维普资讯 http://www.cqvip.com

浮点数编码小生境遗传算法的研究*)

崔明义
(河南财经学院计算机科学系　郑州 450002)

摘　要　小生境在增加遗传算法群体的多样性,提高遗传算法的局部搜索能力方面具有良好的性能。迄今为止,有关小生境遗传算法的研究都是基于二进制编码,缺乏以浮点数编码为研究对象的相应成果。而浮点数编码在提高遗传算法的性能和遗传算法的推广应用中,具有其它编码所无法比拟的优势。本文以浮点数编码为研究对象,研究小生境遗传算法的机理,分析在遗传操作中小生境的生成、合并和分离的动态过程,探索其方法。本文的研究和实验结果表明,浮点数编码小生境遗传算法的性能是可靠的,方法是可行的。

关键词　小生境,浮点数编码,遗传算法

Research on Niche Genetic Algorithm of Float Number Code

CUI Ming-Yi
(Dept. of Computer Science, Henan University of Finance & Economics, Zhengzhou 450002)

Abstract　Niche has better performance in increasing the population diversity of genetic algorithm (GA), in improving local researching performance of it. So far, research results of relating to niche GA are all on binary code, there are no almoston float number code. But in improving the performance of GA and extending GA'sapplication, float number code is superior to other codes. In this paper, the mechanism of niche GA is researched by it on float number code. Dynamic process is analyzed by it on niche forming and merging and separating in inherit operation. The method is explored by it. The results of its research and experiment indicated that the performance of niche GA of float number code is reliable. The method is feasible.

Keywords　Niche, Float number code, Genetic algorithm

1　引言

所谓小生境遗传算法,就是模拟自然界的某些物种在一种特定的环境中生存进化的机理,在遗传算法用固定规模群体中的一些子群体,分别承担求解问题的一些子任务。与基本遗传算法不同的是:基本遗传算法的群体只存在竞争,而小生境遗传算法群体中既有竞争,也有子群体间的协作。

Cavicchio[1]最早提出了基于预选择机制(Preselection)的小生境实现方法。1975 年,De Jong[2]提出了基于排挤机制(Crowding)的小生境实现方法。1987 年,Goldberg[3,4]提出了基于共享机制(Sharing)的适应度共享小生境实现方法,并在 1990 年用于玻耳兹曼(Boltzmann)锦标赛选择。20 世纪 90 年代,Horn[5~7]对小生境遗传算法的有限 Markov 链进行了分析,提出了基于资源共享的小生境实现方法,并对搜索、优化和机器学习中的小生境遗传算法进行了分析研究。这些研究,都是基于二进制编码的遗传算法实现的。而浮点数编码在提高遗传算法的性能和遗传算法的推广应用中,具有其它编码所无法比拟的优势。但迄今为止,对浮点数编码的小生境遗传算法研究较少,缺乏浮点数编码遗传算法的小生境性能分析和机理的研究。本文着重分析适应度共享的浮点数小生境遗传算法机制,研究浮点数遗传算法在优化过程中小生境的生成、合并和分离机理,探索其在优化领域的优势,为遗传算法的基础和应用研究拓展更为广阔的空间。

2　浮点数小生境遗传算法的机理

小生境是较直接协作的更高水平,展示了许多复杂的通常只在成熟和直接协作中出现的交互现象(如专门化、共栖共生、互惠共生、Lotka-Voltera 震荡、长期进化稳定性、竞争和协作的嵌套等现象)。随着群体规模和小生境重叠的变化漂移,某些小生境会瞬间消失,而新的小生境又同时产生。从开始没有小生境到小生境重叠的不断增加,产生了从协作到竞争的"相转换"。这是一个小生境产生、维持和收敛的往复过程。

在遗传算法中引入小生境,目的是为了避免群体中的最优个体取代所有竞争个体的复制,防止"早熟",提高其局部搜索能力。小生境导致了搜索空间的恢复压力,平衡了选择的收敛压力[8]。小生境遗传算法中稳定状态的群体分布体现了问题解的分布表达式。这些问题的高质量的解(而非优化解)存在于平衡分布中。小生境遗传算法在防止"早熟"、提高局部搜索性能和维持理想模式(即积木块)等都有明显的优势。适应度共享为显性小生境,常用于多峰函数优化[9]。

2.1　浮点数适应度共享

浮点数适应度共享小生境的构造是通过降低目标适应度值来实现的。具体思想是:计算参数空间中某个体 i 与任意其他个体 j 的欧几里得距离,判断两个个体是否在估计小生境半径 σ_m 确定的圆内,求出其共享值 $sh(d)$,将个体 i 所有 $sh(d)$ 之和 $m_i=\Sigma sh(d)$ 作为小生境数。共享值 $sh(d)$ 的计算如下:

$$sh(d_{i,j})=\begin{cases}1-\left(\dfrac{d_{i,j}}{\sigma_m}\right)^{\alpha_m}, & d_{i,j}<\sigma_m\\ 0, & \text{otherwise}\end{cases} \quad (1)$$

将该共享值加到两个小生境数 m_i 和 m_j 中。个体由 σ_m

*)本课题受到河南省自然科学基金(0411014500)和河南省高校杰出科研人才创新工程项目(2004KYCX014)的资助。崔明义　教授,博士,研究方向为计算智能、软件和网络安全。

维普资讯 http://www.cqvip.com

分成小生境内外两部分。参数 σ_{ni} 和 σ_{ni} 依据适应度景观的先验知识或特别目标确定。在缺少先验知识的情况下，α_{ni} 可设为 1。这时，出现三角共享函数。图 1 表示了共享函数随 α_{ni} 变化的势曲线族。随着一对个体的欧几里得距离从 0 到 σ_{ni} 变化，共享值则从 1 到 0 呈单调减少。所以，参数 σ_{ni} 直接影响着递减函数的形状，即小生境的形态。对于个体 i，小生境数 m_i 可按下式计算：

$$m_i=\sum_{j=1}^{N}sh(d_{i,j}) \tag{2}$$

这里，N 是群体规模。个体 i 的共享适应度依据下式计算：

$$f_{sh,i}=\frac{f_i}{m_i} \tag{3}$$

适应度共享分布在多峰的区域内，且与峰的高度成正比。这就使在参数空间的不同小生境最优值附近形成较稳定的子群体，并由此花费时间复杂度为 $O(n^2)$ 的计算代价，且受限于 α_{ni}。

2.2 浮点数编码不同小生境的形成

与自然小生境类似，参数空间的不同小生境由相应的子群体构成。每个小生境由其质心和半径所确定。小生境的质心和半径在遗传过程中不断变化。这样，小生境内部的个体数量和个体编码也在不断变化，同一个个体可能属于一个或同时属于多个小生境的成员。

假定小生境 j 的质心和半径分别是 C_j 和 α_{ni}^j，在该小生境内的个体 i 的表现型值为 v_i，则

$$C_j-\alpha_{ni}^j\leqslant v_i\leqslant C_j+\alpha_{ni}^j \tag{4}$$

式(4)说明，在多维空间中，若某个体在由 C_j 和 α_{ni}^j 定义的区域内，则该个体是这个小生境的成员。一个小生境存储其相关信息，如该小生境生成的代数、初始质心、组成当前小生境的成员个体等。

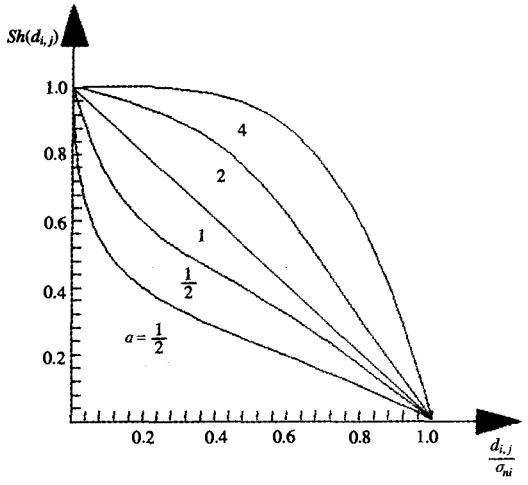


图 1 适应度共享函数曲线

假设 β_{ni} 为小生境的内径，用于决定两个小生境是否合并。由于 σ_{ni} 和 β_{ni} 均是不断变化的，因此有可能产生非均等的超量小生境。为了限制小生境半径的无限增大或减小，把它们限定在一定的范围：$\sigma_{\min}\leqslant\sigma_{ni}\leqslant\sigma_{\max}$ 和 $\beta_{\min}\leqslant\beta_{ni}\leqslant\beta_{\max}$。$\sigma_{\max}$ 和 $\beta_{\max}$ 可以预先给定，$\sigma_{\min}$ 和 $\beta_{\min}$ 取决于 σ_{ni} 的初始值，而 σ_{ni} 取决于群体规模和参数空间的大小。群体规模越大，σ_{ni} 的初始值越小，生成的小生境就越多。

使用固定的小生境半径或者在具有固定表现型空间的遗传算法中使用最小小生境半径，所产生的小生境都是有限的，这特别适合合并小生境的情况。在第一代，对于群体中的每个个体，都将其小生境作为小生境集合的元素，则其质心集中在该个体上。在初始代和后来的几代中，执行以下操作：

(1)依据下式重新计算当前小生境集中每一个小生境的质心：

$$C_j=C_j+\frac{\sum_{i=1}^{n_j}(v_i-C_j)*f_i}{\sum_{i=1}^{n_j}f_i} \tag{5}$$

这里，C_j 是小生境 j 的质心，v_i 是个体 i 的表现型的值，f_i 是个体 i 的适应度，n_j 是小生境 j 的个体数。每个小生境的质心都将移动到小生境内最适合生存的个体的最高密度上。如果一个小生境没有成员，则它将从当前小生境集中除去。

(2)重新计算小生境成员。如果一个个体不是任何小生境的成员，将生成一个以该个体为质心的一个新小生境。

(3)当前小生境集中的每一个小生境与集合中的每一个其它小生境进行两次比较：

①第一次，如果一个小生境的质心在另一个小生境质心的 β_{ni} 内，则两个小生境合并。

②第二次，如果一个小生境的质心在另一个小生境质心的 β_{ni} 之外，且两个小生境的半径有重叠，则两个小生境分离。

合并和分离一直用于遗传过程中，直到小生境集合不再发生变化而趋于稳定。

(4)重新计算小生境成员，然后使每一个小生境的每个个体共享函数。

2.3 共享函数

可取小生境的成员个数作为小生境数。共享函数 $m_{ni,i}$ 如下定义：

$$m_{ni,i}=\begin{cases}n_j, & i\in j\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

这里，n_j 是小生境 j 的成员数。小生境 j 中的每个个体的共享适应度可按下式计算：

$$f_{ni,i}=\frac{f_i}{m_{ni,i}} \tag{7}$$

2.4 小生境合并

如果一个小生境 j 的质心在另一个小生境 i 的 β_{ni} 内，则将两个小生境合并成一个新的小生境。即

$$C_i+\beta_{ni}\geqslant C_j\geqslant C_i-\beta_{ni} \tag{8}$$

新的小生境质心从两个小生境的最初质心移动到两个小生境中每个个体的平均加权距离上。最初质心依据下式计算：

$$C_{ni}=C_j+\frac{C_i-C_j}{2} \tag{9}$$

这里，$C_i>C_j$。两个小生境的权值和新小生境质心的定义如下：

$$\begin{aligned}w_i&=\sum_{a=1}^{n_i}(v_a-C_{ni})*f_a \quad \forall a\in i\\ w_j&=\sum_{a=1}^{n_j}(v_a-C_{ni})*f_b \quad \forall b\in j\\ C_{new}&=C_{ni}+\frac{w_i+w_j}{\sum_{a=1}^{n_i}f_a+\sum_{b=1}^{n_j}f_b}\end{aligned} \tag{10}$$

这里，n_i 和 n_j 分别是小生境 i 和 j 的成员数，f_a 和 f_b 分别是个体 a 和 b 的适应度，v_a 和 v_b 分别是个体 a 和 b 的表现型值。新的小生境半径可通过新小生境质心与两个原小生境 i 和 j 的质心比较求得。考虑 3 种情况：

(1)若 $C_{new}<C_{ni}$，即新质心距小生境 i 较近。这时，$\alpha_{ni_{new}}$

维普资讯 http://www.cqvip.com

是到小生境 i 或 j 最左边界的距离，而 $\beta_{ni_{new}}$ 是由 β_{ni} 与 $\alpha_{ni_{new}}$ 和 α_{ni} 的 1/2 运算得到。

$$\alpha_{ni_{new}}=C_{new}-(C_i-\sigma_{ni}^i)$$

$$\beta_{ni_{new}}=\beta_{ni}^i+\frac{\sigma_{ni_{new}}-\sigma_{ni}^i}{2} \tag{11}$$

否则

$$\alpha_{ni_{new}}=C_{new}-(C_j-\sigma_{ni}^j)$$

$$\beta_{ni_{new}}=\beta_{ni}^j+\frac{\sigma_{ni_{new}}-\sigma_{ni}^j}{2} \tag{12}$$

(2)若 $C_{new}>C_{ni}$，即新质心距小生境 j 较近。这时，$\sigma_{ni_{new}}$ 是到小生境 i 或 j 最右边界的距离，而 $\beta_{ni_{new}}$ 是由 β_{ni} 与 $\sigma_{ni_{new}}$ 和 σ_{ni} 的 1/2 运算得到。如果 $C_i+\sigma_{ni}^i>C_j+\sigma_{ni}^j$，则

$$\sigma_{ni_{new}}=(C_i+\sigma_{ni}^i)-C_{new}$$

$$\beta_{ni_{new}}=\beta_{ni}^i+\frac{\sigma_{ni_{new}}-\sigma_{ni}^i}{2} \tag{13}$$

否则

$$\sigma_{ni_{new}}=(C_j+\sigma_{ni}^j)-C_{new}$$

$$\beta_{ni_{new}}=\beta_{ni}^j+\frac{\sigma_{ni_{new}}-\sigma_{ni}^j}{2} \tag{14}$$

(3)新质心与最初质心相等，即 $C_{new}=C_{ni}$。这时，$\sigma_{ni_{new}}$ 是小生境 i 和 j 最远边界间距的 1/2。$\beta_{ni_{new}}$ 是小生境 i 或 j 的 β_{ni} 最大值与 $\sigma_{ni_{new}}$ 的 1/2 和小生境 i 或 j 的 σ_{ni} 最大值之和。

$$\sigma_{ni_{new}}=\frac{(C_{right}+\sigma_{ni_{new}})-(C_{left}-\sigma_{ni_{new}})}{2} \tag{15}$$

在上述三种情况中，如果 $\sigma_{ni_{new}}>\sigma_{\max}$，则 $\sigma_{ni_{new}}=\sigma_{\max}$。类似地，对于 $\beta_{ni_{new}}>\beta_{new}$，则 $\beta_{ni_{new}}=\beta_{\max}$。

$$\beta_{ni_{new}}=\beta_{ni_{largest}}+\frac{\alpha_{ni_{new}}-\alpha_{ni_{largest}}}{2} \tag{16}$$

2.5 小生境分离

如果小生境 i 和 j 的半径重叠，则应将其分离而非合并。即

$$C_i-\sigma_{ni}^i<C_j-\sigma_{ni}^j<C_i+\sigma_{ni}^i \tag{17}$$

每个小生境的 σ_{ni} 值将一定程度地减少，以避免重叠。对最大平均适应度的小生境半径以比例减少。

$$Crossover=(C_i+\sigma_{ni}^i)-(C_j-\sigma_{ni}^j)$$

$$\sigma_{ni}^i=\sigma_{ni}^i-Crossover*\frac{\bar{f}_j}{\bar{f}_i+\bar{f}_j}$$

$$\sigma_{ni}^j=\sigma_{ni}^j-Crossover*\frac{\bar{f}_i}{\bar{f}_i+\bar{f}_j} \tag{18}$$

这里，$\bar{f}_i$ 和 $\bar{f}_j$ 分别是小生境 i 和 j 中所有个体的平均适应度。两个小生境的 β_{ni} 值减少小生境半径变化的 1/2。

$$\beta_{ni}^i=\beta_{ni}^i-\frac{\sigma_{ni_{new}}^i-\sigma_{ni_{old}}^i}{2} \tag{19}$$

这里，$\sigma_{ni_{old}}^i$ 和 $\sigma_{ni_{new}}^i$ 分别是小生境 i 的老小生境半径和新小生境半径。如果 $\sigma_{ni}<\sigma_{\min}$，则 $\sigma_{ni}=\sigma_{\min}$。类似地，对于 β_{ni}，如果 $\beta_{ni}<\beta_{\min}$，则 $\beta_{ni}=\beta_{\min}$。

3 实例

为检验上述浮点数小生境方法的可靠性，本文取著名的遗传算法测试函数：

$$\max f(x)=e^{-2\times\ln 2\times((x-0.1)/0.8)^2}\times\sin^6(5\pi x),x\in[0,1] \tag{20}$$

对上述方法进行了测试验证。随机取 10 组，每组规模为 100 的[0,1]的随机数。建立小生境，第一代小生境数为 1000，经过调整参数和合并，第二代为 50 个小生境。经过 80 代遗传运算，小生境不断合并或分离，最后形成 5 个小生境。运行结果如图 2、图 3 所示。在图 2，显示了函数有 5 个近似极大值点 $x^*=\{0.098,0.309,0.496,0.703,0.887\}$，其相应的最优解为 $f(x^*)=\{0.9970,0.8566,0.7036,0.4519,0.2305\}$。这个结果接近于理论值，证明这种方法是可靠的。在图 2 和图 3 中，分别显示了第 2 代和 80 代各 50 个和 5 个小生境的个体数及相应半径的变化曲线，不难看出，小生境的成员数总是随着小生境的半径变化，且小生境的半径越大，其成员数也越多，这与生物小生境的生活习性越普通、环境越宽容，则群体个数越多的特点相吻合。这说明，本文阐述的浮点数小生境的生成、合并和分离机理是可行的。

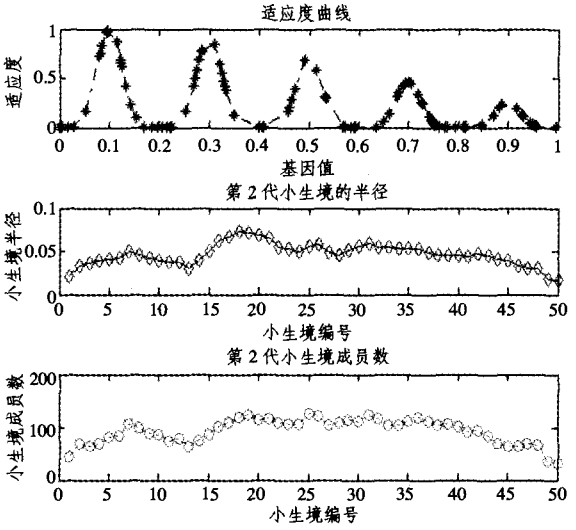


图 2 测试函数仿真结果和第 2 代小生境成员和半径变化曲线

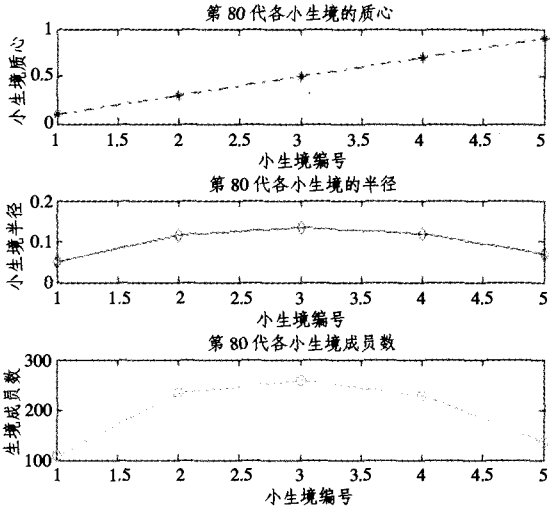


图 3 第 80 代小生境的质心、半径和成员数变化曲线

结束语 迄今为止，在小生境遗传算法的研究中，国内外文献大都是以二进制代码的遗传算法为对象进行研究，对浮点数小生境遗传算法的研究较少，也缺乏相应的研究成果。而浮点数编码在提高遗传算法的性能和遗传算法的推广应用中，具有其它编码所无法比拟的优势。因此，开展对浮点数编码遗传算法的运行机理包括小生境的研究，对于遗传算法理

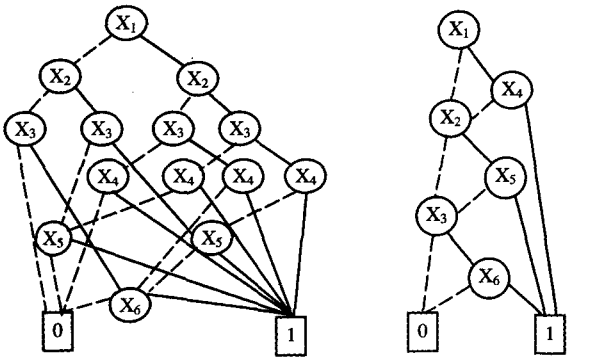
（下转第 288 页）

维普资讯 http://www.cqvip.com

尔函数所共享，这些共享的结点可以节省布尔函数在计算机内部的存储空间。

3 简化 BDD 大小的方法—变量顺序

尽管 BDD 是表示布尔函数的有效方法，但是，表示一个布尔函数的 BDD 的结构和大小依赖于变量的顺序。例如下图给出了布尔函数 $f_1=(x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$ 的在两种变量顺序下的 BDD。



变量顺序为 $x_1<x_2<x_3<x_4<x_5<x_6$　　变量顺序为 $x_1<x_4<x_2<x_5<x_3<x_6$

图 1　变量顺序对 BDD 大小的影响

(a)的变量顺序为 $x_1<x_2<x_3<x_4<x_5<x_6$，其结点总数为 16，(b)的变量顺序为 $x_1<x_4<x_2<x_5<x_3<x_6$，其结点总数却为 8。从此例可以发现，布尔函数在不同的变量顺序下生成的 BDD 的结构和结点数有很大的差别。可以说，变量排序也是一种有效的减小 BDD 规模的方法。

近年来，人们对 BDD 的排序进行了大量的研究工作。目前的方法大致上可分为三种类型。第一种是启发式变量排序，第二种是动态变量排序，第三类是最优化变量排序法。第三类的结果最好。这种方法只适用于变量数较少的系统。动态变量排序是指对已建好的 BDD 的变量次序进行调整以减少 BDD 的规模，该方法往往能找到一个更好的变量次序。目前许多的 BDD 包都应用了动态变量排序方法[1]。

4 SBDD 与变量重排序相结合

分析下面两个布尔函数：

$f_1=(x_1 \wedge x_2) \vee (x_3 \wedge x_4) \vee (x_5 \wedge x_6)$

$f_2=(x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$

当这两个布尔函数的变量顺序为 $x_1<x_2<x_3<x_4<x_5<x_6$ 时，布尔函数 f_1 有 8 个结点而布尔函数 f_2 有 16 个结点。它们以 SBDD 的形式存储有 21 个结点，确实是比原来两个布尔函数的总共的结点数(8+16=24)减少了。

但是，当布尔函数 f_1 的变量顺序为 $x_1<x_2<x_3<x_4<x_5<x_6$ 时，布尔函数 f_1 的结点数为 8，而当布尔函数 f_2 的变量顺序为 $x_1<x_4<x_2<x_5<x_3<x_6$ 时，布尔函数 f_2 的结点数为 8。这两个布尔函数分别存储总共有 16 个结点，比以 SBDD 的形式存储的结点数少。所以，SBDD 是可以减少结点数目，但并不一定是存储 BDD 结点的最小数目。我们提出一种算法，用来求两个布尔函数的最小结点数。这个算法把 SBDD 与动态变量排序结合在一起。

bddnodenum(f,order)为布尔函数 f 在变量顺序 order 下的 BDD 结点总数。

算法：

使用动态变量再排序，求布尔函数 f_1，f_2 分别在各自变量顺序下的最小结点数

bddnodenum(f_1,order1)

bddnodenum(f_2,order2)

使用动态变量再排序，求以 SBDD 的形式存储布尔函数 f_1，f_2 的最小结点数

bddnodenum(SBDD,order3)

比较布尔函数 f_1，f_2 所形成的 SBDD 的结点数和布尔函数 f_1，f_2 的总结点数

if (bddnodenum (SBDD, order) < (bddnodenum (f_1, order1)

+bddnodenum(f_2,order2)))

f_1，f_2 以 SBDD 的形式存储在一个 unique_table 表中

else

分别存储 f_1，f_2

按以上提出的算法思想，能够有效简化 BDD 的大小。

参 考 文 献

1　韩俊刚，杜慧敏. 数字硬件的形式化验证. 北京：北京大学出版社，2001

2　Bryant R. Graph-based Algorithms for Boolean Functions Manipulation. IEEE Transaction on Computers, 1996, C-35(8):677～691

3　Andersen H R. An Introduction to Binary Decision Diagrams.

4　Matsuura M, Sasao T, Butler J T, Iguchi Y. Bi-Partition of Shared Binary Decision Diagrams

(上接第 227 页)

论的发展和推广应用都是很有意义的。本文对此进行了探索，取得了一些经验。随着这方面研究的不断深入和拓展，其成果必将推动遗传算法在工程和优化领域的应用，为遗传算法理论的发展和推广奠定更坚实的基础。

参 考 文 献

1　Cavicchio D J. Adaptive search using simulated evolution:[Ph D thesis]. University of Michigan, Ann Arbor, University Microfilms, 1970. 25～0199

2　De Jong K A. An analysis of the behavior of a class of genetic adaptive systems:[Ph D thesis]. University of Michigan, Ann Arbor. Dissertation Abstracts International, 5140B. University Microfilms, 1975,36(10):76～9381

3　Goldberg D E, Richardson J. Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette J, ed. Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: Lawrence Erlbaum Associates, 1987. 41～49

4　Goldberg D E. A note on Boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. Complex Systems,1990,4: 445～460

5　Horn J. Finite Markov chain analysis of genetic algorithms with niching. In: Forrest S, ed. Proceedings of the Fifth International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann, 1993. 110～117

6　Horn J, Goldberg D E, Deb K. Implicit niching in a learning classi-er system: nature's way. Evolutionary Computation, 1994,2(1):37～66

7　Horn J. Genetic algorithms (with niching) in search, optimization, and machine learning. In: Whitley L D, Vose M D, eds. Foundations of Genetic Algorithms, 1996,4

8　Horn J. Finite Markov chain analysis of genetic algorithms with niching. In:Forrest S, ed. Proceedings of the Fifth International Conference on Genetic Algorithms. San Mateo, CA: Morgan Kaufmann,1993. 110～117

9　Im Chang-Hwan, Kin Hong-Kyu, Jung Hyun-Kyo . A Novel Algorithm for Multimodal Function Optimization Based on Evolution Strategy. IEEE Transactions on Magnetics, 2004, 40(2): 1224～1227