控制与决策
Control and Decision

基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法

方伟, 吴昀霖, 朱书伟

引用本文:
方伟, 吴昀霖, 朱书伟. 基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法[J]. 控制与决策, 2026, 41(4): 1077-1088.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2025.0110

您可能感兴趣的其他文章

Articles you may be interested in

基于度量学习和典型相关分析的亲缘关系识别网络

Kinship relationship recognition network based on metric learning and canonical correlation analysis

控制与决策. 2021, 36(8): 1977-1983 https://doi.org/10.13195/j.kzyjc.2019.1798

基于种群演化的超参数异步并行搜索

Asynchronous parallel hyperparameter search with population evolution

控制与决策. 2021, 36(8): 1825-1833 https://doi.org/10.13195/j.kzyjc.2019.1743

基于预防维护的单机调度问题

Single-machine scheduling problem with preventative maintenance activities

控制与决策. 2021, 36(2): 395-402 https://doi.org/10.13195/j.kzyjc.2019.0626

基于分类特征约束变分伪样本生成器的类增量学习

Class incremental learning based on variational pseudo-sample generator with classification feature constraints

控制与决策. 2021, 36(10): 2475-2482 https://doi.org/10.13195/j.kzyjc.2020.0228

基于迁移学习灰支持向量回归机的交互式进化计算

Interactive evolutionary computation based on transfer learning grey support vector regression

控制与决策. 2021, 36(10): 2399-2408 https://doi.org/10.13195/j.kzyjc.2020.0420

基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法

方 伟†，吴昀霖，朱书伟

(江南大学 人工智能与计算机学院，江苏 无锡 214122)

摘 要: 贝叶斯网络 (BN) 是一种概率图模型, 用于表示不确定的因果关系. 由于解空间的数量随着变量数量增长呈超指数增长, 使得贝叶斯网络结构学习 (BNSL) 成为 NP 难问题. 遗传算法 (GA) 可以高效地在空间中搜索更多可能的结构组合, 在 BNSL 问题中取得了诸多成果, 但是仍然存在过早收敛, 结构准确率不高等问题. 鉴于此, 提出一种基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法 (DM-GABN). 在去环阶段, 使用翻转-删除-修复混合操作代替删除边以保留更多样的基因型; 在选择算子阶段, 根据当前种群多样性动态调整种群年龄阈值, 淘汰衰老个体, 维持合理的种群年龄结构; 在交叉策略中, 引入生物学的基因型频率概念, 保护低频结构的同时利用互信息限制搜索空间大小并引导搜索. 在 10 个标准 BN 数据集上对 DM-GABN 进行实验评估, 并与包含最先进方法在内的 10 种 BNSL 方法进行对比. 实验结果显示, 所提出方法学习的 BN 结构准确率更高, 算法收敛速度更快.

关键词: 贝叶斯网络；遗传算法；结构学习；种群多样性；互信息；基因型频率

中图分类号: TP18 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2025.0110

引用格式: 方伟, 吴昀霖, 朱书伟. 基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法 [J]. 控制与决策, 2026, 41(4): 1077-1088.

Diversity and mutual information mixed guidance GA for Bayesian network structure learning

FANG Wei†，*WU Yun-lin*，*ZHU Shu-wei*

(School of Artificial Intelligence and Computer Science，Jiangnan University，Wuxi 214122，China)

Abstract: A Bayesian network (BN) is a probability graph model which can be used to express the causal relationship between variables. Due to the solution space growing exponentially with the increasing number of nodes, Bayesian network structure learning (BNSL) has become an NP-hard problem. The genetic algorithm (GA) can efficiently search more possible structural combinations in the search space, and it has achieved many satisfying results in recent years in the BNSL problem, but there are still some problems, such as premature convergence and low accuracy. To solve these problems, we propose a diversity and mutual information mixed guidance GA for Bayesian network structure learning (DM-GABN). In the cycle removal phase, this paper uses flip edges instead of deleting to preserve more diverse genotypes. In the selection phase, we remove older individuals and maintain a reasonable population age structure to improve population diversity. We introduce the concept of biological genotype frequency into the crossover strategy, and protect the structure with low genotype frequency. In addition, we use mutual information to guide the search process. Experimental results on ten widely used benchmark networks and compared with ten BNSL algorithms including the most advanced methods show the proposed algorithm DM-GABN outperforms other algorithms regarding structural accuracy and convergence speed.

Keywords: Bayesian network； genetic algorithm； structure learning； population diversity； mutual information； genotype frequency

收稿日期: 2025-01-21；**录用日期:** 2025-04-04.

基金项目: 国家自然科学基金项目 (62473176, 62106088, 62206113)；船舶结构安全全国重点实验室项目 (450324300).

责任编委: 王凌.

†通信作者. E-mail: fangwei@jiangnan.edu.cn.

0 引 言

贝叶斯网络 (Bayesian network, BN) 是一种概率图模型, 被广泛应用于数据挖掘[1]、机器学习[2]、安全生产[3] 等多个领域, 常用于表示多变量间的因果关系. 贝叶斯网络结构学习 (Bayesian network structure learning, BNSL) 是从数据中学习并推理变量间的依赖关系, 由于解空间的数量随着变量数量增长呈超指数增长, BNSL 成为 NP 难的问题. 传统的构建贝叶斯网络结构方法为专家手动构建, 但是, 得到的结果受专家知识领域的影响较大, 且这种方法难以构建复杂网络.

为提高 BNSL 的效率和结果的准确性, 研究者们提出了两种从数据中自动学习贝叶斯网络结构的方法, 分别为基于约束 (constraint-based, CB) 的方法和基于评分搜索 (score-and-search, SS) 的方法. CB 方法利用条件独立 (conditional independence, CI) 测试来判断每对节点间的依赖关系从而构建 BN, 但是, CB 方法因使用大量的 CI 测试导致算法效率较低, 且数据的噪声和也会影响算法结果的准确度; SS 方法是通过评分函数和搜索策略对贝叶斯网络结构进行学习, 在大量解中寻求评分更高的结果, SS 方法效率受搜索策略影响较大, 算法易陷于局部最优. 因此, 研究者们提出了结合以上两种方法优点的混合算法, 首先使用 CB 方法限制搜索空间, 再使用 SS 方法在限制后的空间内搜索最优的解. 由于遗传算法 (genetic algorithm, GA) 可以高效地在空间中搜索更多可能的结构组合, 近年来有多名学者使用 GA 来解决 BNSL 问题[4]. Dai 等[5] 提出的 MIIGA 使用互信息 (mutual information, MI) 限制了搜索空间; Yan 等[6] 提出了 MIGA, 在初始化种群时利用 MI 生成初始种群, 并在交叉节点由 MI 引导决定子代结构. 但是, 以上方法在各环节中没有充分地利用 MI 引导搜索过程, 且存在遗传算法常见的过早收敛的问题. 种群多样性过低是该问题产生的一大原因, 随着种群的迭代, 多样性较低的种群后代将会拥有趋于相同的基因型, 导致算法无法搜索到更优的个体.

鉴于此, 本文提出一种基于种群多样性和互信息混合引导的 GA 算法 (diversity and mutual information mixed guidance GA for Bayesian network structure learning, DM-GABN) 用于解决该问题. 本文引入生物学中的基因型频率概念, 在遗传算法的交叉算子中对种群中基因型频率较低的结构进行保护; 在去环的算子中使用翻转-删除-修复混合策略代替删除, 保留更多样的基因; 在选择算子中, 根据种群多样性动态调整年龄阈值, 淘汰衰老个体, 在种群中留出更多空间用于生成新个体, 维持种群合理的年龄结构. 最后, 在算法的多个环节中使用互信息限制搜索空间并引导搜索过程.

1 相关工作

1.1 贝叶斯网络结构学习

BN 是一种有向无环图 (directed acyclic graphical model, DAG), 用于表示大型的、复杂的和不确定的因果关系. BNSL 是从数据中构建出一种表达数据间关系的图结构, 数据中的变量转换为图的节点, 参数间的依赖关系用图形中的边来表示, 这使得随机变量间不确定的关系有了清晰直观的结构化描述. 由于避免了专家手动构造网络结构的人力资源浪费和人工误差, 近年来, 从数据中自动学习网络结构的方法成为 BNSL 方法的主流策略.

CB 方法通常假定在因果关系充足的情况下, 返回与数据中独立关系一致的 DAG 合集, 分为全局学习方法和局部学习方法. 全局学习方法主要思路是学习整体的图形结构. Spirtes 等[7] 在 1989 年提出了 SGS 算法, 对网络中每对节点的每个可能的条件集均使用了 CI 测试, 判断节点间的独立性并确定边的方向, 该算法效率较低但是相对稳定, 其核心逻辑被大多数 CB 方法沿用; Spirtes 等[8] 随后提出的经典算法 PC 是 CB 方法中最具代表性的方法, 减少了大量不必要的 CI 测试, 显著提高了效率; Cheng 等[9] 在 1999 年提出了 TPDA 算法, 定量地使用了 MI 测试判断节点间的条件独立, 获得了与 PC 方法相似的准确性和效率. 局部学习方法则是分别了解与每个变量有关的局部结构, 如父节点或相邻节点, 然后合并为完整的图结构. 1999 年, Margaritis 等[10] 提出的 (grow shrink, GS) 算法是第 1 个利用马尔可夫毯概念来减少邻接阶段中 CI 测试数量的算法. 后续, 有学者提出了 IAMB 方法[11], 优化了马尔可夫毯发现, 使其可以处理数千个节点, 并使用互信息确定节点在生长阶段被纳入马尔可夫毯的顺序, 减少不必要的 CI 测试. Inter-IAMB 将生长与搜索阶段相结合[12], 能够处理合成网络.

SS 方法是通过评分函数搜索优秀的结构, 需要在大量可行解中寻找评分最高的解. 最早应用于 BNSL 的搜索算法之一, 是 Cooper 等[13] 提出的 K2 算法, 该算法假设节点排序是已知的, 并对有序的节点列表进行处理, 贪婪地从列表中为排序靠前的候选父节点添加边, 以最大限度地提高了 K2 得分; Heckerman 等[14] 提出了一种消除具有预定义节点排

序限制的方法, 即在 DAG 空间上的通用爬山 (hill climbing, HC) 贪婪搜索算法, 它是一个较为简单且最常用的搜索策略; Tabu 算法在 HC 的基础上[15], 新增了最近搜索过的 DAG, 鼓励算法进入新的搜索区域避免出现局部最优的情况.

混合算法中最为经典且高效的是由 Tsamardinos 等[16] 提出的 MMHC 算法, 其在 CB 阶段使用 MMPC[11] 对局部进行了约束, 从而构建了无向图搜索空间, 随后在 SS 阶段使用禁忌爬山法在搜索空间中学习 DAG 网络结构, 取得了很好的效果. 近年来, Constantinou[17] 提出的 SaiyanH 将结构学习分为 3 个阶段: 首先利用局部学习确定连通的骨架图作为搜索空间, 然后使用 CB 方法获得了骨架图中的边, 最后使用 SS 方法进一步对结构进行了修改.

1.2　基于遗传算法的 BNSL

GA 是基于自然选择和遗传学思想的自适应启发式算法, 是进化算法中最常使用的算法之一, 可应用于网络优化[18]、动态优化[19-20] 等领域. 标准遗传算法的主要算子有种群初始化、选择、交叉、变异等. 基于 GA 的 BNSL 方法一般整体结构与混合方法相同, 先使用 CB 方法限制搜索空间后, 再利用 SS 方法搜索最佳结构. 种群初始化时生成N个可能的贝叶斯网络结构个体, 并通过对个体与标准网络的拟合程度进行适应度评估; 根据适应度选择合适的个体进入下一代; 基于模拟生物遗传中染色体交换的过程, 选择两个个体交叉产生后代; 变异过程是通过一定随机概率修改个体的部分基因来引入新的变化; 多次迭代后, 选择适应度最高的个体为最终的贝叶斯网络.

GA 既保留了自然选择中的随机性, 同时, 也利用了种群中的历史信息将搜索引导至搜索空间内性能更好的区域. 因此, GA 可有效地学习贝叶斯网络结构问题, 其既能够高效地在空间内搜索更多可能的结构组合, 又可以利用种群信息、专家知识、先验知识等信息引导搜索过程. Dai 等[5] 提出了 MIIGA 方法, 使用 MI 在 CB 阶段限制了搜索空间, 并在 SS 阶段令子代从父代中继承优秀的结构; Contaldi 等[21] 提出的 AESL-GA 使用知识驱动从父代开始对搜索空间动态调整; Yan 等[6] 提出了 MIGA, 在初始化种群利用 MI 生成初始种群, 并在交叉节点由 MI 引导决定子代结构.

1.3　互信息

在信息论中, MI 可以衡量两个变量间共享的信息量. 两个节点X与Y间的互信息可定义为

$$\mathrm{MI}(X,Y)=\sum_{x,y}P(x,y)\ln\frac{P(x,y)}{P(x)P(y)}. \tag{1}$$

其中: $P(x)$和$P(y)$分别为X和Y事件发生的概率, 若X和Y完全独立, 则$P(x,y)=P(x)P(y)$, 互信息为零. 互信息越大, X与Y间的相互依赖性越强. 互信息在图像处理[22]、神经网络学习[23]、机器学习[24-26]、数据挖掘[27-28]、信号处理[29] 等领域有较为广泛的应用.

2　本文方法

为解决遗传算法在学习贝叶斯网络结构问题中过早收敛的问题, 本文提出一种基于种群多样性和互信息混合引导的贝叶斯网络结构学习算法 DM-GABN. 种群多样性有助于种群快速适应环境的变化, 并帮助算法实现充分的全局探索, 避免算法陷入局部最优, 所提出算法做出如下改进.

在传统的有向图去环策略中, 通常使用删除边的方式破坏环路, 但是, 删除操作会将可能正确的结构排除在外, 导致种群的基因型数量减少. 因此, DM-GABN 方法提出了一种利用 MI 引导的翻转-删除-修复的去环策略, 尽可能保留更多的基因型, 避免去环操作对个体产生较大的影响, 同时提升种群多样性.

一些启发式算法的研究表明, 算法的性能对于种群中个体最大年龄较为敏感[30-33], 因此, DM-GABN 方法提出了一种基于动态寿命机制的二次选择算子, 根据种群多样性调节寿命阈值, 淘汰衰老个体, 空出更多种群空间用于生成新个体, 以此提高种群的多样性.

遗传算法中, 交叉策略对于种群多样性影响较大, DM-GABN 方法引入了生物中种群基因型频率的概念, 用于衡量种群多样性[34]. 为了将种群基因型频率维持在一个合理的范围区间, 本文设计一种基于 MI 与种群基因型频率共同指导的交叉算子, 该方法在父代结构产生冲突时, 将基因型频率较低的结构保留至下一代. 此外, 算法中设计了一种交叉保留概率, 使得 MI 引导算法将可能正确的结构的保留概率在一个合理的范围内. 算法的流程如图 1 所示.

2.1　种群基因型频率和种群互信息

在 DM-GABN 算法的多个算子中, 均使用了种群基因型频率和种群互信息, 对此将给出它们的定义.

算法使用邻接矩阵对贝叶斯网络结构编码, 将个体中的一个边结构视为一个基因, 种群基因型频

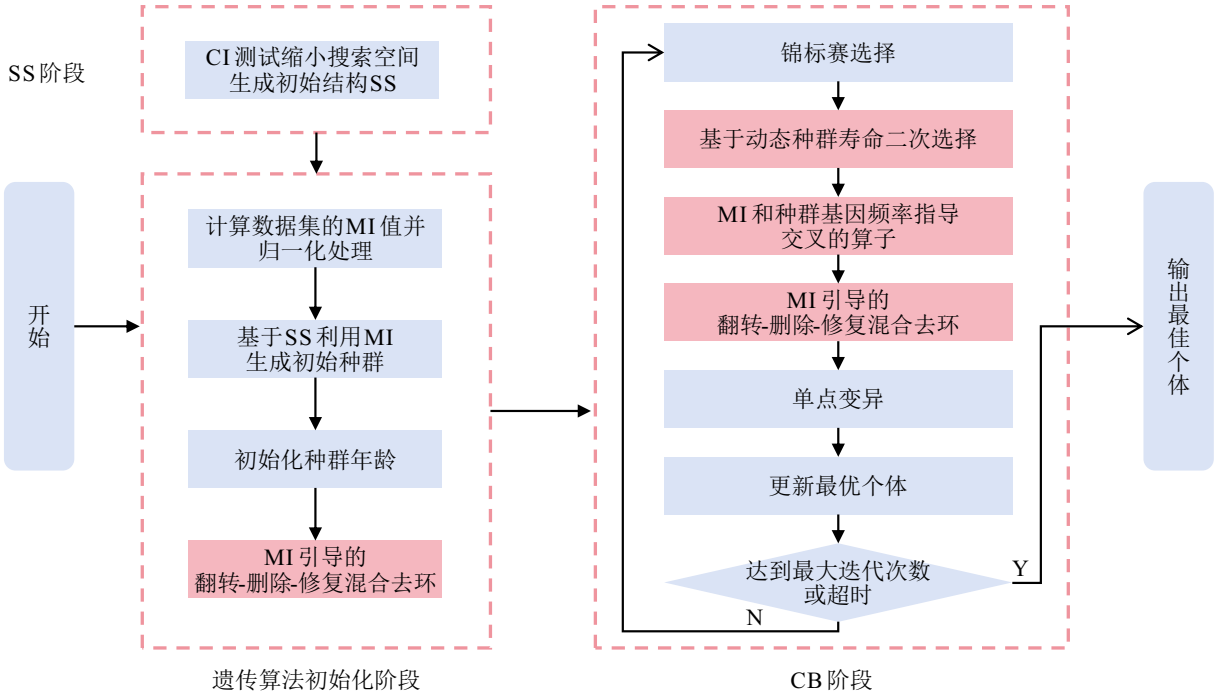

图1 DM-GABN 算法流程

率即为某个边结构在种群所有个体中出现的次数. 如N个大小的种群中, 有n个个体包含边A-B, 则结构A-B的种群基因型频率为$F = n/N$.

互信息值表示两个节点间的依赖关系, 若互信息值高, 则意味着在贝叶斯网络结构中, 两个节点间有依赖关系的可能性较高; 反之, 则两个节点间有依赖关系的可能性较低. 为消除互信息特征间的尺度差异, 再依次计算搜索空间中每对节点间的互信息值后, 对互信息进行如下归一化处理, 种群互信息取值范围经过归一化后为 (0, 1), 使用其作为算法中节点关系参考因素之一:

$$\mathrm{MI}_{\mathrm{norm}}(X,Y) = \max\left\{\frac{\mathrm{MI}(i,j)}{\mathrm{MI}_{\max}(i)}, \frac{\mathrm{MI}(i,j)}{\mathrm{MI}_{\max}(j)}\right\}. \quad (2)$$

2.2 基于 MI 引导的翻转-删除-修复混合去环策略

在算法的初始化、变异、交叉等环节中, 算法对个体边结构修改时, 出于算法运行效率考量并未考虑是否会生成环路, 需要对网络中的环路进行处理, 使其符合贝叶斯网络结构有向无环的要求. 检测到环路后, 传统方法是随机删除环路中的一条边, 或是按照节点顺序和搜索顺序删除边. 环路出现的情况是没有规律的、随机的, 这样的方法可能会将正确的结构排除, 同时, 也减少了种群中的基因类型. 为了最低程度地降低该算子对于个体的评分影响, 本文提出了一种基于互信息引导的翻转-删除-修复混合去环策略.

所提出策略分为 3 个阶段: 第 1 阶段是找出当前个体网络结构中的所有环路, 使用类似于拓扑排序的方式, 循环删除所有入度为 0 的节点和与之相关的边, 得到的最终队列即为当前结构的所有环路. 第 2 阶段会对环路中所有边的互信息进行升序排序, 若最小的边互信息小于阈值α, 则认为两个节点间有依赖关系的可能性较小, 将删除该边; 若互信息最低的边大于阈值α, 则保留该边将其翻转.

在一些复杂环路中, 进行一次翻转的操作可能并不能完全地去除环路, 如图 2 所示. 环路中的每条边均有较高的 MI 值, 因此, 首先选取 MI 值最低的边$B \to E$进行翻转时, 该操作虽然消除了节点B、C、E间的环路, 但是, 此时B、D、E间形成了新的环路, 且节点B、D、E、C间的环路一直存在, 若选择翻转 MI 值排序列表中的第 2 条边$B \to D$, 则会得到一个无环的网络结构.

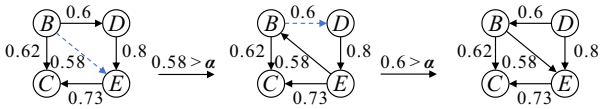

图2 基于 MI 引导的翻转和删除去环方法示例

简单的环路有限次翻转操作即可去除环路, 复杂环路翻转操作可能会形成新的环路, 需要多次操作甚至仅靠翻转无法去除环路. 为减少无效的操作和算法复杂度, DM-GABN 对每个个体的翻转次数设定了上限, 上限值由环路边的数量决定. 当超出环路翻转次数上限时, 算法将恢复个体的初始结构, 对环路中所有边的 MI 值进行排序, 并从 MI 值最小的边开始删除, 直至不存在环. 对于每个个体, 循环执行寻找环路和去除环路两个操作, 直至该个体中不

存在有效环.

在前两个阶段, 删除操作可能会破坏局部结构的连贯性, 导致较好的局部结构被破坏, 因此, 在第3阶段将对局部结构进行修复, 尝试重新连接MI值较高的边恢复关键的依赖关系. 每次记录删除的边, 当个体中不再存在环路时, 将所有被删除边和与之相连的边按照MI值从高到底排序, 并逐一添加, 若添加后没有形成环路, 则将其保留; 否则, 删除.

经过以上混合策略, 可以最大程度地保留优秀的结构, 避免去环操作对个体造成较大影响.

2.3　基于动态寿命机制的二次选择算子

一些研究表明, 淘汰衰老的个体增加了种群多样性[30-33], 有助于避免启发式算法陷于局部最优的情况发生. 为提高种群的多样性, 利用个体寿命在选择算子中淘汰衰老个体, 本文提出了一种基于寿命机制的二次选择策略. 该策略共有两次选择过程. 首先为初始化种群个体年龄为1, 每次迭代种群个体年龄设置加1, 设定最大寿命为T. 第1次选择使用了锦标赛选择方法, 从$2N$的种群中通过比较个体适应度选出N个个体; 第2次选择对N个个体进行寿命的筛选, 淘汰掉年龄超过T的个体, 此时种群大小由N变为N^*.

最大寿命年龄越小, 搜索过程越贴近随机搜索, 搜索策略将会无效. 过高的最大寿命由于淘汰的个体较少, 限制了策略的有效性. 若个体的最大寿命为T, 该个体在遗传算法过程中已经产生了$T \sim 2T$个后代, 其各结构的基因已充分地传递给后代, 将其淘汰后, 则新个体有较大概率为种群贡献更多样的基因. 因此, 本文设计了一种通过种群多样性动态调整寿命的方法. 根据种群基因型频率判断种群多样性, 将每条边的种群基因型频率P按照种群大小N归一化为F_i, 所有边的F_i构成一个向量$F = [F_1, F_2, \ldots, F_M]$. 通过向量的方差来表示种群的多样性高低, 若趋近于0, 则表示所有边的基因型频率趋近于均值, 此时种群同质化较高, 即多样性较低; 若方差趋近于最大值, 则表示边的基因型频率值差异较大, 如某些边高频出现, 某些边则低频出现, 象征着种群多样性较高, 计算公式如下所示:

$$\sigma^2 = \frac{1}{M}\sum_{i=1}^{M}(F_i - \mu)^2, \tag{3}$$

$$\mu = \frac{1}{M}\sum_{i=1}^{M}F_i. \tag{4}$$

其中: u为F_i的平均值, M为搜索空间内边的数量. 在极端情况下, 种群中有一半边的$F_i = 1$, 一半边的$F_i = 0$, 则此时种群多样性达到最极值, 即0.25; 相反, 最小值为0. 根据该极值, 本文设计了一种动态调整寿命T的策略, 计算方式如下:

$$T = T_{\min} + (T_{\max} - T_{\min}) \times \frac{\sigma^2 - \sigma^2_{\min}}{\sigma^2_{\max} = \sigma^2_{\min}}. \tag{5}$$

本文将最小寿命淘汰值$T_{\min}$定为2, 最大寿命淘汰值$T_{\max}$为5. 若种群多样性较高, 则动态降低寿命T; 若种群多样性较低, 则动态提高T, 加速淘汰衰老个体, 通过控制种群寿命, 可将种群的年龄结构维持在一个合理的水平. 经过改进的二次选择算子, 充分利用了种群当前的多样性信息, 使得算法充分探索了搜索空间, 避免算法陷入局部最优.

2.4　基于种群基因型频率和种群互信息值的交叉策略

交叉策略对种群多样性起着重要的作用, 在个体均较为相似的种群中, 交叉生成的子代也与父代较为接近, 因此, 如何通过交叉策略产生更多样的子代是研究的重点. DM-GABN提出了一种基于种群基因型频率和MI的交叉策略, 分为选择父代和父代交叉两个过程.

在选择父代时, 为种群的每个个体随机匹配一个个体进行交叉, 此时种群的规模将从N^*变为$2N^*$. 若二次选择时淘汰了个体, 则随机抽取个体进行交叉, 直至当前种群个体数量为$2N$. 在父代交叉过程时, 新个体的结构继承两个父代相同的结构, 当父代结构表现冲突时, 即某个结构仅有一个父代存在, 或是两个父代均存在但是方向不一致时, 将由种群基因型频率和MI值决定该结构是否保留至子代. 当该结构的种群基因型频率$F_{\min}$小于设定最小值的10%时, 则将该结构保留至子代, 这样可以避免某些基因型在迭代的过程被淘汰, 导致种群的多样性降低; 反之, 则计算该结构的交叉保留概率P_{cross}. 当大于随机数rand时, 则保留该结构至子代; 否则, 删除. 尽管MI可以在一定程度上表示两个节点间的依赖关系, 可依此推测某个结构存在的可能性, 但是MI的引导性并不完全准确, 因此, 在交叉保留概率中设置了一定的随机性, 避免完全依赖MI造成错误判断的情况, 公式如下所示:

$$P_{\text{cross}}(X_i, X_j) = 0.25 + 0.5 \times \text{MI}(X_i, X_j). \tag{6}$$

经过缩放, 保留概率的范围为(0.25, 0.75). 这样, MI较高的结构也有概率在遗传过程中被排除, MI较低的结构也有概率被保留, 图3为两个父代结构的交叉过程示例.

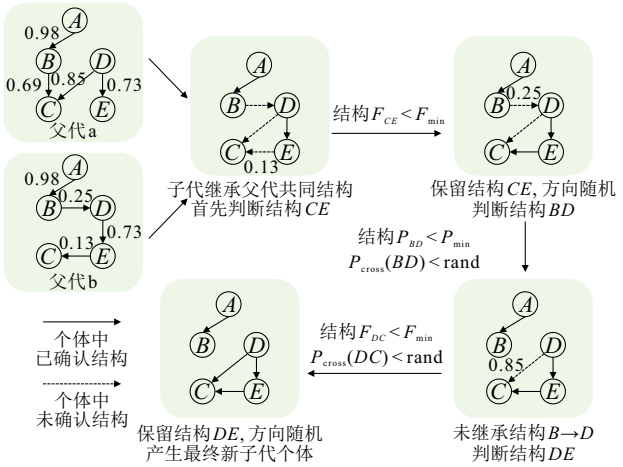

图3 基于种群基因型频率和种群互信息值的交叉方法示例

子代首先继承两个父代相同的结构. 在两个父代中, 结构$B \to D$、$E \to C$和$D \to E$存在冲突, 对其依次进行判断. 首先判断结构$E \to C$的种群基因型频率是否小于$F_{\min}$, 若小于$F_{\min}$, 则保留该结构并赋予随机方向; 再判断结构$B \to D$的种群基因型频率, 若大于$F_{\min}$, 则利用P_{cross}判断该结构是否保留. 若$P_{\text{cross}} > \text{rand}$, 则保留该结构并赋予随机方向, 至此生成一个新的子代.

2.5 算法时间复杂度

本节根据 DM-GABN 算法的伪代码分析算法的时间复杂度.

算法 1 DM-GABN 算法伪代码.

step 1: 初始化搜索空间.

step 2: 使用 CI 测试缩小搜索空间得到初始结构SS.

step 3: 根据数据集计算搜索空间的种群互信息MI.

step 4: 使用MI引导SS生成初代种群, 初始化年龄.

step 5: 利用 MI 引导, 通过翻转-删除-修复边策略去除个体环路.

step 6: 开始M次迭代.

step 7: 锦标赛选择出适应度更高的个体.

step 8: 计算当前最大寿命T, 淘汰寿命超过T的个体.

step 9: 计算当前种群的基因型频率.

step 10: 基于种群基因型频率和MI指导交叉过程生成后代.

step 11: 更新种群年龄.

step 12: 每个个体单点变异.

step 13: 利用 MI 引导, 通过翻转-删除-修复边策略去除个体环路.

step 14: 评估种群中每个个体的适应度值.

step 15: 比较最优个体, 更新种群最优个体g_{best}.

step 16: 判断当前是否超出运行时间或达到最大迭代次数M, 若为假, 则转至 step 7; 否则, 输出最优个体g_{best}.

假设网络有n个节点, m条边, 种群大小为N, 遗传算法迭代次数为M. 算法各阶段的时间复杂度如下: 初始搜索空间时, CI 测试的时间复杂度为$O(n\log_2 n)$. 去环算子在第 1 阶段对每个个体的每个节点做遍历, 时间复杂度为$O(Nn)$, 第 3 阶段最坏的情况下对每条边做遍历, 故总时间复杂度为$O(Nn+Nm)$. 选择算子中锦标赛选择时间复杂度为$O(N\log_2 N)$, 二次选择对每个个体遍历一次, 总的时间复杂度为$O(N\log_2 N+N)$; 交叉算子需要判断个体的每条边, 时间复杂度为$O(Nm)$; 单点变异的时间复杂度为$O(Nm)$; 计算个体适应度的时间复杂度为$O(Nn^2)$; 综上, 单次迭代的时间复杂度为$O(Nn^2+Nm)$. DM-GABN 算法的时间复杂度为$O(n^2+m)\times N^2$.

3 实验结果与分析

3.1 实验设置

所提出算法使用 Matlab 和 Bayes Net Toolbox for Matlab 进行实验. 实验过程中使用的两台服务器如下: 1) 服务器 1: 两颗英特尔至强 E5-2 640 v4 处理器 (10 核心, 主频 2.4 GHz), 内存 160 GB, 操作系统为 Ubuntu 21.04; 2) 服务器 2: 两颗英特尔至强金牌 5218R 处理器 (20 核心, 主频 2.1 GHz), 内存 512 GB, 操作系统为 CentOS7. 本文对比算法选取了 hybrid-SLA[35]、SIGA[36]、AESL-GA[21]、EKGA-BN、MIGA[8] 五种基于遗传算法的 BNSL 方法, MMHC[16]、Inter-IAMB[12] 两种混合 BNSL 方法、以及精确算法 GOBNILP[37] 和 SaiyanH[17].

基本参数设置为种群数量$N=100$, 最大迭代次数$M=100$, 锦标赛算子$K=4$.

为了验证所提出 DM-GABN 算法的有效性, 选取了小型、中型、大型和超大型网络的 10 个模型, ANDE 模型使用 100、200、500 的规模, 其余均使用 500、1 000、2 000、5 000 的规模.

3.2 评价指标介绍

为了评估算法的收敛性以及所学习到网络结构的准确性, 本文使用以下评价指标对算法结果进行比较.

1) BIC 评分: 选取算法最终输出的最优解的 BIC

评分作为最终比较分数. BIC 分数越大, 所评价网络结构与标准网络结构拟合程度越高, 但是, 并不能代表其是更准确的网络结构. BIC 公式如下所示:

$$S_{\mathrm{BIC}} = S_{\mathrm{LL}}(G, D) - \frac{\log N}{2} \times F, \tag{7}$$

$$S_{\mathrm{LL}} = (G, D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \log \frac{N_{ijk}}{N_{ij}}, \tag{8}$$

$$F = \sum_{i=1}^{n} (r_i - 1) q_i. \tag{9}$$

其中: N为样本量; S_{LL}为对数似然函数值, 描述图G对数据D的拟合程度; 式 (7) 中第 2 项F是一个描述模型复杂程度的惩罚项.

2) F_1评分: F_1分数越高, 所评价网络结构越准确, 其公式如下所示:

$$F_1 = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}. \tag{10}$$

其中: precision 为所评价网络结构中正确的有向边在网络中的比重, recall 为所评价网络中正确的有向边在标准网络结构中的比重.

3) 汉明距离 (Hamming distance, HD): 用于表示所评价网络结构与标准网络结构的差异边数, 其值越小, 所评价网络结构与标准结构差距越小.

3.3　算子有效性测试

在 Sachs、Insrance、Water 三个数据集上进行了消融实验. 首先使用 MIGA 算法作为基准算法, 然后在基准算法上替换基于 MI 引导的翻转-删除-修复混合去环算子, 接着替换基于动态寿命机制的二次选择算子, 最后用基于种群基因型频率和种群互信息值的交叉算子替换原算法算子, 形成了所提出 DM-GABN 算法, 表 1 给出了结果的平均值和标准差值.

表1　DM-GABN 算法消融实验结果

数据集	Sachs-1000			Insrance-1000			Water-1000		
	F_1 Score	HD	BIC Score	F_1 Score	HD	BIC Score	F_1 Score	HD	BIC Score
Baseline	0.915 (0.043)	4.13 (1.64)	−7670.54 (55.89)	0.757 (0.029)	22.25 (2.67)	−16152.15 (198.49)	0.457 (0.018)	47.68 (1.18)	−15402.90 (86.88)
Baseline + 去环算子	0.918 (0.041)	4.03 (1.45)	−7671.02 (56.49)	0.771 (0.029)	20.9 (2.44)	−16098.48 (131.59)	0.465 (0.011)	46.98 (0.73)	−15402.85 (88.72)
Baseline + 去环算子 + 选择算子	0.919 (0.041)	**3.92** **(1.35)**	−7670.90 (56.42)	0.772 (0.041)	20.52 (2.50)	−16082.665 (86.63)	0.465 (0.015)	46.98 (0.90)	−15402.85 (82.07)
DM_GABN	**0.920** **(0.040)**	3.98 (1.37)	**−7670.68** **(56.69)**	**0.779** **(0.026)**	**19.98** **(2.17)**	**−16056.34** **(113.39)**	**0.468** **(0.013)**	**46.77** **(0.78)**	**−15399.09** **(87.86)**

在 3 个数据集上, DM-GABN 均获得了最佳的F_1分数、HD 和 BIC 分数, 表明其搜索到了更为准确的结构, 同时也是与标准网络差距最小、拟合程度最高的结构. 去环算子在 Insrance 数据集有所降低, 可能是该方法得到的结构更为复杂, 影响了效果, 但是去环算子保留了更多的结构, 提升了这些结构的种群基因型频率, 为交叉算子做出更好的铺垫, 最终 DM-GABN 搜索到了最佳的结构. 从标准差也可以看出, 3 种算子同时提升了算法的稳定性. 综上, 验证了 DM-GABN 所提出算子的有效性.

3.4　算法性能测试和分析

第 3.3 节的消融实验验证了所提出 3 个算子的有效性, 为了深入探究 DM-GABN 在性能上的表现, 本节在 10 个数据集上, 将 DM-GABN 与其余 10 个对比算法进行性能上的比较, 实验结果的F_1评分和 HD 分别如表 2 和表 3 所示, 其中最优的结果由加粗字体表示. 基于遗传算法的 BNSL 方法可以在每代输出当前的最佳结果, 据此制作了收敛图用于评价这些方法的收敛性能.

F_1分数更注重于对边的预测的准确性, HD 更关注于整体结构间的差异, 因此, 算法在这二者的表现上有所差异.

在小型数据集 Asia 和 Sachs 上, 由于搜索空间较小, 基于 GA 的 BNSL 方法均能够获得较好的结果, 在F_1和 HD 两个指标上没有较大的差距, 且搜索到的结果相比于其他方法均更为接近正确结构.

在中型数据集上, DM-GABN 已经开始展现优势, 在大多数数据集上均获得了最优的结果, 其中在 Alarm、Insrance、Water 上表现最佳. 在 Alarm 数据集上, DM-GABN、MIGA 和 GOBNILP 算法均可获得较优的结果. 在 Barley 数据集上, GOBNILP 算法获得了最佳的结果, DM-GABN 与 MIGA 虽然获得了优于其他方法的结果, 但是相比于 GOBNILP 仍然有较大的距离. Barley 数据集相比于其他数据集, 参数量明显提升, GOBNILP 算法在这样的数据集上

表2 算法综合性能在F_1上的对比实验

数据集	DM-GABN	MIGA	SIGA	AESL-GA	EKGA-BN	hybrid-SLA	PSX	MMHC	Inter-IAMB	SaiyanH	GOBNILP
AS-1000	**0.907**	**0.907**	0.906	0.896	0.899	0.906	**0.907**	0.682	0.471	0.756	0.742
	(0.071)	**(0.071)**	(0.071)	(0.082)	(0.076)	(0.071)	**(0.071)**	(0.107)	(0.081)	(0.095)	(0.126)
AS-2000	**0.930**	**0.930**	**0.930**	**0.930**	**0.930**	0.927	0.927	0.768	0.459	0.778	0.812
	(0.021)	**(0.021)**	**(0.021)**	**(0.021)**	**(0.021)**	(0.025)	(0.025)	(0.103)	(0.049)	(0.086)	(0.081)
AS-5000	0.940	0.940	0.940	0.920	0.940	0.931	**0.941**	0.806	0.470	0.733	0.826
	(0.033)	(0.033)	(0.033)	(0.072)	(0.033)	(0.056)	**(0.033)**	(0.083)	(0.068)	(0.118)	(0.122)
AL-1000	**0.902**	0.847	0.689	0.598	0.727	0.573	0.466	0.698	0.670	0.775	0.844
	(0.031)	(0.059)	(0.069)	(0.070)	(0.050)	(0.053)	(0.043)	(0.106)	(0.037)	(0.031)	(0.029)
AL-2000	**0.927**	0.898	0.730	0.664	0.781	0.613	0.531	0.670	0.701	0.823	0.872
	(0.018)	(0.037)	(0.071)	(0.056)	(0.052)	(0.060)	(0.052)	(0.085)	(0.026)	(0.051)	(0.014)
AL-5000	**0.947**	0.917	0.718	0.631	0.811	0.579	0.474	0.755	0.680	0.851	0.883
	(0.024)	(0.035)	(0.065)	(0.050)	(0.087)	(0.048)	(0.052)	(0.113)	(0.030)	(0.036)	(0.032)
BA-1000	0.414	0.400	0.299	0.169	0.322	0.230	0.164	0.365	0.246	运算	**0.552**
	(0.020)	(0.029)	(0.027)	(0.049)	(0.020)	(0.029)	(0.049)	(0.020)	(0.029)	超时	**(0.029)**
BA-2000	0.504	0.500	0.371	0.203	0.405	0.306	0.208	0.425	0.240	运算	**0.626**
	(0.018)	(0.019)	(0.030)	(0.033)	(0.044)	(0.037)	(0.042)	(0.072)	(0.021)	超时	**(0.020)**
BA-5000	0.648	0.620	0.428	0.234	0.478	0.349	0.246	0.624	0.230	运算	**0.706**
	(0.025)	(0.029)	(0.022)	(0.044)	(0.043)	(0.042)	(0.051)	(0.076)	(0.016)	超时	**(0.018)**
IN-1000	0.777	0.757	**0.791**	0.577	0.647	0.649	0.552	0.659	0.374	0.509	0.755
	(0.027)	(0.029)	**(0.052)**	(0.049)	(0.045)	(0.059)	(0.062)	(0.080)	(0.030)	(0.061)	(0.018)
IN-2000	**0.848**	0.825	0.750	0.635	0.703	0.654	0.585	0.710	0.420	0.604	0.791
	(0.018)	(0.026)	(0.056)	(0.057)	(0.055)	(0.060)	(0.062)	(0.060)	(0.026)	(0.066)	(0.030)
IN-5000	**0.888**	0.864	0.793	0.657	0.766	0.706	0.571	0.754	0.453	0.618	0.812
	(0.012)	(0.032)	(0.051)	(0.048)	(0.071)	(0.049)	(0.049)	(0.065)	(0.031)	(0.059)	(0.025)
WA-1000	**0.466**	0.457	0.441	0.424	0.416	0.435	0.422	0.418	0.332	0.358	0.414
	(0.014)	(0.018)	(0.023)	(0.027)	(0.036)	(0.025)	(0.029)	(0.060)	(0.028)	(0.036)	(0.034)
WA-2000	**0.518**	0.507	0.499	0.465	0.454	0.482	0.458	0.459	0.372	0.403	0.485
	(0.016)	(0.018)	(0.791)	(0.040)	(0.028)	(0.023)	(0.033)	(0.056)	(0.031)	(0.047)	(0.048)
WA-5000	**0.544**	0.542	0.536	0.493	0.510	0.517	0.449	0.544	0.373	0.431	0.518
	(0.018)	(0.018)	(0.026)	(0.035)	(0.031)	(0.021)	(0.041)	(0.054)	(0.018)	(0.055)	(0.045)
HA-1000	**0.697**	0.683	0.556	0.397	0.575	0.497	0.409	0.603	0.252	0.378	0.530
	(0.039)	(0.042)	(0.059)	(0.046)	(0.058)	(0.051)	(0.049)	(0.06)	(0.024)	(0.040)	(0.042)
HA-2000	**0.741**	0.709	0.589	0.403	0.606	0.483	0.400	0.685	0.254	0.451	0.615
	(0.042)	(0.043)	(0.041)	(0.056)	(0.056)	(0.051)	(0.036)	(0.062)	(0.022)	(0.034)	(0.042)
HA-5000	**0.782**	0.726	0.602	0.428	0.630	0.495	0.442	0.697	0.321	0.452	0.647
	(0.034)	(0.045)	(0.054)	(0.042)	(0.051)	(0.042)	(0.047)	(0.055)	(0.015)	(0.044)	(0.059)
HE-1000	**0.574**	0.562	0.529	0.452	0.510	0.518	0.431	0.396	0.167	0.392	0.399
	(0.029)	(0.023)	(0.020)	(0.027)	(0.033)	(0.020)	(0.034)	(0.041)	(0.015)	(0.050)	(0.033)
HE-2000	**0.665**	0.658	0.608	0.495	0.610	0.571	0.476	0.473	0.165	0.473	0.507
	(0.013)	(0.015)	(0.023)	(0.034)	(0.023)	(0.033)	(0.029)	(0.037)	(0.015)	(0.036)	(0.023)
HE-5000	**0.749**	0.742	0.682	0.486	0.688	0.587	0.480	0.575	0.170	0.578	0.593
	(0.011)	(0.012)	(0.020)	(0.031)	(0.023)	(0.019)	(0.040)	(0.050)	(0.015)	(0.061)	(0.019)
WIN-1000	**0.739**	0.699	0.521	0.458	0.576	0.468	0.419	0.422	0.369	0.396	运算
	(0.016)	(0.039)	(0.056)	(0.042)	(0.051)	(0.044)	(0.032)	(0.059)	(0.019)	(0.032)	超时
WIN-2000	**0.771**	0.734	0.533	0.459	0.597	0.467	0.404	0.472	0.362	0.465	运算
	(0.024)	(0.029)	(0.027)	(0.055)	(0.055)	(0.043)	(0.034)	(0.048)	(0.020)	(0.045)	超时
WIN-5000	**0.800**	0.738	0.506	0.430	0.605	0.439	0.371	0.518	0.361	0.588	运算
	(0.777)	(0.039)	(0.035)	(0.037)	(0.046)	(0.033)	(0.032)	(0.055)	(0.024)	(0.055)	超时
PA-1000	**0.350**	0.250	0.044	0.044	0.123	0.054	0.063	0.232	0.123	0.199	运算
	(0.027)	(0.032)	(0.013)	(0.013)	(0.023)	(0.016)	(0.016)	(0.022)	(0.009)	(0.017)	超时
PA-2000	**0.409**	0.318	0.042	0.042	0.097	0.059	0.064	0.319	0.132	0.227	运算
	(0.045)	(0.029)	(0.015)	(0.015)	(0.017)	(0.013)	(0.016)	(0.016)	(0.009)	(0.018)	超时
PA-5000	**0.503**	0.432	0.405	0.062	0.110	0.066	0.066	0.431	0.151	0.227	运算
	(0.040)	(0.031)	(0.013)	(0.013)	(0.015)	(0.015)	(0.015)	(0.013)	(0.013)	(0.019)	超时
AN-100	**0.459**	0.451	0.405	0.360	0.403	0.376	0.346	0.322	0.174	运算	运算
	(0.019)	(0.022)	(0.019)	(0.023)	(0.023)	(0.020)	(0.023)	(0.029)	(0.009)	超时	超时
AN-200	**0.561**	0.559	0.503	0.414	0.510	0.456	0.405	0.446	0.272	运算	运算
	(0.017)	(0.017)	(0.021)	(0.020)	(0.023)	(0.017)	(0.021)	(0.039)	(0.015)	超时	超时
AN-500	**0.666**	0.659	0.553	0.444	0.582	0.495	0.420	0.576	0.370	运算	运算
	(0.013)	(0.010)	(0.015)	(0.018)	(0.022)	(0.022)	(0.028)	(0.020)	(0.011)	超时	超时

表3　算法综合性能在 HD 上的对比实验

数据集	DM-GABN	MIGA	SIGA	AESL-GA	EKGA-BN	hybrid-SLA	PSX	MMHC	Inter-IAMB	SaiyanH	GOBNILP
AS-1000	1.75	1.75	1.78	1.90	1.88	1.78	**1.70**	4.95	5.88	3.85	4.10
	(1.08)	(1.08)	(1.07)	(1.27)	(1.13)	(1.07)	**(1.07)**	(1.80)	(0.63)	(1.59)	(2.23)
AS-2000	1.40	1.40	1.40	**1.38**	1.5	**1.38**	1.43	3.60	5.88	3.55	2.95
	(0.37)	(0.37)	(0.37)	**(0.38)**	(0.59)	**(0.38)**	(0.46)	(1.69)	(0.38)	(1.40)	(1.32)
AS-5000	1.25	1.25	1.25	1.58	1.25	1.43	**1.18**	3.05	5.78	4.25	2.80
	(0.54)	(0.54)	(0.54))	(1.27)	(0.54)	(0.86)	(0.58)	(1.36)	(0.54)	(1.89)	(2.09)
AL-1000	**10.58**	16.28	29.98	37.15	26.08	40.13	49.65	28.00	30.18	19.85	15.15
	(3.23)	(5.62)	(5.82)	(5.85)	(4.21)	(4.13)	(4.08)	(9.92)	(3.21)	(2.69)	(2.89)
AL-2000	**8.25**	12.25	28.25	33.10	23.08	39.15	47.52	30.80	27.175	15.90	12.4
	(1.70)	(3.64)	(6.53)	(5.18)	(4.66)	(6.10)	(5.59)	(7.85)	(2.22)	(4.65)	(1.36)
AL-5000	**6.30**	10.60	31.75	37.73	21.83	46.33	57.45	22.90	28.10	13.50	11.10
	(2.63)	(3.52)	(6.68)	(5.00)	(8.42)	(5.26)	(6.81)	(10.90)	(1.75)	(3.25)	(3.13)
BA-1000	66.75	68.30	76.425	94.475	74.70	84.53	100.05	82.90	116.75	运算	**60.10**
	(2.34)	(2.75)	(2.46)	(4.31)	(3.85)	(3.42)	(4.17)	(8.80)	(3.54)	超时	**(3.59)**
BA-2000	70.33	69.80	73.30	106.58	74.70	104.53	116.38	83.55	117.45	运算	**61.00**
	(1.85)	(2.48)	(2.91)	(4.97)	(3.31)	(6.07)	(4.20)	(9.04)	(3.85)	超时	**(43.15)**
BA-5000	47.33	51.08	72.00	94.50	67.05	82.43	95.65	54.35	118.00	运算	**43.15**
	(3.04)	(3.45)	(2.84)	(4.60)	(4.83)	(5.02)	(6.61)	(10.95)	(2.80)	超时	**(2.63)**
IN-1000	**20.30**	22.25	27.98	36.93	31.58	31.68	39.53	30.00	60.38	44.00	22.5
	(2.28)	(2.65)	(4.43)	(4.12)	(3.80)	(5.12)	(5.06)	(6.92)	(3.80)	(6.05)	(1.57)
IN-2000	**14.85**	17.20	24.13	33.08	28.10	32.53	39.55	26.50	49.90	36.05	19.63
	(1.66)	(2.62)	(4.96)	(4.98)	(5.01)	(5.28)	(5.63)	(5.54)	(2.27)	(6.26)	(2.77)
IN-5000	**11.50**	14.525	21.35	32.83	24.00	29.75	43.00	23.30	45.50	35.45	18.20
	(1.14)	(3.79)	(5.10)	(4.61)	(7.04)	(4.89)	(4.52)	(6.23)	(2.81)	(5.63)	(2.32)
WA-1000	**46.93**	47.68	49.05	50.35	50.93	49.48	51.13	55.40	66.58	65.25	64.05
	(0.88)	(1.18)	(1.70)	(2.16)	(2.91)	(2.00)	(2.71)	(5.87)	(3.83)	(4.39)	(3.72)
WA-2000	**44.30**	45.20	46.05	48.65	49.48	47.80	49.80	52.35	55.60	59.85	58.15
	(1.62)	(1.62)	(1.85)	(3.12)	(2.66)	(2.16)	(2.79)	(5.51)	(3.06)	(5.08)	(5.20)
WA-5000	**42.83**	43.08	43.65	47.13	45.75	45.53	52.98	45.55	54.90	57.45	55.1
	(1.49)	(1.49)	(2.18)	(3.03)	(2.61)	(1.74)	(3.76)	(5.49)	(1.54)	(5.49)	(4.74)
HA-1000	**36.925**	38.95	36.93	71.13	51.13	60.80	70.13	49.05	128.35	76.30	59.85
	(4.44)	(4.84)	(6.67)	(5.16)	(6.29)	(6.06)	(5.91)	(7.19)	(5.82)	(4.81)	(5.55)
HA-2000	**31.98**	36.18	52.95	75.75	48.83	66.78	76.55	40.15	126.58	67.35	50.10
	(5.08)	(4.79)	(4.88)	(8.15)	(6.15)	(6.11)	(5.95)	(5.21)	(6.51)	(3.89)	(5.47)
HA-5000	**27.33**	34.58	53.16	76.30	47.63	70.43	77.05	39.55	105.73	67.65	46.50
	(4.23)	(5.51)	(7.51)	(6.48)	(6.28)	(5.77)	(8.28)	(7.35)	(2.83)	(5.12)	(7.81)
HE-1000	**78.60**	80.45	86.08	96.90	89.30	87.18	100.18	107.20	112.35	121.45	113.00
	(4.78)	(3.38)	(3.65)	(4.05)	(5.12)	(3.09)	(5.44)	(7.28)	(0.94)	(24.04)	(6.66)
HE-2000	**65.15**	66.13	75.05	94.60	74.68	81.43	98.90	98.15	112.35	103.25	94.85
	(2.35)	(2.80)	(3.65)	(5.91)	(4.38)	(5.76)	(4.94)	(6.66)	(1.05)	(6.87)	(4.42)
HE-5000	**51.23**	53.08	65.88	103.475	64.25	85.65	107.83	83.40	111.93	86.80	80.75
	(1.95)	(2.14)	(4.52)	(7.06)	(4.37)	(4.44)	(8.53)	(9.71)	(1.165)	(23.99)	(3.59)
WIN-1000	**54.00**	61.39	98.95	108.60	85.63	112.55	125.10	122.95	91.25	153.6	运算
	(5.67)	(7.63)	(10.75)	(8.28)	(9.96)	(9.41)	(7.52)	(13.31)	(2.44)	(10.34)	超时
WIN-2000	**49.12**	56.93	104.23	112.28	84.93	122.90	136.73	115.70	91.45	133.45	运算
	(5.35)	(6.37)	(5.94)	(10.73)	(12.08)	(10.35)	(8.64)	(11.55)	(1.94)	(17.10)	超时
WIN-5000	**44.90**	58.8	120.78	121.3	87.75	136.8	151.1	109.5	91.15	95.95	运算
	(3.87)	(9.12)	(7.87)	(7.87)	(9.80)	(8.86)	(9.13)	(13.28)	(2.31)	(21.00)	超时
PA-1000	**189.18**	201.25	257.10	313.35	251.55	349.60	336.18	288.3	276.18	296.20	运算
	(6.14)	(8.02)	(6.42)	(6.36)	(6.36)	(8.09)	(8.34)	(9.74)	(5.49)	(12.03)	超时
PA-2000	**190.45**	192.53	272.4	325.25	280.28	371.96	352.66	252.05	275.65	287.10	运算
	(10.23)	(7.60)	(6.20)	(9.40)	(7.60)	(9.68)	(11.13)	(6.92)	(4.07)	(10.04)	超时
PA-5000	**161.75**	173.03	295.63	345.30	325.08	403.08	299.35	208.00	250.18	259.60	运算
	(12.663)	(8.07)	(9.77)	(8.21)	(12.10)	(8.67)	(8.02)	(5.10)	(7.17)	(17.85)	超时
AN-100	332.73	**322.75**	347.60	355.85	346.85	357.95	352.88	380.95	750.98	运算	运算
	(11.61)	**(12.93)**	(9.63)	(13.88)	(12.04)	(13.35)	(12.50)	(18.04)	(16.49)	超时	超时
AN-200	278.60	**266.45**	299.43	337.83	293.65	329.35	338.35	322.40	460.80	运算	运算
	(11.60)	**(9.23)**	(13.33)	(15.51)	(14.42)	(11.50)	(14.73)	(23.96)	(18.24)	超时	超时
AN-500	218.58	**211.34**	285.13	341.43	263.05	329.20	356.70	253.25	297.45	运算	运算
	(10.45)	**(7.95)**	(11.85)	(14.34)	(15.19)	(14.57)	(20.54)	(12.19)	(3.66)	超时	超时

优势更为明显, 而 SaiyanH 方法无法在一定时间内获得结果.

在节点更多的 Hailfinder、HEPAR、Win95pts 数据集上, DM-GABN 与其余算法间的差距逐渐变大, 在 Win95pts 数据集上最为明显, GOBNILP 算法在超过 70 个节点的数据集开始, 已经无法在有效时间内获得搜索结果. MIGA 与 DM-GABN 结果较为接近, 但是, DM-GABN 获得的结构拥有更高的边预测的准确性, 与正确结构的差异更小. 在数据量增多的情况下, 一些算法的性能开始下降, 但是, DM-GABN 在性能提升的同时与其他算法拉开了更大的差距. 在超大型数据集 Pathfider 上, 由于变量和参数的激增, 几乎所有算法均难以预测到较为准确的结构, 但是, DM-GABN 仍然获得了这些方法中最优的结果, 且与第 2 名拉开了一定差距. 相比于 MIGA, DM-GABN 的 F_1 分数提升了 40%, 表明搜索到了更为准确的结构. 在 ANDE 数据集, 当数据量较少时, DM-GABN 的搜索结果准确地预测了更多的边, 但是, 其与正确结构间的差距稍逊于 MIGA 算法.

图 4 为不同遗传算法在数据集上的收敛表现. 由图 4 可见: 在 Alarm 数据集中, 所有算法最终获得的结果均较为接近, DM-GABN 和 MIGA 在迭代的开始可以更快地找到更优的结构; 在 Insrance 等中型数据集上, DM-GABN 仍然保持着较为明显的收敛性能优势, 在迭代开始便搜索到相比于其他方法更优的结果, 在后期仍然没有陷入局部最优, 搜索到了更好的结果; 在大型数据集 HEPAR、Win95pts、ANDE 上, 随着数据量增多, DM-GABN 则可获得最佳的结果, 尤其在 Win95pts 和 ANDE 数据集中更为明显.

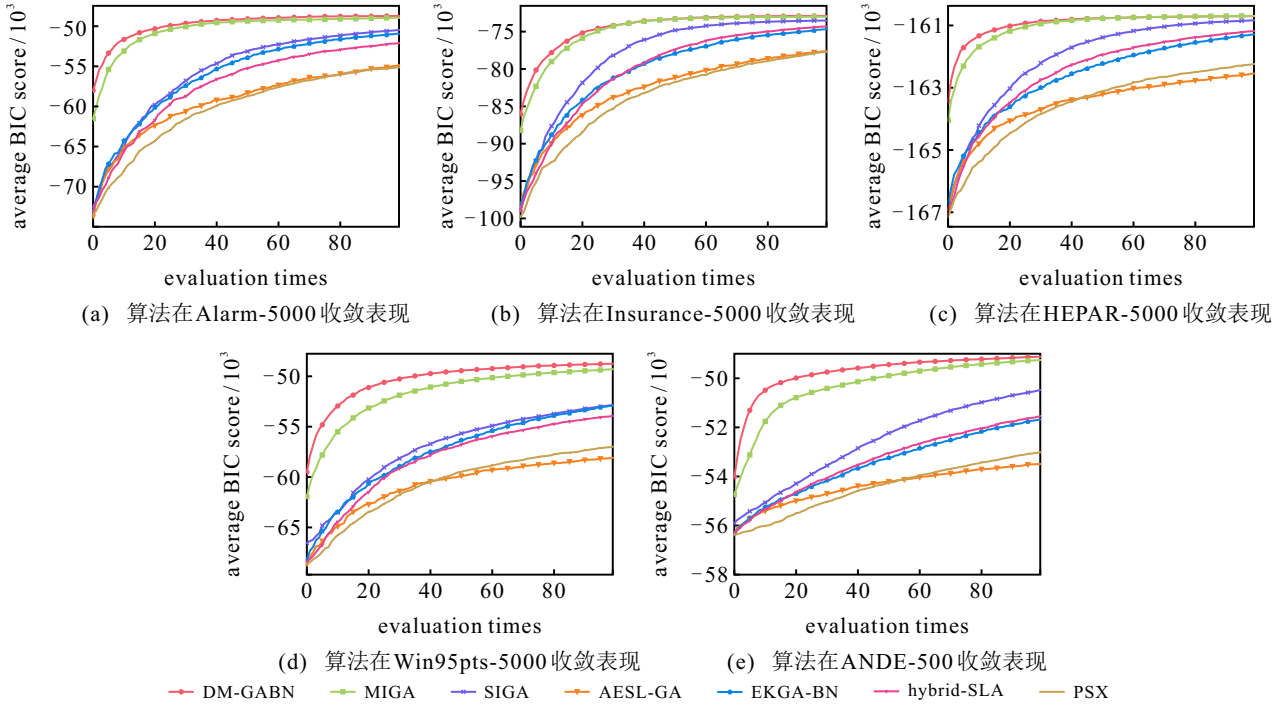

图4 不同遗传算法在数据集上收敛表现

本文在 10 个标准数据集上验证了 DM-GABN 的性能, 实验结果表明: 在小型网络 Asia-5000 上, 算法在 100 次迭代内收敛至接近最优解, 耗时为 60 s; 在中型网络 water-5000 上, 算法有了较为明显的提升, 可在 421 s 内找到更优的网络; 在大型网络 Pathfider-5000 上, DM-GABN 可在 4 125 s 相较于 MIGA 提升了 16.44% 的准确性, 但是在大型或超大型数据集上, 运行时间仍然较长, 主要原因是编码方式邻接矩阵的遍历较为耗时. 因此, 在节点数 200 以下的数据集上表现更好, 未来可通过分布式或并行计算提升算法效率和速度.

综上所述, 由于 DM-GABN 所提出 3 个算子的有效性, 所提出算法在搜索结果的准确性和收敛性上均有显著提升, 领先于其余的对比算法, 在一定程度上改善了基于 GA 的 BNSL 问题的过早收敛问题. 这种改善在 Insrance、HEPAR、Win95pts 等数据集的收敛图上尤为明显. 在大部分的数据集上, DM-GABN 均可获得最佳的结果. 在提升种群多样性的同时, 随着节点数量的提升, 构建的结构复杂度也随之提升, 但是, DM-GABN 在处理参数量较大的数据集时仍有很明显的提升空间.

4 结 论

本文基于遗传算法, 提出了由种群多样性和互

信息引导的贝叶斯网络结构学习算法. 该算法在选择算子和交叉算子上提升了更多基因型的种群基因型频率, 在选择算子中淘汰了年龄较大的个体, 提升了种群多样性, 从而有效地提升了算法的收敛性能; 同时, 在选择算子和交叉算子中利用互信息引导算法搜索. 通过在 10 个数据集上的实验, 多项指标表明了 DM-GABN 方法不仅提升了搜索结果的准确度精度, 且改善了遗传算法易陷入局部最优的问题.

参考文献 (References)

[1] Nayak N R, Kumar S, Gupta D, et al. Network mining techniques to analyze the risk of the occupational accident via Bayesian network[J]. International Journal of System Assurance Engineering and Management, 2022, 13(1): 633-641.

[2] Mihaljević B, Bielza C, Larrañaga P. Bayesian networks for interpretable machine learning and optimization[J]. Neurocomputing, 2021, 456: 648-665.

[3] 毛腾, 褚菲, 王建文, 等. 基于分布式混合贝叶斯网络的煤泥浮选过程安全运行与产品质量一体化控制方法[J]. 控制与决策, 2025, 40(2): 497-506.
(Mao T, Chu F, Wang J W, et al. An integrated safe operation and product quality control method for coal slurry flotation process based on distributed hybrid Bayesian network[J]. Control and Decision, 2025, 40(2): 497-506.)

[4] 汪春峰, 张永红. 基于无约束优化和遗传算法的贝叶斯网络结构学习方法[J]. 控制与决策, 2013, 28(4): 618-622.
(Wang C F, Zhang Y H. Bayesian network structure learning based on unconstrained optimization and genetic algorithm[J]. Control and Decision, 2013, 28(4): 618-622.)

[5] Dai J G, Ren J, Du W C, et al. An improved evolutionary approach-based hybrid algorithm for Bayesian network structure learning in dynamic constrained search space[J]. Neural Computing and Applications, 2020, 32(5): 1413-1434.

[6] Yan K F, Fang W, Lu H Y, et al. Mutual information-guided GA for Bayesian network structure learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(8): 8282-8299.

[7] Spirtes P, Glymour C, Scheines R. Causality from probability[J]. Evolving Knowledge in Natural and Artificial Intelligence, 1989, 181-199.

[8] Spirtes P, Glymour C. An algorithm for fast recovery of sparse causal graphs[J]. Social Science Computer Review, 1991, 9(1): 62-72.

[9] Cheng J, Bell D, Liu W. Learning Bayesian networks from data: An efficient approach based on information theory[J]. Artif Intell, 1999, 137(1/2): 43-90.

[10] Margaritis D, Thrun S. Bayesian network induction via local neighborhoods[J]. Advances in Neural Information Processing Systems, 1999, 12: 505-511.

[11] Tsamardinos I, Aliferis C F, Statnikov A. Time and sample efficient discovery of Markov blankets and direct causal relations[C]. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, 2003: 673-678.

[12] Tsamardinos I, Aliferis C F, Statnikov A R, et al. Algorithms for large scale Markov blanket discovery[C]. Proceedings of the 16th International Florida Artificial Intelligence Research Society Conference. Augustine, 2003: 376-381.

[13] Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data[J]. Machine Learning, 1992, 9: 309-347.

[14] Heckerman D, Geiger D, Chickering D M. Learning Bayesian networks: The combination of knowledge and statistical data[J]. Machine Learning, 1995, 20: 197-243.

[15] Bouckaert R R. Bayesian belief networks: From construction to inference[D]. University of Utrecht, 1995.

[16] Tsamardinos I, Brown L E, Aliferis C F. The max-min hill-climbing Bayesian network structure learning algorithm[J]. Machine Learning, 2006, 65: 31-78.

[17] Constantinou A C. Learning Bayesian networks that enable full propagation of evidence[J]. IEEE Access, 2020, 8: 124845-124856.

[18] 陈克斌, 鲁云军, 韩梦瑶, 等. 一种基于双编码遗传算法的机动微波接力网组网方法[J]. 控制与决策, 2020, 35(12): 2915-2922.
(Chen K B, Lu Y J, Han M Y, et al. Mobile microwave relay network construction method based on double coding genetic algorithm[J]. Control and Decision, 2020, 35(12): 2915-2922.)

[19] 朱光宇, 张德颂. 基于强化学习的遗传算法求解一种新的钻削路径优化问题[J]. 控制与决策, 2024, 39(2): 697-704.
(Zhu G Y, Zhang D S. Genetic algorithm based on reinforcement learning for a novel drilling path optimization problem[J]. Control and Decision, 2024, 39(2): 697-704.)

[20] 武燕, 刘小雄, 池程芝. 动态多目标优化的预测遗传算法[J]. 控制与决策, 2013, 28(5): 677-682.
(Wu Y, Liu X X, Chi C Z. Predictive multiobjective genetic algorithm for dynamic multiobjective optimization problems[J]. Control and Decision, 2013, 28(5): 677-682.)

[21] Contaldi C, Vafaee F, Nelson P C. Bayesian network hybrid learning using an elite-guided genetic algorithm[J]. Artificial Intelligence Review, 2019, 52: 245-272.

[22] Pluim J P W, Maintz J B A, Viergever M A. Mutual-information-based registration of medical images: A survey[J]. IEEE Transactions on Medical Imaging, 2003, 22(8): 986-1004.

[23] Cha J, Lee K, Park S, et al. Domain generalization by mutual-information regularization with pre-trained

models[C]. European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2022: 440-457.

[24] Zhou H F, Wang X Q, Zhu R R. Feature selection based on mutual information with correlation coefficient[J]. Applied Intelligence, 2022, 52(5): 5457-5474.

[25] Zhang P, Liu G X, Song J Z. MFSJMI: Multi-label feature selection considering join mutual information and interaction weight[J]. Pattern Recognition, 2023, 138: 109378.

[26] Alalhareth M, Hong S C. An improved mutual information feature selection technique for intrusion detection systems in the Internet of medical things[J]. Sensors, 2023, 23(10): 4971.

[27] He Z Y, Xu X F, Deng S C. *K*-ANMI: A mutual information based clustering algorithm for categorical data[J]. Information Fusion, 2008, 9(2): 223-233.

[28] Li J L, Liu Z F. Attribute-weighted outlier detection for mixed data based on parallel mutual information[J]. Expert Systems with Applications, 2024, 236: 121304.

[29] Lee S, Oh C, Wong Y, et al. Universal spreading of conditional mutual information in noisy random circuits[J]. Physical Review Letters, 2024, 133(20): 200402.

[30] Oliveto P S, Sudholt D. On the runtime analysis of stochastic ageing mechanisms[C]. Proceedings of the Annual Conference on Genetic and Evolutionary Computation. Vancouver, 2014: 113-120.

[31] Horoba C, Jansen T, Zarges C. Maximal age in randomized search heuristics with aging[C]. Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation. Montreal, 2009: 803-810.

[32] Jansen T, Zarges C. Analyzing different variants of immune inspired somatic contiguous hypermutations[J]. Theoretical Computer Science, 2011, 412(6): 517-533.

[33] Jansen T, Zarges C. On the role of age diversity for effective aging operators[J]. Evolutionary Intelligence, 2011, 4: 99-125.

[34] Benfey P N, Mitchell-Olds T. From genotype to phenotype: Systems biology meets natural variation[J]. Science, 2008, 320(5875): 495-497.

[35] Fang W, Zhang W J, Ma L, et al. An efficient Bayesian network structure learning algorithm based on structural information[J]. Swarm and Evolutionary Computation, 2023, 76: 101224.

[36] Jose S, Liu S M, Louis S, et al. Towards a hybrid approach for evolving Bayesian networks using genetic algorithms[C]. IEEE the 31st International Conference on Tools with Artificial Intelligence. Portland, 2019: 705-712.

[37] Cussens J. Bayesian network learning with cutting planes[J/OL]. 2012, arXiv: 1202.3713.

作者简介

方伟 (1980−), 男, 教授, 博士, 主要研究方向为智能优化理论、方法与应用、机器学习中的智能优化技术和优化与调度, E-mail: fangwei@jiangnan.edu.cn;

吴昀霖 (1997−), 女, 硕士生, 主要研究方向为智能优化与机器学习、贝叶斯网络结构学习, E-mail: 6223114008@stu.jiangnan.edu.cn;

朱书伟 (1990−), 男, 讲师, 博士, 主要研究方向为演化学习、智能优化、数据知识协同优化、数据挖掘和智能计算, E-mail: zhushuwei@jiangnan.edu.cn.