

基于混合樽海鞘-差分进化算法的贝叶斯网络结构学习算法

刘彬^{1,2}, 范瑞星^{1,2}, 刘浩然^{1,2}, 张力悦^{1,2}, 王海羽^{1,2}, 张春兰^{1,2}

(1. 燕山大学信息科学与工程学院, 河北 秦皇岛 066004; 2. 河北省特种光纤与光纤传感重点实验室, 河北 秦皇岛 066004)

摘要: 针对目前利用启发式算法学习贝叶斯网络结构易陷入局部最优、寻优效率低的问题, 提出一种基于混合樽海鞘-差分进化算法的贝叶斯网络结构学习算法。该算法在种群划分阶段提出自适应的规模因子平衡局部搜索与全局搜索, 在子种群更新阶段利用改进的变异算子与交叉算子构建樽海鞘搜索策略与差分搜索策略, 更新不同的子种群, 在合并子种群阶段利用两点变异算子增加种群多样性。由算法的收敛性分析可知, 通过种群的迭代搜索可以找到最佳结构。实验结果表明, 与其他算法相比, 所提算法收敛精度与寻优效率均有提升。

关键词: 贝叶斯网络结构学习; 樽海鞘算法; 差分进化算法; 自适应

中图分类号: TP18

文献标识码: A

doi: 10.11959/j.issn.1000-436x.2019124

Bayesian network structure learning algorithm based on hybrid binary salp swarm-differential evolution algorithm

LIU Bin^{1,2}, FAN Ruixing^{1,2}, LIU Haoran^{1,2}, ZHANG Liyue^{1,2}, WANG Haiyu^{1,2}, ZHANG Chunlan^{1,2}

1. School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China

2. Hebei Province Key Laboratory of Special Optical Fiber and Optical Fiber Sensing, Qinhuangdao 066004, China

Abstract: Aiming at the disadvantages of Bayesian network structure learned by heuristic algorithms, which were trapping in local minimums and having low search efficiency, a method of learning Bayesian network structure based on hybrid binary salp swarm-differential evolution algorithm was proposed. An adaptive scale factor was used to balance local and global search in the swarm grouping stage. The improved mutation operator and crossover operator were taken into salp search strategy and differential search strategy respectively to renew different subswarms in the update stage. Two-point mutation operator was adopted to improve the swarm's diversity in the stage of merging of subswarms. The convergence analysis of the proposed algorithm demonstrates that best structure can be found through the iterative search of population. Experimental results show that the convergence accuracy and efficiency of the proposed algorithm are improved compared with other algorithms.

Key words: Bayesian network structure learning, salp swarm algorithm, differential evolution algorithm, adaptive factor

1 引言

贝叶斯网络(BN, Bayesian network)是结合图论和概率论来表示因果知识的概率图模型, 是用于不确定领域中推理和预测的最佳方式之一^[1]。贝叶斯网络可以用图论的语言直观地揭示问题的结构,

并利用该结构降低概率推理的计算复杂度。由于贝叶斯网络直观易懂, 在风险分析、机器学习、信息学等研究领域^[2-3]都有应用。

贝叶斯网络的构建包含结构学习、参数学习和推理学习。结构学习是基础与核心, 完备数据下的结构学习方法主要有3种: 基于依赖性测试的方法^[4]、

收稿日期: 2018-08-28; 修回日期: 2019-05-05

通信作者: 刘浩然, liuhaoran@ysu.edu.cn

基金项目: 河北省自然科学基金资助项目(No.F2019203320); 国家自然科学基金资助项目(No.51641609)

Foundation Items: The Natural Science Foundation of Hebei Province (No.F2019203320), The National Natural Science Foundation of China (No.51641609)

基于评分搜索的方法^[5]和混合方法^[6]，其中常见的结构学习方法是基于评分搜索的方法，即在所有节点的结构空间内按照一定的搜索策略及评分准则构建贝叶斯网络结构。

基于评分搜索的方法学习贝叶斯网络结构是一种 NP 问题^[7]，国内外学者通常利用启发式算法来解决此类问题。Tsamardinos 等^[8]提出了一种基于依赖性测试和爬山算法的最大最小爬山 (MMHC, max-min hill-climbing) 算法，该算法虽然改善了检索策略，降低了搜索空间复杂度，但由于搜索空间的缩小易导致算法陷入局部最优。刘浩然等^[9]提出了基于最大支撑树 (MWST, most weight supported tree) 和蚁群算法 (ACO, ant colony optimization) 的混合搜索节点序算法 (MAK, MWST-ACO-K2)，该算法在处理小型网络时可取得较理想的结果，但是与其他基于节点序搜索算法类似，需要对种群中所有个体运行 K2 算法得到对应的适应度值，在大网络中存在时间复杂度较高、结果较差等问题。Wang 等^[1]提出了基于离散水循环算法的贝叶斯结构学习算法 (BEWCA-BN, binary encoding water cycle algorithm for BN structures learning)，该算法根据逻辑算子提出了改进的水循环算法更新个体，其中蒸发策略虽然提高了算法跳出局部最优的能力，但在小网络中通常需要花费更多的时间寻找最优解。Contaldi 等^[10]提出了基于精英遗传算法的贝叶斯结构学习算法 (AESL-GA, adaptive elite-based structure learner using genetic algorithm)，该算法采用自适应的控制参数来避免参数设置对结果的影响，在小网络中学习到了较优的网络结构，但在大网络中由于缩小搜索空间导致学习到的结果不太理想。

上述启发式算法应用于贝叶斯网络结构学习时由于参数设置对搜索过程的影响，存在搜索效率较低、易陷入局部最优等问题，而 Seyedall 等^[11]提出的樽海鞘算法 (SSA, salp swarm algorithm) 具有参数较少、操作简单、易于实现等优点。SSA 将种群划分为引领者 (leader) 与跟随者 (follower)，通过引领者领导跟随者形成 slaps 链进行种群寻优。在处理复杂问题时由于引领者对跟随者的引领作用，容易使整个种群过早地聚集于局部最优解^[12]。

本文将 SSA 应用到贝叶斯网络结构学习，同时为了避免 SSA 陷入局部最优，将全局寻优能力强的差分进化 (DE, differential evolution) 算法^[13]与 SSA 融合，构建了一种混合樽海鞘-差分进化 (HBSS-DE,

hybrid binary salp swarm-differential evolution) 算法。该算法首先设置规模因子将种群划分^[14]为较好种群和较差种群。然后构建樽海鞘搜索策略更新较好种群，提高算法的收敛速度；构建差分搜索策略更新较差种群，避免陷入局部最优，并在合并子种群时利用变异算子增大搜索范围。最后通过种群的迭代搜索最佳结构。

2 HBSS-DE 算法的构建

HBSS-DE 算法利用最大支撑树与爬山算法建立初始种群 P_0 ，然后将 P_0 降序排列更新为种群 P ，并设置规模因子将种群 P 划分为较好种群 P_1 与较差种群 P_2 。构建樽海鞘搜索策略更新 P_1 ，建立差分搜索策略更新 P_2 。合并为种群 P 时利用两点变异算子增加种群的多样性，并根据规模因子重新划分 P 进入下次迭代。迭代结束输出种群 P 中评分最高的樽海鞘个体，即最佳的贝叶斯结构。

2.1 初始种群的建立及划分

根据数据样本计算目标网络各节点间的互信息，利用 2 个节点间的互信息可以得出这 2 个节点是否相关^[15]，不存在相关关系的节点必然不存在因果关系。以变量 X 、 Y 为例，互信息 $I(X, Y)$ 为

$$I(X, Y) = \sum_{x, y} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} \quad (1)$$

其中， $p(x, y)$ 为变量 X 和变量 Y 的联合概率， $p(x)$ 为变量 X 的概率， $p(y)$ 为变量 Y 的概率。

根据互信息计算各节点之间的权重，利用最大支撑树原则生成一个候补结构^[16]，可以有效地缩小搜索的空间。该候补结构中，除树形结构之外的节点利用爬山算法中的加边、减边、转边算子得到樽海鞘种群。其中每个樽海鞘代表的贝叶斯网络结构可用如图 1 所示的邻接矩阵表示。

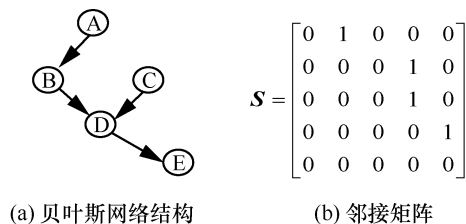


图 1 樽海鞘个体

采用贝叶斯信息准则 (BIC, Bayesian information criterion) 评分函数对樽海鞘个体评价。根据 BIC 评分函数的可分解性，当 BN 中的局部结构改

变时，为了减少重复计算的次数，只需利用式(2)计算改变的局部结构 G_1 的评分 $f_{\text{new}}(G_1, D)$ ，然后代入式(3)即可得到樽海鞘个体的评分 $f(G, D)$ 。

$$f_{\text{new}}(G_1, D) = \sum_{i \in G_1} f(X_i, \pi(X_i)) = \sum_{i \in G_1} \left(\sum_{j=1}^{q_i} \sum_{k=1}^{r_i} m_{ijk} \text{lb} \frac{m_{ijk}}{\sum_{k=1}^{r_i} m_{ijk}} - \frac{q_i(r_i-1)}{2} \text{lb} m \right) \quad (2)$$

$$f(G, D) = f_{\text{new}}(G_1, D) + f_{\text{old}}(G_2, D) \quad (3)$$

其中， X_i 表示节点， $\pi(X_i)$ 表示 X_i 的父节点； m_{ijk} 表示数据中满足 $\pi(X_i)$ 组合，即取值为 j 且 $X_i = k$ 的样本数； q_i 表示 $\pi(X_i)$ 取值共有 q_i 种组合； r_i 表示 X_i 共有 r_i 种取值； D 表示数据的样本； G 表示邻接矩阵， G_2 表示 G 中未改变的局部结构， $f_{\text{old}}(G_2, D)$ 表示 G_2 的评分。

选取樽海鞘种群中高于平均评分的个体作为初始种群 P_0 。文献[14]在种群划分阶段将初始种群随机分成 2 个相等的子种群更新个体，由于个体分布随着种群迭代动态变化，导致迭代后期无法提高算法的局部搜索能力。为了保证种群全局搜索与局部搜索的平衡，本文根据种群的进化情况建立自适应划分种群的规模因子 q ，具体的形式为

$$q = \left\lfloor \text{NP} \cdot K \cdot \sqrt{\frac{t}{T_m}} \right\rfloor \quad (4)$$

其中， NP 为 P_0 的种群规模， T_m 为最大迭代次数， t 为迭代次数， $\lfloor * \rfloor$ 为向下取整函数， K 为自适应调整规模因子 q 的参数，是根据个体评分与该种群平均评分的相对值建立的，具体的形式为

$$K = \frac{f_{\text{max}} - \bar{f}}{f_{\text{max}} - f_{\text{min}}} \quad (5)$$

其中， f_{max} 为 P 的最高评分值， \bar{f} 为 P 的平均评分值， f_{min} 为 P 的最低评分值。

由于在迭代初期，种群的个体分布较分散，即 K 的值较小，因此 q 的值较小保证算法的全局搜索。随着迭代的进行，种群的个体分布趋于集中，即 K 的值也逐渐增大，因而 q 的值较大保证算法的局部搜索。

根据樽海鞘个体的评分将 P_0 降序排列更新为种群 P ，利用规模因子 q 将种群划分为较好种群 P_1 与较差种群 P_2 ，具体形式为

$$\begin{aligned} P_1 &= \{P(i) \mid i \leq \text{NP}_1\} \\ P_2 &= \{P(i) \mid \text{NP}_1 < i \leq \text{NP}\} \end{aligned} \quad (6)$$

其中， P_1 的种群规模 $\text{NP}_1 = q$ ， P_2 的种群规模 $\text{NP}_2 = \text{NP} - \text{NP}_1$ 。

2.2 子种群更新

为了提高 HBSS-DE 算法的收敛速度，利用自适应的变异算子与交叉算子改进 SSA 的引领者与跟随者，建立樽海鞘搜索策略更新较好种群 P_1 。同时为了提高算法的收敛精度，改进 DE 算法中的变异算子与交叉算子，建立差分搜索策略更新较差种群 P_2 。

樽海鞘搜索策略利用自适应的变异算子更新引领者 $x_{k_1}^t$ ($k_1=1$ 表示评分最佳的个体)。根据 $x_{k_1}^t$ 与个体 x_r^t 的差异性，即利用式(7)的异或操作得到扰动矩阵 L_1^t ，将 L_1^t 中不为 0 的位置代入式(8)确定为变异位置集合 R 。将属于 R 的变异位置利用式(9)得到变异位置的值 $L_1^{t+1}(i, j)$ 。

$$L_1^t = 2e^{-\left(\frac{t}{T_m}\right)^2} \times (x_{k_1}^t \oplus x_r^t) \quad (7)$$

$$R = \{(i, j) \mid L_1^t(i, j) \neq 0\} \quad (8)$$

$$L_1^{t+1}(i, j) = \begin{cases} x_1^t(i, j) + L_1^t(i, j), & \alpha_2 > 0.5 \\ x_1^t(i, j) - L_1^t(i, j), & \text{其他} \end{cases} \quad (9)$$

其中， x_r^t 为 P_1 中随机选取的个体，并且 $x_r^t \neq x_{k_1}^t$ ， \oplus 为异或操作， \times 为矩阵点乘操作， $L_1^t(i, j)$ 为 L_1^t 中的具体位置的值， $L_1^{t+1}(i, j)$ 为 L_1^{t+1} 中的具体位置的值，选择概率 α_2 为 [0,1] 之间的随机数。

$L_1^{t+1}(i, j)$ 利用式(10)的传递函数转化为变异概率 $\alpha_3(i, j)$ 。根据 $\alpha_3(i, j)$ 判断 $x_{k_1}^t$ 的变异位置是否变异，即利用式(11)确定 $x_{k_1}^{t+1}(i, j)$ 是加边操作还是减边操作。

$$\alpha_3(i, j) = \frac{1}{1 + e^{-L_1^{t+1}(i, j)}} \quad (10)$$

$$x_{k_1}^{t+1}(i, j) = \begin{cases} 1, & (i, j) \in R \text{ 且 } \alpha_3(i, j) > \text{rand} \\ 0, & (i, j) \in R \text{ 且 } \alpha_3(i, j) \leq \text{rand} \\ x_{k_1}^t(i, j), & (i, j) \notin R \end{cases} \quad (11)$$

其中， rand 为 [0,1] 之间的随机数，“1”为加边操作，“0”为减边操作， $x_{k_1}^{t+1}(i, j)$ 为变异位置的值。

利用自适应的交叉算子更新跟随者 $x_{k_1}^t$ ($k_1 \in [2, \text{NP}_1]$)。由于在迭代初期，个体的差异较

大,采用单点交叉法可获得较广的搜索空间;而在迭代后期,个体差异较小,采用两点交叉法扩大搜索空间。因而根据种群的迭代过程在交叉算子中引入自适应因子。结合交叉概率的上下限构建如式(12)所示的自适应因子Cr。

$$Cr = Cr_{\min} + M(Cr_{\max} - Cr_{\min}) \quad (12)$$

其中, Cr_{\max} 为交叉概率上限; Cr_{\min} 为交叉概率下限; M 为控制交叉方式的参数,它是根据个体的差异,即当代种群中个体最佳评分与平均评分的相对值建立的,具体的形式如式(13)所示。

$$M = \frac{f_{\max} - \bar{f}}{f - f_{\min}} \quad (13)$$

其中, f_{\max} 为 P 中最高评分值, \bar{f} 为 P 的平均评分值, f_{\min} 为 P 中最小评分值。由式(13)可知,在算法的迭代初期, M 的值较小,因而Cr的值较小,算法能以较大的概率选择单点交叉法更新个体。随着迭代的进行, M 的值逐渐增大, Cr的值也逐渐增大,樽海鞘个体以较大的概率选择两点交叉法更新个体。

将自适应因子代入式(14)的交叉算子中更新 P_1 中的跟随者个体 $\mathbf{x}_{k_1}^t$, 即

$$\mathbf{x}_{k_1}^{t+1} = \begin{cases} F_1(\mathbf{x}_{k_1-1}^{t+1}, \mathbf{x}_{k_1}^t), Cr < \text{rand} \\ F_2(\mathbf{x}_{k_1-1}^{t+1}, \mathbf{x}_{k_1}^t), \text{其他} \end{cases} \quad (14)$$

其中, F_1 是单点交叉,即随机选择一列的位置,然后交换 2 个邻接矩阵对应列的位置; F_2 是两点交叉,即交换 2 个邻接矩阵对应两列的位置。以图 2 所示的单点交叉为例,交换矩阵 A 和矩阵 C 的最后一列,得到矩阵 E 。

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, E = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图2 单点交叉

1) 差分搜索策略

改进的变异算子先从 P_2 中随机选取 3 个樽海鞘个体 $\mathbf{x}_{r_1}^t$ 、 $\mathbf{x}_{r_2}^t$ 、 $\mathbf{x}_{r_3}^t$ (其中 $\mathbf{x}_{r_1}^t \neq \mathbf{x}_{r_2}^t \neq \mathbf{x}_{r_3}^t$), 然后将 $\mathbf{x}_{r_2}^t$ 、 $\mathbf{x}_{r_3}^t$ 代入式(15)产生差分矩阵 $W_{k_2}^t$, 根据式(8)得到 $\mathbf{x}_{r_1}^t$ 的变异位置集合 R 。利用式(16)将 $\mathbf{x}_{r_1}^t$ 中的变异位置值转化为变异概率 $c_1(i, j)$, 并代入式(17)得到 $\mathbf{x}_{r_1}^t$ 的

变异位置值 $V_{k_2}^t(i, j)$ 。

$$W_{k_2}^t = \mathbf{x}_{r_2}^t \oplus \mathbf{x}_{r_3}^t \quad (15)$$

$$c_1(i, j) = \frac{1}{1 + \exp\left[\frac{-2b(x_{r_1}(i, j) + FW_{k_2}^t(i, j) - 0.5)}{1 + 2F}\right]} \quad (16)$$

$$V_{k_2}^t(i, j) = \begin{cases} 1, (i, j) \in R \text{ 且 } c_1(i, j) > \text{rand} \\ 0, (i, j) \in R \text{ 且 } c_1(i, j) \leq \text{rand} \\ x_{r_1}^t(i, j), (i, j) \notin R \end{cases} \quad (17)$$

其中, $k_2 \subseteq [1, NP_2]$; r_1 、 r_2 、 r_3 为 $[1, NP_2]$ 之间的随机整数, 并且 $r_1 \neq r_2 \neq r_3$; rand 为 0~1 的随机数; $c_1(i, j)$ 为变异概率; $b=20$ 为带宽因子^[17]; $F=0.5$ 为缩放比例因子; $W_{k_2}^t(i, j)$ 为变异之后对应点的值。

改进的交叉算子是通过式(12)得到 P_2 的自适应因子Cr, 代入式(18)将变异个体 $V_{k_2}^t$ 与原个体 $\mathbf{x}_{k_2}^t$ 交叉得到更新个体 $\mathbf{x}_{k_2}^{t+1}$, 即

$$\mathbf{x}_{k_2}^{t+1} = \begin{cases} F_1(V_{k_2}^t, \mathbf{x}_{k_2}^t), Cr < \text{rand} \\ F_2(V_{k_2}^t, \mathbf{x}_{k_2}^t), \text{其他} \end{cases} \quad (18)$$

2) 选择操作

根据 DE 算法的贪婪选择策略, 在原个体 \mathbf{x}_k^t ($k \in [1, NP]$) 与更新个体 \mathbf{x}_k^{t+1} 中, 选择评分较高的个体进入下次迭代。利用式(3)得到更新个体 \mathbf{x}_k^{t+1} 的评分 $f(\mathbf{x}_k^{t+1})$, 并代入式(19)选择适应度值更优的个体进入下次迭代。

$$\mathbf{x}_k^{t+1} = \begin{cases} \mathbf{x}_k^{t+1}, f(\mathbf{x}_k^{t+1}) \geq f(\mathbf{x}_k^t) \\ \mathbf{x}_k^t, \text{其他} \end{cases} \quad (19)$$

其中, $f(\mathbf{x}_k^{t+1})$ 为更新个体的评分, $f(\mathbf{x}_k^t)$ 为原个体的评分, \mathbf{x}_k^{t+1} 为进入下次迭代的个体。

2.3 合并子种群

为了能够用更少的时间搜索到最佳网络结构, 文献[14]将更新之后的子种群 P_1 、 P_2 合并为种群 P , 使种群中的个体在搜索空间中彼此共享位置信息。但是种群个体的分布情况是动态变化的, 即在算法的迭代初期, 个体分布比较分散, 种群 P_1 、 P_2 中相同的个体较少; 随着迭代的进行, 个体趋于集中, 种群 P_1 、 P_2 中相同的个体也逐渐增多, 种群 P 的多样性易丧失。

由于迭代后期种群的多样性遭到破坏, 算法易陷入局部最优。为保证算法迭代后期有能力跳出局

部最优，本文将种群 P_1 、 P_2 中相同的个体通过两点变异增加种群的多样性。根据式(20)将 P_2 中与 P_1 评分相同的个体保存到种群 P_3 ，评分不同的个体保存到种群 P_4 。

$$\begin{aligned} P_3 &= \{P_2(\mathbf{x}_{k_2}^{t+1}) \mid f(\mathbf{x}_{k_2}^{t+1}) = f(\mathbf{x}_{k_1}^{t+1})\} \\ P_4 &= \{P_2(\mathbf{x}_{k_2}^{t+1}) \mid f(\mathbf{x}_{k_2}^{t+1}) \neq f(\mathbf{x}_{k_1}^{t+1})\} \end{aligned} \quad (20)$$

其中， $k_1 \subseteq [1, NP_1]$ ， $k_2 \subseteq [1, NP_2]$ ， $f(\mathbf{x}_{k_1}^{t+1})$ 、 $f(\mathbf{x}_{k_2}^{t+1})$ 分别是种群 P_1 、 P_2 中每个个体的评分。

将 P_3 中的个体通过两点变异增加种群多样性，其中两点变异是对邻接矩阵中任意一列的 2 个元素进行取反，如图 3 所示对矩阵 Q 的右上角 2 个元素进行取反，得到矩阵 W 。

$$Q = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad W = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

图 3 两点变异

将更新种群 P_3 与 P_1 、 P_4 合并成新的种群 P_0 ，根据个体评分降序排列更新种群 P ，利用规模因子重新划分种群进入下一次迭代。迭代结束后保留种群 P 中的评分最佳的樽海鞘个体，即最优的贝叶斯结构。

2.4 HBSS-DE 算法的整体流程

本文 2.1 节~2.3 节介绍了算法的原理，接下来，对算法的具体步骤进行简要描述。

步骤 1 输入数据样本，通过互信息建立最大支撑树，初始化参数 T_m ， $t=1$ 。

步骤 2 利用爬山算法对最大支撑树进行初步搜索得到樽海鞘种群，并选择高于平均值的樽海鞘个体建立初始种群 P_0 。

步骤 3 将 P_0 通过评分降序排序更新为种群 P ，并通过自适应因子划分种群 P 为较好种群 P_1 与较差种群 P_2 。

步骤 4 种群 P_1 中采用樽海鞘搜索策略进行更新，提高算法的收敛速度。

步骤 5 种群 P_2 中采用差分搜索策略进行更新，提高算法的收敛精度。

步骤 6 将 2 个子种群中相同个体利用两点变异增加种群的多样性，变异操作之后合并为种群 P_0 。

步骤 7 若满足 $t < T_m$ ，则 $t=t+1$ ，并转至步骤 3；否则，输出最高评分值对应的贝叶斯结构。

2.5 HBSS-DE 算法的收敛性分析

根据 Solis 等^[18]提出的概率测度法分析随机搜索算法 HBSS-DE 的收敛性。

引理 1 随机搜索算法 HBSS-DE 满足 $f(N(z, \xi)) \geq f(z)$ ，若 $\xi \in S_{\text{gbest}}$ ，则 $f(N(z, \xi)) \geq f(\xi)$ 。其中， f 为 HBSS-DE 算法解决最大化问题的适应度函数； N 为产生较优最新解的算子； z 为搜索空间 S 的最优解空间 S_{gbest} 中的樽海鞘个体，也是在 S_{gbest} 上产生可接受函数值的上确界； ξ 为算法在 S 上随机生成的樽海鞘个体。

证明 根据 2.2 节中选择操作的描述，可将 HBSS-DE 算法中对当前最优解的选择算子 N 定义为

$$N(\mathbf{G}^t, \mathbf{x}_k^t) = \begin{cases} \mathbf{G}^t, & f(N(\mathbf{x}_k^t)) < f(\mathbf{G}^t) \\ N(\mathbf{x}_k^t), & f(N(\mathbf{x}_k^t)) \geq f(\mathbf{G}^t) \end{cases} \quad (21)$$

其中， $N(\mathbf{x}_k^t)$ 为樽海鞘个体 \mathbf{x}_k^t 第 t 次迭代更新之后的个体， \mathbf{G}^t 为第 t 次迭代全局最优解的樽海鞘个体。由式(21)可知， \mathbf{G}^t 所对应的适应度值是单调不减的，且逐渐向解空间的上确界收敛。证毕。

引理 2 HBSS-DE 算法的最优解空间 S_{gbest} 的概率测度大于 0，即 $L[S_{\text{gbest}}] > 0$ 。

证明 假设 HBSS-DE 算法的贝叶斯结构群为 $X = \{x_1, x_2, \dots, x_n\}$ ， $x \in S$ ，其中搜索空间 S 是可列集或有限集，显然它的 Lebesgue^[19]总是大于 0，即 $L[S] > 0$ ，HBSS-DE 算法的最优解空间 S_{gbest} 属于 S 的一个 Borel^[19]子集，由 HBSS-DE 算法的最优解空间 S_{gbest} 的定义可知， $L[S_{\text{gbest}}] > 0$ ^[20]。证毕。

引理 3 HBSS-DE 算法中，当满足 $L[S_{\text{gbest}}] > 0$ 时，式(22)成立。

$$\prod_{t=1}^{\infty} (1 - \mu_t(S_{\text{gbest}})) = 0 \quad (22)$$

其中， $\mu_t(*)$ 为第 t 次迭代结果的概率测度。

证明 当满足 $L[S_{\text{gbest}}] > 0$ 时，有 $0 < \mu_{i,t}(S_{\text{gbest}}) < 1$ ，可知由 μ_t 产生的对 S_{gbest} 的概率测度为

$$\mu_t(S_{\text{gbest}}) = 1 - \prod_{i=1}^n (1 - \mu_{i,t}(S_{\text{gbest}})) \quad (23)$$

将 $\mu_t(S_{\text{gbest}})$ 代入式(22)，可得

$$\prod_{t=1}^{\infty} (1 - \mu_t(S_{\text{gbest}})) = \prod_{t=1}^{\infty} \left(\prod_{i=1}^n (1 - \mu_{i,t}(S_{\text{gbest}})) \right) = 0 \quad (24)$$

至此,引理 3 得证。根据 Solis 等^[18]提出的概率测度法可知,当 HBSS-DE 算法满足引理 1~引理 3 时,式(25)成立。

$$\lim_{t \rightarrow \infty} P[x^t \in S_{gbest}] = 1 \quad (25)$$

其中, $\lim_{t \rightarrow \infty} P[x^t \in S_{gbest}] = 1$ 表示在第 t 次的结果 x^t 落到 S_{gbest} 里的概率值为 1,即算法经过有限次迭代后,HBSS-DE 算法中一定有存在于 S_{gbest} 的个体。寻找最优解空间 S_{gbest} 之后,根据 HBSS-DE 算法的迭代性原则,在以后的所有迭代中,种群中所有个体都向 S_{gbest} 靠拢,最终收敛于 S_{gbest} 。

3 实验仿真

为了验证 HBSS-DE 算法的性能,在操作系统为 Windows7,处理器为 Intel i3 3.40 GHz CPU,内存为 4 GB,软件环境为 Matlab 2010a,基于贝叶斯网络工具箱 FullBNT-1.0.7 中的 ASIA 网络、CAR 网络、ALARM 网络进行仿真实验。其中,标准的 ASIA 网络具有 8 个节点、8 条边,标准的 CAR 网络具有 12 个节点、9 条边,标准的 ALARM 网络具有 37 个节点、46 条边。3 种标准网络是通过训练实际数据建立的贝叶斯网络,其中节点之间的边表示因果关系。在 3 种网络中随机生成数据量为 500、1 000、3 000、5 000 个的数据样本,因为数据是随机产生的,为降低数据的随机性对实验结果准确性的影响,每种样本容量分别产生 10 个,且每个数据单独运行 10 次,即每组数据运行 100 次后取平均值作为实验结果。

随着节点的增加,采用评分搜索的方法学习全局最优解是一种 NP 问题^[7],而全局最优解与较其稍差的局部最优解的实际效益相差不大^[18],因而采用足够好的解 S_{gbest} 来代替全局最优解,将 NP 问题转化为 N-问题。HBSS-DE 算法在 3 种网络中的实验结果分别如表 1~表 3 所示,其中 f_{best} 是 100 次实验中的最佳 BIC 评分, f_{av} 是 100 次实验的平均 BIC 评分, f_{sn} 是标准网络的 BIC 评分。利用 $\sigma = \left| \frac{f_{sn} - f_{av}}{f_{sn}} \right|$ 验证算法的局部最优解 S_{gbest} 是否接近全局最优解。

由表 1~表 3 可知,在小网络(如 ASIA 网络、CAR 网络)中,当数据量 > 500 个时 HBSS-DE 算法搜索到的最佳结构评分 f_{best} 与标准网络的评分 f_{sn} 几乎相同,即在小网络中可以搜索标准网络,这是

因为小网络中节点较少、搜索空间较小。而在大网络中(如 ALARM 网络),HBSS-DE 算法的 f_{best} 比较接近 f_{sn} ,这是因为随着节点的增多,搜索空间变大,算法收敛于接近全局最优解的局部最优解。在 ASIA 网络中 σ 的平均值为 0.004 91,在 CAR 网络中 σ 的平均值为 0.010 4,在 ALARM 网络中 σ 的平均值为 0.023 9,总体而言 HBSS-DE 算法的局部最优解 S_{gbest} 接近全局最优解。

表 1 ASIA 网络 HBSS-DE 算法的 BIC 评分

数据量/个	BIC 评分		
	f_{best}	f_{av}	f_{sn}
500	-1 139.83	-1 143.31	-1 133.84
1 000	-2 243.91	-2 265.09	-2 243.91
3 000	-6 822.17	-6 825.32	-6 822.17
5 000	-11 228.53	-11 243.58	-11 228.53

表 2 CAR 网络 HBSS-DE 算法的 BIC 评分

数据量/个	BIC 评分		
	f_{best}	f_{av}	f_{sn}
500	-3 387.12	-3 392.34	-3 357.18
1 000	-6 980.96	-7 083.95	-6 980.94
3 000	-19 610.25	-19 930.82	-19 610.25
5 000	-33 565.68	-33 965.16	-33 565.65

表 3 ALARM 网络 HBSS-DE 算法的 BIC 评分

数据量/个	BIC 评分		
	f_{best}	f_{av}	f_{sn}
500	-5 221.83	-5 310.02	-5 196.92
1 000	-10 148.52	-10 180.08	-10 017.97
3 000	-29 227.87	-29 780.76	-28 993.16
5 000	-48 228.58	-48 920.58	-47 753.58

根据文献[21],列出分析算法性能的评价指标。

Ext (execution time): 找到最佳网络所需要的时间(单位为 s)。

TP (true positive): 真正例,即预测结构与标准结构中相同且方向为正的边。

FP (false positive): 假正例,即预测结构中边的方向为正,标准结构中为负。

FN (false negative): 假负例,即预测结构中边的方向为负,标准结构中为正。

TPR (true positive rate): 正确率,即预测结构中 TP 与标准结构中边的比值, $TPR = \frac{TP}{TP+FN}$ 。

FPR (false positive rate): 精确率, 即预测结构中 TP 与预测结构全部边的比值, $FPR = \frac{TP}{TP+FP}$ 。

F (F-measure): 综合评价指标, 即正确率和召回率的调和平均值, $F = 2FPR \frac{TPR}{FPR+TPR}$ 。

W: 100 次实验中 HBSS-DE 算法的实验值 ψ 与其他算法的实验值 ε 相比提升的百分比, 即

$$W = \frac{\sum_{l=1}^{100} |\psi_l - \varepsilon_l|}{100 \varepsilon_l}。$$

将 HBSS-DE 算法与 AESL-GA、MAK、BEWCA-BN、MMHC 算法进行对比实验。其中本文算法自适应因子的上下限分别为 0.85、0.2。根据文献[1]设 BEWCA-BN 尺度因子 c 的上下限分别为 0.9、0.7, 搜索强度参数 d 的上下限分别为 0.2、0.01。根据文献[7]设 MMHC 独立性测试的置信度 α 为 0.01, 统计阈值设置为 0.05。根据文献[8]设 MAK 信息素强度系数为 1, 信息素蒸发系数为 0.7, 启发式因子权重为 2。根据文献[9]设 AESL-GA 独立

性测试的置信度为 0.01, 精英个体阈值 α 为 0.09。实验结果如下: 表 4~表 6 是不同算法在不同网络的平均 BIC 评分, 图 4~图 6 是不同算法在不同网络的结构对比, 表 7~表 9 是不同算法在不同网络的执行时间对比。

由表 4~表 6 可知, 当数据量为 500 时, 在 ASIA 网络与 CAR 网络中, 本文算法和 BEWCA-BN 算法均能找到相对较优的评分, 但随着数据量的增大, 本文算法相比于其他算法能学习到更优的评分。根据 W 计算, 当数据量为 500、1 000、3 000、5 000 时, 本文算法与其他算法相比, BIC 评分的平均提高值分别为 W_{BIC}^{500} 、 W_{BIC}^{1000} 、 W_{BIC}^{3000} 、 W_{BIC}^{5000} , 然后利用 $\frac{W_{BIC}^{500} + W_{BIC}^{1000} + W_{BIC}^{3000} + W_{BIC}^{5000}}{4}$ 得到整体 BIC 提升百分比。本文算法在 ASIA 网络中的 BIC 评分比 MAK、MMHC、AESL-G、BEWCA-BN 算法平均提升了 3.71%、4.34%、3.85%、1.86%, 在 CAR 网络中平均提升了 4.41%、3.64%、2.49%、1.47%, 在 ALARM 网络中平均提升了 6.96%、9.08%、10.02%、3.84%。总体而言, HBSS-DE 算法能够

表 4 不同算法在 ASIA 网络中不同数据量下的平均 BIC 评分

数据量/个	BIC 评分				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	-1 143.31	-1 256.04	-1 285.34	-1 294.32	-1 143.32
1 000	-2 265.06	-2 653.03	-2 294.76	-2 310.48	-2 265.85
3 000	-6 825.32	-7 069.02	-6 898.79	-6 948.57	-6 894.61
5 000	-11 243.58	-12 109.55	-11 956.57	-11 789.65	-1 1678.23

表 5 不同算法在 CAR 网络中不同数据量下的平均 BIC 评分

数据量/个	BIC 评分				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	-3 392.34	-3 498.95	-3 405.89	-3 425.87	-3 398.72
1 000	-7 083.95	-7 393.02	-7 209.63	-7 298.65	-7 144.06
3 000	-19 930.82	-21 970.46	-20 446.78	-20 324.85	-20 881.64
5 000	-33 965.16	-35 461.82	-35 192.46	-34 780.25	-34 455.11

表 6 不同算法在 ALARM 网络中不同数据量下的平均 BIC 评分

数据量/个	BIC 评分				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	-5 310.02	-5 680.04	-5 792.48	-5 852.14	-5 514.29
1 000	-10 180.08	-10 311.08	-10 194.25	-10 249.75	-10 245.48
3 000	-29 780.76	-30 308.01	-29 894.56	-30 014.25	-29 989.65
5 000	-48 920.58	-49 259.15	-48 925.67	-49 652.37	-48 958.73

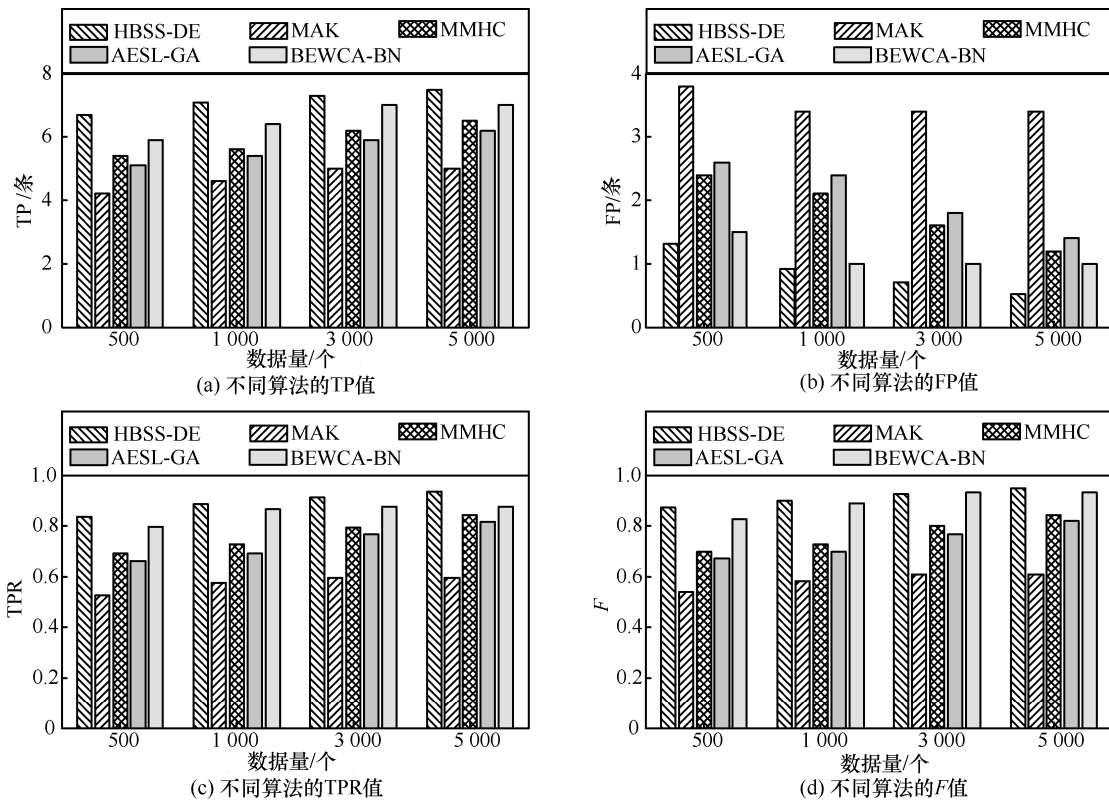


图4 ASIA网络不同数据下的结构对比

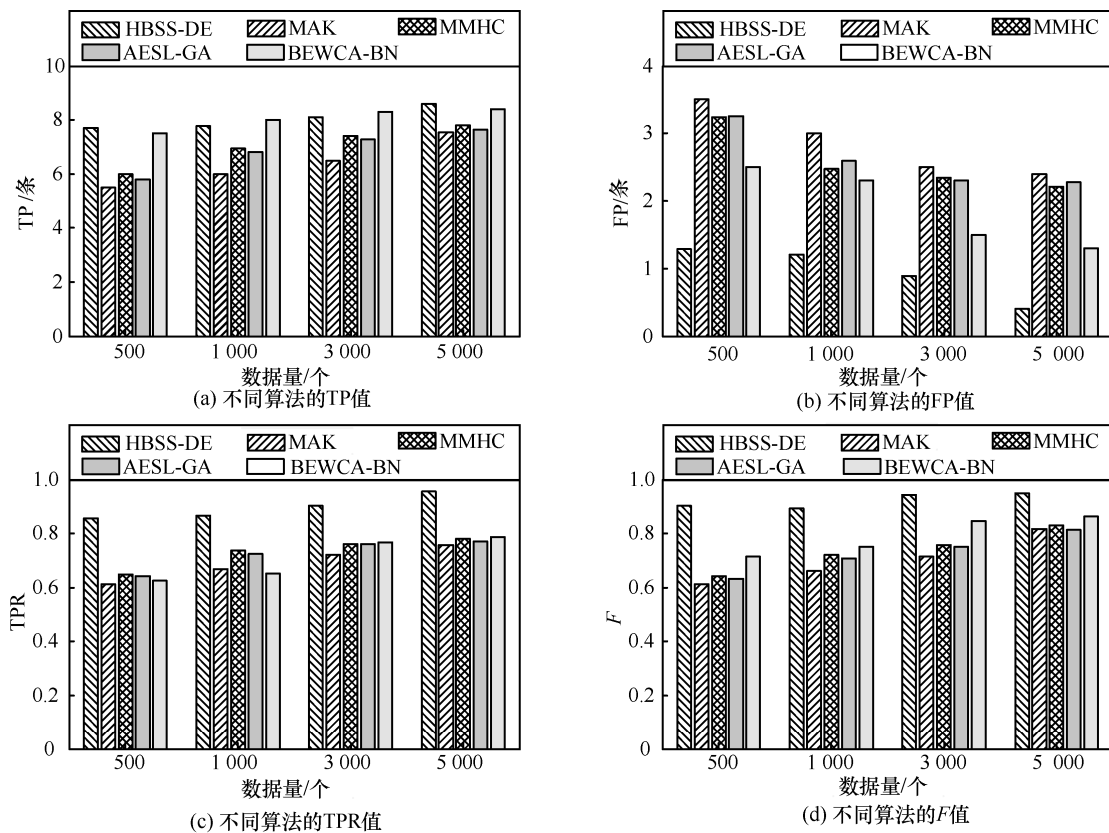


图5 CAR网络不同数据下的结构对比

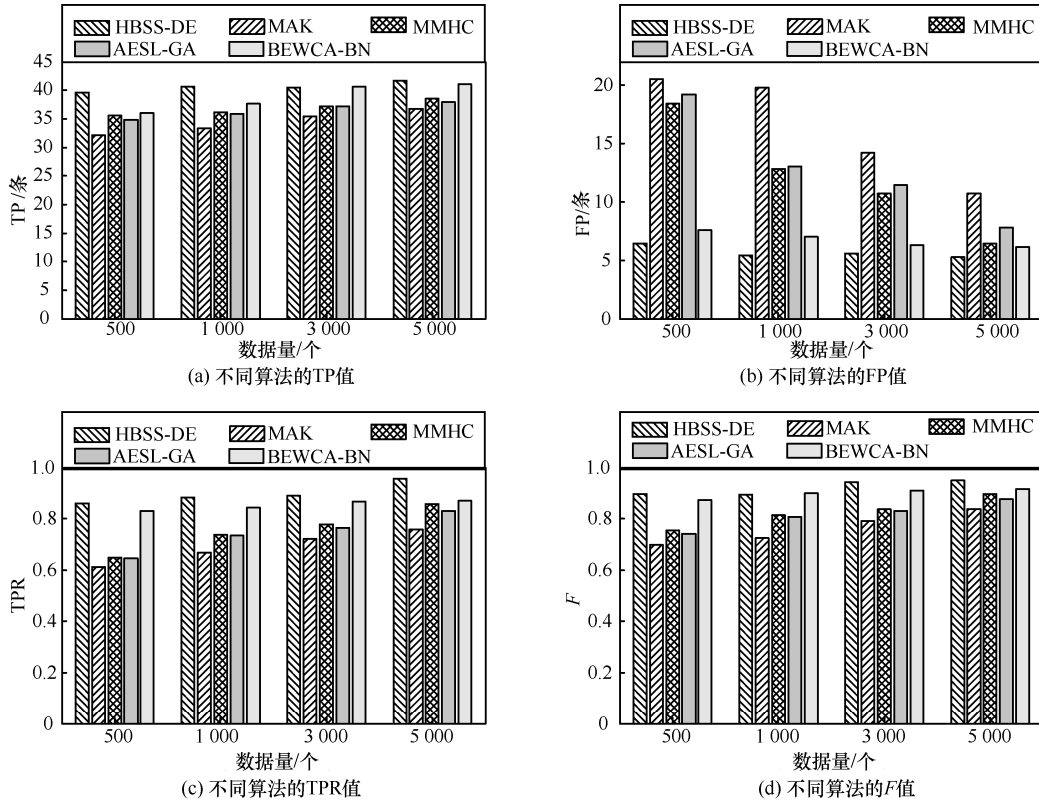


图 6 ALARM 网络不同数据下的结构对比

表 7 不同算法在 ASIA 网络中不同数据量下的执行时间

数据量/个	执行时间/s				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	2.11	3.62	4.52	1.58	3.85
1 000	2.37	3.74	4.85	2.21	4.14
3 000	2.85	4.32	5.62	2.89	4.69
5 000	3.03	5.31	7.67	3.52	4.43

表 8 不同算法在 CAR 网络中不同数据量下的执行时间

数据量/个	执行时间/s				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	5.23	7.04	10.24	8.47	7.79
1 000	5.84	7.72	10.79	8.52	7.95
3 000	6.08	9.57	11.58	9.72	9.82
5 000	6.85	11.77	12.28	10.48	10.11

表 9 不同算法在 ALARM 网络中不同数据量下的执行时间

数据量/个	执行时间/s				
	HBSS-DE	MAK	MMHC	AESL-GA	BEWCA-BN
500	118.37	314.16	2 305.48	304.24	125.47
1 000	128.54	356.18	2 465.75	325.49	139.84
3 000	171.25	505.57	3 180.27	490.24	172.43
5 000	235.48	948.81	4 587.58	578.59	195.48

找到评分更优的贝叶斯网络结构，在大型网络中比其他算法的提升效果更佳。这是由于本文算法可以在较好个体周围局部搜索，在较差个体周围全局搜索，并通过规模因子保证了算法的局部搜索与全局搜索的平衡，即在迭代前期全局搜索，迭代后期局部搜索。同时对于 2 个子种群中相同个体利用两点变异策略增加了种群的多样性，避免陷入局部最优。

由图 4~图 6 可知，本文算法的 TP 值在 ASIA 网络中比 MAK、MMHC、AESL-GA、BEWCA-BN 算法平均多了 1.49、1.34、1.02、0.79，在 CAR 网络中比其他算法平均多了 2.21、1.71、1.23、0.56，在 ALARM 网络中比其他算法平均多了 7.41、4.11、4.81、3.65。本文算法在 ASIA 网络、CAR 网络、ALARM 网络中 FP 的值均小于其他算法，且在 3 个网络中的 TPR 值均大于其他算法。根据 W 计算，当数据量为 500、1 000、3 000、5 000 时，本文算法与其他算法相比， F 值平均提高的百分比分别为 W_F^{500} 、 W_F^{1000} 、 W_F^{3000} 、 W_F^{5000} ，然后利用 $\frac{W_F^{500} + W_F^{1000} + W_F^{3000} + W_F^{5000}}{4}$ 计算整体提高百分比。

本文算法的 F 值在 ASIA 网络中比 MAK、MMHC、AESL-GA、BEWCA-BN 算法平均提高了 22.54%、17.89%、15.17%、8.57%，在 CAR 网络中比其他算法平均提高了 25.74%、19.29%、13.32%、9.64%，在 ALARM 网络中比其他算法平均提高了 24.71%、15.19%、10.91%、6.36%。总体而言，HBSS-DE 算法能够找到更准确的贝叶斯网络结构，这是由于在较差种群中采用差分搜索策略，即在较差个体周围进行全局搜索增加了种群的多样性；通过自适应因子保证种群在迭代前期的全局搜索能力强；同时合并子种群时的两点变异增大种群的搜索范围，增加了种群跳出局部最优的能力。

由表 7~表 9 可知，当 ASIA 网络的数据量为 500 和 1 000 时，AESL-GA 运行时间最短，HBSS-DE 算法运行时间比 AESL-GA 略长一些，这是由于在小数据量下 HBSS-DE 算法为了提高算法的全局搜索，通过规模因子将种群划分为 2 个子种群更新个体，并将子种群中相同的个体利用两点交叉变异扩大搜索的范围。虽然避免了陷入局部最优，但同时增加了运行时间。根据 W 计算，当数据量为 500、1 000、3 000、5 000 时，HBSS-DE 算法与其他算法相比，执行时间平均缩短的百分比分别为 W_{ts}^{500} 、

W_{ts}^{1000} 、 W_{ts}^{3000} 、 W_{ts}^{5000} ，其整体缩短的百分比根据 $\frac{W_{ts}^{500} + W_{ts}^{1000} + W_{ts}^{3000} + W_{ts}^{5000}}{4}$ 得到。与 MAK 算法、

MMHC 算法、BEWCA-BN 算法相比，HBSS-DE 算法在 ASIA 网络中的运行时间平均缩短了 15.25%、21.77%、5.32%。在 CAR 网络中，HBSS-DE 算法比 MAK、MMHC、AESL-GA、BEWCA-BN 算法平均缩短了 18.57%、13.26%、15.18%、5.12%。当 ALARM 网络数据量为 5 000 时，HBSS-DE 算法比 BEWCA-BN 算法耗时略长，这是由于节点较多的网络搜索的范围较大，在迭代的过程中差分搜索策略虽然增大了搜索的空间，提高了算法的全局搜索能力；合并子种群时的两点交叉变异虽然增加了种群的多样性，避免了算法陷入局部最优，但是差分搜索策略与两点交叉变异也增加了算法的搜索时间。HBSS-DE 算法在 ALARM 网络中比 MAK、MMHC、AESL-GA、BEWCA-BN 算法平均缩短了 28.23%、92.63%、19.04%、1.42%。总体而言，HBSS-DE 算法在大型网络中比其他算法的搜索效率的优势更明显。这是因为樽海鞘搜索策略的局部搜索能力强，能够快速收敛到最优解。而自适应因子保证了迭代后期的局部搜索，提高了搜索效率；同时在评价个体时仅计算个体改变的部分结构，减少在评价个体时消耗的无用时间。

4 结束语

本文提出了基于混合樽海鞘-差分算法的贝叶斯网络结构学习算法 HBSS-DE。该算法通过设置规模因子，将种群划分为 2 个子种群，利用构建的樽海鞘搜索策略与差分搜索策略分别更新不同的子种群，在合并子种群时利用两点交叉变异增加种群的多样性。仿真实验证明，HBSS-DE 算法能够找到最佳的贝叶斯网络结构，其中规模因子平衡了局部搜索与全局搜索，樽海鞘搜索策略提高了算法的寻优效率，差分搜索策略提高了算法的收敛精度。与其他算法相比，HBSS-DE 算法的收敛精度与寻优效率均有提升。

参考文献:

[1] WANG J, LIU S. Novel binary encoding water cycle algorithm for solving bayesian network structures learning problem[J]. Knowledge-Based Systems, 2018, 150: 95-110.
 [2] XUAN J, LU J, ZHANG G, et al. Bayesian nonparametric relational topic model through dependent Gamma processes[J]. IEEE Transac-

- tions on Knowledge & Data Engineering, 2017, 29(7): 1357-1369.
- [3] 耿杨, 邵苏杰, 郭少勇, 等. 基于可见损伤持续时间贝叶斯网络的视频 QoE 评估方法[J]. 通信学报, 2017, 38(6): 136-141.
GENG Y, SHAO S J, GUO S Y, et al. Bayesian network-based video QoE assessment method using image sustained damage analysis[J]. Journal on Communications, 2017, 38(6): 136-141.
- [4] ALONSO J I, OSSA L D L, GÁMEZ J A, et al. On the use of local search heuristics to improve GES-based Bayesian network learning[J]. Applied Soft Computing, 2018, 64: 366-376.
- [5] SCANAGATTA M, CORANI G, CAMPOS C P D, et al. Approximate structure learning for large Bayesian networks[J]. Machine Learning, 2018, 15: 1-19.
- [6] LIU H, ZHOU S, LAM W, et al. A new hybrid method for learning bayesian networks: Separation and reunion[J]. Knowledge-Based Systems, 2017, 121: 185-197.
- [7] YANG C, JI J, LIU J, et al. Structural learning of Bayesian networks by bacterial foraging optimization[J]. International Journal of Approximate Reasoning, 2016, 69(C): 147-167.
- [8] TSAMARDINOS I, BROWN L E, ALIFERIS C F. The max-min hill-climbing Bayesian network structure learning algorithm[J]. Machine Learning, 2006, 65(1): 31-78.
- [9] 刘浩然, 孙美婷, 李雷, 等. 基于蚁群节点寻优的贝叶斯网络结构算法研究[J]. 仪器仪表学报, 2017, 38(1): 143-150.
LIU H R, SUN M T, LI L, et al. Bayesian network structure algorithm based on ant colony node optimization[J]. Chinese Journal of Scientific Instrument, 2017, 38 (1): 143-150.
- [10] CONTALDI C, VAFAEE F, NELSON P C. Bayesian network hybrid learning using an elite-guided genetic algorithm[J]. Artificial Intelligence Review, 2018, 293: 1-28.
- [11] SEYEDALL M, AMIR H G, SEYEDEH Z M, et al. Salp swarm algorithm: a bio-inspired optimizer for engineering design problems[J]. Advances in Engineering Software, 2017, 114(1): 163-191.
- [12] SAYED G I, KHORIBA G, HAGGAG M H. A novel chaotic salp swarm algorithm for global optimization and feature selection[J]. Applied Intelligence, 2018, 48: 3462-3481
- [13] 刘会宇, 韩继红, 袁霖, 等. 基于双变异策略的自适应骨架差分进化算法[J]. 通信学报, 2017, 38(8): 201-212.
LIU H Y, HAN J H, YUAN L, et al. Self-adaptive bare-bones differential evolution based on bi-mutation strategy[J]. Journal on Communications, 2017, 38(8): 201-212.
- [14] WANG G G, GANDOMI A H, ALAVI A H, et al. A hybrid method based on krill herd and quantum-behaved particle swarm optimization[J]. Neural Computing and Applications, 2016, 27(4): 989-1006.
- [15] JIAO J, VENKAT K, WEISSMAN T. Mutual information, relative entropy and estimation error in semi-martingale channels[J]. IEEE Transactions on Information Theory, 2017, PP(99): 1.
- [16] 刘浩然, 吕晓贺, 李轩, 等. 基于 Bayesian 改进算法的回转窑故障诊断模型研究[J]. 仪器仪表学报, 2015, 36(7): 1554-1561.
LIU H R, LV X H, LI X, et al. A study of fault diagnosis model of rotary kiln based on improved algorithm of Bayesian[J]. Chinese Journal of Scientific Instrument, 2015, 36(7): 1554-1561.
- [17] GOUDOS S. Antenna design using binary differential evolution: application to discrete-valued design problems[J]. IEEE Antennas & Propagation Magazine, 2017, 59(1): 74-93.
- [18] SOLIS F, WETS R. Minimization by random search technologies[J]. Mathematics of Operations Research, 1981, 6: 19-30.
- [19] ZAKHAROV V K, RODIONOV T V. Naturalness of the class of Lebesgue-Borel-Hausdorff measurable functions[J]. Mathematical Notes, 2014, 95(3-4): 500-508.
- [20] 陈志敏, 田梦楚, 吴盘龙, 等. 基于蝙蝠算法的粒子滤波法研究[J]. 物理学报, 2017, 66(5): 41-50.
CHEN Z M, TIAN M C, WU P L, et al. Research on particle filtering based on bat algorithm [J]. Journal of Physics, 2017, 66(5): 41-50.
- [21] LIU F, ZHANG S W, GUO W F, et al. Inference of gene regulatory network based on local bayesian networks[J]. Plos Computational Biology, 2016, 12(8): e1005024.

[作者简介]



刘彬 (1953-), 男, 黑龙江哈尔滨人, 博士, 燕山大学教授、博士生导师, 主要研究方向为无线传感网络、工业故障检测。



范瑞星 (1993-), 男, 河北邯郸人, 燕山大学硕士生, 主要研究方向为智能算法、贝叶斯网络、故障诊断。



刘浩然 (1980-), 男, 黑龙江哈尔滨人, 博士, 燕山大学教授、博士生导师, 主要研究方向为无线传感网络、工业故障检测。

张力悦 (1994-), 男, 河北唐山人, 燕山大学硕士生, 主要研究方向为智能算法、贝叶斯网络、故障诊断。

王海羽 (1993-), 男, 黑龙江鹤岗人, 燕山大学硕士生, 主要研究方向为智能算法、贝叶斯网络、故障诊断。

张春兰 (1992-), 女, 河北衡水人, 燕山大学硕士生, 主要研究方向为智能算法、贝叶斯网络、故障诊断。