进行了硬件实现。设计结合 FPGA 自身的特点，充分利用算法的并行特性，同时采用了流水线结构的设计思想，大大提高了系统的运行速度。测试结果表明，这种硬件实现的遗传算法有效地缩短了运行时间，为实时应用提供了可能。

参 考 文 献

[1] 潘正君，康立山，陈毓屏. 演化计算[M]. 北京：清华大学出版社，1998: 3－4.

[2] Petridis V, Kazarlis S, Bakirtzis A. Varying fitness function in genetic algorithm constrained optimization: The cutting stock and unit commitment problems[J]. *IEEE Trans. on SMC Part B: Cybernetics* , 1998, 28 (5): 629－639.

[3] Shaunna M, Tom L, Abdulla H. A gene- tic algorithm environme-mnt for star pattern recognition[J]. *Journal of Intelligent and Fuzzy Systems*, 1998, 6(1): 3－16.

[4] Bhandarkar S M , Zhang H.Image seg- ment using evolutionary computation[J]. *IEEE Trans. on Evolutionary Computation*,1999,

3 (1): 1 – 21.

[5] 唐加福，汪定伟，高振，等. 面向非线性规划问题的混合式遗传算法[J]. 自动化学报，2000，26(3): 401 – 404.

[6] 骆晨钟，邵惠鹤. 采用混沌变异的进化算法[J]. 控制与决策，2000，15(5): 557 – 560.

[7] Scrinvas M, Patnaik L M. Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms . *IEEE Trans. on SMC*, 1994, 24(4): 656 – 667.

[8] 王蕾，沈庭芝，招扬. 一种改进的自适应遗传算法[J]. 系统工程与电子技术, 2002, 24(5): 75 – 78.

[9] Z.米凯利维茨 .演化程序——遗传算法和数据编码的结合[M]. 北京：科学出版社, 2001: 257 – 258.

房　磊：男，1977 年生，博士生，研究领域为模糊逻辑、遗传算法和数字系统设计.

张焕春：男，1940 年生，教授，博士生导师，研究领域为计算机测控数据融合、现代测控系统集成技术等.

经亚枝：女，1942 年生，副教授，研究领域为计算机测控、信号处理和数据采集等.