

免疫遗传算法的逻辑电路进化设计研究

朱金仁, 史旭华*

(宁波大学 信息科学与工程学院, 浙江 宁波 315211)

摘要: 针对电路进化设计时传统进化算法收敛速度慢且易陷入局部最优解等问题, 模拟人体免疫系统的机制, 设计了一种改进的免疫遗传算法用于逻辑电路的进化设计. 首先建立电路进化设计模型, 对种群中抗体进行多目标适应度评估; 然后改进抗体的选择机制并将精英抗体作为记忆单元保存; 最后引入自适应交叉、变异策略保持抗体多样性并提高算法的收敛性. 实验结果表明: 该算法具有较强的全局搜索能力, 能有效地减少搜索到全局最优解的迭代次数, 并设计出新颖、高效的电路结构.

关键词: 免疫遗传算法; 进化设计; 多目标评估; 选择策略

中图分类号: TP391.9

文献标志码: A

文章编号: 1001-5132 (2015) 02-0036-06

1992 年日本和瑞士科学家提出了演化硬件^[1]的概念, 演化硬件是指硬件电路能够通过与外界环境的交互作用自适应地调整和改变自身结构, 具有自适应、自修复的特性. 当前欧美、日本等一些国家在演化硬件的理论以及应用研究上处于领先地位, 在人工脑、图像压缩、滤波器设计、电路设计等领域有实际应用. 受限于硬件平台, 国内只有少数的科研机构从事演化硬件的研究^[2], 基本停留在演化硬件的理论研究阶段, 与国外存在一定的差距.

作为演化硬件的重要分支, 电路进化设计是一种将进化计算技术应用于电路设计的方法. 该方法以进化算法作为搜索引擎, 以可编程器件作为评估和实现载体, 通过进化得到满足预期功能的电路结构. 与传统电路设计方法相比, 该方法不依赖于先验知识并能设计更加新颖、优化的电路结构, 成为当前研究的热点. 进化算法作为搜索引擎, 其效率直接影响进化收敛速度^[3], 赵曙光等^[4]设计自适应遗传算法(Genetic Algorithm, GA)用于逻辑电路的进化设计, 并通过偶校验器、乘法器等进化设计实验证明了所提方法的有效性, 但在如何避

免局部最优解及减少算法的迭代次数方面有所欠缺; 张义国等将免疫算法(Immune Algorithm, IA)原理融入到逻辑电路设计中^[5-6], 通过非选择机制去掉未成熟抗体, 设定生存阈值来避免高亲和度的抗体在非选择的过程中被删除掉, 避免抗体集陷入局部最优. 实验证明此方法具有比 GA 更好的求解性能, 但对个体的评估只考虑了电路功能的符合度, 进化所得电路利用了更多的逻辑门电路.

针对电路进化设计时传统进化算法收敛速度慢而且容易陷入局部最优的问题, 笔者设计了一种改进的自适应免疫遗传算法(Adaptive Immune Genetic Algorithm, AIGA)用于逻辑电路的进化设计. 对种群中抗体进行多目标评估, 并改进抗体的选择机制; 同时引入自适应交叉、变异策略, 保持抗体多样性以提高算法的收敛性. 利用该法进行 2 位乘法器的设计, 并与 GA、IA 等方法相比较, 结果表明该法能有效地减少迭代次数, 提高算法的收敛速度; 同时设计出更加新颖、高效的电路结构.

1 问题描述

FPGA 由可编程逻辑模块(Configurable Logic

收稿日期: 2014-02-15.

宁波大学学报(理工版)网址: <http://journalg.nbu.edu.cn/>

基金项目: 国家自然科学基金(61273146); 浙江省自然科学基金(LY14F030004); 宁波市自然科学基金(2012A610005).

第一作者: 朱金仁(1986—), 男, 安徽安庆人, 在读硕士研究生, 主要研究方向: 电路进化设计. E-mail: zhujinren309@sina.com

*通信作者: 史旭华(1967—), 女, 浙江宁波人, 博士/副教授, 主要研究方向: 智能计算及应用等. E-mail: shixuhua@nbu.edu.cn

Block, CLB)、输入输出模块及内部连线资源构成, 具有在线可编程及内部结构可动态重配置的特性, 因此被广泛地用作电路进化设计. 我们依据 FPGA 的内部结构, 建立了电路进化设计模型, 如图 1 所示. 图 1 中 CLB 组成阵列单元, 每个单元可表示一种逻辑功能, 并且单元间可互连构成逻辑电路. 因此, 电路进化设计可看成是一个黑匣子问题, 在已知待进化电路的输入及输出的前提下, 对 CLB 阵列按一定方式编码形成代表电路结构特征的染色体, 通过进化算法搜索满足功能要求的高效、新颖的电路结构.

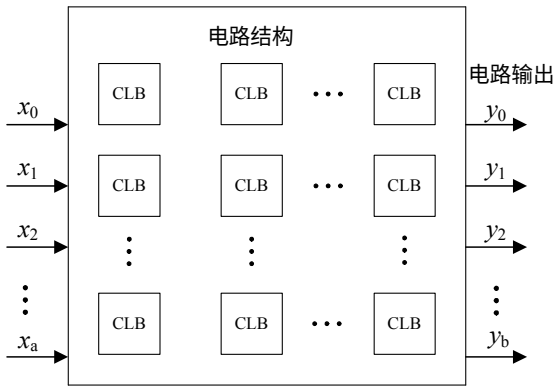


图 1 逻辑电路进化设计模型

2 基于 AIGA 的电路进化设计

2.1 免疫遗传算法原理

免疫遗传算法是在免疫算法中加入遗传算子, 是一种改进的、具有免疫功能的遗传算法. 在算法中, 把待求解的问题看作抗原, 问题的解作为抗体, 对抗原亲和力高的优秀抗体进行记忆, 遇到同类抗原时能快速求解. 与遗传算法相比, 增加了抗原识别、优秀抗体记忆和自我调节功能, 没有降低遗传算法的鲁棒性; 免疫遗传算法兼顾了搜索速度、全局搜索能力和局部搜索能力.

2.2 电路结构编码

电路进化设计的前提是如何将待求解电路拓扑结构编码成进化算法能执行的形式, 现在常用的编码方法有: CGP (Cartesian Genetic Programming)^[7]、基因表达式编程^[8]、有向图^[9]等. 笔者采用 CGP 方法, 模拟 FPGA 的内部结构, 建立一个 $G = R \cdot C$ 的阵列单元, 如图 1 所示. 每个 CLB 是一个阵列节点单元 $G_{i,j}$ ($0 \leq i < R$, $0 \leq j < C$), 具有输入输出且实

现特定的功能. 图 2 表示一个 2 输入 1 输出的节点单元. 针对不同的电路进化设计问题, 节点功能具有不同的选择项. 我们选用了 2 输入 1 输出逻辑门 NOT、XOR、AND、OR 等 4 种, 每个节点单元完成功能对应的编码, 同时节点之间的连接方式也可以编程. 阵列中节点单元的输入来自于待进化电路的原始输入或者是阵列中其他节点单元的输入; 在组合逻辑设计中, 为了防止输出反馈, 规定第 i 列节点单元的输入只能来自于第 j ($j < i$) 列的输出, 阵列中一些节点单元的输入作为整个阵列的输出. 节点单元阵列的大小则依据所要进化设计电路的功能复杂度. 每个节点单元按以下规定进行编码: 节点的输入和功能构成单个节点的编码, 则将 $R \cdot C$ 个节点单元连起来再加上阵列的输出就是整个电路拓扑结构的染色体编码.

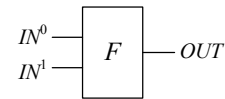


图 2 节点单元

图 2 中 IN^0 、 IN^1 表示输入的来源, F 表示此节点单元的功能, OUT 表示输出, 因此单个节点可以编码为: IN^0, IN^1, F . 按照先行后列的次序将整个阵列单元的编码连接起来, 再加上输出单元就构成了电路拓扑结构的编码. 整个阵列编码可以用 (1) 式表示:

$$X = [IN_0^0, IN_0^1, F_0, \dots, IN_i^0, IN_i^1, F_i, \dots, IN_G^0, IN_G^1, F_G], \quad (1)$$

式中, IN_i^0, IN_i^1 分别表示第 i 个节点的 2 个输入, F_i 表示第 i 个节点的功能.

2.3 适应度评价

在逻辑电路的进化设计过程中, 对表示电路特征染色体的评估直接影响算法进程以及能否得到最优解, 逻辑电路的进化设计主要考虑 3 个指标: 电路的功能符合度、门电路的利用率以及电路工作速度.

(1) 电路的功能符合度定义为进化电路的输出与真值表的符合程度. 对于一个 a 输入 b 输出的电路, 有 2^a 种输入组合, 对每种输入组合若它对应的 b 输出均正确, 其适应度加 1, 则最大适应度为:

$$f = 2^a \cdot b.$$

对每个抗体适应度的计算公式为:

$$f_1(X) = \sum_{i=1}^{2^a} \sum_{j=1}^b (1 - p_{i,j} \otimes q_{i,j}), \quad (2)$$

其中: $p_{i,j}$ 表示输入第 i 个输入组合时实际输出的第 j 位, $q_{i,j}$ 表示与该输入相对应的期望输出的第 j 位.

(2) 门电路的利用率定义为电路中有效单元个数与阵列规模 G 的比值. 电路进化设计时阵列中的一些节点单元是冗余的, 主要有 2 种情况: 一种是某些节点单元没有被利用, 另一种是某些节点链路的起点和终点的信号没有改变, 并且链路中的节点单元没有被其他节点单元利用. 这样的冗余单元越多, 进化电路中所利用的实际节点单元个数则越少, 利用率越高. 设 G_0 表示进化设计时冗余单元的个数, 则节点利用率公式为:

$$f_2(X) = \frac{G - G_0}{G} \times 100\%. \quad (3)$$

(3) 电路工作速度由逻辑电路最长信号链路中电路节点单元的个数决定. 当信号通过一个门电路后会有一定时间的延时, 因此电路的工作速度与链路的长度有关, 链路长度由门电路的个数决定. 设链路长度为 L , 则电路的工作速度与 L 成反比, 计算公式为:

$$f_3(X) = \frac{1}{L}. \quad (4)$$

综上所述, 电路进化设计是典型的多目标优化问题^[10]. 对于多目标优化问题, 通常用加权法将多目标优化问题转化为单目标优化问题, 对于目标函数 $f_1(X)$, $f_2(X)$, $\dots$, $f_3(X)$, 依据各个目标函数的重要性, 分别给出权系数 a_1 、 a_2 、 a_3 , 且满足 $a_1 + a_2 + a_3 = 1$, 则可将多目标优化问题转化为单目标问题, 抗体的适应度函数为:

$$F(X) = \sum_{i=1}^3 a_i f_i(X). \quad (5)$$

2.4 AIGA 的进化机制

2.4.1 抗体选择策略

设计一种改进的赌轮盘选择方法, 在选择适应度高的抗体同时, 抑制浓度过大的抗体, 保持抗体的种群多样性, 避免算法的局部收敛. 定义一个新的概念抗体相似度, 假设表示电路结构的抗体 X_a 与抗体 X_b 的适应度分别为 $F(X_a)$ 、 $F(X_b)$, 当其满足(6)式

$$1 - \alpha \leq \frac{F(X_a)}{F(X_b)} \leq 1 + \alpha, \quad (6)$$

则说明抗体 X_a 与抗体 X_b 相似, 其中 α 为适当小的正常数, 例如 $\alpha = 0.04$, 则说明抗体 X_a 与抗体 X_b 有 96% 以上的相似度. 假设与优秀抗体 X_i 相似的抗体个数有 n , 种群规模为 N , 则其抗体浓度为:

$$\beta = \frac{n}{N}, \quad (7)$$

抗体 X_i 的选择概率为:

$$p(X_i) = \varepsilon \frac{F(X_i)}{\sum_i F(X_i)} + (1 - \varepsilon)\beta, \quad (8)$$

其中: ε 为 0.5~1 之间的常数. 此抗体选择策略将适应度与抗体浓度相关联, 选择高适应度抗体同时抑制浓度过高的抗体进入下一代, 保证抗体的多样性.

2.4.2 交叉、变异操作

电路结构的编码是以节点为基本单位构成, 为保证抗体的有效性, 算法中以一个节点的编码作为整体进行单点交叉, 如图 3 所示, 保证群体多样性同时, 算法能够全局搜索.

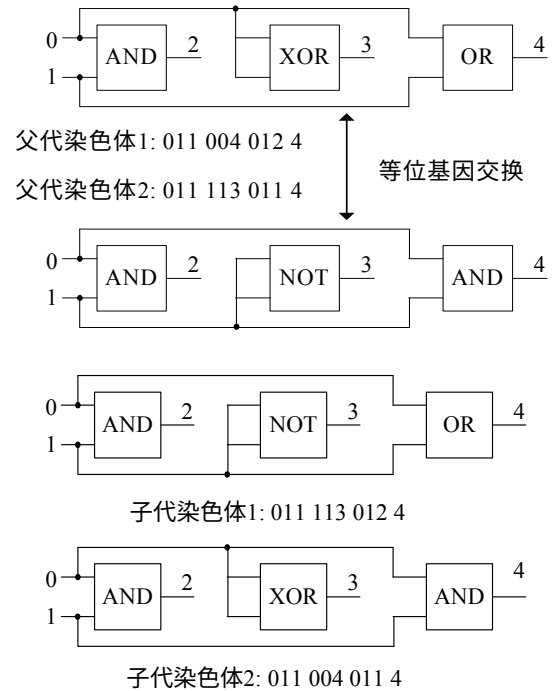


图3 交叉算子

同时, 为了保证种群的多样性, 对染色体的某个基因进行单点变异, 如图 4 所示.

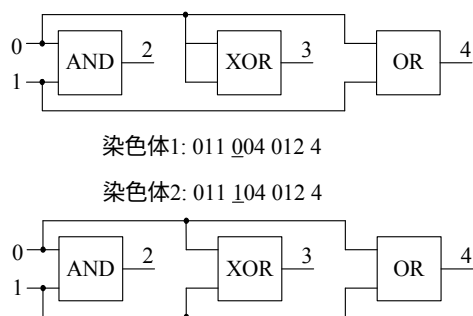


图 4 变异算子

2.5 交叉变异概率自适应

算法中的交叉、变异概率直接影响着算法的收敛速度和效率, 交叉率过大则易破坏高适应度的抗体, 过小则搜索过程相对缓慢; 变异率同样存在这样的问题, 变异率过大易破坏高适应度的抗体结构, 过小则不能够保持种群的多样性. 算法初期, 抗体适应度增长很快, 存在 Stalling effect 现象^[11], 后期适应度增长缓慢直至满足要求, 因此在进化过程的前期需要大的交叉变异概率, 在进化过程的中后期, 由于适应度较高, 不能轻易破坏优秀抗体, 因此需要较小的交叉变异概率. 鉴于此, 设计了与抗体进化代数相关的交叉概率:

$$p_c = p_{c,\max} e^{-t/T}, \quad (9)$$

其中: $p_{c,\max}$ 为初始交叉概率, t 为当前进化代数, T 为最大进化代数.

变异算子在算法中产生新抗体, 具有抑制早熟维持多样性功能, 将种群中每代抗体的变异率与抗体适应度相关联, 抗体越优变异率越小, 相反变异率越大, 其公式为:

$$p_m = \exp\left(-\frac{F_{\max}(X) - F(X)}{F_{\max}(X)}\right) \frac{1}{1 + t/T} p_{m,\max}, \quad (10)$$

其中: $p_{m,\max}$ 为初始变异概率, t 为当前进化代数, T 为最大进化代数.

2.6 算法流程

AIGA 算法用于组合电路进化设计的步骤为:

Step1: 将待进化电路的期望输出及其约束作为抗原, 用 CGP 建立电路结构模型, 随机产生 N 个抗体, 并且初始化进化代数、交叉变异概率等相关参数.

Step2: 计算每个抗体的适应度值, 采用精英保留策略, 将高适应度抗体作为记忆单元存贮.

Step3: 判断种群是否为第一代, 若是则跳转

到 Step5, 若不是继续以下步骤.

Step4: 再次计算每个抗体的适应度值, 用精英抗体替换适应度小的抗体, 若出现比精英抗体更优秀的抗体, 则将其复制替换原有的精英抗体.

Step5: 计算抗体浓度.

Step6: 计算抗体被选择进行交叉与变异操作的概率.

Step7: 计算自适应交叉与变异概率.

Step8: 满足适应度最优解或达到预设定的进化代数, 输出种群中优秀抗体, 若无则回到 Step4.

3 实验及结果分析

我们将电路结构编码、抗体适应度多目标评估融入免疫遗传算法, 设计新的抗体选择方法, 避免逻辑电路进化设计中陷入局部最优解, 同时引入自适应交叉与变异策略, 并采用精英保留策略替换适应度低的抗体, 优化种群中抗体分布, 从而减少迭代次数. 通过 2 位乘法器的进化设计实验, 验证提出的 AIGA 在逻辑电路进化设计中的可行性.

(1) 实验条件. PC Intel(R)、Core(TM) 2.5 GH 处理器, 32 位 Windows 7 操作系统. C 语言编程实现算法. 2 位乘法器逻辑真值表(表 1).

表 1 2 位乘法器真值表

输 入				输 出			
A_1	A_0	B_1	B_0	F_1	F_2	F_1	F_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

表1中 A_1 、 A_0 为被乘数, B_1 、 B_0 为乘数, F_3 、 F_2 、 F_1 、 F_0 为乘积. 实验中各参数设置如下: 初始种群规模 $N=100$, 算法执行最大代数为 20 000 代, 初始交叉概率 $p_{c,\max}=0.75$, 初始变异概率 $p_{m,\max}=0.15$, 所选用的候选门集有 NOT、XOR、AND、OR 共 4 种. 实验选用 6 行 4 列的阵列单元, 多目标评估时权系数 $a_1=a_2=a_3=1/3$.

(2)实验结果及其分析. 利用 GA、IA 以及本文设计的 AIGA 进行 2 位乘法器实验, 选取其中成功的 20 次实验作比较, 如图 5 所示.

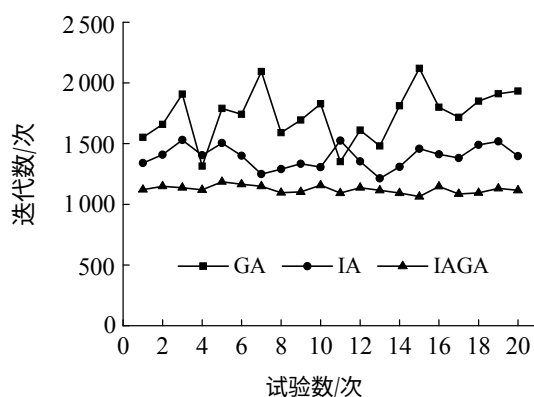


图5 收敛代数比较

同等条件下, 相比于 GA、IA, AIGA 能有效地减少搜索到全局最优解的迭代次数, 且波动性较小, 趋于稳定在 1120 代左右. 利用 AIGA 设计的 2 位乘法器电路如图 6 所示. 使用了 7 个逻辑门电路, 最大链路长度为 4, 而利用 GA、IA 及传统的卡诺图等方法设计 2 位乘法器所需要的门电路个数见表 2.

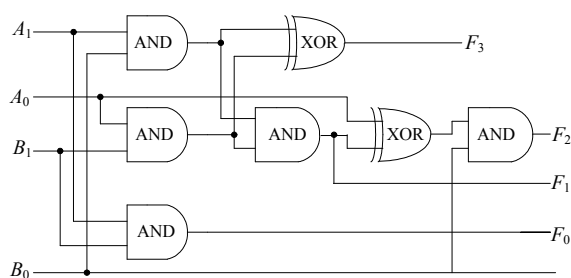


图6 AIGA 设计的 2 位乘法器

其中:

$$F_3 = (A_1 \cdot B_0) \oplus (A_0 \cdot B_1),$$

$$F_2 = B_0 \cdot (A_0 \oplus F_1),$$

$$F_1 = (A_1 \cdot B_0) \cdot (A_0 \cdot B_1),$$

$$F_0 = A_1 \cdot B_1. \quad (11)$$

表2 AIGA 与 GA、IA 及卡诺图法比较

比较对象	GA	IA	卡诺图	AIGA
门个数	8	11	12	7
链路长	3	4	4	4

相比于其他方法, AIGA 所使用的逻辑门个数更少, 更加合理地利用了门电路资源, 设计出新颖、高效的电路结构. 但相比于文献[4]中改进的 GA, 在工作速度方面有所欠缺. 另外, 由于采取了多目标评估以及自适应策略, 在逻辑电路进化设计中, AIGA 算法计算复杂度要高于 GA 及 IA, 因此进化设计过程中需要消耗更多的计算时间, 这也是后续工作需要改进之处.

4 结语

本文设计的 AIGA 可应用于逻辑电路的多目标进化设计, 与 GA、IA 等方法相比能够有效地避免局部收敛, 减少搜索到全局最优解的迭代次数, 并能设计出优化、简洁的逻辑电路模块, 为逻辑电路进化设计的研究提供了新的方法. 而如何提高可进化逻辑电路的规模以及评估速度将是下一步需要继续研究的课题.

参考文献:

- [1] Yao X, Higuchi T. Promises and challenges of evolvable hardware[J]. IEEE Transactions on Applications and Reviews, 1999, 29(1):87-97.
- [2] 纪震, 田涛, 朱泽轩, 等. 进化硬件研究进展[J]. 深圳大学学报: 理工版, 2011, 28(3):255-263.
- [3] 莫宏伟, 徐立芳. 基于 Memetic 算法的电路演化设计研究[J]. 电子学报, 2013, 41(5):1036-1040.
- [4] 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电路门级进化方法[J]. 计算机辅助设计与图形学学报, 2004, 16(4):402-406.
- [5] 张义国, 罗文坚, 王煦法. 基于免疫原理的逻辑电路设计算法[J]. 计算机工程与应用, 2006, 11(1):38-40; 62.
- [6] Xu Haiqin, Ding Yongsheng, Huang Zhihua. Adaptive immune genetic algorithm for logic circuit design[C]// Proceedings of the First ACM/SIGEVO Summit on Genetic and Evolutionary Computation, ACM, 2009: 639-644.
- [7] Harding S, Miller J F, Banzhaf W. Developments in cartesian genetic programming: Self-modifying CGP[J].

- Genetic Programming and Evolvable Machines, 2010, 11(3/4):397-439.
- [8] 吴江, 唐常杰, 李大勇, 等. 基于多目标基因表达式编程的电路演化算法[J]. 吉林大学学报: 信息科学版, 2011, 29(3):245-261.
- [9] 陶砚蕴, 林家骏, 徐萃华. 有向图基因表达式程序的电路演化模型[J]. 计算机辅助设计与图形学学报, 2010, 22(1):94-99.
- [10] 刘楠楠, 史旭华. 基于抗体浓度的克隆选择多目标优化算法及其应用[J]. 宁波大学学报: 理工版, 2013, 26(3):57-61.
- [11] 梁后军. 电路进化设计算法研究[D]. 合肥: 中国科学技术大学, 2009.

Research on Evolutionary Design of Logic Circuit Based on Immune Genetic Algorithm

ZHU Jin-ren, SHI Xu-hua*

(Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China)

Abstract: In order to overcome the problems with the traditional algorithms in designing logic circuits such as slow evolution speed and premature convergence, by simulating the body's immune system mechanisms, an improved immune genetic algorithm is proposed for the evolutionary design of logic circuit. Firstly, it takes multi-object fitness evolution to assess the individuals of population. Then, the individuals with high fitness are stored by elitist strategy as a memory unit for improving the selection mechanism of the individual. Finally, by adopting the adaptive crossover and mutation strategy, the diversity of individuals is maintained and the convergence of the algorithm is improved. The search ability of the proposed method is tested using the two-bit multiplier. Compared with the genetic algorithms and immune algorithms, the experiment results show that proposed design has strong global search capability, effectively reducing the convergence algebra of optimal solution.

Key words: immune genetic algorithm; evolutionary design; multi-object assessment; selection policy

(责任编辑 史小丽)