doi:10.3969/j.issn.1673-4785.2009.06.003

遗传算法中的联结关系

周树德[1],孙增圻[2]
(1.中国电子科学研究院,北京 100041;2.清华大学 计算机系,北京 100084)

摘 要:进化计算领域的一个根本问题是哪些问题适合遗传算法求解,为此需要研究问题的结构对算法性能的影响.变量之间的联结关系是问题的本质属性,决定了遗传算法求解问题的难度.如果某个变量对函数值的影响非线性依赖于其他变量,则认为这些变量之间存在联结关系不,对遗传算法的联结关系这一理论问题进行了深入研究,给出了分析一般离散问题联结结构的理论基础,通过分析傅里叶系数与函数子空间的关系,提出了检测黑箱问题联结结构的确定性和随机性算法,通过试验分析说明了算法的正确性和有效性.

关键词:遗传算法;联结关系;适应值函数;傅里叶分析
中图分类号:TP18 **文献标识码**:A **文章编号**:1673-4785(2009)06-0483-07

Linkage in genetic algorithms

ZHOU Shu-de[1], SUN Zeng-qi[2]
(1. China Academy of Electronic and Information Technology, Beijing 100041, China; 2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: One of the most challenging and fundamental problems in the field of evolutionary computation is identification of the classes of problems for which genetic algorithms are especially well (or ill) suited. This is closely related to the question of how the structure of the fitness landscape affects the performance of genetic algorithms. The linkage is referred to as a nonlinear interaction between variables. This is the intrinsic characteristic of the optimization problem, determining the degree of difficulty in solving it. The authors focused on the linkage problem with genetic algorithms and were able to establish a theoretical foundation for the analysis of linkage structures. Based on Fourier analysis of problem structure, it was proven that mask strings with nonzero Fourier coefficients accurately reflect linkage structure. A deterministic and stochastic algorithm for identifying the linkage structure of black-box problems was discussed and experimental results verified its correctness and efficiency.

Keywords: genetic algorithm; linkage; fitness function; Fourier analysis

通常的遗传算法适用于以下问题:1)问题规模不大,变量数一般为几个、十几个或几十个;2)离线和非实时,问题求解允许足够长的时间;3)问题不复杂.一句话,通常的遗传算法很难求解大规模的复杂问题.对于这类问题,随着问题规模的增大,遗传算法需要指数级的计算量和指数级的群体规模[1-2].遗传算法的模式理论是遗传算法寻优的理论基础.但随着人们对遗传算法的深入认识,模式理论的局限性也逐渐显现[2-3].遗传算法通过染色体编码表示一个解向量,问题变量对应于染色体中的基因.根据模式理论,选择操作使适应值高的模式呈指数形式增长,而交叉操作和变异操作则对模式有破坏作用.交叉操作的破坏作用与问题的结构及编码方式有非常直接的关系.为此需要研究问题的结构对算法性能的影响.很多大规模复杂问题难于求解的一个主要问题就是变量之间的联结关系[2].变量之间的联结关系是问题的本质属性,它决定了遗传算法求解问题的难度[4-5].遗传算法中的联结关系是目前遗传算法理论研究的热点和难点问题.

收稿日期:2009-04-15.
基金项目:国家自然科学基金资助项目(60736023,60674053,90716021).
通信作者:周树德. E-mail:shudezhou@gamil.com.

1 遗传算法与联结关系

考虑2个变量 x 和 y,如果 x 的赋值对函数值 f 的影响与 y 的赋值有关系,则称 x 和 y 之间存在联结关系. 例如 $f_1(x,y)=x+y$,x 和 y 之间无联结关系,这时 x 和 y 可分别进行优化;若 $f_2(x,y)=xy$,则 x 和 y 之间存在联结关系,这时必须同时考虑 x 和 y 2个变量进行优化.

联结关系的复杂程度直接影响问题的求解难度. 在遗传算法中,传统的交叉操作并没有考虑联结关系,因而使性能受到影响. 如果在遗传算法编码中,具有联结关系的变量基因位置比较靠近,称为紧致编码,那么遗传算法就很容易求得问题的最优解. 如果具有联结关系的变量基因位置很分散,称为松散编码,那么交叉操作则很容易造成基因流失和早熟收敛. 图1表示了单点交叉与基因位置的关系. 由图1可以看出:当紧致编码时,交叉操作后很容易保持原来的基因组特性;当松散编码时,交叉操作将以很大的概率破坏掉原来的基因组特性. 紧致编码使得低阶的模式块容易通过交叉操作构成高阶的模式块,而松散编码使得交叉操作很容易破坏掉有用的模式块[2,6].

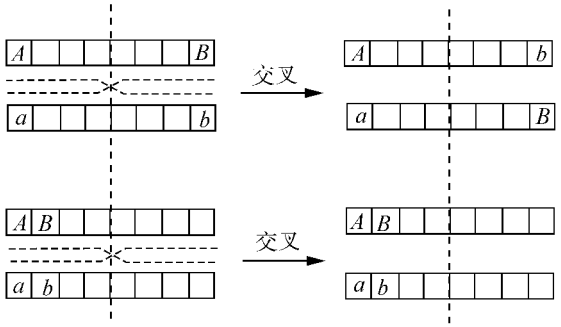


图1 编码方式对交叉操作的影响

Fig. 1 The influence of coding to crossover operator

遗传算法的编码方式对算法的性能有着很重要的影响. 对于一些简单问题,遗传算法可以搜索到较好的解,但对于一些复杂问题,必须利用问题结构的先验知识才能有效地对问题求解[1].

一般情况下,具有联结关系的变量越多,问题的最优解越难找到. 具有联结关系的变量集合称作联结集合或联结块. 对于联结块 S,当且仅当不存在变量 $x_i \notin S$ 使 $S\cup\{x_i\}$ 仍是联结块时,称 S 为最大联结块(极大联结块). 如果问题至多有 k 个变量相互关联,k 称作问题的阶数,并称问题的联结集合是 k 阶限定的. 例如,若 $f(x_3,x_2,x_1,x_0)=x_3x_2+x_2x_1+x_0+x_1$,则 $\{x_3,x_2\}$,$\{x_2,x_1\}$,$\{x_0\}$ 是联结块,且都是最大联结块(极大联结块),该问题的阶数为2,并称问题是二阶限定的. 注意这里 $\{x_3,x_2\}$ 和 $\{x_2,x_1\}$ 存在交叠.

2 二值编码的联结关系检测

设每一个基因位表示一个变量,并设每个变量均为离散取值. 本节首先考虑离散取值仅为0或1的二值取值的简单情况,然后再推广到多值取值的一般情况.

对于给定二进制编码的黑箱问题,关键是如何检测哪些变量之间存在联结关系. 下面介绍2种检测算法.

2.1 近似检测算法

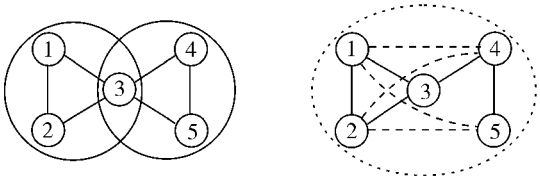
考虑一个待优化的函数 $f(x_0,x_1,\cdots,x_{L-1})$,每个变量的取值为0或1的二值编码. 设二进制字符串 $x=x_0x_1\cdots x_{L-1}$,选择2个变量 x_i 和 x_j,通过扰动2个变量的赋值来观测函数 f 的变化,进而判断 x_i 和 x_j 是否关联. 设

$$\Delta f_i = f(\cdots\bar{x}_i\cdots) - f(\cdots x_i\cdots),$$
$$\Delta f_j = f(\cdots\bar{x}_j\cdots) - f(\cdots x_j\cdots),$$
$$\Delta f_{ij} = f(\cdots\bar{x}_i\cdots\bar{x}_j\cdots) - f(\cdots x_i\cdots x_j\cdots).$$

式中:$\bar{x}_i=1-x_i$,$\bar{x}_j=1-x_j$.

如果 $\Delta f_{ij}=\Delta f_i+\Delta f_j$,即 x_i 和 x_j 对 f 的影响是线性叠加关系,则认为 x_i 和 x_j 不存在联结关系. 如果 $\Delta f_{ij}\neq\Delta f_i+\Delta f_j$,则认为 x_i 和 x_j 存在联结关系. 该算法的计算复杂度为 $O(L^2)$.

该算法的主要特点是比较直观,容易理解,它每次对2个变量进行联结检测,对于联结关系无交叠或交叠简单的问题很有效. 它的缺点是不精确,尤其是对于联结关系有交叠的复杂情况很难处理. 例如图2(a)所示的联结关系,采用上述近似检测算法可能得出如图2(b)所示的错误结果,其中虚线所示为误判的联结关系.



(a)真实的联结关系 (b)误判的联结关系

图2 真实的联结关系和误判的联结关系

Fig. 2 True linkage and false linkage (denoted by dashed line)

2.2 联结关系检测算法

针对上述近似检测算法所存在的不足,Heckendorn等人提出了以沃尔什变换为工具的二进制编码联结关系检测算法[7-8]. 该检测算法具有严格的理

论基础,因而它是一种精确的检测算法.

任意给定函数 $f(\boldsymbol{x})$, $\boldsymbol{x}$ 为二进制变量,通过沃尔什变换,$f(\boldsymbol{x})$ 可以表示为如下的线性组合

$$f(\boldsymbol{x}) = \sum_{i=0}^{2^L-1} \omega_i \psi_i(\boldsymbol{x}).$$

式中:$\psi_i(\boldsymbol{x})$ 是 $f(\boldsymbol{x})$ 的沃尔什基函数,它定义为 $\psi_i(\boldsymbol{x}) = (-1)^{\|i \wedge \boldsymbol{x}\|}$. $\|i \wedge \boldsymbol{x}\|$ 表示 $(i \wedge \boldsymbol{x})$ 中 1 的个数. ω_i 是 $f(\boldsymbol{x})$ 的沃尔什系数,它可通过下式计算:

$$\omega_i = \frac{1}{2^L} \sum_{\boldsymbol{x}} f(\boldsymbol{x}) \psi_i(\boldsymbol{x}).$$

可以证明[7],沃尔什系数 ω_i 可以用来判断变量的联结关系. 准确地说,沃尔什系数 ω_i 的非 0 位说明了相应位的变量具有联结关系. 例如,

$$f(x_1,x_2,x_3,x_4) = w_{0000}\psi_{0000}(x_1,x_2,x_3,x_4) + w_{0001}\psi_{0001}(x_1,x_2,x_3,x_4) + w_{0010}\psi_{0010}(x_1,x_2,x_3,x_4) + w_{0011}\psi_{0011}(x_1,x_2,x_3,x_4) + \cdots + w_{1111}\psi_{1111}(x_1,x_2,x_3,x_4).$$

式中:$w_{0000} \neq 0, w_{0001} \neq 0, w_{0010} \neq 0, w_{0011} \neq 0, , w_{1000} \neq 0, w_{0100} \neq 0, w_{1100} \neq 0$. 说明 x_1 和 x_2 之间具有联结关系,x_3 和 x_4 之间具有联结关系,即 $\{x_1,x_2\}$ 和 $\{x_3,x_4\}$ 是最大联结块(极大联结块). 因而也说明函数 f 一定可以分解为 $f(x_1,x_2,x_3,x_4) = g_1(x_1,x_2) + g_2(x_3,x_4)$.

3　一般离散问题的联结结构检测

上面讨论了变量二值编码时联结关系的检测方法,由于这时变量只能取 0 或 1,因此这种情况有很大的局限性. 因此,下面讨论一般离散编码情况,即变量可以多值编码时联结关系的检测方法.

假定待优化问题的每个变量的定义域是有限集合 $Z_M = \{0,1,\cdots,M-1\}$,L 维问题的定义域表示为 Z_M^L. 对于一般的离散问题 $f(\boldsymbol{x}): Z_M^L \to R$,问题是如何分析和检测 L 维向量 $\boldsymbol{x}$ 的分量 $x_0,x_1,\cdots,x_{L-1}$ 之间的联结关系.

下面介绍后文分析中要用到的“掩码串”的概念. 设 $m \in Z_M^L$ 表示一个掩码串,它的非 0 元素可用来表征所标识的变量. 例如在图 3 中,$m \in \{0,1,2,3\}^6$,$m = 020130$ 表示所对应的变量集合是 $\{x_1,x_2,x_4\}$;掩码串 $m = 100002$ 表示所对应的变量集合是 $\{x_0,x_5\}$.

可以看出,包含 k 个变量的集合可以对应 $(M-1)^k$ 个不同的掩码串. 如果掩码串中非 0 数字的个数为 k,则称其为 k 阶掩码串. 例如 01102 为 3 阶掩码串.

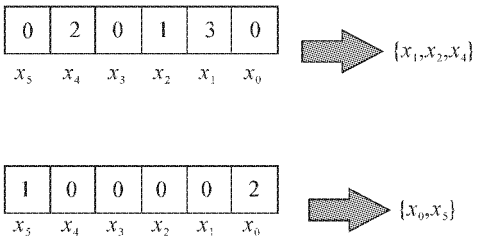


图 3　用掩码串表示变量集合

Fig. 3　Mask strings and the corresponding variable sets

3.1　基于傅里叶变换的联结结构分析

这里基于傅里叶变换对多值编码的联结结构分析,可以看成是前面基于沃尔什变换对二值编码的联结结构分析的推广.

设给定多值编码的函数 $f(\boldsymbol{x}): Z_M^L \to R$,通过傅里叶变换,可将其分解为 M^L 个正交子空间中傅里叶基函数的加权和.

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{j} \in Z_M^L} \omega_{\boldsymbol{j}} \psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x}).$$

式中:$\psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x})$ 是 $f(\boldsymbol{x})$ 的傅里叶基函数,它定义为

$$\psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x}) = \mathrm{e}^{\frac{2\pi \mathrm{i}}{M}(\boldsymbol{x} \cdot \boldsymbol{j})}.$$

式中:$\boldsymbol{x},\boldsymbol{j} \in Z_M^L$. 这里点乘运算定义为 $\boldsymbol{x} \cdot \boldsymbol{y} = \oplus_i (x_i \oplus y_i)$,二元运算 $\oplus$ 定义为 $\boldsymbol{x} \oplus \boldsymbol{y} = (x_0 + y_0 \bmod M, x_1 + y_1 \bmod M, \cdots, x_{L-1} + y_{L-1} \bmod M)$,二元运算 $\otimes$ 定义为 $\boldsymbol{x} \otimes \boldsymbol{y} = (x_0 y_0 \bmod M, x_1 y_1 \bmod M, \cdots, x_{L-1} y_{L-1} \bmod M)$. 例如,$L=3, M=3, \boldsymbol{j}=012, \boldsymbol{x}=121$,则

$$\psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x}) = \mathrm{e}^{\frac{2\pi \mathrm{i}}{M}(0 \oplus 2 \oplus 2) \bmod 3)} = -\frac{1}{2} + \mathrm{i}\frac{\sqrt{3}}{2}.$$

在空间 Z_M^L 中一共有 M^L 个这样的正交基函数. 值得注意的是,每个基函数的值仅仅依赖于在 $\boldsymbol{j}$ 中非 0 的那些变量. 例如,$M=3, L=5, \boldsymbol{j}=12001$,则傅里叶基函数 $\psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x})$ 仅仅依赖于 $\boldsymbol{x}$ 中的 $x_0 \cdot x_3 \cdot x_4$.

傅里叶表达式中的 $\omega_{\boldsymbol{j}}$ 是基函数 $\psi_{\boldsymbol{j}}^{(M)}(\boldsymbol{x})$ 的傅里叶系数,一共有 M^L 个,每一个字串 $\boldsymbol{j}$ 对应一个 $\omega_{\boldsymbol{j}}$,它可由下式来进行计算:

$$\omega_{\boldsymbol{j}} = \frac{1}{M^L} \sum_{\boldsymbol{x} \in Z_M^L} f(\boldsymbol{x}) \bar{\psi}_{\boldsymbol{j}}^{(M)}(\boldsymbol{x}). \quad (1)$$

式中:$\bar{\psi}_{\boldsymbol{j}}^{(M)}(\boldsymbol{x}) = \mathrm{e}^{-\frac{2\pi \mathrm{i}}{M}(\boldsymbol{x} \cdot \boldsymbol{j})}$. 类似二值编码情况下沃尔什系数 ω_i 与变量联结结构有严格的对应关系,这里傅里叶系数 $\omega_{\boldsymbol{j}}$ 也与变量联结结构有严格的对应关系.

可以证明[4],对任意给定函数 $f(\boldsymbol{x}): Z_M^L \to R$,$\omega_{\boldsymbol{j}}$ 表示它的傅里叶系数 $\boldsymbol{j} \in Z_M^L$,则 $\omega_{\boldsymbol{j}} \neq 0$ 当且仅当掩码串 $\boldsymbol{j}$ 所标识的变量之间存在联结关系. 例如,$\omega_{001121} \neq 0$,当且仅当变量 $x_0 \cdot x_1 \cdot x_2 \cdot x_3$ 之间存在联结关系.

下面通过一个具体例子来说明上面的结论. 考

虑函数 $f(x_0,x_1,x_2,x_3)=x_0x_1+x_1x_2+x_3$，其中 $\{x_0,x_1,x_2,x_3\}\in\{0,1,2\}^4$，显见，$f$ 的联结集合为 Ø，$\{x_0\}$、$\{x_1\}$、$\{x_2\}$、$\{x_3\}$、$\{x_0,x_1\}$、$\{x_1,x_2\}$ 计算得到它的傅里叶系数 ω_j 如表 1 所示，表中未列出的掩码串的 ω_j 均等于 0. ω_j 非 0 的掩码串反映了该问题的联结结构. 例如，$\omega_{0011}\neq 0$，则说明 0011 所标识的变量 x_0 和 x_1 之间存在联结关系；$\omega_{0210}\neq 0$，则说明 0210 所对应的变量集 $\{x_1,x_2\}$ 是联结集合.

表 1 傅里叶系数 ω_j 与联结结构的关系

Table 1 The relationship between Fourier Coefficients ω_j and the linkage structure

掩码串	傅里叶系数 ω_j	联结集合
0000	3	Ø
0001	−0.500 0 +0.288 7i	$\{x_0\}$
0002	−0.500 0 −0.288 7i	
0010	−1.000 0 +0.577 4i	$\{x_1\}$
0020	−1.000 0 +0.577 4i	
0011	0.166 7 −0.288 7i	$\{x_0,x_1\}$
0012	0.333 3	
0021	0.333 3	
0022	0.166 7 +0.288 7i	
0100	−0.500 0 +0.288 7i	$\{x_2\}$
0200	−0.500 0 −0.288 7i	
0110	0.166 7 −0.288 7i	$\{x_1,x_2\}$
0120	0.333 3	
0210	0.333 3	
0220	0.166 7 +0.288 7i	
1000	−0.500 0 +0.288 7i	$\{x_3\}$
2000	−0.5000 −0.288 7i	

根据上面的分析，可以通过计算适应度函数的傅里叶系数来确定变量之间的联结关系. 但是值得注意的是，按照式(1)来计算傅里叶系数 ω_j 需要遍历整个定义域空间，显然这个计算工作量很大. 下面将在上述分析的基础上，进一步介绍有效的联结关系检测算法.

需要注意的是，在一般离散域中，一个联结集合可能对应多个不同的掩码串(其傅里叶系数非 0). 例如在表 1 中，联结集合 $\{x_1,x_2\}$ 对应 4 个掩码串 0110、0120、0210、0220. 只要其中有一个掩码串的傅里叶系数非 0，就可以说明 x_1 和 x_2 存在联结关系. 在空间 Z_M^L 中，一个 k 阶联结集合至少对应一个、至多对应 $(M-1)^k$ 个傅里叶系数非 0 的掩码串. 这与二值编码的情况是不同的，二值编码情况的联结块与掩码串是一一对应的.

3.2 联结关系检测的确定性算法

假定问题是 k 阶限定的，也即最多有 k 个变量是相互关联的. 在这种假设下，根据前面的结论，显然当 $\|\boldsymbol{j}\|>k$ 时，傅里叶系数 $\omega_j=0$. 其中 $\|\boldsymbol{j}\|$ 表示字符串 $\boldsymbol{j}$ 中非 0 位置的个数，例如 $\boldsymbol{j}=01020021$，$\|\boldsymbol{j}\|=4$.

对于给定函数 $f(\boldsymbol{x}):Z_M^L\to R$，若 $k\ll L$，由于 $\|\boldsymbol{j}\|>k$ 时，傅里叶系数 $\omega_j=0$，则可大大减少计算 ω_j 的工作量.

根据前面计算 ω_j 的一般公式及这里问题 k 阶限定的假设，可以推得[4]：

$$\omega_i=\begin{cases}\dfrac{1}{\|S(\boldsymbol{i})\|}\displaystyle\sum_{\boldsymbol{x}\in S(\boldsymbol{i})}f(\boldsymbol{x})\bar{\psi}_i^{(M)}(\boldsymbol{x}) & ,\|\boldsymbol{i}\|=k;\\ \dfrac{\displaystyle\sum_{\boldsymbol{x}\in S(\boldsymbol{i})}f(\boldsymbol{x})\bar{\psi}_i^{(M)}(\boldsymbol{x})-\sum_{j:J\supseteq i\&\|j\|>\|i\|}\omega_j\sum_{\boldsymbol{x}\in S(\boldsymbol{i})}\psi_j^{(M)}(\boldsymbol{x})\bar{\psi}_i^{(M)}(\boldsymbol{x})}{\displaystyle\sum_{\boldsymbol{x}\in S(\boldsymbol{i})}\psi_i^{(M)}(\boldsymbol{x})\bar{\psi}_i^{(M)}(\boldsymbol{x})} & ,\|\boldsymbol{i}\|<k;\\ 0 & ,\|\boldsymbol{i}\|>k.\end{cases}\tag{2}$$

式中：$S(\boldsymbol{i})$ 是 $\boldsymbol{i}$ 的模板集，它是将 $\boldsymbol{i}$ 的非 0 位置变为通配符“*”后的字符串集合. 例如，若 $\boldsymbol{i}=01020\in Z_3^5$，则 $\boldsymbol{i}$ 的模板集为 $S(\boldsymbol{i})=0*0*0$，其中 * 遍历 0、1、2，即 $S(i)=0*0*0=\{00000,00010,00020,01000,01010,01020,02000,02010,02020\}$. 首先根据式(2)中的第 1 个式子计算 $\|\boldsymbol{i}\|=k$ 时的 ω_i，它需要遍历 $\boldsymbol{i}$ 的模板集 $S(\boldsymbol{i})$，而在式(1)的一般计算公式中，计算 ω_i 需要遍历整个定义域空间 Z_M^L. 当 $k\ll L$ 时，将大大减小计算 ω_i 的工作量. 然后根据式(2)中的第 2 个式子计算 $\|\boldsymbol{i}\|=k-1$ 时的 ω_i，这时式中需要的 $f(\boldsymbol{x})$ 和 ω_j 都是前面已经计算过的，无需重新计算. 然后依次计算 $\|\boldsymbol{i}\|=k-1,k-2,\cdots,0$ 时 ω_i. 这样就将所有傅里叶系数 ω_i 计算出来了.

在式(2)的计算中，如果注意到 $\omega_i=\bar{\omega}_{\bar{i}}$ 的事实，还可以进一步提高计算效率. 其中 $\bar{\boldsymbol{i}}$ 是 $\boldsymbol{i}$ 的逆元，它满足 $\bar{\boldsymbol{i}}\oplus\boldsymbol{i}=0$.

可以证明[4],该算法所需要的适应值函数计算次数的上限为 $\sum_{i=0}^{k}(M-1)^i\binom{L}{i}$. 下面通过一个具体例子来说明该算法的计算工作量. 设问题为

$$f(\boldsymbol{x}) = g(x_0,x_1,x_2) + g(x_2,x_3,x_4) + g(x_4,x_5,x_6) + g(x_1,x_7,x_8).$$

式中:$g(x_i,x_j,x_k)=\begin{cases}100, & x_i=\alpha_i,x_j=a_j,x_k=a_k;\\ x_ix_jx_k, & \text{其他.}\end{cases}$

显然该问题的联结集合为 $\{x_0,x_1,x_2\}$、$\{x_2,x_3,x_4\}$,$\{x_4,x_5,x_6\}$、$\{x_6,x_7,x_8\}$、

对于该例,若采用前面的一般计算公式,计算傅里叶系数需要计算 $3^9=19\,683$次函数评价,而利用确定性算法仅需计算1 512次函数评价. 显然利用该算法可大大减少计算工作量.

3.3　联结关系检测的随机性算法

确定性算法具有严格的理论基础,并给出了最坏情况下的复杂度估计,而且可以准确地给出联结关系的检测结果,这些是该算法的优点. 它的不足是需要先验知识 k,当 k 较大时计算量仍然较大.

为了提高算法的效率,下面将给出一种随机算法. 该算法的基本思想是,基于傅里叶分析定义“探针”算子,该探针的值反映了联结关系,进而给出一种自低价到高价的检测算法. 该随机算法比确定算法效率更高,且不需要先验知识 k.

定义离散域探针为

$$P(f,\boldsymbol{m},\boldsymbol{c}) = \frac{1}{M^{\|\boldsymbol{m}\|}}\sum_{\boldsymbol{i}\in S(\boldsymbol{m})}\bar{\psi}_k^{(M)}(\boldsymbol{i})f(\boldsymbol{i}\oplus\boldsymbol{c}).$$

式中:$\boldsymbol{m}$ 是掩模串,$S(\boldsymbol{m})$ 是 $\boldsymbol{m}$ 的模板集,$\boldsymbol{c}\in B(\boldsymbol{m})$. $B(\boldsymbol{m})$ 是 $\boldsymbol{m}$ 的背景集,它是将 $\boldsymbol{m}$ 的非零位置变为 0 及将 $\boldsymbol{m}$ 的 0 位置变为通配符“ * ”后的字符串集合. 如 $\boldsymbol{m}=01201$,则 $\boldsymbol{m}$ 的模板集 $S(\boldsymbol{m})=0**0*$,$\boldsymbol{m}$ 的背景集 $B(\boldsymbol{m})=*00*0$.

可以证明[4],探针 $P(f,\boldsymbol{m},\boldsymbol{c})$ 具有如下性质:

性质 1　任意给定函数 $f(\boldsymbol{x}):Z_M^L\to R$,如果 $\omega_{\boldsymbol{m}}$ 是它的一个极大非零傅里叶系数,那么对任意的背景串 $\boldsymbol{c}\in B(\boldsymbol{a})$,均有 $P(f,\boldsymbol{m},\boldsymbol{c})=\omega_{\boldsymbol{m}}$.

性质 2　任意给定函数 $f(\boldsymbol{x}):Z_M^L\to R$,如果 $\omega_{\boldsymbol{m}}$ 是一个极大非零傅里叶系数,那么对于任何 $\boldsymbol{m}$ 的子串 $\boldsymbol{a}$($\boldsymbol{m}\supseteq\boldsymbol{a}$)都存在背景串 $\boldsymbol{c}\in B(\boldsymbol{a})$ 使得 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$.

下面通过一个例子来说明性质 2. 如图 4 所示,对于任意函数 $f:\{0,1,2\}^5\to R$,如果 ω_{01201} 是极大非零傅里叶系数,那么掩码串 01201 的任何子串,都必然存在背景串 $\boldsymbol{c}$,使得它的探针值非零. 例如一定存在背景串 $\boldsymbol{c}_1$ 使得二阶子串 01200 的探针值 $P(f,01200,\boldsymbol{c}_1)\neq 0$,一定存在背景串 $\boldsymbol{c}_2$ 使得一阶子串 00001 的探针值 $P(f,00001,\boldsymbol{c}_2)\neq 0$.

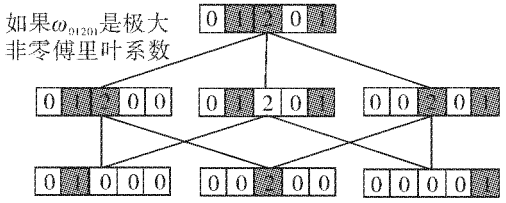


图 4　探针性质 2 的直观解释

Fig. 4　Explanation of the property 2 of probe operator

性质 3　任意给定函数 $f(\boldsymbol{x}):Z_M^L\to R$,如果掩模串 $\boldsymbol{m}\in Z_M^L$ 所标识的变量之间存在联结关系,那么当且仅当存在 $\boldsymbol{c}\in B(\boldsymbol{m})$ 使得 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$.

根据性质 3,需要遍历所有 $\boldsymbol{c}\in B(\boldsymbol{m})$ 来计算 $P(f,\boldsymbol{m},\boldsymbol{c})$,才能判断是否存在 $\boldsymbol{c}\in B(\boldsymbol{m})$ 使得 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$. 这个计算工作量是很大的. 实际计算时,随机选择背景串 $\boldsymbol{c}\in B(\boldsymbol{m})$,计算它的探针值 $P(f,\boldsymbol{m},\boldsymbol{c})$,如果探针值 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$,则表明存在 $\boldsymbol{c}\in B(\boldsymbol{m})$ 使得 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$;否则继续随机选择背景串 $\boldsymbol{c}$ 来计算探针值,至多进行 N_p 次,如果 N_p 次的探针值都为 0,则认为不存在 $\boldsymbol{c}$ 使得该掩码串的探针值非零. 这里 N_p 是需要设定的参数,它表示对掩码串 $\boldsymbol{m}$ 进行随机探测的次数. 显然 N_p 越大,成功检测的概率便越大. 性质 4 可以帮助估计这个概率.

性质 4　设 $f(\boldsymbol{x}):Z_M^L\to R$ 是 k 阶限定问题,$\boldsymbol{m}\in Z_M^L$ 且 $\|\boldsymbol{m}\|=j,j\leqslant k$,如果 $\boldsymbol{c}\in B(\boldsymbol{m})$ 是随机选择的背景串,那么 $P(f,\boldsymbol{m},\boldsymbol{c})\neq 0$ 的概率至少是 M^{j-k}.

考虑函数 $f(x_0,x_1,x_2)=g_1(x_0,x_1)+g_2(x_2)$,其中 $(x_0,x_1,x_2)\in\{0,1,2\}^3$,有 2 个极大联结集合 $\{x_0,x_1\}$ 和 $\{x_2\}$,该函数傅里叶系数非零的有:ω_{000}、ω_{100}、ω_{200}、ω_{010}、ω_{020}、ω_{001}、ω_{002}、ω_{011}、ω_{022},其他的傅里叶系数都等于 0. 如图 5 所示,按照自低阶到高阶的步骤,首先检测空掩码串 000,得到非零的探针值. 然后检测 1 阶掩码串,以 001 为例进行说明:因为 001 所有子串(此情况下只有一个 0 阶子串 000)均有不为 0 的探针值,所以需要计算 001 的探针值,计算后得到非零的探针值. 当所有 1 阶掩码串都被检

测后，开始考虑 2 阶掩码串. 以 011 为例，它有 2 个 1 阶子掩码串 010 和 001，它们均有不为 0 的探针值，因此需要计算 011 的探针值，计算后得到非零的探针值，图中用下划实线标示. 图中下划虚线的掩码串，表示它的探针值为 0. 当所有 2 阶掩码串都被检测后，进而考虑 3 阶掩码串. 由于对任意的 3 阶掩码串，都存在探针值为 0 的 2 阶子掩码串，根据性质 2 可以推断，所有 3 阶掩码串的探针值一定为 0. 根据最后计算得到的探针值不为 0 的掩码串(000,001,002,010,020,100,200,011,022)，运用上面的探针性质 3 就可以判断出变量之间的联结关系. 显然这里可以判断出该问题存在 2 个极大联结集合 $\{x_0, x_1\}$ 和 $\{x_2\}$.

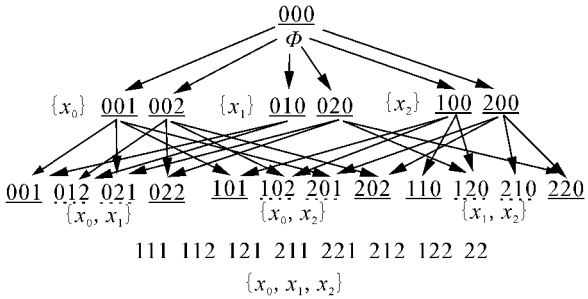


图 5 自低阶到高阶的联结关系检测的随机算法

Fig. 5 The bottom-up random linkage detection algorithm

根据以上探针性质可以给出自低阶到高阶的联结关系检测的随机算法. 可以证明[4]，该随机算法所需要的函数评价次数的理论估计为 $O(L^2 \ln L)$. 下面通过一个例子来说明该算法的执行过程.

下面再给出一个标准测试问题来验证该算法的有效性. 设给定五阶陷阱问题：

$$f_{5\text{trap}}(\boldsymbol{x}) = \sum_{i=0}^{\frac{L}{5}-1} f(x_{5i}, x_{5i+1}, x_{5i+2}, x_{5i+3}, x_{5i+4}).$$

式中：$f(x_{5i}, x_{5i+1}, \cdots, x_{5i+4}) = \begin{cases} 9 - \sum_{k=0}^{4} x_{5i+k}, \sum_{k=0}^{4} x_{5i+k} < 9; \\ 10, \sum_{k=0}^{4} x_{5i+k} = 10. \end{cases}$

变量的定义域为 $\{0,1,2\}^L$，问题的维数设定为 $L = 100$. 该函数中每个子函数 f 有 2 个峰值 9 和 10，函数值除了在 22222 处取得最优值外，在其他空间使得函数值趋向 9 这个局部极小值. 每个子函数的 5 个变量之间都存在联结关系.

在计算中分别设定随机探测的次数 $N_p = 10$、25、35，每种情况运行算法 20 次，考察算法检测 2 阶联结集合的成功率. 实验结果见表 2.

表 2 不同探针操作次数 N_p 下检测 $f_{5\text{trap}}(\boldsymbol{x})^2$ 阶联结集合

Table 2 Using probe operator of differenct N_p to detect the order-2 linkage of $f_{5\text{trap}}(\boldsymbol{x})$

探针次数 N_p	成功率/%	函数评价次数
10	77.5	1 748 025
25	98.0	4 321 269
35	100.0	6 032 286

表 2 表明，对于 100 维的五阶陷阱问题，设置 $N_p = 35$，可以以 100% 的成功率检测到二阶联结关系. 因此，在该问题的随机算法中，对于 2、3、4 阶联结关系，设置探针次数为 35；对于五阶联结关系，根据探针性质 1，设置探针次数为 1 即可. 问题维数分别设为 50、100、150 和 200. 试验结果表明，该随机算法能正确地检测出五阶陷阱问题的所有极大联结集合. 算法所需要的函数评价次数如图 6 所示. 可以看出，所需函数评价次数随着维数的增大几乎线性增长. 试验结果所体现的计算复杂度比理论结果 $O(L^2 \ln L)$ 还要好.

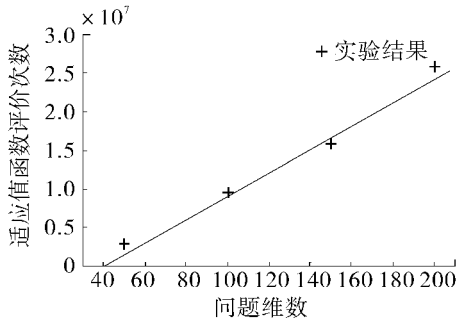


图 6 随机算法检测五阶陷阱问题所需函数评价次数随维数的变化关系

Fig. 6 Scalability of the random algorithm on order-5 trap problems

与前面介绍的确定性算法相比，这里介绍的随机算法不需要预先假定问题是 k 阶限定的，而且需要的计算工作量也比确定性算法少. 其缺点是需要设置探针次数 N_p，且一般不能保证 100% 的成功率.

4 结束语

对遗传算法中重要的理论问题——联结关系进行了系统研究，基于对适应值函数的傅里叶分析，理论证明了联结关系的数学性质，并给出了 2 类检测黑箱问题联结关系的算法. 一旦利用任何一种算法检测出问题的联结结构，在进行遗传算法染色体编

码时,可有目的地将有联结关系的变量编排在一起,而获得紧致编码,这样便于有效地解决传统遗传算法难以解决大规模复杂问题的难题.

参考文献:

[1]THIERENS D. Scalability problems of simple genetic algorithms[J]. Evolutionary Computation, 1999, 7(4):331-352.

[2]GOLDBERG D E. The design of innovation: lessons from and for competent genetic algorithms[M]. Boston: Kluwer Academic Publishers, 2002:88-95.

[3]MENON A. Frontiers of evolutionary computation[M]. London: Kluwer Publisher, 2004:23-27.

[4]周树德. 遗传算法中联结关系检测的理论和方法研究[D]. 北京:清华大学,2007.
ZHOU Shude. Research on theory and methods for linkage detection in genetic algorithms[D]. Beijing:Tsinghua University,2007.

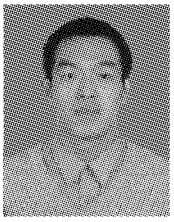
[5]DAVIDOR Y. Epistasis variance: a viewpoint on GA-hardnes[C]//Proceedings of the 6th International Conference on Genetic Algorithms. San Mateo: Morgan Kaufmann, 1995:133-138.

[6]CHEN Y P. Extending the scalability of linkage learning genetic algorithms: theory and practice[D]. Champaign: UIUC, 2004.

[7]HECKENDORN R B. Embedded landscapes[J]. Evolutionary Computation, 2002, 10(4):345-369.

[8]HECKENDORN R B, WRIGHT A. Efficient linkage discovery by limited probing[J]. Evolutionary Computation, 2004, 12(4):517-545.

作者简介:



周树德,男,1980年生,博士,主要研究方向为智能优化理论和算法、复杂系统分析与设计. 发表学术论文10余篇.

孙增圻,男,1943年生,教授,博士生导师,主要研究方向为智能控制、机器人、模糊系统和神经网络、计算机控制理论及应用等. 发表学术论文300余篇.

2010年Web信息系统与挖掘、人工智能与计算智能国际会议

The 2010 International Conference on Web Information Systems and Mining, the 2010 International Conference on Artificial Intelligence and Computational Intelligence

The 2010 International Conference on Web Information Systems and Mining (WISM'10) and the 2010 International Conference on Artificial Intelligence and Computational Intelligence (AICI'10). WISM'10-AICI'10 aims to provide a high-level international forum for scientists and researchers to present the state of the art of web information systems, web mining, artificial intelligence, computational intelligence, with their applications for addressing world problems of various kinds. WISM'10-AICI'10 is multi-disciplinary in which a wide range of theory and methodologies are being investigated and developed to tackle complex and challenging problems.

All accepted papers will appear in conference proceedings published by the Springer's LNCS/LNAI and the IEEE-CS, respectively. (All accepted papers at WISM'10-AICI'10 are indexed by EI Compendex and ISTP). Selected good papers will appear in SCI/EI indexed international journals, such as the Journal of Web Engineering, Journal of Information and Computation Science.

Website:http://wism-aici2010.njupt.edu.cn.