第 27 卷第 8 期
2007 年 8 月

计算机应用
Computer Applications

Vol. 27 No. 8
Aug. 2007

文章编号:1001-9081(2007)08-1948-04

构建在联接探索和分解分布上分布估计算法的扩展算法

姜 群,王 越,欧 阳
(重庆工学院 计算机科学与工程学院,重庆 400050)
(jq@cqit.edu.cn)

摘 要:遗传算法(GA)在解决变量间存在较大相互作用优化问题时缺乏有效性,一种解决问题的途径是分布估计算法(EDA)。分解分布算法是一种近似高阶相互作用的 EDA,它用分解 Boltzmann 分布来产生新的解。运用联接探测及分解分布给出一个以高概率找到最优解的新算法。该算法能解决一些分布估计算法难于处理的问题。实验证明了算法的可行性和有效性。

关键词:k-强性;适应度;麦克斯韦-玻尔兹曼;分解
中图分类号: TP18 **文献标志码**:A

Algorithm extended to an estimation of distribution algorithm based on linkage detection and factorization

JIANG Qun, WANG Yue, OU Yang
(*School of Computer Science and Engineering, Chongqing Institute of Technology, Chongqing* 400050, *China*)

Abstract: Genetic Algorithm (GA) has been found to be lack of effectiveness in solving optimization problems where there is a large amount of interaction between variables, one approach to solve this problem is Estimation of Distribution Algorithms (EDA). Factorized distribution algorithm is an EDA that uses approximation of higher-order interaction, and it uses a factorization of the Boltzmann distribution for the generation of new solutions. A new algorithm which finds the optimum with high probability based on the linkage detection and factorization was given. The algorithm can solve the problems which EDA may have difficulties to deal with. Experimental results prove that the new algorithm is feasible and effective.

Key words: k-epistatic; fitness; Boltzmann; factorization

0 引言

在比特串上的加性分解适应度函数(Additively Decomposable Function, ADF)可写为一些仅依赖于少量比特的简单子函数之和。子函数的支持集是它所依赖的比特集。一个加性分解适应度函数叫 k-强性的,如果它的子函数的支持集的最大规模为 k。文献[1,2]已提出仅由适应度函数是 k-强性的已知条件,通过抽样适应度函数来确定 ADF 分解的算法。本文将探索一种把该算法扩展到分布估计算法(Estimation of Distribution Algorithm, EDA)的途径。

Boltzmann 概率分布可描述为对任何适应度函数 $f(x)$[3]:

$$p(x) = \frac{e^{uf(x)}}{\sum_{y} e^{uf(x)}} \tag{1}$$

参数 u 为乘幂因子,随着 u 趋于无穷,最优点集的概率趋于1。因此,在 u 足够大时,从 Boltzmann 分布中抽样可高概率找到最优点,但 u 不够大时则不能以高概率找到最优点。因为计算式(1)的分母需要整个搜寻空间的最优点求和。文献[3]证明了分解定理,给出了在某些允许一个 ADF 的 Boltzmann 分布被分解的 ADF 支持集上的条件(称为交叉性)。只要 Boltzmann 分布能分解,便可有效地抽样。

本文算法由三步组成:1)用联接探测算法[1,2]确定适应度函数的 ADF 结构;2)分解适应度函数的 Boltzmann 分布;3)从Boltzmann 分布中抽样。如果 Boltzmann 分布的乘幂因子足够大,则能在每次抽样中高概率找到最优解。

算法可进一步描述如下:假设搜寻空间是长度为 l 的二元串集,若已知适应度函数是 k-强性的,可运用联接探测算法[2]。给定一个成功的概率 $\delta < 1$,算法找到适应度函数的 ADF 结构的概率至少为 δ(假设 k 和 δ 是固定的),需用 $O(l^2 \log l)$ 次函数评价。若该 ADF 结构满足交叉性,可用分解定理获得 Boltzmann 分布的分解,计算分解的时间复杂性为 $O(2^k l)$。然后需选择一个乘幂因子,要获得一个已知的最优解概率所必需的乘幂因子取决于最优解与次优解的适应度分离程度。实际上,可增加乘幂因子直到通过抽样获得一个(最优解的)高概率。抽样的复杂性为 $O(l)$。

本算法的优点在于 ADF 的子函数也许重叠,也许有不同的数值范围(度量标准),而基于从种群中取样的分布估计算法也许难于处理 ADF 适应度函数有不同数值范围的子函数。同时算法可获取多个最优点,从 Boltzmann 分布中取样,所有最优点有相同被选取的概率。此外,最优点数量能从任何一个最优点的概率来估计。因而,算法适用于多目标的优化问题。

在算法的实现中,如果 Boltzmann 分布的乘幂因子很大,可能有浮点上溢和下溢问题。我们实现的算法解决了这个问题。

2 联接探测分解算法

如上所述,联接探测分解算法是联接探测算法和分解定

收稿日期:2007-02-27。
基金项目:重庆市自然科学基金资助项目(CSTC2006BB2397);重庆市教委科学技术研究基金资助项目(KJ060611)。
作者简介:姜群(1959-),女,重庆人,副教授,硕士,主要研究方向:进化计算; 王越(1961-),男,北京人,教授,博士,主要研究方向:人工智能; 欧阳(1959-),男,重庆人,教授,硕士,主要研究方向:人工智能。

理的结合,可描述如下:

1)用联接探测算法确定适应度函数的 ADF 结构;

2)用分解定理分解适应度函数的 Boltzmann 分布;

3)取样 Boltzmann 分布 r 次。

2.1　联接探测算法

联接探测算法是一个随机算法,它确定一个 k-强性适应度函数的子函数在固定长度二元串上的分解。已知一个期望的概率为 $0 < \delta < 1$,该算法确定函数的一个加性分解的概率不小于 δ。算法基于摄动原理对串进行摄动探测比特间的相互作用。为了探测二比特间的作用,从一个基础串(背景串)开始,分别翻转每个比特和同时翻转二比特,如果分别翻转比特的作用之和与同时翻转二比特的作用不同,那么探测到二比特间的作用。这种测试也称为探针。一个j阶探针探测j比特集中比特间的相互作用需要 2^j 次函数评价。如果 $j < k$,需若干探针(用不同的基础串)测试已知j比特间的相互作用。

联接探测算法不仅探测哪个比特子集作用于适应度函数,它还构造适应度函数的一个 ADF 分解。实现方法:如果比特间的相互作用与适应度函数有关,则该比特集称为强性的。如果一个比特集是强性的,而且没有超集是强性的,则在该比特集上的任何探测结果恰好是适应度函数的沃尔什(Walsh)系数,其二进制码位数标为该比特集的掩码。只要确定了沃尔什系数,通过用与确定最强性集的沃尔什系数相同的基础串来探测这些子集以决定子集掩码相应的沃尔什系数。当一个 ADF 子函数的沃尔什系数决定后,再用沃尔什逆变换获取子函数的标准描述,其结果是适应度函数的一个 ADF 分解的完全描述。

用摄动方法确定适应度函数结构的优点之一是摄动对 ADF 子函数的相对数值范围不敏感。由于摄动是为测试一个比特集的相互作用而特别设计的,所以不会受到来自于不涉及这些比特的子函数的噪声的影响。另一个优点是对所需的函数评价次数有一个严格的复杂性分析[2]。如果子函数个数随串长线性增长,且需要一个找到所有子函数的全局成功概率,则函数评价次数为 $O(l^2\log l)$,这里 l 为串长,适应度函数评价次数随强性次数 k 指数级增长。

联接探测算法的最差情况是当子函数为“干草堆里的针”函数或者为线性函(除了一个特殊值外)。但子函数不属于这些情况,仅需要很少的探针。例如,若子函数是随机的,理论上探测一个比特集的相互作用仅需一次探针。

2.2　计算分解

本算法的分解部分假设已从联接探测步骤知道适应度函数的 ADF 结构。从文献[3]分解定理的证明可知怎样把适应度函数的 Boltzmann 分布写成条件概率的乘积,以及怎样有效计算所需的条件概率表。

设 X 表示长度为 l 的比特串集合,用二元运算符 $\wedge$ 表示按位“与”运算,$\oplus$ 表示按位“加”运算。设 $L = \{0,1,2,\cdots,l-1\}$ 表示比特位置集合,用L的子集s的掩码表示子集s:掩码是二元串 m,使得 $m_i = 1$ 对所有 $i \in s$ 和 $m_i = 0$ 对所有 $i \notin s$。有时用同样的符号表示一个集合和它的掩码。让 $Xm = \{x \in X: x \wedge m = x\}$,换言之,$Xm$ 是长度为l的也许只在被 m 掩码的位置上包含 1 的比特串集。

如果 $s \subset L$,Π_s 是一个投影映射,它把一个比特串映射到 s 对应比特位置的子串。例如,$l = 5$,$s = \{1,3\}$,$\Pi_s x = \Pi_s(x_0x_1x_2x_3x_4) = x_1x_3$。设 $x_s = x \wedge s$(这里 s 为一个掩码),那么 $\Pi_s x$ 是一个长度为 $|s|$ 的串,而 x_s 是一个长度为 l 的串,而且 $\Pi_s x = \Pi_s x_s$。

若 $f = \sum_{i=1}^{n} f_i(\Pi_{s_i}x)$ 是一个 ADF,设 $e_i(x) = e^{uf_i(\Pi_{s_i}x)}$,如果 $p(x)$ 是由式(1) 定义的 Boltzmann 概率分布:

$$p(x) = \frac{\prod_{i=1}^{n} e_i(x)}{\sum_{y}\prod_{i=1}^{n} e_i(y)} \tag{3}$$

也就是一个 ADF 的 Boltzmann 分布是乘法可分解。

设 $p(\Pi_{s_i}x)$ 表示边缘概率,即:

$$p(\Pi_{s_i}x) = \sum_{y \in X_{L\setminus s_i}} p(x_{s_i} \oplus y) \tag{4}$$

这里 $x_{s_i} \in X_{s_i}$,使得 $\Pi_{s_i}x = \Pi_{s_i}x_{s_i}$。在下列已知条件下,一个 ADF 的 Boltzmann 分布 $p(x)$ 能分解,可由 ADF 的子函数有效地计算 $p(x)$[3]。已知一个适应度函数f能写成一个子函数为一些 f_i 的 ADF,f_i 的支持掩码是 s_i,那么计算新的集合 d_i、b_i 和 c_i 如下:

$$d_i := \bigcup_{j=1}^{i} s_j \tag{5}$$

$$b_i := s_i \setminus d_{i=1} \tag{6}$$

$$c_i := s_i \cap d_{i-1} \tag{7}$$

假设 $d_0 := \varnothing$,如果 $j < i$,$c_i \subseteq s_j$,i 称为 s_i 的后继。

定理 1　分解定理[3]。设 $f = \sum_{i=1}^{n} f_i(\Pi_{s_i}x)$ 是一个 ADF,$p(x)$ 是它的 Boltzmann 分布,如果:

$$b_i \neq 0,\ \forall i = 1,\cdots,n;\ d_n = L \tag{8}$$

$$\forall i \geqslant 2,\ \exists j < i \text{ 使得 } c_i \subseteq s_j \tag{9}$$

那么:

$$p(x) = \prod_{i=1}^{n} p(\Pi_{b_i}x \mid \Pi_{c_i}x) \tag{10}$$

这里$p(\Pi_{b_i}x \mid \Pi_{c_i}x)$是$\Pi_{b_i}x$在已知$\Pi_{c_i}x$条件下的概率。条件(8、9) 定义了交叉性。直观地,它指支持集有一个由后继函数决定的树形结构。

从文献[3] 可知,如果f可表示为一个有交叉性的 ADF,$S(i)$ 是 s_i 的后继集,那么对 $\forall x \in X$:

$$p(\Pi_{b_i}x \mid \Pi_{c_i}x) = \frac{e_i(x_{b_i} \oplus x_{c_i})\prod_{j \in S(i)}\sum_{z \in X_{b_j}} e_j(z \oplus x_{b_i})}{\sum_{w \in X_{b_i}} e_i(w \oplus x_{c_i})\prod_{j \in S(i)}\sum_{z \in X_{b_j}} e_j(z \oplus w)} \tag{11}$$

例:假设一个适应度函数f,它能写为如下 ADF 的形式:

$$f(x_0x_1x_2x_3) = f_1(x_0x_1) + f_2(x_1x_2) + f_3(x_2x_3)$$

表 1　适应度值

x_ix_j	$f_1(x_0x_1)$	$f_2(x_1x_2)$	$f_3(x_2x_3)$	e^{uf_1}	e^{uf_2}	e^{uf_3}
00	2.36	0.73	1.49	10.6	2.08	4.44
01	0.69	0.14	0.94	1.99	1.15	1.15
10	0.95	0.27	0.14	2.59	1.31	2.56
11	1.64	0.41	1.08	5.16	1.51	2.94

表 1 表示每个子函数f_i(已知 $\Pi_{s_i}x$) 的适应度和 e^{uf_i} 的值(在 $u = 1$ 时)。

例如,用式(3) 可计算选择个体 $x = 0110$ 的概率:

$$P(0110)=\frac{e^{uf_1(01)}\cdot e^{uf_2(11)}\cdot e^{uf_3(10)}}{F_u}=\frac{1.99\times 1.51\times 2.56}{347.84}=0.02 \tag{12}$$

用分解概率可计算个体的概率,第一步是构造如下所需集合:

假设 $d_0=\varnothing$,用式(5 ~ 7) 计算其他 d_i、b_i 和 c_i 如下:

$d_1=\{0,1\}$　　$b_1=\{0,1\}$　　$c_1=\{\ \}$

$d_2=\{0,1,2\}$　　$b_2=\{2\}$　　$c_2=\{1\}$

$d_3=\{0,1,2,3\}$　　$b_3=\{3\}$　　$c_3=\{2\}$

由于集合 b_1、b_2 和 b_3 非空,且 $c_2\subseteq s_1$,$s_3\subseteq s_2$,满足交叉性,所以分解式定理成立。显然 s_1 的后继是2,s_2 的后继是3,而 s_3 没有后继,可用式(11) 计算条件概率表。例如:

$$p(x_3=0\,|\,x_2=1)=\frac{e_3(0000\oplus 0010)}{e_3(0000\oplus 0010)+e_3(0001\oplus 0010)}=0.47$$

$$p(x_2=1\,|\,x_1=1)=\frac{e_2(0010\oplus 0100)\cdot\sum_{z\in X_{b_3}}e_3(z\oplus 0010)}{\sum_{w\in X_{b_2}}e_2(w\oplus 0010)+\sum_{z\in X_{b_3}}e_3(z\oplus w)}=0.53$$

概率分布 $p(x_3\,|\,x_2)$、$p(x_2\,|\,x_1)$ 和 $p(x_0x_1)$ 分别列在表2 ~ 4 中。

表2　$p(x_3\,|\,x_2)$ 的分布

x_2	$p(x_3=0\,|\,x_2)$	$p(x_3=1\,|\,x_2)$
0	0.79	0.21
1	0.47	0.53

表3　$p(x_2\,|\,x_1)$ 的分布

x_1	$p(x_2=0\,|\,x_1)$	$p(x_2=1\,|\,x_1)$
0	0.65	0.35
1	0.47	0.53

表4　集合1 的概率分布

x_0x_1	$p(x_0x_1)$
00	0.55
01	0.09
10	0.13
11	0.23

从式(10) 选取个体 $x=x_0x_1x_2x_3$ 的概率为:

$$p(x)=p(x_0x_1)p(x_2\,|\,x_1)p(x_3\,|\,x_2) \tag{13}$$

如 $p(0110)=p(x_0=0,x_1=1)p(x_2=1\,|\,x_1=1)p(x_3=0\,|\,x_2=1)=0.09\times 0.53\times 0.47=0.02$,这个结果与直接计算式(12) 是一致的。

2.3　选择乘幂因子

Boltzmann 分布依赖于乘幂因子参数。当乘幂因子趋向无穷,最优点集的概率趋于1。因此,本算法由决定乘幂因子和从 Boltzmann 分布中抽样组成。为此需要考虑:1)乘幂因子应取多大;2)在计算高乘幂因子时怎样避免浮点上溢和下溢问题。

设置乘幂因子的目的是使 Boltzmann 分布中的最优点集的概率至少是次最优点集概率的 C 倍。

假设一个最优点的适应度是次优点适应度的 $1+\varepsilon$ 倍,设次优点的数量是最优点的 K 倍,如果要使最优点集的概率至少是次优点集概率的 C 倍,则需要选乘幂因子满足下列不等式:

$$e^{u(1+\varepsilon)}\geqslant CKe^{u}\Leftrightarrow e^{u\varepsilon}\geqslant CK\Leftrightarrow u\geqslant\frac{\ln C+\ln K}{\varepsilon}$$

显然 K 以 2^l-1 为上界。

引理2　如果最优点的适应度是次优点适应度的 $1+\varepsilon$ 倍,如选取二次乘幂因子 u,使得 $u\geqslant\frac{\ln C+\ln 2}{\varepsilon}$,则式(1) 中最优点集的概率至少是次优点集概率的 C 倍。

唯一使 $K=2^l-1$ 的方法是有一个“干草堆里的针”适应度,它不是 $k<l$ 的 ADF。假设 K 以 2^kl 为上界,在这个假设下,足以能选取 u 使得:

$$u\geqslant\frac{\ln C+k\ln 2+\ln l}{\varepsilon} \tag{14}$$

实际上,也许有一些 ε 的预先知识,如限制适应度为一个整数,并已知适应度不超过100,则可选取 $\varepsilon=1/100$。

选用适当的 C,比如2,然后从 Boltzmann 分布中反复抽样来增加找到最优点的概率。显然,如果用标准的浮点运算实现式(10、11),则式(14) 可能会导致产生上溢或下溢的乘幂因子,因此,用对数形式来存储这些量。在计算式(11) 时,量 $e_i(x_{s_i})=e^{uf_i(\Pi_{s_i}x)}$ 以对数形式存储为 $uf_i(\Pi_{s_i}x)$。计算式(11) 包括乘法和求和运算。计算乘法时,我们加上存储值。计算求和时,转换存储值为指数形式,相加后转换其结果为对数形式。

2.4　从 Boltzmann 分布中抽样

算法最后一步是用以上所述分解的计算和存储方法抽样 Boltzmann 分布。抽样运用公式(10)顺次从 $i=1$ 到 $i=n$。由于 $c_1=\varnothing$,第一步依据概率分布 $p(\Pi_{b_1}x)$ 选取 $\Pi_{b_1}x$。注意,确定 $\Pi_{c_2}x$ 需要 $\Pi_{b_1}x$,所以下一步根据概率分布 $p(\Pi_{b_2}x\,|\,\Pi_{c_2}x)$ 选取 $\Pi_{b_2}x$。在第 i 步,已确定了 $\Pi_{c_i}x$,根据概率分布 $p(\Pi_{b_i}x\,|\,\Pi_{c_i}x)$ 选取 $\Pi_{b_i}x$。当 x 选定,可用累积式(10) 的乘积因数计算 $p(x)$。显然,一次抽样的时间复杂性为 $\Theta(l)$。

由于算法从准确的 Boltzmann 分布抽样,抽取任何最优点的概率一样。这样,如果 α 是 Boltzmann 分布中最优点集的概率,x 是任何最优点,则最优点的个数为 $\alpha/p(x)$。如乘幂因子足够大,以致 α 近似于1,则 $1/p(x)$ 是最优点个数的一个较好的近似。

3　交叉性不满足时的解决途径

交叉性是适应度函数的 ADF 分解子函数支持集上一个很强的条件。若找不到具有交叉性的有序支持集,有以下解决途径:1)扩展支持集直到满足交叉性,例子可参考文献[4]合并启发式算法;2)忽略一些 ADF 的相关性,为了满足交叉性,或把一些点从 ADF 支持集中去掉,或从 ADF 中去掉一些子函数。这样起着在适应度函数中忽略变量相关性的效果。例如,若想知道两变量间的作用强度,首先看适应度函数的二进制码位数标为构成这两个变量的集合的掩码的沃尔什系数。如想知道 x_1 和 x_5($l=5$) 之间的作用大小,先看二进制码位数标为二元串 01001 的沃尔什系数。我们也可查看二进制码位数标为构成这两个变量的集合的掩码的沃尔什系数的大小获取涉及这两个变量的高阶相互作用信息。

4　实验结果

第一类测试函数是一个 ADF,它的子函数为随机放置针

的“干草堆里的针”函数。也就是除了一个随机选取的串外,对其他所有串,子函数的值是 1。而对这个串,随机选取值 $1+\varepsilon$(正针)或 $1-\varepsilon$(负针)。我们的实验取 $\varepsilon=0.1$,强性参数 $k=4$,有 $l-k+1$ 个子函数,第 i 个函数 f_i 的支持集是 $\{i-1,i\cdots,i+k-2\}$,所以,在 f_i 和 f_{i+1} 间有一个 $k-1=3$ 的重叠。结果这个测试函数可有很多最优点。一般地,不是所有正针子函数同时有针集。因此,最优点值在 $l-k+1$ 和 $(l-k+1)(1+\varepsilon)$ 之间的某处,并且几乎所有情况都是次优点比最优点少 ε。

如果乘幂因子足够大,则从任何最优点的概率可确定最优点的近似个数,所有最优点的概率是相等的。当从分解中取样时,很容易累积由抽样产生的点的概率。

作为一个实例,运行 $l=20$,乘幂因子为 100,有 1424 个最优点,由分解决定的最优点概率为 7.01528×10^{-4}。注意,$1/1424=7.02247191011\times10^{-4}$,若乘幂因子为 10000,最优点的概率为 $7.02247191009\times10^{-4}$。图 1 显示对测试函数运行不同长度串的函数评价次数。根据文献[6]所给出的确定 j 阶探测次数 N 的公式:

$$N\geqslant\begin{cases}\ln(1-\delta^{1/J})/\ln(1-2^{j-k}), & \text{如果 } j<k\\ 1, & \text{如果 } j=k\end{cases}\qquad(15)$$

这里 J 是 j 阶强性集的个数,这些集以 $l\binom{k}{j}$ 为界。取成功概率 δ 为 0.9999,如 $l=64$,$k=4$ 和 $j=2$,则 $N=52.7$。

图 1 是常数次数 $l^2\log l$ 在文献[2]里预测的函数评价次数的复杂性。图 1 经验性地验证了算法的联接探测步骤函数评价次数的时间复杂性为 $O(l^2\log l)$。

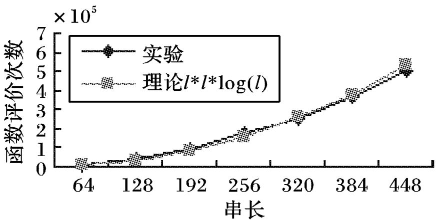

图 1　重叠随机针的函数评价次数

4　结语

本文先介绍了对一类在固定长度二元串上的适应度函数来说,从 Boltzmann 分布抽样是可行的。这类适应度函数需满足以下性质:1)它必须是 k-强性的,即相互作用的变量(或比特)最多为 K 个。满足该条件的适应度函数有 NK-fitness,K 阶联接陷阱以及骗函数等。一个 k-强性适应度函数是加性分解成一些依赖于最多 k bit 的子函数;2)加性分解子函数的支持集必须满足交叉性,即这些支持集有一个树形交叉性。

如果适应度函数是 k-强性的,则本文算法的第一步(联接探测算法取自文献[1,2])从适应度函数中抽样确定 k-强性结构。

这个随机的算法对任何事先给定的成功概率都获得成功。另外,算法计算函数的加性分解,如该加性分解满足交叉性,则进一步计算对任何给定的乘幂因子适应度函数的 Boltzmann 分布的分解。本算法的最后一步用这个分解从 Boltzmann 中抽样。如果选取足够大乘幂因子,算法将以高概率找到最优点。而且,算法还对最优点数量作准确的估计,所有最优点被等概率地抽样。

当 k 为常数时,本算法给出一个严格的复杂性分析,联接探测需要 $O(l^2\log l)$ 次函数评价,构建分解时间复杂性为 $O(l)$,从 Boltzmann 分布中抽样时间复杂性为 $O(l)$。

本算法的实现克服了在乘幂因子很大时可能出现的浮点溢出问题。给出了在一个函数的最优点和次优点适应度分离时选取乘幂因子的公式以及最优点与次优点的数量比。实验证明本算法是可行的。

参考文献:

[1] HECKENDORN R E, WRIGHT A H. Efficient linkage discovery by limited probing[C]// ERICK CANTU P A Z. Genetic and Evolutionary Computation (GECCO 2003), LNCS 2724. [S. l]: Springer-Verlag, 2003: 1003 - 1014.

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

[3] MUHLENBEIN H, MAHNIG T, RODRIGUEZ A O. Schemata, distributions and graphical models in evolutionary optimization[J]. Journal of Heuristics, 1999, 5(2): 215 - 247.

[4] MUHLENBEIN H, HONS R. The estimation of distributions and the maximum relative entropy principle[J]. Evolutionary Computation, 2005, 13(1): 1 - 28.

[5] MUHLENBEIN H, MAHNIG T. Convergence theory and application of the factorized distribution algorithm[J]. Journal of Computing and Information Technology, 1999, 7(1): 19 - 32.

[6] MUHLENBEIN H, MAHNIG T. FDA-a scalable evolutionary algorithm for the optimization of additively decomposed functions[J]. Evolutionary Computation, 1999, 7(4): 353 - 376.

(上接第 1947 页)

[3] KURAMOCHI M, KARYPIS G. Finding frequent patterns in a large sparse graph[J]. Data Mining And Knowledge Discovery, 2005, 11(3): 243 - 271.

[4] YAN X, HAN J W. gSpan: Graph-based substructure pattern mining[C]// Proceedings of IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan. Washington, DC: IEEE Press, 2002: 721 - 724.

[5] DEHASPE L, TOIVONEN H. Discovery of frequent datalog patterns[J]. Data Mining and Knowledge Discovery, 1999, 3(1): 7 - 36.

[6] RAEDT L D, KRAMER S. The levelwise version space algorithm and its application to molecular fragment finding[C]// Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01). San Francisco, CA: Morgan Kaufmann Pulishers, 2001, 2: 853 - 862.

[7] COOK D J, HOLDER L B. Graph-based data mining[J]. IEEE Intelligent Systems, 2000, 15(2): 32 - 41.

[8] GRUNWALD P. A tutorial introduction to the minimum description length principle[EB/OL]. [2006 - 12 - 26]. http://homepages.cwi.nl/pdg/ftp/mdlintro.pdf.

[9] BANDYOPADHYAY S, MAULIK U, COOK D J, *et al.* Enhancing structure discovery for data mining in graphical databases using evolutionary programming[C]// Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference. [S. l.]: AAAI Press, 2002: 232 - 236.

[10] HAN J W, KAMBER M. 数据挖掘概念与技术[M]. 2 版. 北京:机械工业出版社, 2006.

[11] 李敏强,寇纪淞,林丹,等. 遗传算法的基本理论与应用[M]. 北京: 科学出版社,2002.