

基于演化深度强化学习的符号网络影响最大化研究*

马里佳¹, 洪华平¹, 林秋镇¹, 李坚强¹, 公茂果²

¹(深圳大学 计算机与软件学院, 广东 深圳 518060)

²(西安电子科技大学 电子工程学院, 陕西 西安 710071)

通信作者: 李坚强, E-mail: lijq@szu.edu.cn



摘要:近年来,随着互联网信息传播以及新型冠状病毒 COVID-19 传播链阻断等重大应用问题的出现, 社会网络影响最大化问题的研究受到了科学界广泛关注. 影响最大化问题旨在根据特定应用问题的传播模型, 识别出最优影响种子节点集, 最大化其信息传播影响. 现有影响最大化算法主要针对单连接影响传播模型, 将影响最大化问题模拟为离散的影响力种子节点组合选取优化问题. 然而, 这些算法具有较高的计算时间复杂度, 且无法解决具有大规模冲突关系的符号网络影响最大化问题. 针对上述问题, 首先, 构建适用于符号网络的正负影响传播模型以及影响最大化优化模型. 其次, 通过引入由神经网络构成的 deep Q network 来选取种子节点集, 将离散的种子节点组合选取问题转化为更易优化的网络权重连续优化问题. 最后, 提出基于演化深度强化学习的符号网络影响最大化算法 SEDRL-IM. 该算法将演化算法的个体视作策略, 结合演化算法的无梯度全局搜索以及强化学习的局部搜索特性, 实现对 deep Q network 权重优化问题的有效搜索, 从而找到最优影响种子节点集. 在基准符号网络以及真实社交网络数据集上的大量实验结果表明, 所提算法在影响传播范围与求解效率上都优于经典的基准算法.

关键词: 符号网络; 影响最大化; 演化算法; 深度强化学习

中图法分类号: TP18

中文引用格式: 马里佳, 洪华平, 林秋镇, 李坚强, 公茂果. 基于演化深度强化学习的符号网络影响最大化研究. 软件学报, 2023, 34(11): 5084-5112. <http://www.jos.org.cn/1000-9825/6728.htm>

英文引用格式: Ma LJ, Hong HP, Lin QZ, Li JQ, Gong MG. Influence Maximization for Signed Networks Based on Evolutionary Deep Reinforcement Learning. Ruan Jian Xue Bao/Journal of Software, 2023, 34(11): 5084-5112 (in Chinese). <http://www.jos.org.cn/1000-9825/6728.htm>

Influence Maximization for Signed Networks Based on Evolutionary Deep Reinforcement Learning

MA Li-Jia¹, HONG Hua-Ping¹, LIN Qiu-Zhen¹, LI Jian-Qiang¹, GONG Mao-Guo²

¹(School of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China)

²(School of Electronic Engineering, Xidian University, Xi'an 710071, China)

Abstract: In recent years, the research on influence maximization (IM) for social networks has attracted extensive attention from the scientific community due to the emergence of major application issues, such as information dissemination on the Internet and the blocking of COVID-19's transmission chain. IM aims to identify a set of optimal influence seed nodes that would maximize the influence of information dissemination according to the propagation model for a specific application issue. The existing IM algorithms mainly focus on unidirectional-link influence propagation models and simulate IM issues as issues of optimizing the selection of discrete influence seed node combinations. However, they have a high computational time complexity and cannot be applied to solve IM issues for signed networks with large-scale conflicting relationships. To solve the above problems, this study starts by building a positive-negative influence propagation model and an IM optimization model readily applicable to signed networks. Then, the issue of selecting discrete seed node

* 基金项目: 国家自然科学基金 (U1713212, 61672358, 61572330, 61772393, 61836005); 广东省自然科学基金 (2017A030313338); 国家重点研发计划 (2020YFA0908700)

收稿时间: 2021-12-03; 修改时间: 2022-02-09, 2022-06-16; 采用时间: 2022-06-20; jos 在线出版时间: 2023-05-18

CNKI 网络首发时间: 2023-05-19

combinations is transformed into one of continuous network weight optimization for easier optimization by introducing a deep Q network composed of neural networks to select seed node sets. Finally, this study devises an IM algorithm based on evolutionary deep reinforcement learning for signed networks (SEDRL-IM). SEDRL-IM views the individuals in the evolutionary algorithm as strategies and combines the gradient-free global search of the evolutionary algorithm with the local search characteristics of reinforcement learning. In this way, it achieves the effective search for the optimal solution to the weight optimization issue of the Deep Q Network and further obtains the set of optimal influence seed nodes. Experiments are conducted on the benchmark signed network and real-world social network datasets. The extensive results show that the proposed SEDRL-IM algorithm is superior to the classical benchmark algorithms in both the influence propagation range and the solution efficiency.

Key words: signed network; influence maximization (IM); evolutionary algorithm (EA); deep reinforcement learning (DRL)

1 引言

随着互联网和 Web 技术的兴起, 各大网络社交平台如 Facebook、Twitter 和微博产生了海量的观测性社交数据. 这为信息传播、网络欺诈、谣言散发、传染病控制等领域的研究提供了前所未有的机遇. 在社交网络中, 人们可通过言论发表、视频推荐、线上交流、新闻发布、消息转发、朋友圈评论、信息互赞等方式逐步在社交网络中散播信息, 进而在用户传播并不断扩大其影响^[1]. 由于受到爆发式营销、商品推荐、舆论调控等应用领域的驱动, 影响最大化 (influence maximization, IM) 问题的研究受到了广泛关注. IM 问题的概念首先由 Domingos 等人^[2]提出, 被用于分析市场用户的营销方案. 该问题研究给定营销传播模型下, 如何从市场中选择最合适的用户集参与营销传播, 实现营销影响最大.

随着 IM 问题研究的逐步深入, 各种影响传播模型被陆续提出, 用于研究在不同激活条件下的影响传播方式, 如独立级联 (independent cascade, IC) 模型、线性阈值 (linear threshold, LT) 模型和时间感知 (time detection, TD) 模型. 近年来, Kempe 等人^[3]和 Gong 等人^[4]将 IM 问题拓展到复杂网络领域, 并结合网络的拓扑结构知识进行了深入研究. 如 Kempe 等人^[3]利用贪心策略等局部搜索方法选择度较大的节点作为种子节点, 并将 IM 问题建模为离散组合优化问题. 然而, 基于 IC 模型和 LT 模型的离散组合 IM 问题被验证具有 NP-hard 特性与非凸优化特性. 基于拓扑结构的贪心局部搜索由于其复杂度较高, 使得其无法处理大规模网络. 为了解决此类问题, 一些工作尝试通过优化贪心算法的伸缩性来提升计算性能. 如 Leskovec 等人^[5]提出的 CELF (cost-effective lazy-forward) 算法, 利用 IM 问题的子模特性减少影响评估次数, 使计算速度提升约 700 倍. Ohsaka 等人^[6]和 Goyal 等人^[7]分别提出 PMC (pruned Monte-Carlo) 算法和 CELF++ 算法, 通过对重复的影响扩散路径进行剪枝, 进一步提高了求解效率. 李国良等人^[8]提出基于树的算法模型, 通过挖掘子模特性与计算增益上界来避免冗余计算. 近年来, 启发式与元启发式思想逐渐被引入, 用于提升贪心算法的优化性能. 如易秀双等人^[9]通过引入弱连接边缘理论, 加强不同社区的信息传播与边缘节点搜索. Chen 等人^[10]利用度的中心性, 提出 DegreeDiscount 算法来选择种子节点. Morone 等人^[11]和 Wang 等人^[12]分别引入社区信息和网络渗透理论, 提出基于社区感知 (community genetic awareness, CGA) 和集体影响感知 (collective intelligence, CI) 算法来选择种子节点.

互联网信息传播、隐私保护与新型冠状病毒传播链阻断等重大应用问题的出现, 使得符号化社交网络的 IM 问题受到广泛关注. 符号化社交网络中用户间具有冲突正负边关系, 其正负边分别表示用户间的积极关系 (如友好、联盟、合作关系) 与消极关系 (如敌对、分裂、竞争关系). 此类网络的冲突化关系, 对信息传播等应用领域产生了巨大影响. 如在产品销售中, 厂家倾向于与好评率高的门店合作销售产品, 通过好评用户群传播正面评价提高产品销量. 同时厂家取消与差评率高的门店合作, 减少产品负面评价传播. 随着网络冲突关系的广泛出现, 符号网络的 IM 问题^[13]近年来被广泛研究. 如 Li 等人^[14]首次提出符号网络的影响传播模型, 指出用户在交流观点时倾向于赞同友好方观点, 并容易反驳敌对方观点.

符号网络 IM 问题主要研究正影响和负影响在影响传播过程中的生成方式和相互作用. 如 Li 等人^[15]将正负影响建模为对立关系, 并以正负影响之差构成影响范围评估. Liu 等人^[16]通过结合 IC 模型, 定义符号网络的影响传播影响为正负活跃节点的数量之差. 田家堂等人^[17]通过结合 LT 模型的“影响累积”特性, 选择累计影响多的节点为种子节点. 然而, 符号网络的 IM 问题同样具有 NP-hard 特性与非凸优化特性, 这使得经典基于贪心局部搜索的 IM 优化算法难以有效找到较优解集. 为了解决此问题, Liang 等人^[18]提出基于贪心策略的 RLP (R-greedy with

live-edge and propagation-path) 算法, 通过引入基于活边以及传播路径的加速技术来提高求解速率. Li 等人^[19]则提出基于个体全局搜索的模拟退火 (simulated annealing, SA) 算法. Gong 等人^[20]提出 DPSO (discrete particle swarm optimization) 算法, 通过设计具有快速收敛性与全局搜索能力的粒子群算法, 实现符号网络的影响节点集选取. Şimşek 等人^[21]提出通过重构影响最大化函数, 使得的状态空间曲面具备斜率, 并采用群体智能算法 GWO (grey wolf optimizer) 算法来优化求解过程. Yin 等人^[22]通过假设每个节点可以同时接收正负影响, 提出随机游走策略 SPR (signed-pagerank) 算法来选择种子节点.

值得注意的是, 上述算法主要是将 IM 问题转化为离散的组合优化问题进行求解. 然而, 离散组合 IM 优化问题的 NP-hard 特性使得这些算法在求解过程中具有较高计算代价, 导致其无法高效处理大规模网络. 为了解决上述问题, Dai 等人^[23]通过引入深度强化学习思想 (deep reinforcement learning, DRL)^[24], 提出 S2V-DQN 算法用于求解. 该算法首先使用神经网络构建 DQN (deep Q network), 通过网络特征降维学习网络的属性特征. 其次, 该算法通过特征训练 DQN 获得每个节点的种子分数, 并利用贪心策略选择种子节点. 该算法能将离散的组合选点问题转化为对 DQN 的连续网络权重优化问题. 然而, 该算法过于依赖贪心策略, 导致其容易陷入局部最优解. 为了尽可能摆脱局部最优解, Wang 等人^[25]引入演化算法 (evolutionary algorithm, EA)^[26], 提出了混合模因算法 MA-IM 求解无符号网络的 IM 问题. 该算法将种子节点序列视为“基因序列”进行演化, 通过 EA 算法中的交叉和变异演化操作增强“基因序列”的多样性, 实现对解空间的针对性全局探索. 我们结合网络社区信息^[27], 提出 CMA-IM 算法对解空间进行全局搜索与社区贪心局部搜索, 使得算法能快速地寻找 IM 问题的较优解.

然而, 上文提到的算法^[23-27]在求解 IM 问题时很难在求解有效性和高效性上取得较好平衡. 近年来, 演化深度强化学习算法通过结合 EA 与 DRL 的策略搜索, 能够实现对非凸优化问题进行较好求解. 比如 Open-AI ES^[28]、DRL-GA^[29]、SGES^[30]、G2N^[31]等算法通过使用演化算法代替基于策略梯度的策略优化算法来优化策略, 从而实现对非凸优化问题的快速有效求解. EPG^[32]、ERL^[33]、CERL^[34]、EDO-CS^[35]与 CEM-RL^[36]算法将策略视为可演化的个体, 结合 EA 的无梯度优化与种群搜索特性以及 DRL 的梯度优化特性, 优化策略种群, 实现对非凸优化问题的快速有效求解. 为了高效求解网络的 IM 问题, 我们提出了一种基于结合 EA 算法和 DRL 思想的演化深度强化学习算法 EDRL-IM (evolutionary deep reinforcement learning for IM)^[37]. 该算法首先采用 DQN 生成评估节点影响力的 Q 值, 结合动态马尔可夫决策选择种子节点参与影响传播, 将影响分数作为决策奖励对 DQN 进行反向更新. 这些过程可将离散的组合选点过程转化为 DQN 的连续网络权重优化过程. 接下来, EDRL-IM 算法将每个 DQN 的网络权重视作策略个体, 通过结合 EA 的全局搜索能力与 DRL^[26]的反向梯度优化局部搜索能力, 快速有效地对所建模的连续 IM 问题进行求解.

为了高效地求解符号网络中的 IM 问题, 本文尝试将 EDRL-IM 算法中的 EA 算法和 DRL 算法的混合优化思想拓展到符号网络中, 进而提出求解符号网络 IM 问题的演化深度强化学习算法 SEDRL-IM (signed EDRL-IM). 该算法首先结合符号网络的正负拓扑关系, 构建基于符号 IC 模型的影响传播模型, 模拟现实社交网络中正负影响的扩散变化过程; 其次, 该算法通过 DQN 生成评估节点影响力的 Q 值, 结合动态马尔可夫决策选择种子节点参与影响传播, 将符号网络的 IM 离散组合优化问题转化为基于 DQN 的连续网络权重优化问题; 最后, 该算法将每个 DQN 的网络权重视作策略个体, 通过结合 EA 的全局搜索能力与 DRL 的反向梯度优化局部搜索能力, 根据符号网络影响扩散的适应值函数和马尔可夫决策奖励函数对策略优化进行监督, 实现对 DQN 权重优化问题解的局部和全局探索, 找到最优影响种子节点集. 在基准符号网络以及真实社交网络数据集上的大量实验结果表明, 该算法在影响传播范围与求解效率上优于其他经典的基准算法.

本文第 2 节介绍相关的影响传播模型以及符号网络上的 IM 问题. 第 3 节介绍求解 IM 问题的相关性算法研究. 第 4 节介绍 DQN 和马尔可夫决策在符号网络中选取种子节点集的过程和原理. 第 5 节介绍本文所提 SEDRL-IM 算法的框架和原理. 第 6 节介绍本文的实验设计, 包括实验数据集和对比算法. 第 7 节对实验结果进行分析与验证. 第 8 节总结工作总结与展望.

2 问题定义与建模

本文中 $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ 被用于表示符号网络数据集. 其中 $\mathcal{V} = \{v_i\}_n$ 为符号网络的节点集, n 为节点的数量; $\mathcal{E} = \{e_{ij}\}_m$

为符号网络的边集, 其中 e_{ij} 表示节点 v_i 指向节点 v_j 的边, m 为边的数量; $A \in [a_{ij}]_{m \times n}$ 表示符号网络的邻接矩阵, 其中 $a_{ij} \in [-1, 1]$ 表示 e_{ij} 的权重: $a_{ij} > 0$, $a_{ij} < 0$ 与 $a_{ij} = 0$ 分别表示节点 v_i 和节点 v_j 之间存在正边、负边以及不存在边. $\mathcal{N}_i = \{v_j \in \mathcal{V} | a_{ij} \neq 0\}$ 表示符号网络中任意节点 v_i 的邻居节点集. 在影响传播模型中, 符号网络中的每条边 e_{ij} 都具有对应的传播概率 p_{ij} , 其中 p_{ij} 表示节点 v_j 被节点 v_i 激活的概率.

2.1 基于独立级联的影响传播模型

影响传播模型描述了复杂网络^[38]中节点集如何通过边激活邻居节点的传播影响过程. 其经典模型主要包括 IC 模型^[39]、LT 模型^[40]和 TD 模型^[41]. 本文主要基于 IC 模型, 结合符号网络来模拟现实社交网络的正负影响扩散过程. IC 模型主要作用于拥有正关系的网络. 即当任意节点 v_i 被激活后, 其根据传播概率 e_{ij} 来尝试激活邻居节点 v_j 来传播影响, 被激活的节点则变为活跃状态. 上述过程一直重复直至没有节点可被激活为止. 具体而言, 在任意 t 时刻下, 节点被激活并传播影响的具体原理可表示如下.

- (1) 在 t 时刻, 最新被激活的节点 v_i 以 p_{ij} 概率尝试激活其邻居节点 $v_j \in \mathcal{N}_i$ 实现影响传播.
- (2) 如果某一节点 v_j 被激活, 则该节点将在 $(t+1)$ 时刻转变为活跃状态, 并具备在 $(t+1)$ 时刻激活其邻居节点扩散影响的能力. 在 $(t+1)$ 时刻后, 该节点将成为无激活能力的活跃节点.
- (3) 上述过程一直重复直至没有节点可被激活为止. 在影响传播过程的每个时刻, 网络的影响分数等于活跃节点的数量.

图 1 展示了 IC 模型下以节点 v_5 与 v_6 在 t 时刻为最新被激活节点的传播影响过程, 其箭头表示影响传播方向. 具体而言, 在 t 时刻, v_5 与 v_6 节点以传播概率 0.4、0.5、0.3 与 0.2 分别影响其邻居节点 v_1 、 v_3 、 v_4 与 v_4 . 在 $(t+1)$ 时刻, v_1 与 v_4 被激活, 并以相应边的传播概率影响其邻居节点 v_2 与 v_3 . 在 $(t+2)$ 时刻, v_2 节点被激活, 并无法影响产生新的激活节点. 因此, 在 $(t+2)$ 时刻后, 传播过程结束, 其传播影响分数为活跃节点的数量 5.

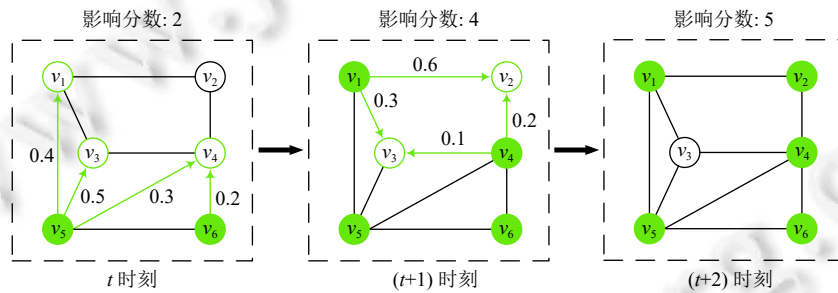


图 1 IC 模型下的影响传播过程

2.2 基于符号网络的独立级联模型

上述 IC 模型展示了正相关关系网络的影响传播过程和原理. 然而, 符号网络同时含有正负相关关系, 导致任意节点在激活其他节点时会同时传播正负影响. 因此, 传统的 IC 模型无法适用于符号网络. 本文尝试结合正负相关关系, 提出基于符号网络的独立级联 (signed network independent cascade, SNIC) 模型.

SNIC 模型定义了符号网络中正负影响的生成方式和相互作用, 用来模拟现实社交网络的正负影响传播过程. 在 SNIC 模型中, 社交理论被引入对正负影响的传播过程进行建模: 假设符号网络中用户 U_i 和用户 U_j 之间为积极关系 (如友好关系), 用户 U_i 和用户 U_k 之间为消极关系 (如敌对关系), 当用户 U_i 向用户 U_j 与 U_k 分享观点时, 则正影响的传播表现为: U_j 倾向于赞同 U_i 的观点并传播相同的观点; 负影响的传播表现为: U_k 倾向于反驳 U_i 的观点并传播相反的观点. 基于这个社交理论, SNIC 模型假定正负影响传播的关系是相互对立、相互抵消与非叠加. 因此, SNIC 模型下任意 t 时刻节点被激活并传播正负影响的具体原理可表示如下.

- (1) 在 t 时刻, 任意节点 v_i 以 p_{ij} 绝对值大小的概率, 尝试激活自身的邻居节点 $v_j \in \mathcal{N}_i$ 来扩散影响.
- (2) 在 $(t+1)$ 时刻, 节点 v_j 若被通过正边和负边激活, 则分别转变为正与负活跃状态.

(3) 上述过程一直重复直至没有节点可被激活为止. 影响分数表示为正负活跃节点的数量之差.

图 2 展示了 SNIC 模型下以节点 v_5 与 v_6 为被激活节点的传播影响过程, 其绿色与蓝色箭头分别表示正边与负边影响传播过程. 在影响传播过程中, 通过正边被成功激活的节点将变为正活跃状态 (绿色节点表示), 而通过负边被成功激活的节点将变为负活跃状态 (蓝色节点表示). 在 $(t+2)$ 时刻后, 传播过程结束, 其正传播影响节点数为 4, 负传播影响节点数为 2, 因此总的传播影响分数为 2.

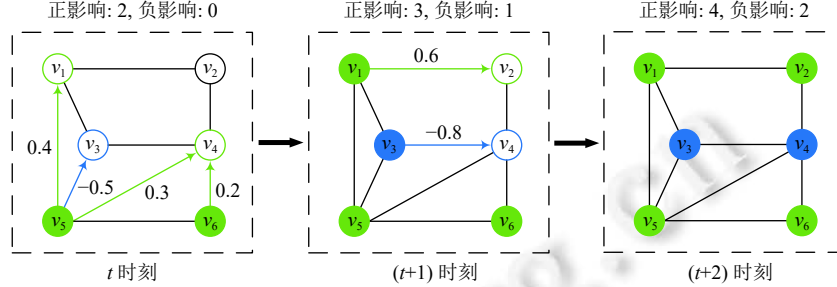


图 2 SNIC 模型下的影响传播过程

2.3 基于 SNIC 模型的影响最大化问题

对大规模网络而言, SNIC 模型下进行正负影响传播过程的影响分数计算代价较高. 为了简化影响分数的计算流程, 本文将 Lee 等人^[42]提出的快速近似 IC 理论拓展到 SNIC 模型中, 将节点的影响范围限制在距离自身两跳范围. 在计算影响分数时, $x = \{x_i \in \{1, 0\}\}_n$ 表示符号网络每个节点的种子节点状态, 比如 $x_i = 1$ 表示节点 v_i 是种子节点, $\mathcal{S} = \{v_i \in \mathcal{V} : x_i = 1\}$ 表示符号网络的种子节点集. $\mathbf{I}(\mathcal{S})$ 表示采用种子节点集 \mathcal{S} 进行影响传播后得到的影响分数, $\mathbf{I}(v_i)_{1\text{-hop}}$ 表示节点 v_i 在一跳以内的影响分数, $\mathbf{I}(v_i)_{2\text{-hop}}$ 表示节点 v_i 在两跳以内的影响分数.

SNIC 模型采用正影响与负影响之差来衡量任意节点的影响分数. 因此, $\mathbf{I}(v_i)_{1\text{-hop}}$ 表示的是节点 v_i 通过正边激活邻居的成功个数和通过负边激活邻居的成功个数之差.

$$\mathbf{I}(v_i)_{1\text{-hop}} = 1 + \sum_{j \in \mathcal{N}_i} p_{ij} \quad (1)$$

其中, \mathcal{N}_i 为节点 v_i 的邻居节点集. p_{ij} 表示边 e_{ij} 的激活概率, 若边 $e_{ij} > 0$ 则 $p_{ij} > 0$; 若边 $e_{ij} < 0$ 则 $p_{ij} < 0$.

在节点 v_i 尝试激活所有邻居节点得到新的活跃节点后, SNIC 模型基于公式 (1), 可将 $\mathbf{I}(v_i)_{2\text{-hop}}$ 表示为被节点 v_i 激活后的活跃节点在一跳以内的影响分数之和.

$$\mathbf{I}(v_i)_{2\text{-hop}} = 1 + \sum_{j \in \mathcal{N}_i} p_{ij} (\mathbf{I}(v_j)_{1\text{-hop}}) \quad (2)$$

然而, 公式 (2) 计入了节点 v_i 激活种子节点的无效影响, 以及节点 v_j 反过来激活节点 v_i 以及其他种子节点的无效影响. 因此, SNIC 模型可通过剔除上述无效影响后计算真实的 $\mathbf{I}(v_i)_{2\text{-hop}}$, 其可计算为:

$$\mathbf{I}(v_i)_{2\text{-hop}}' = 1 + \sum_{j \in \mathcal{N}_i - \mathcal{S}} p_{ij} (\mathbf{I}(v_j)_{1\text{-hop}} - p_{ji}) - \sum_{j \in \mathcal{N}_i - \mathcal{S}} \sum_{u \in \mathcal{N}_j \cap \mathcal{S}} p_{ij} p_{ju} \quad (3)$$

其中, $\mathcal{N}_i - \mathcal{S}$ 表示任意节点 v_i 的邻居节点集在去除种子节点后得到的集合, $\mathcal{N}_j \cap \mathcal{S}$ 表示任意节点 v_j 的邻居节点集在去除非种子节点后得到的集合.

SNIC 模型下符号网络的影响分数 $\mathbf{I}(\mathcal{S})$ 可统计为 k 个种子节点在两跳以内的影响分数之和.

$$\mathbf{I}(\mathcal{S}) = k + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i - \mathcal{S}} p_{ij} (\mathbf{I}(v_j)_{1\text{-hop}} - p_{ji}) - \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i - \mathcal{S}} \sum_{u \in \mathcal{N}_j \cap \mathcal{S}} p_{ij} p_{ju} \quad (4)$$

基于上述公式, 符号网络的 IM 问题可表述为: 如何选取合适的种子节点集 \mathcal{S} 参与正负影响传播, 从而使得符号网络的影响分数 $\mathbf{I}(\mathcal{S})$ 达到最大化. 因此, 符号网络的 IM 问题可建模为如下的离散优化模型.

$$\max \mathbf{I}(\mathcal{S}) = \max \left\{ k + \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i - \mathcal{S}} p_{ij} (\mathbf{I}(v_j)_{1\text{-hop}} - p_{ji}) - \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{N}_i - \mathcal{S}} \sum_{u \in \mathcal{N}_j \cap \mathcal{S}} p_{ij} p_{ju} \right\} \quad (5)$$

3 相关性工作

本文研究主要涉及优化算法搜索框架、演化强化学习算法与影响最大化方法相关性工作.

3.1 基于局部搜索、全局探索以及混合搜索框架的优化算法

优化算法的搜索框架根据对解空间的搜索方式可分为局部搜索、全局探索以及混合搜索, 其差异主要体现在搜索策略、优化性能、解收敛性以及运行效率上^[5,12,19,20,23,25,27,30,43-49], 如表 1 所示。

表 1 各优化算法搜索框架性能分析

| 算法 | 优化性能 | 收敛性 | 稳定性 | 复杂度 | 网络优化的代表性算法 |
|-------|------|-----|-----|-----|--|
| 局部搜索 | 差 | 好 | 好 | 低 | CELF算法 ^[5] 、CI算法 ^[12] 、S2V-DQN算法 ^[23] 、FM算法 ^[43] 、BGLL算法 ^[44] 、GenLouvain算法 ^[30] |
| 全局搜索 | 好 | 差 | 差 | 高 | SA算法 ^[19] 、DPSO算法 ^[20] 、MODPSO算法 ^[45] |
| 混合搜索 | 好 | 好 | 好 | 高 | MLCD算法 ^[46] 、MWDS算法 ^[47] 、MA-IM算法 ^[25] 、CMA-IM算法 ^[27] |
| 自适应搜索 | 好 | 好 | 好 | 高 | 自适应演化策略算法 ^[49] 、自适应差分演化算法 ^[48] |

局部搜索算法主要基于贪心策略对解空间进行邻域搜索. 该类算法通常从一个初始解开始, 结合特定极值问题的特性, 通过邻域扰动来搜索该解的局部邻域解, 并以邻域的最优解作为下次迭代的初始解. 局部搜索算法通过反复执行邻域搜索, 从而求得在单路径上的局部最优解. 该类算法因为搜索方向单一与目的明确, 具有较好的收敛性、稳定性以及运行效率. 但对于非凸优化问题, 该类方法易陷入局部最优. 比如, 在该类算法中, 梯度下降法^[50]是最典型的方法之一. 该算法的主要优化思想是将极值问题在某时刻的负梯度方向作为搜索方向, 从而通过梯度更新来搜索该方向的解空间. 虽然该算法借助梯度来提高收敛速度, 但容易陷入局部最优并出现“锯齿”现象: 当优化问题的绝对梯度呈现平滑变小时, 该策略会导致局部搜索速度越来越慢. 在复杂网络相关优化研究中, 一系列经典的局部搜索算法被提出, 其通过结合网络的节点度与模块性知识来求解复杂网络中的极值问题, 其中包括基于节点整合与模块度优化的网络社区检测算法 FM^[43]、BGLL^[44]与 GenLouvain^[51]; 基于度分布最大化的网络鲁棒性优化算法 CI^[12]以及影响最大化算法 CELF^[5]; 基于强化学习的网络鲁棒性优化方法 S2V-DQN^[23]等. 近年来, 局部搜索算法也逐渐发展为基于全局搜索 (如演化算法)、一阶优化 (如随机梯度下降) 等算法对局部解空间进行最优优化搜索.

全局探索算法主要基于演化策略等元启发式算法对解空间进行试探性的探索. 相比于单方向的局部搜索, 全局搜索算法通常从一组或单个初始解开始, 通过学习自然界的演化方式 (如种群演化、模拟退火、群体行为) 来同时优化这组初始解, 实现对解空间的多路径探索, 进而避免陷入局部最优. 该类算法具有较好的优化性能, 但收敛速度较慢、运行效率较低以及稳定性较弱. 比如, 在该类算法中, 演化算法^[26]是最典型的算法之一, 其主要优化思想是结合自然界的种群迭代演化行为 (如选择、交叉与变异), 实现对极值问题的迭代优化过程, 获得比局部搜索更广的探索视野^[52]. 该算法的优化性能已经在传统组合优化问题如旅行商问题、分类问题、背包问题等得到了广泛验证. 但该算法同样表现出收敛速度较慢以及稳定性较弱的问题. 在复杂网络相关优化研究中, 一系列经典的全局探索算法被提出, 如基于多目标演化优化的社区检测算法 MODPSO^[45]、基于多目标粒子群分解优化的社区检测算法 DPSO^[20]以及基于模拟退火的社区鲁棒性优化算法 SA^[19]等.

混合搜索算法早期主要结合全局搜索与局部搜索对解空间同时进行针对性探索与邻域搜索. 其中, 全局搜索主要对解空间进行多方向探索, 从而避免局部搜索算法陷入局部最优, 保证算法的优化性能^[29]; 局部搜索主要对全局探索得到的多条解路径上进行邻域搜索, 加快全局搜索的收敛性, 保证算法的收敛稳定性. 因此, 相比于全局搜索算法和局部搜索算法, 混合搜索算法具有更好的优化性能、收敛性与稳定性. 但是, 混合搜索算法由于需要同时运行全局和局部搜索, 导致其复杂度高于单一的局部搜索算法和全局探索算法. 在该类算法中, 模因算法^[53]是最典型的算法之一, 其主要优化思想是结合自然界的自然演化行为与社会界的行为干预, 制定演化策略来同时优化一组初始解, 对解空间进行多方向探索, 并利用贪心算法加速每个解路径方向上的搜索速度. 该算法的优化性能已经在旅行商问题、路径规划问题、分类问题等得到了广泛验证. 在复杂网络相关优化研究中, 代表性的混合搜索算法包括基于模因优化的社区检测算法 MLCD^[46]、网络控制优化算法 MWDS^[47]以及影响最大化算法 MA-IM^[25]、CMA-IM^[27]等. 近年来, 混合搜索算法得到了较大的发展. 其中, 随着深度神经网络、强化学习等人工智能算法的

兴起及广泛应用, 基于演化算法 (策略) 与梯度下降的混合算法受到了学界广泛的关注与研究^[28,54]. 其中演化算法 (策略) 作为全局搜索用于优化问题策略或策略梯度, 梯度下降算法作为局部搜索用于加速优化. 此外, 针对实际应用中的大规模问题, 基于贪心策略、贝叶斯优化与梯度下降的局部混合算法也受到了较大的关注^[55]. 这类算法在解决大规模问题上能取得在收敛性、有效性以及效率上的较优折中.

自适应搜索算法主要通过制定自适应参数来进一步平衡混合搜索算法中的局部搜索和全局探索, 能获得较好的收敛稳定性和高效的优化能力. 如自适应差分演化算法^[48]基于神经网络生成交叉概率和比例因子, 通过强化学习动态监督和调整差分演化算法的探索方向, 并借助策略梯度来加快收敛速度. 自适应演化策略算法^[49]基于历史估计梯度自适应采样探索方向, 动态调整演化中的求解路径, 能有效降低演化策略带来的样本复杂度并提高收敛效率.

3.2 演化深度强化学习算法

强化学习算法主要通过奖励对智能体与环境的马尔可夫决策交互过程进行指导, 实现累计奖励的最大化, 求得特定问题的近似最优解. 深度强化学习算法结合深度学习的感知搜索能力和强化学习的决策能力, 通过计算损失函数梯度反向传播优化神经网络权重参数, 实现对极值问题的行动空间进行搜索. 然而, 深度强化学习主要依赖于基于梯度下降的局部搜索, 容易陷入局部最优解^[30]. 为了提高该类算法的优化性能, Kwon 等人^[32]基于问题的对称性, 设计了一种具备低方差贴现因子的损失函数, 使得强化学习在训练过程中可借助动态变化的策略梯度来调整局部搜索的方向, 使得其训练快速而稳定. S2V-DQN 算法通过深度神经网络学习网络节点特征, 结合强化学习思想以影响分数作为奖励, 结合马尔可夫决策高效地选出影响力节点.

演化深度强化学习算法主要结合演化算法与深度强化学习进行策略搜索, 实现对非凸优化问题的有效求解. 从策略搜索角度来说, 演化深度强化学习主要分为两类: 一类是使用演化算法代替基于策略梯度 (如梯度下降、随机梯度下降、ADAM 等) 的策略优化算法来优化策略搜索方向, 如 OpenAI-ES^[28]、DRL-GA^[29]、SGES^[30]、G2N^[31]; 另一类是将策略视为可演化的个体, 结合演化算法的无梯度优化与种群搜索特性以及强化学习的梯度优化特性, 进而优化策略种群, 如 EPG^[32]、ERL^[33]、CERL^[34]、EDO-CS^[35]、CEM-RL^[36]、EDRL-IM^[37]与本文算法 SEDRL-IM. 该类方法可通过引入多样性演化策略或优化目标, 要求演化种群里的策略保持多样性, 如 EDO-CS^[35]. 从演化算法与梯度搜索算法的结合方式角度来说, 演化深度强化学习算法可以分为无混合和混合演化深度强化学习两类. 其中, 无混合演化深度强化学习是使用演化算法来代替梯度搜索来优化策略, 如 Open-AI ES^[28]、DRL-GA^[29]、SGES^[30]、EDO-CS^[35]; 混合演化深度强化学习是结合演化算法和梯度搜索算法来优化策略, 如 G2N^[31]、EPG^[32]、ERL^[33]、CERL^[34]、CEM-RL^[36]、EDRL-IM^[37]与本文方法 SEDRL-IM. 其中, 混合搜索演化深度强化学习的结合循环次序包括外演化内强化优化与外强化内演化优化两种方式. 在上述混合算法中, 仅 G2N^[31]遵循外强化内演化优化这类结合循环次序. 这些方法可通过引入动作奖励与目标优化函数, 转化为基于动作优化的演化深度强化学习^[56]. 近年来, 演化深度强化学习领域已有较为系统的综述性研究^[55-57]. 其中, 文献^[56]主要从演化算法与深度强化学习的结合方式以及优化对象等角度, 对演化深度强化学习进行了系统性的综述. 文献^[55]主要深入分析了演化深度强化学习与无导数优化算法的结合, 及其在优化搜索与探索上的性能. 文献^[57]系统性总结归纳了演化算法、演化策略与深度强化学习在搜索、探索、并行化上的应用特性.

本文提出的演化深度强化学习算法 SEDRL-IM 将策略视为可演化的个体, 并结合遗传算法的无梯度优化与随机搜索特性以及强化学习的梯度优化特性, 并基于外演化内强化优化方式对优化策略种群进行混合优化. SEDRL-IM 通过演化算法优化 DQN 权重参数实现全局优化, 并结合梯度下降法进一步反向优化 DQN 权重进行局部搜索. 后文表 2 总结归纳了本文算法 SEDRL-IM 与相关演化深度强化学习算法的联系和差异.

3.3 求解影响最大化问题的局部搜索和全局探索算法

在求解 IM 问题的研究中, 一系列局部搜索算法被提出. 如 CELF (CELF++) 等算法^[5,7]基于节点度来贪心选取种子节点. CI 算法^[12]通过网络渗透理论获取网络的社区信息, 并基于社区感知和集体影响感知策略来选择种子节点. PageRank^[22]算法通过随机游走策略获取每个节点的 PR 值, 并基于该值贪心选择种子节点. S2V-DQN 算法^[23]

通过 DQN 学习网络特征, 并对每个节点的影响力进行评估, 以评估分数作为选择种子节点的标准. 然而, 这类局部搜索算法容易过早收敛, 从而导致优化能力较差.

表 2 本文算法 SEDRL-IM 与相关演化深度强化学习算法的比较

| 算法 | 优化策略 | 优化策略种群 | 演化方法 | 强化学习方法 | 多样性要求 | 结合方式 | 搜索次序 |
|---------------------------|------|--------|-------|------------------|-------|-------|------------|
| OpenAI-ES ^[28] | 是 | 否 | 演化策略 | 策略优化 (DDPG) | 无 | 无混合搜索 | 无 |
| DRL-GA ^[29] | 是 | 否 | 遗传算法 | 值优化 (Q-learning) | 无 | 无混合搜索 | 无 |
| SGES ^[30] | 是 | 否 | 演化策略 | 策略优化 (DDPG) | 无 | 无混合搜索 | 无 |
| G2N ^[31] | 是 | 否 | 遗传算法 | 策略优化 (A2C) | 无 | 混合搜索 | 外强化, 内演化优化 |
| EPG ^[32] | 否 | 是 | 演化策略 | 策略优化 (DDPG) | 无 | 混合搜索 | 外演化, 内强化优化 |
| ERL ^[33] | 否 | 是 | 演化策略 | 策略优化 (DDPG) | 无 | 混合搜索 | 外演化, 内强化优化 |
| CERL ^[34] | 否 | 是 | 遗传算法 | 策略优化 (TD3) | 无 | 混合搜索 | 外演化, 内强化优化 |
| EDO-CS ^[35] | 否 | 是 | 演化策略 | 策略优化 (DDPG) | 有 | 无混合搜索 | 无 |
| CEM-RL ^[36] | 否 | 是 | 交叉熵算法 | 策略优化 (DDPG) | 无 | 混合搜索 | 外演化, 内强化优化 |
| EDRL-IM ^[37] | 否 | 是 | 遗传算法 | 值优化 (Q-learning) | 无 | 混合搜索 | 外演化, 内强化优化 |
| SEDRL-IM | 否 | 是 | 遗传算法 | 值优化 (Q-learning) | 无 | 混合搜索 | 外演化, 内强化优化 |

为了解决上述问题, 一些基于全局探索的 IM 算法被提出. 如 SA 算法^[19]通过模拟退火算法以一定概率保留当前局部最优解, 从而避免陷入局部最优. DPSO 算法通过粒子群算法以多组解作为起点对解空间进行多路径探索, 从而拥有全局寻找近似最优解的能力. 然而, 该类方法难以在有限迭代次数内收敛到较优解.

为了解决上述问题, 一些混合搜索算法被陆续提出. 如 MA-IM 算法^[25]结合遗传算法的全局搜索特性以及度贪心的局部搜索特性, 能搜索获得较优的影响种子节点. CMA-IM 算法^[27]结合遗传算法的全局搜索特性以及社区度贪心搜索, 能有效避免导致陷入局部最优. EDRL-IM^[37]通过 DQN 来评估网络节点的影响力, 并结合演化算法全局搜索和 DRL 随机梯度下降局部搜索, 实现对单网络 IM 问题解的有效快速搜索. 本文提出的 SEDRL-IM 算法在 EDRL-IM 算法的基础上引入 DNE-SBP 降维算法, 来捕获符号化网络的正负拓扑特征, 通过结合 DQN 与马尔可夫决策有效评估符号化网络中节点的正负影响力, 并结合演化算法与深度强化学习对符号化网络中的 IM 问题进行高效求解. 表 3 展示了本文所提 SEDRL-IM 算法与相关 IM 算法之间的异同.

表 3 本文算法 SEDRL-IM 与相关的影响最大化算法的异同

| 算法 | 主要框架 | 目标函数 (影响力函数) | 传播模型 | 优化模型 | 局部搜索策略 | 全局优化策略 | 处理网络类型 | 特征学习方法 |
|----------------|-------------|--------------|------|------|-----------------------------|--------|--------|--------|
| CELF CELF++ | 贪心选择算法 | 多跳 | IC | 离散 | 贪心选择度最大节点 | 无 | 无符号网络 | 无 |
| S2V-DQN | 深度神经网络 | 两跳 | IC | 连续 | 马尔可夫决策贪心选取评估分数最大节点 | 无 | 无符号网络 | 无 |
| CI | 贪心选择算法 | 多跳 | IC | 离散 | 贪心选择社区影响力最大节点 | 无 | 无符号网络 | 无 |
| Page Rank | 随机游走算法 | 两跳 | IC | 离散 | 贪心选择PR值最高节点 | 无 | 符号化网络 | 无 |
| SA | 模拟退火算法 | 多跳 | IC | 离散 | 无 | 模拟退火 | 无符号网络 | 无 |
| DPSO | 粒子群优化算法 | 多跳 | IC | 离散 | 无 | 粒子群优化 | 符号化网络 | 无 |
| MA-IM | 演化算法 | 两跳 | IC | 离散 | 贪心选择度最大节点 | 演化算法 | 无符号网络 | 无 |
| CMA-IM | 演化算法 | 两跳 | IC | 离散 | 贪心选择社区影响力最大节点 | 演化算法 | 无符号网络 | 无 |
| EDRL-IM | 演化算法和深度神经网络 | 两跳 | IC | 连续 | 马尔可夫决策贪心选取评估分数最大节点 | 演化算法 | 无符号网络 | 表示学习 |
| SEDRL-IM | 演化算法和深度神经网络 | 两跳 | SNIC | 连续 | 马尔可夫决策贪心选取正评估分数最大且负评估分数最小节点 | 演化算法 | 符号网络 | 符号表示学习 |

4 基于连续参数优化的影响最大化求解模型

现有算法主要将 IM 问题建模为基于种子节点选取的离散组合优化问题, 并采用贪心策略来选取种子节点. 然而, 离散组合 IM 优化问题具有 NP-hard 特性, 且基于贪心策略的求解算法在大规模网络中计算代价较高, 导致这些算法无法有效处理符号网络的 IM 问题. 本文通过引入基于 SNIC 模型下的 DQN 种子节点选取过程, 将离散的 IM 问题转化为更易于优化的 DQN 连续参数优化问题. 具体而言, DQN 首先将符号网络的特征表示向量作为神经网络的输入, 输出选取所有节点的 Q 值. 然后, 通过马尔可夫决策选择 Q 值最高的节点作为种子节点参与影响传播. 接下来, DQN 通过影响分数根据误差值的梯度, 引入随机梯度下降算法反向传播更新自身网络权重来生成近似最优的种子集, 让其参与正负影响传播, 使得符号网络的影响分数达到最大化.

4.1 网络降维模块

网络降维^[58]是指通过神经网络提取出离散拓扑网络的低维潜在特征. 本文基于 DNESBP 算法^[59], 对符号网络进行网络降维, 获得一个 d 维的特征表示向量 $H = [h_{ij}]_{n \times d}$, $h_{ij} \in [-1, 1]$. DQN 可通过输入 H , 充分学习符号网络的正负拓扑特性^[59]和结构平衡特性^[60].

DNESBP 算法首先采用由 l 层神经元构成的神经网络来构建自编码器对符号网络的邻接矩阵 A 进行降维, 并获得表示学习向量.

$$H^{(i)} = \tanh(X^{(i)}(W^{(i)})^T + B^{(i)}), i = 1, \dots, l \quad (6)$$

其中, $X^{(0)} = A$ 且 $H^{(i)}, i \neq 0$ 为通过第 i 次降维得到的向量, $H = H^{(l)} = [h_{ij}]_{n \times d}$ 表示最终降维得到的特征表示向量. \tanh 函数为用于生成正负特征值的激活函数, $W^{(i)}$ 为自编码器第 i 层的网络权重矩阵, $B^{(i)}$ 为自编码器第 i 层网络权重的偏置向量.

接下来, DNESBP 算法通过构建 l 层神经网络的解码器, 对 H 进行重构, 并获得重构矩阵 \hat{A} .

$$\hat{X}^{(i)} = \tanh(\hat{H}^{(i)}(\hat{W}^{(i)})^T + \hat{B}^{(i)}), i = 1, \dots, l \quad (7)$$

其中, $\hat{H}^{(0)} = H^{(0)}$ 且 $\hat{X}^{(l)} = \hat{A}$, $\hat{W}^{(i)}$ 为解码器第 i 层的网络权重矩阵, $\hat{B}^{(i)}$ 为解码器第 i 层网络权重的偏置向量.

进一步地, DNESBP 算法通过 F 范数计算重构损失 $\|\hat{A} - A\|_F^2$ 来监督降维过程, 从而使得 H 充分捕获并学习符号网络的正负拓扑特性. 其中详细的损失函数和原理请参考文献 [59].

4.2 DQN 模块

在本文所提 SEDRL-IM 算法中, DQN^[61]网络结构由两层神经网络构成. SEDRL-IM 算法以 H 作为 DQN 的输入, 输出 $Q = [q_1, q_2, \dots, q_n]$, 其中 q_i 表示节点 v_i 对应的 Q 值^[51]. 在每次选取种子节点时, $S = [s_1, s_2, \dots, s_n]$ 被用于表示当前的种子节点序列向量, 其中 $s_i = 1$ 与 $s_i = 0$ 分别表示节点 v_i 已被选为和未被选为种子节点.

值得注意的是, S 的状态会根据种子节点的选取过程而发生变化. 针对这种情况, 本文将种子节点序列向量 S 与特征表示向量 H 通过拼接, 生成能表示当前符号网络真实状态的新向量 $[H, S]$. 这使得 DQN 在选取种子节点时, 能动态地学习每个时刻种子节点集的特征, 进而更好地模拟符号网络中动态变化的影响扩散过程^[62].

DQN 以 $(d+1)$ 维的向量 $[H, S]$ 为输入, 通过其网络结构计算出 Q 值.

$$Q = \text{DQN}(W_1, W_2, [H, S]) = \text{ReLU}([H, S] \cdot W_1) \cdot W_2 \quad (8)$$

其中, $W_1 \in [-1, 1]_{(d+1) \times z}$ 表示神经网络第 1 层的连续权重矩阵, $W_2 \in [-1, 1]_{z \times 1}$ 表示第 2 层的连续权重向量, $(d+1)$ 和 z 分别表示第 1 层和第 2 层的神经元维数, ReLU 函数为消除负特征值的激活函数.

4.3 马尔可夫决策模块

根据 DQN 模块输出的 Q 值, SEDRL-IM 算法可通过马尔可夫决策^[63]来迭代选取 Q 值最高的节点作为种子节点 (具体描述如算法 1 所示). 具体而言: 在第 $(t-1)$ 次决策过程中, 在获得种子节点序列 S_t 后, 马尔可夫决策过程将 $[H, S_t]$ 作为 DQN 新的输入, 并通过贪心策略 (如以下公式所示) 得到 Q 值最高的新动作 a_t , 并选取相应节点作为该决策过程生成的种子节点.

$$a_t = \operatorname{argmax}_a Q(S_t, a) \quad \text{s.t.} \quad Q(S_t, a) = \text{DQN}(W_1, W_2, [H, S_t]) \quad (9)$$

其中, a 表示马尔可夫决策选取种子节点时所有可能发生的“动作”构成的集合, a_t 则表示为马尔可夫决策选取 Q 值最高的节点作为种子节点时的“动作”。

在第 t 次决策中, 若节点 v_k 具有最高的 Q 值, 马尔可夫决策则将其选为种子节点并令 $s_k = 1$, 并更新种子节点序列 S_{t+1} , 然后根据如下 $O(S)$ 函数将当前种子节点序列 $S_{t+1} = [s_1, s_2, \dots, s_n]$ 对应转换为种子节点集 $\mathcal{S}_{t+1} = \{v_i \in V : x_i = 1\}$, 并计算出当前符号网络的影响分数 $\mathbf{I}(\mathcal{S}_{t+1})$ 。

$$\mathbf{I}(\mathcal{S}_{t+1}) = \mathbf{I}(O(S_{t+1})) \quad \text{s.t.} \quad O(S) = \{v_i \in V : s_i = 1\} \quad (10)$$

算法 1. MSS (Markov seed selection) 算法。

输入: 参数 (W_1, W_2) , 种子节点数 k 以及特征表示向量 H ;
输出: 种子节点序列 \mathcal{S} , 影响分数 $\mathbf{I}(\mathcal{S})$, 四元组 $\{(S_t, a_t, r_t, S_{t+1})\}_k$ 。

1. 初始化: $t \leftarrow 1$ 以及 $S_t \leftarrow []$
2. 设置 DQN 的网络权重 (W_1, W_2)
3. while $t \leq k$ do
4. DQN 将 $[H, S_t]$ 作为输入, 并通过正向传播输出 $Q(S_t, a)$
5. 基于公式 (9) 选择 Q 值最高的节点作为种子节点, 并获得 S_{t+1}
6. 计算 $r_t = \mathbf{I}(O(S_{t+1})) - \mathbf{I}(O(S_t))$
7. end while
8. $\mathcal{B} \leftarrow \{(S_t, a_t, r_t, S_{t+1})\}_k$
9. return $\mathcal{S}, \mathbf{I}(\mathcal{S})$

接下来, 马尔可夫决策计算决策“奖励” r_t , 即采取“动作” a_t 前后的影响分数之差, 如下所示:

$$r_t = \mathbf{I}(O(S_{t+1})) - \mathbf{I}(O(S_t)) \quad (11)$$

其中, $\mathbf{I}(O(S_t))$ 表示在 SNIC 模型下选取 S_t 作为种子节点集来参与正负影响传播得到的影响分数。

在每次决策过程中, 马尔可夫决策将获得的数据以四元组 (S_t, a_t, r_t, S_{t+1}) 的形式存入缓存池 \mathcal{B} 中。为了加速 DQN 的网络权重优化, SEDRL-IM 算法在进行 n 次迭代训练后, 从缓存池 \mathcal{B} 中采样 n 组四元组 (S_t, a_t, r_t, S_{t+1}) , 通过随机梯度下降算法反向更新 DQN 的网络权重 W_1 和 W_2 。

通过 DQN 模块和马尔可夫决策模块, SEDRL-IM 算法可将符号网络选取 k 个种子节点的离散组合优化 IM 问题转化为如下对 DQN 的连续参数优化问题。

$$\max \mathbf{I}(\mathcal{S}) = \max \mathbf{I}(O(S)) = \mathbf{I}(W_1, W_2) \quad \text{s.t.} \quad W_1, W_2 \in \text{DQN} \quad (12)$$

后文图 3 展示了基于 DQN 模块进行马尔可夫决策来选取种子节点的具体流程示意图。图 3 中, S_t 记录在 t 时刻每个节点的种子节点状态, q_i 记录第 i 个节点的 Q 值。具体过程描述如下: 首先, 马尔可夫决策采用 DNESBP 算法对符号网络 \mathcal{G} 进行降维, 获得特征表示向量 H 。其次, 在每次选取种子节点时, DQN 以 $[H, S]$ 为输入, 输出 Q 值。接下来, 马尔可夫决策过程根据 Q 值决策出种子节点。如图 3 所示, 在 t 时刻, 马尔可夫决策通过 $a_t = 3$ 选出 $q_3 = 0.88$ 的节点 v_3 作为种子节点来参与图 2 的影响传播, 从而得到影响分数 $\mathbf{I}(O(S_t)) = 2.54$ 以及决策奖励 $r_t = \mathbf{I}(O(S_t)) - 0 = 2.54$ 。在选取节点 v_3 为种子节点并得到新的种子节点序列 S_{t+1} 后, 马尔可夫决策将 $[H, S_{t+1}]$ 输入 DQN, 并输出新的 Q 值, 进而在 $(t+1)$ 时刻基于 $a_{t+1} = 1$ 选取 $q_1 = 0.85$ 的节点 v_1 作为新的种子节点, 并得到 $\mathbf{I}(O(S_{t+1})) = 6.10$ 以及 $r_{t+1} = \mathbf{I}(O(S_{t+1})) - \mathbf{I}(O(S_t)) = 3.56$ 。

5 基于演化深度强化学习的影响最大化求解模型

基于 DQN 求解 IM 问题的算法因为梯度下降法的贪心特性, 容易陷入局部最优解。本文受到 EDRL-IM 算法启发, 提出求解符号网络 IM 问题的演化深度强化学习 SEDRL-IM 算法 (框架如算法 2 所示)。

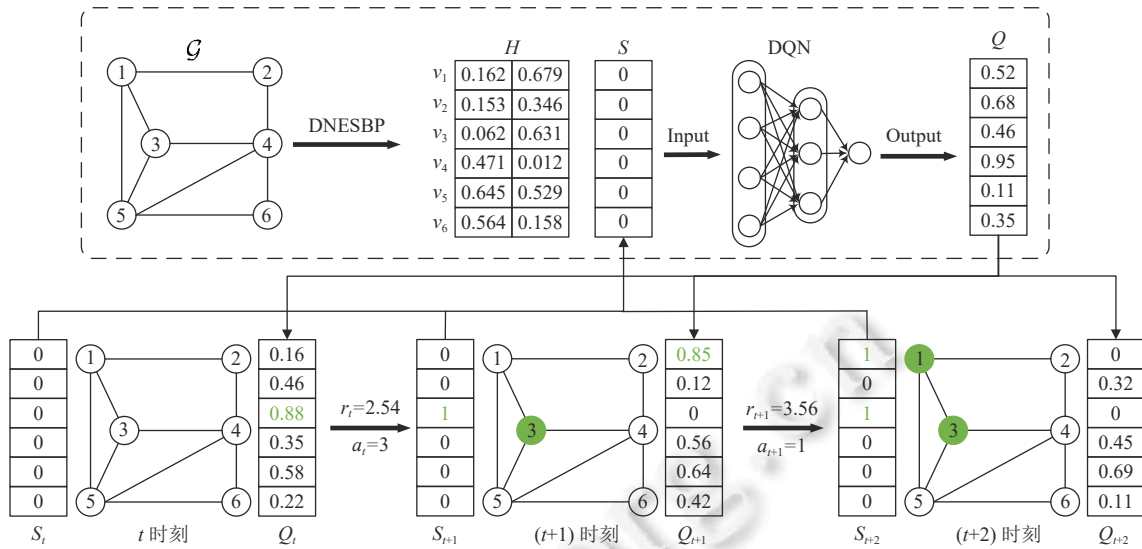


图 3 基于 DQN 的马尔可夫决策流程

算法 2. SEDRL-IM 算法框架.

输入: 初始种群大小 n_o , 种群大小 n_p , 最大迭代数 n_g , 交叉变异概率 $p_{crossover}$, 随机变异概率 $p_{mutation}$.

1. $\{\mathcal{P}_i(0)\}_{n_o} \leftarrow \text{Initialization}(n_o)$
2. for $q=0$ to n_g-1 do
3. $\mathcal{P}(q)_{EA} \leftarrow \text{EA}(n_p, \mathcal{P}(q), p_{crossover}, p_{mutation})$
4. $\mathcal{P}(q)_{DRL} \leftarrow \text{NWU}(\mathcal{P}(q)_{EA})$
5. $\mathcal{S}, \mathbf{I}(\mathcal{S}) \leftarrow \text{MSS}(\mathcal{P}(q) \cup \mathcal{P}(q)_{EA} \cup \mathcal{P}(q)_{DRL}, k, H)$
6. 从 $\mathcal{P}(q) \cup \mathcal{P}(q)_{EA} \cup \mathcal{P}(q)_{DRL}$ 中取出 $\mathbf{I}(\mathcal{S})$ 最高且大小为 $n_p/2$ 的种群 $\mathcal{P}(q)_{Opt}$
7. 从种群集合 $\mathcal{P}(q) \cup \mathcal{P}(q)_{EA} \cup \mathcal{P}(q)_{DRL} - \mathcal{P}(q)_{Opt}$ 中随机取出大小为 $n_p/2$ 的种群 $\mathcal{P}(q)_{Random}$
8. $\mathcal{P}(q+1) \leftarrow \mathcal{P}(q)_{Opt} \cup \mathcal{P}(q)_{Random}$
9. end for

SEDRL-IM 算法将策略视为可演化的个体, 结合演化算法的无梯度优化与种群搜索特性以及强化学习的梯度优化搜索特性, 优化策略种群. 其中, 每个策略个体表示一个可优化的 DQN 网络权重序列. 具体而言, 在每次迭代中, SEDRL-IM 算法通过遗传算法的交叉和变异对 DQN 网络权重参数进行全局搜索; 通过强化学习的随机梯度下降算法对种群中的一个较优个体进行局部搜索, 并引入 DRL 思想的 n -step Q-learning 技术, 利用缓存池存储的四元组数据反向传播更新 DQN 的网络权重, 加速 DQN 的训练收敛和演化; 通过马尔可夫决策、SNIC 模型以及两跳影响传播分数 $\mathbf{I}(\mathcal{S})$ 来评估与筛选 DQN 网络; 通过贪心策略选择适应值 $\mathbf{I}(\mathcal{S})$ 最高的部分种群并同时随机选取另一部分种群作为下一次进行演化的种群, 保障种群的演化以及多样随机性. 后续章节将逐步介绍 SEDRL-IM 算法的具体实现细节.

5.1 DQN 种群的演化过程

SEDRL-IM 算法将每个个体 (或基因序列) 视作 DQN 网络权重序列策略, 试图利用遗传算法的演化操作 (如初始化、交叉、变异、选择等操作) 对一组种群序列同时进行迭代优化, 展开对解空间的全局探索, 进而找出能使得 $\mathbf{I}(\mathcal{S})$ 最大化的网络权重序列.

5.1.1 种群解或“基因序列集”的表达

SEDRL-IM 算法使用 $\mathcal{P} = \{\mathcal{P}_i(q)\}_{n_p}$ 表示第 q 次迭代演化过程中的 DQN 种群解集合 (基因序列集), 其中 n_p 表示基因序列的数量. 那么 $\mathcal{P} = \{\mathcal{P}_i(q)\}_{n_p}$ 中任一 DQN 解 (基因序列) $\mathcal{P}_i(q)$ 皆对应 DQN 的一个网络权重序列 (W_1^i, W_2^i) , 可表达如下:

$$\mathcal{P}_i(q) = (W_1^i, W_2^i), W_1^i \in [w_{ju}^i]_{(d+1) \times z}, W_2^i \in [w_{ju}^i]_{z \times 1} \quad (13)$$

其中, $W_1^i = [w_{ju}^i]_{(d+1) \times z}, w_{ju}^i \in [-1, 1]$ 和 $W_2^i = [w_{ju}^i]_{z \times 1}, w_{ju}^i \in [-1, 1]$ 分别表示 DQN 中第 1 层的网络权重矩阵和第 2 层的网络权重向量.

5.1.2 DQN 种群的初始化操作

SEDRL-IM 算法首先对 DQN 网络权重序列种群进行随机初始化, 得到初始 DQN 种群解集合 $\mathcal{P} = \{\mathcal{P}_i(0)\}_{n_o}$. 该随机初始化操作保障了初始 DQN 种群解的多样性, 其具体执行过程如算法 3 所示.

算法 3. Initialization 算法.

输入: 初始种群大小 n_o ;

输出: 初始种群 \mathcal{P} .

1. 令 $\mathcal{P} = \{\mathcal{P}_i(0)\}_{n_o}$ 且 $\mathcal{P}_i(0) = (W_1^i, W_2^i), i = 1, 2, \dots, n_o$
2. for $i = 1$ to n_o do
3. for w_{ju}^i in W_1^i do
4. 令 $w_{ju}^i = \text{Random}()$ 来随机分配位于 -1 到 1 的值
5. end for
6. for w_{ju}^i in W_2^i do
7. 令 $w_{ju}^i = \text{Random}()$ 来随机分配位于 -1 到 1 的值
8. end for
9. end for

具体而言, 对于任意 $\mathcal{P}_i(0) = (W_1^i, W_2^i)$ 中每个维度的值 w_{ju}^i 和 w_{ju}^i , SEDRL-IM 算法采用如下函数来随机赋予位于 -1 到 1 的值作为初始值.

$$w_{ju}^i = \text{Random}() \quad (14)$$

其中, Random 函数用于生成位于 -1 到 1 的随机值.

5.1.3 DQN 种群的演化操作

在获得初始 DQN 种群 $\mathcal{P} = \{\mathcal{P}_i(0)\}_{n_o}$ 后, SEDRL-IM 算法将影响分数 $\mathbf{I}(\mathcal{S})$ 作为适应值函数对 DQN 进行迭代演化如交叉、变异与选择操作, 具体描述如算法 4 所示.

(1) 交叉: 该操作首先基于适应值 $\mathbf{I}(\mathcal{S})$ 对种群集合 $\mathcal{P}(q)_C$ 进行排序, 其次, 依照 50% 比例来将 $\mathcal{P}(q)_C$ 划分为具有“优秀基因”的集合 $\mathcal{P}_i^b(q)_C = [x_i]_{(d+1) \times z+z}$ 和具有“一般基因”的集合 $\mathcal{P}_j^w(q)_C = [x_j]_{(d+1) \times z+z}$. 接下来, 该操作为每对将要交叉的种群 $\mathcal{P}_i^b(q)_C$ 和 $\mathcal{P}_j^w(q)_C$ 随机分配一个基因置换序列 $M = \{0, 1\}_{(d+1) \times z+z}$, 用于执行单点交叉操作, 得到新的种群 $\mathcal{P}_i(q)'_C = [x'_i]_{(d+1) \times z+z}$ 和 $\mathcal{P}_j(q)'_C = [x'_j]_{(d+1) \times z+z}$. 对于任意一个基因位 u , 该交叉过程可表达如下:

$$\begin{cases} x'_i = x_i, x'_j = x_j, & \text{if } M_u = 0 \\ x'_i = x_j, x'_j = x_i, & \text{if } M_u = 1 \end{cases} \quad (15)$$

算法 4. EA 算法.

输入: 种群大小 n_p , 父代种群 (q) , 交叉变异概率 $p_{\text{crossover}}$, 随机变异概率 p_{mutation} ;

输出: 后代种群 $\mathcal{P}(q)_{\text{EA}}$.

1. 将 $\mathcal{P}(q)$ 随机划分为两个种群 $\mathcal{P}(q)_C, \mathcal{P}(q)_M$
2. $\mathbf{S}, \mathbf{I}(\mathbf{S}) \leftarrow \text{MSS}((q)_C, k, H)$
3. 根据 $\mathbf{I}(\mathbf{S})$ 对 $\mathcal{P}(q)_C$ 进行降序排序
4. 将 $\mathcal{P}(q)_C$ 中前 50% 的种群 $\{\mathcal{P}_i^b(q)_C\}$ 和后 50% 的种群 $\{\mathcal{P}_j^w(q)_C\}$ 组成 $\{(\mathcal{P}_i^b(q)_C, \mathcal{P}_j^w(q)_C)\}$
5. for $\{(\mathcal{P}_i^b(q)_C, \mathcal{P}_j^w(q)_C)\}$ 中的每一对 $(\mathcal{P}_i^b(q)_C, \mathcal{P}_j^w(q)_C)$ do
6. if 随机生成一个数 p_1 且 $p_1 \geq p_{\text{crossover}}$ do
7. 为 $\mathcal{P}_i^b(q)_C$ 和 $\mathcal{P}_j^w(q)_C$ 中每一对基因 (x_i, x_j) 随机分配一个值 0 或 1, 从而构成序列 M
8. for $(\mathcal{P}_i^b(q)_C, \mathcal{P}_j^w(q)_C)$ 中的每一对 (x_i, x_j) do
9. if (x_i, x_j) 对应序列 M 中的值为 1 do
10. 置换 x_i 和 x_j 的值, 得到新的 $(\mathcal{P}_i(q)'_C, \mathcal{P}_j(q)'_C)$
11. end if
12. end for
13. end if
14. end for
15. 将 $\{(\mathcal{P}_i(q)'_C, \mathcal{P}_j(q)'_C)\}$ 汇集为 $\mathcal{P}(q)'_C$
16. for $\mathcal{P}(q)_M$ 中的每一个 $\mathcal{P}_i(q)_M$ do
17. for $\mathcal{P}_i(q)_M$ 中的每一个权重参数 y do
18. if 随机生成一个数 p_2 且 $p_2 \geq p_{\text{mutation}}$ do
19. $y = N(-1, 1)$, 得到新的 $\mathcal{P}_i(q)'_M$
20. end if
21. end for
22. end for
23. $\mathbf{S}, \mathbf{I}(\mathbf{S}) \leftarrow \text{MSS}(\mathcal{P}(q)'_C \cup \mathcal{P}(q)'_M, k, H)$
24. 从 $\mathcal{P}(q)'_C \cup \mathcal{P}(q)'_M$ 中选出 n_p 个适应值 $\mathbf{I}(\mathbf{S})$ 最高的种群作为 $\mathcal{P}(q)_{\text{EA}}$

(2) 变异: 该操作对于每个将要变异的种群 $\mathcal{P}_i(q)_M = [y_i]_{(d+1) \times z}$ 中的基因 y_i , 通过如下 N 函数进行随机基因突变, 进而增强种群基因序列的多样性, 并得到 $\mathcal{P}_i(q)'_M$:

$$y_i = N(-1, 1) \quad (16)$$

其中, N 函数是用于生成位于 -1 到 1 的值的正态随机函数, 且服从于概率密度函数 $e^{-y^2/2} / \sqrt{2\pi}$.

(3) 选择竞争: 该操作基于贪心策略从 $\mathcal{P}(q)'_C \cup \mathcal{P}(q)'_M$ 中筛选出适应值 $\mathbf{I}(\mathbf{S})$ 最高的种群 $\mathcal{P}(q)_{\text{EA}}$, 进而将具有“优秀基因”的 $\mathcal{P}(q)_{\text{EA}}$ 作为下一代需要演化的种群.

图 4 展示了 DQN 种群的具体演化流程. 图中采用权重序列表示 DQN 的基因序列, 权重序列中的每一权重代表一个基因. 在交叉过程中, 交叉模块为任意两个权重序列 $\mathcal{P}_i(q)$ 和 $\mathcal{P}_j(q)$ 中的每对权重生成一个值为 0 或 1 的随机数用于判断是否进行置换. 如在图 4 中, 交叉模块为 $(\mathcal{P}_i(q), \mathcal{P}_j(q))$ 中第 2 对权重 (0.2, 0.8) 和倒数第 2 对权重 (0.3, 0.7) 生成的随机数皆为 1, 因此将置换这两对权重并保留其他权重, 得到新的权重序列 $\mathcal{P}_i(q)'$ 和 $\mathcal{P}_j(q)'$. 在变异过程中, 变异模块将对 $\mathcal{P}_i(q)$ 中的所有权重依次采用 N 函数进行随机正态突变. 如在图 4 中, 变异模块对 $\mathcal{P}_i(q)$ 的第 1 个权重值 0.5 输入 N 函数来生成新的权重值 0.9.

5.2 基于 n -step Q-learning 加速 DQN 的训练和演化

对大规模网络而言, DQN 的网络权重序列数非常庞大, 单一的演化操作很难在有限的迭代次数内找到较优解, 且收敛速度较慢. 为了解决上述问题, 本文基于 DRL 思想, 提出基于 n -step Q-learning 技术^[64]来进一步加速 DQN 的训练和演化, 其具体描述如算法 5 所示, 流程如图 5 所示.

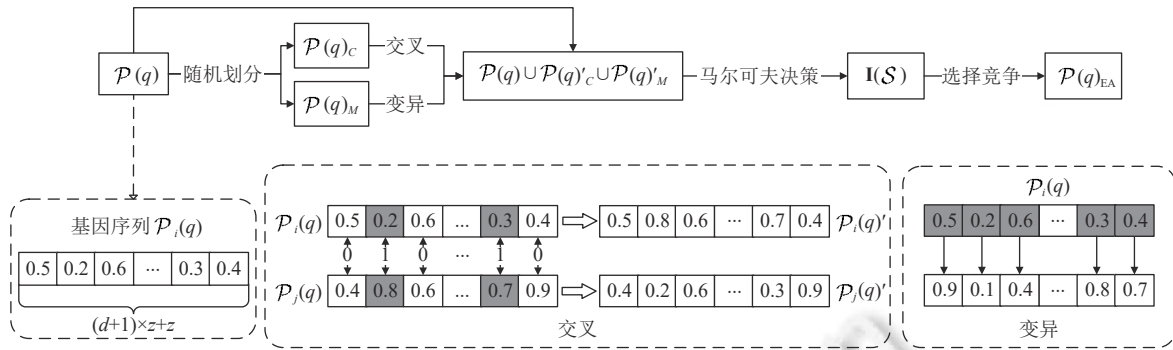


图4 DQN 种群的网络权重序列迭代演化流程

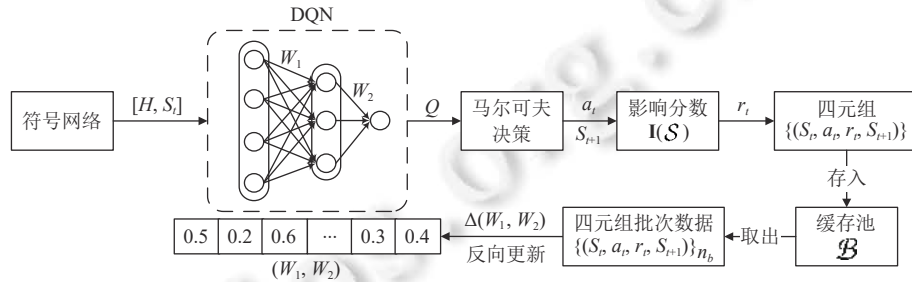


图5 基于 n -step Q-learning 技术反向更新 DQN 的过程

SEDRL-IM 算法在马尔可夫决策迭代选取种子节点过程中, 将迭代 n_q 次训练中的历史四元组数据 $\{(S_t, a_t, r_t, S_{t+1})\}_{n_q}$ 存储进缓存池 \mathcal{B} 中. 在迭代训练 n 次后, SEDRL-IM 算法基于 n -step Q-learning 技术从缓存池中分批次采样四元组数据 $\{(S_t, a_t, r_t, S_{t+1})\}_{n_b}$, 计算 DQN 的目标损失函数 $L(W_1, W_2)$:

$$L(W_1, W_2) = E_{C \in \mathcal{B}} \left[\left(\sum_{i=0}^{n_b-1} r_{t+i} + \gamma \max_{a^*} Q(S_{t+n}, a^*; W_1, W_2) - Q(S_t, a_t; W_1, W_2) \right)^2 \right] \quad (17)$$

其中, γ 表示用于决定奖励 r_{t+i} 重要程度的超参数, E 表示期望函数. \mathcal{B} 表示用于存储四元组数据的缓存池, $C = \{(S_t, a_t, r_t, S_{t+1})\}_{n_b}$ 表示每一批次从缓存池 \mathcal{B} 取出来更新 DQN 的四元组数据, 其批次大小为 n_b .

基于上述目标损失函数 $L(W_1, W_2)$, SEDRL-IM 算法可进一步计算损失函数的梯度 $\Delta(W_1, W_2)$, 并采用如下随机梯度下降算法来反向更新 DQN 的网络权重 (W_1, W_2) , 从而加快 DQN 的网络权重向较优权重收敛拟合:

$$\begin{cases} (W_1, W_2)' = (W_1, W_2) - \Delta(W_1, W_2) \\ \text{s.t. } \Delta(W_1, W_2) = \alpha \cdot \left[\sum_{i=0}^{n_b-1} r_{t+i} + \gamma \max_{a^*} Q(S_{t+n}, a^*; W_1, W_2) - Q(S_t, a_t; W_1, W_2) \right]^2 \nabla_{(W_1, W_2)} Q(S_t, a_t; W_1, W_2) \end{cases} \quad (18)$$

其中, α 表示反向更新 DQN 时的学习率, $\nabla_{(W_1, W_2)}$ 表示 (W_1, W_2) 的偏移量.

算法 5. NWU (network weight update) 算法.

输入: 缓存池 \mathcal{B} , 数据批次大小 n_b , 种群 $\mathcal{P}(q)_{EA}$;

输出: 种群 $\mathcal{P}(q)_{DRL}$.

1. 从缓存池 \mathcal{B} 中随机取出大小为 n_b 的数据 $C = \{(S_t, a_t, r_t, S_{t+1})\}_{n_b}$
2. 根据公式 (17) 算出损失值 $L(W_1, W_2)$
3. 根据公式 (18) 算出 $L(W_1, W_2)$ 的梯度 $\Delta(W_1, W_2)$

4. 基于 $\Delta(W_1, W_2)$ 反向更新 $\mathcal{P}(q)_{EA}$ 中最优种群对应的 (W_1, W_2) , 从而得到新的种群 $\mathcal{P}(q)_{DRL}$

5.3 算法复杂度分析

由算法 2 所示, 所提算法 SEDRL-IM 主要由马尔可夫影响力节点选择 MSS()、初始化 Initialization()、演化全局搜索 EA() 以及增强学习局部搜索 NWU() 一系列操作构成. 现对各操作的计算复杂度分析如下.

MSS(): 由算法 1 可知, 马尔可夫影响力节点选择主要选择 k 个影响力节点, 其中每个影响力节点选择需要通过 DQN 的正向传播获得累计奖励 Q 值以及目标奖励 r 增量值. 每次 DQN 正向传播的计算复杂度为 $O((d+1) \times z + z)$, 其中 $(d+1)$ 与 z 分别表示优化 DQN 的第 1 层和第 2 层的神经元维数. 每次目标奖励 r 增量值的计算复杂度为 $O(\bar{k} \times \bar{k})$, 其中 \bar{k} 为处理网络的节点平均连接度. 因此, 整个影响力节点选择过程的时间复杂度为 $O(k \times [(d+1) \times z + z + \bar{k} \times \bar{k}])$, 可简写为 $O(k \times (d \times z + \bar{k} \times \bar{k}))$.

Initialization(): 由算法 3 可知, 初始化操作主要是对 n_0 个种群解中的每个 DQN 网络权重进行随机初始化操作. 因此, 其计算复杂度可表达为 $O(n_0 \times [(d+1) \times z + z])$, 可简写为 $O(n_0 \times d \times z)$, 其中 n_0 表示种群解规模.

EA(): 由算法 4 可知, 演化全局搜索操作主要在每次迭代中对种群解中每对父代解进行 0-1 交叉操作, 对每个个体每基因位进行随机变异操作. 因此, 该演化操作在每次迭代中的交叉变异计算复杂度为 $O(n_0 \times [(d+1) \times z + z])$. 此外, 该演化操作在每次迭代中对每个解需要执行影响力节点选择过程操作, 其计算复杂度为 $O(n_0 \times k \times [(d+1) \times z + z + \bar{k} \times \bar{k}])$. 因此, 演化全局搜索操作在每次迭代中的计算复杂度为 $O(n_0 \times k \times [(d+1) \times z + z + \bar{k} \times \bar{k}] + n_0 \times [(d+1) \times z + z])$, 可简写为 $O(n_0 \times k \times (d \times z + \bar{k} \times \bar{k}))$.

NWU(): 由算法 5 可知, 增强学习局部搜索操作利用规模为 n_b 的批次数据对种群解中一个较好解进行 DQN 权重反向传播优化, 其中每次传播需要对权重方差进行计算公式 (17) 与公式 (18). 因此, 增强学习局部搜索操作在每次迭代中的计算复杂度为 $O(n_b \times [(d+1) \times z + z + \bar{k} \times \bar{k}])$, 可简写为 $O(n_b \times (d \times z + \bar{k} \times \bar{k}))$.

SEDRL-IM 总共需要迭代 n_g 次, 因此总计算复杂度可表达为 $O(n_g \times (n_0 \times k + n_b) \times (d \times z + \bar{k} \times \bar{k}))$.

6 实验设计

本节对 SEDRL-IM 算法和其余 5 个经典 IM 算法在两个基准符号网络数据集以及 7 个来自真实世界的符号网络数据集上进行比较测试, 并从影响分数和种子节点分布等方面评估比较各个算法的性能.

6.1 数据集

6.1.1 基准网络数据集

为了验证不同社区分布以及度分布对符号网络影响传播过程的影响, 本节采用 GN 和 LFR 两种模块化的符号网络来模拟现实社交网络, 并对所有影响最大化算法进行测试, 分析不同算法在求解 IM 问题时对不同程度异构的符号网络的处理性能.

GN 基准网络^[65]是一类社区均匀分布的模块化网络, 其具有 128 个节点且每个节点的度为 16. 为了遵循符号网络的结构平衡特性, 实验中 GN 基准网络的社区内负边比例和正边比例均被设置为 15%. 在该基准网络中, 所有节点平均地分布在 4 个社区中, 且所有社区之间具有数量比例为 ω 的边, 其中 ω 为超参数. ω 值越大表明网络越复杂. 为了测试所有算法在不同复杂程度下符号网络的处理性能, 本节将在 $\omega = 0.1$ 和 $\omega = 0.4$ 的 GN 基准网络上对 SEDRL-IM 算法和其他算法进行测试.

LFR 基准网络^[66]是在 GN 基准网络的基础上, 考虑节点度和社区规模异质的扩展网络. 在该类网络中, 其节点度和社区规模服从幂律分布且分别受到参数 β 和 τ 的控制. 相比于 GN 基准网络, LFR 基准网络能更真实地模拟现实异构的符号网络. 因此, 本节将在 $\omega = 0.1$ 和 $\omega = 0.3$ 的 LFR 基准网络上对 SEDRL-IM 算法和其他算法进行测试, 其中每个 LFR 网络的节点数设置为 500, $\beta = 1$, $\tau = 2$.

6.1.2 真实网络数据集

7 个来自选举合作、商业交易以及生物反馈等现实应用领域的真实网络将被用来验证所提 SEDRL-IM 算法在真实网络系统上的处理效率和性能有效性. 7 个真实符号网络的基本结构特性如表 4 所示.

表 4 真实网络的详细信息

| 真实网络 | 节点数 | 连边数 | 正边数 | 负边数 |
|------------|------|--------|-------|-------|
| War | 166 | 1433 | 1295 | 138 |
| EGFR | 329 | 779 | 515 | 264 |
| Macrophage | 678 | 1425 | 947 | 478 |
| Yeast | 690 | 1080 | 860 | 220 |
| E.coli | 1461 | 3215 | 1879 | 1336 |
| Soc-alpha | 3783 | 24186 | 22650 | 1536 |
| Wiki | 7115 | 100693 | 78603 | 22090 |

War 网络是从军事战争项目^[67]中提取出的军事关系网络,其反映了 166 个国家在 2000 年到 2010 年期间构成的联盟与对抗关系; EGFR 网络是由 Oda 等人^[68]建模的表皮生长因子受体网络,其每个节点代表受体细胞而节点之间的边则代表正/负反馈关系; Macrophage 网络是由 Oda 等人^[69]建模的由巨噬细胞作为节点与激活/分离行为作为边的分子相互作用网络; Yeast 网络记录了酿酒酵母基因调控网络中 690 个转录因子在结合位点的 1080 次增强/抑制行为^[70]; 类似的, E.coli 网络记录了大肠杆菌基因调控网络中 1461 个转录因子在结合位点的 3215 次增强/抑制行为^[71]; Soc-alpha 网络描述了比特币 Alpha 平台中 3783 个交易用户的信任/不信任操作; Wiki 网络描述了 7115 个选举人对其他候选人的支持/反对态度^[72]。

6.2 对比算法

为了验证所提 SEDRL-IM 算法的性能,基于贪心策略的 Degree 算法^[3]、基于随机游走的 PageRank 算法^[22]、基于 DRL 思想的 S2V-DQN 算法^[23]、基于混合搜索算法的 MA-IM 算法^[25]以及 CMA-IM 算法^[27]被选为对比算法,并进行综合测试,这 5 种经典算法的原理描述如表 5 所示。

表 5 对比算法列表

| 算法 | 算法描述 |
|--------------------------|---|
| Degree ^[3] | 基于贪心策略直接选取 k 个度最大的节点作为种子节点的启发式算法 |
| PageRank ^[22] | 基于随机游走策略迭代计算每个节点的 PR 值,并在游走结束后选出 k 个 PR 值最大的节点作为种子节点的启发式算法 |
| S2V-DQN ^[23] | 采用 struc2vec 进行降维并基于 ϵ -degree 采样训练 DQN 来生成每个节点的种子分数,最后迭代选取 k 个种子分数最高的节点作为种子节点的启发式算法 |
| MA-IM ^[25] | 结合基于轮盘的度策略、随机策略以及距离选择策略生成初始种子节点集后,采取模因演化算法对种子节点集进行演化从而寻找最优种子节点集的启发式算法 |
| CMA-IM ^[27] | 基于网络的节点度和社区信息进行局部搜索生成初始种子节点集后,采取基于演化算法和局部搜索的模因算法来寻找社区内的最优种子节点集的启发式算法 |

本文将从求解效率、解空间的局部搜索能力和全局探索能力 3 个方面对比分析各个算法的性能。

6.3 消融实验、自适应实验及对比实验设置

为了验证 SEDRL-IM 算法中 EA 与 DRL 结合的有效性,本文设置了消融实验,比较只保留 DRL 模块的 SEDRL-IM(DRL) 算法,只保留 EA 模块的 SEDRL-IM(EA) 算法以及结合 EA 和 DRL 的 SEDRL-IM 算法的性能。此外,本文通过与其他对比算法的性能比较,验证了 SEDRL-IM 算法在解决符号网络 IM 问题上的有效性。

本文中所有实验均在装有 Linux 系统的服务器上运行,其硬件配置为 Intel (R), Xeon (R), E5-2630 v4 CPU (2.20 GHz) 处理器以及 512 GB 的内存。本文所提 SEDRL-IM 算法在上述服务器部署的 Python 3.7、Torch 1.8.1+cu111 环境下运行,其中 DQN 在 Keras 1.0.8、TensorFlow 1.14.0 环境下运行。

表 6 展示了所提 SEDRL-IM 算法和各对比算法在实验中关于演化算法和 DQN 网络的具体参数。此外,本文所提 SEDRL-IM 算法的代码以及实验数据都将发布在线上网站 <https://github.com/hhp19980613/SEDRL-IM>。

表 6 所提 SEDRL-IM 算法和对比算法的具体实验参数设置

| 算法参数 | Degree | PageRank | S2V-DQN | MA-IM(multi) | CMA-IM | SEDRL-IM(EA) | SEDRL-IM(DRL) | SEDRL-IM |
|----------------|--------|----------|---------|--------------|--------|--------------|---------------|----------|
| 迭代数 n_p | 无 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| 演化种群规模 n_g | 无 | 无 | 无 | 100 | 100 | 100 | 无 | 100 |
| 交叉概率 p_c | 无 | 无 | 无 | 0.8 | 0.8 | 0.8 | 无 | 0.8 |
| 变异概率 p_m | 无 | 无 | 无 | 0.2 | 0.2 | 0.2 | 无 | 0.2 |
| 隐藏层维数 l | 无 | 无 | 64 | 无 | 无 | 64 | 64 | 64 |
| 训练批次大小 n_b | 无 | 无 | 512 | 无 | 无 | 无 | 512 | 512 |
| 训练折扣率 γ | 无 | 无 | 0.8 | 无 | 无 | 无 | 0.8 | 0.8 |
| 训练学习率 α | 无 | 无 | 0.001 | 无 | 无 | 无 | 0.001 | 0.001 |
| 训练步数 n_t | 无 | 无 | 4 | 无 | 无 | 无 | 4 | 4 |

进一步地, 受到文献 [49] 的启发, 本文尝试在 SEDRL-IM 算法的基础上加入自适应探索策略, 平衡 SEDRL-IM 算法在求解符号网络 IM 问题时的全局探索和局部搜索. 以下是目前考虑的 5 种自适应策略.

(1) Adaptive_1 算法主要引入文献 [73] 中的自适应策略: 在每次迭代演化结束后, 通过如下公式自适应地调整演化过程中交叉概率 p_c 和变异概率 p_m .

$$p_c = \begin{cases} k_1 \frac{f_{\max} - f'}{f_{\max} - f_{\text{avg}}}, & \text{if } f' \geq f_{\text{avg}} \\ k_3, & \text{else} \end{cases}, \quad p_m = \begin{cases} k_2 \frac{f_{\max} - f}{f_{\max} - f_{\text{avg}}}, & \text{if } f \geq f_{\text{avg}} \\ k_4, & \text{else} \end{cases} \quad (19)$$

其中, $k_1 = k_3 = 0.8$, $k_2 = k_4 = 0.2$, f_{\max} 为迭代种群中最优个体的适应值, f_{avg} 为迭代种群中个体的平均适应值, f' 为交叉操作完成后最优个体的适应值, f 为突变操作完成后种群个体的适应值.

(2) Adaptive_2 算法主要引入文献 [74] 中的自适应策略: 在每次迭代结束后, 对当前迭代中通过演化算法进行全局搜索后得到的平均适应值增量 AFI_{GS} , 和执行强化学习算法局部搜索后得到的平均适应值增量 AFI_{LS} , 进行如下自适应评估.

$$AFI = \frac{\Delta Fitness}{\Delta FEs} \quad (20)$$

其中, $\Delta Fitness$ 为在局部或全局探索前后的最优适应值增量, ΔFEs 为局部搜索和全局探索中的适应值评估次数.

(3) Adaptive_3 算法主要引入文献 [47,75] 中的自适应策略: 在每次迭代演化结束后, 依据种群的当前适应值 f_i 、最优适应值 f_{\max} 及最差适应值 f_{\min} 对该种群的相对适应值 F_i 进行评估.

$$F_i = \frac{f_i - f_{\min}}{f_{\max} - f_{\min}} \quad (21)$$

(4) Adaptive_4 算法主要引入文献 [76] 中的自适应策略: 在每次迭代演化结束后, 基于上一次迭代得到平均适应值 $f_{\text{avg}}(t-1)$ 和当前的平均适应值 $f_{\text{avg}}(t)$ 对当前种群的总体适应值增量 RAF 进行评估.

$$RAF = \frac{f_{\text{avg}}(t)}{f_{\text{avg}}(t-1)} \quad (22)$$

(5) Adaptive_5 算法主要引入文献 [77] 中的自适应策略: 即每次演化结束后保留比例为 η 的精英种群和保留比例为 $(1-\eta)$ 的随机种群, 其中保留比例 η 由基于当前种群的平均适应值 f_{avg} 和最优适应值 f_{\max} 自适应计算所得, 并采用 K 参数用来限制 η 的取值范围.

$$\eta = K \times \frac{f_{\max}}{f_{\max} - f_{\text{avg}}} \quad (23)$$

7 结果分析

本节将在 2 个基准符号网络和 7 个真实符号网络上, 对所有算法进行 IM 问题性能测试 (见后文图 6-图 13, 表 7 和表 8). 其中, SNIC 模型的传播概率统一设定为 0.1, 种子集大小规模分别设置为 $k = \{10, 20, 30, 40\}$. 图 6 与图 7 分别展示了所有算法在基准符号网络上的平均性能排名和性能比较; 图 9 与表 7 分别展示了所有算法在真实

符号网络上的平均性能排名和性能比较; 其中, 性能排名以所有算法在所有网络数据集上进行 20 次迭代测试后的对比排名平均值为依据, 从 1 到 8 名进行排名, 其中性能最好的算法排名为 1. 此外, 本节还进行了自适应实验, 并在表 8 和图 13 中分别记录了相应实验得到的性能和排名.

7.1 基准网络数据集上的结果与分析

图 6 展示了所有算法在 4 个基准网络的平均性能排名. 结果显示: SEDRL-IM 算法的总体性能排名 (1.188) 远靠前于 CMA-IM (3.250)、S2V-DQN (3.438)、PageRank (3.750)、MA-IM (4.250)、Degree (5.313). 这充分验证了 SEDRL-IM 算法在求解模块化符号网络中 IM 问题的有效性.

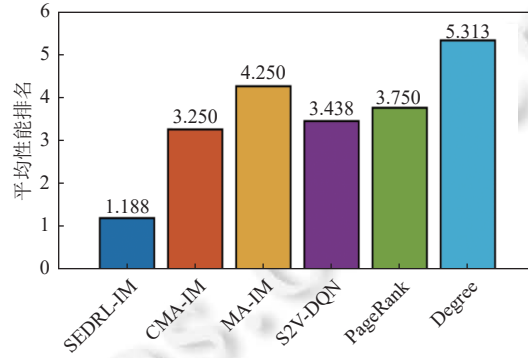


图 6 所有算法在基准网络上的平均性能排名

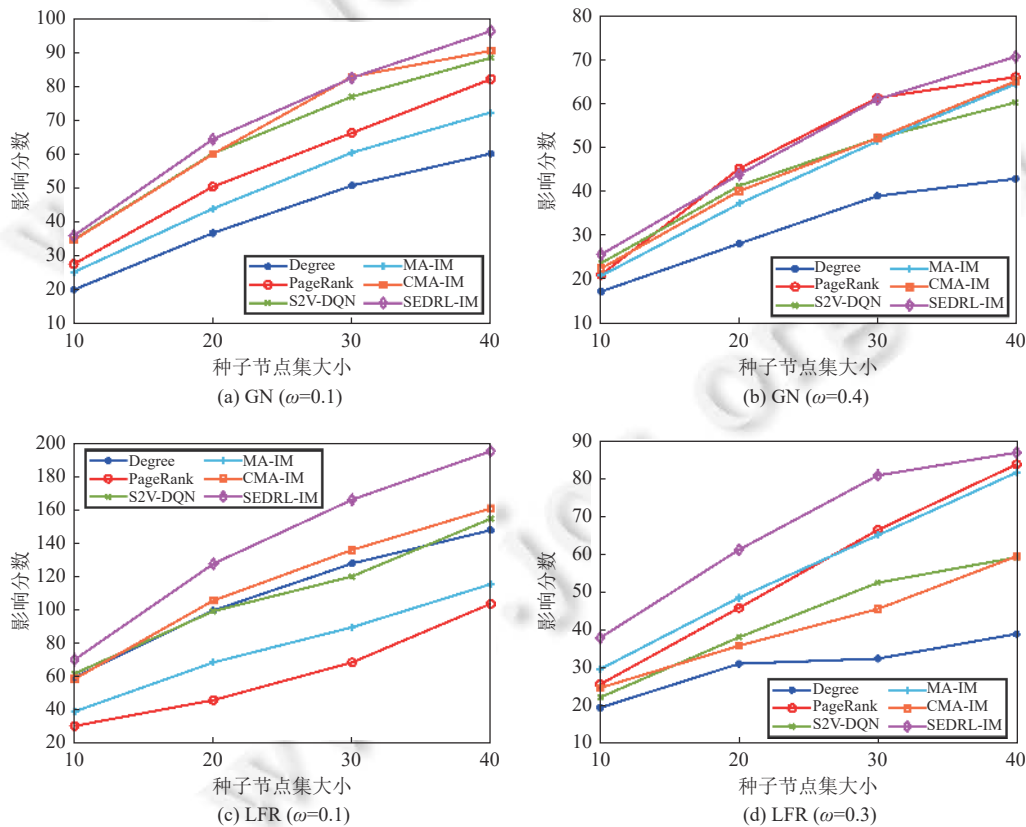


图 7 不同种子集大小下所有算法在基准网络上的性能表现

图 7(a) 和图 7(b) 分别展示了所有算法在 $\omega = 0.1$ 和 $\omega = 0.4$ 的 GN 基准网络上的测试结果; 图 7(c) 和图 7(d) 分别展示所有算法在 $\omega = 0.1$ 和 $\omega = 0.3$ 的 LFR 基准网络上的测试结果; 从图 7 中可以看出, 在 4 个数据集上的 16 个实例中, SEDRL-IM 算法在 13 个实例中的影响分数高于其他算法, 并且其性能差异随着种子集大小 k 变大而增大. 这些对比结果充分说明了 SEDRL-IM 算法在选取能最大限度传播正影响的种子节点上的有效性.

图 7(a) 和图 7(c) 展示了 CMA-IM 算法能获得较好性能, 其性能仅次于 SEDRL-IM 算法. 这是因为 GN 和 LFR 基准网络具有明显的社区结构, CMA-IM 算法可以有效利用这些社区信息挖掘符号网络的结构平衡特性, 并根据这些特性进行针对性局部搜索. 图 7(b) 展示了 PageRank 算法的较好性能, 其平均性能仅次于 SEDRL-IM 算法, 远优于其他算法. 这是因为 PageRank 算法的随机游走范围能够涵盖小规模基准 GN 网络的整个网络. 然而, 对于较大规模的 LFR 网络, PageRank 算法性能变得不稳定且逐步差于其他算法. 这是因为 PageRank 算法的随机性导致其无法在较大规模的异构网络上进行有效全局探索.

图 7 展示了 Degree 算法在种子节点集较小情况下对 LFR 网络能取得较好性能. 这是因为该算法可以充分利用 LFR 网络的异构信息和正负拓扑特性来选取种子节点. 然而, 对于 GN 网络和种子节点集较大的 LFR 网络, Degree 算法性能较差. 这是因为该算法对度均匀分布的 GN 网络无法有效利用节点度信息进行种子节点挖掘. 此外, Degree 算法的贪心策略容易使其陷入局部最优. 图 7 还展示了 MA-IM 与 CMA-IM 算法通过结合基于 EA 算法的全局搜索和基于贪心策略的局部搜索, 能有效避免其陷入局部最优. 因此, 它们的性能在图 7(a)、图 7(b) 和图 7(d) 中性能都优于 Degree 算法. 图 7 还展示了相比于 Degree 算法, S2V-DQN 算法能更好地采用随机梯度下降的局部搜索方式找到更优的影响节点集. 这是因为该算法利用了 struct2vec 降维算法来捕获符号网络的正负拓扑关系等特征信息. 然而, S2V-DQN 算法也很容易陷入局部最优. 因此, S2V-DQN 算法在性能上略低于 CMA-IM 算法、MA-IM 算法和本文所提 SEDRL-IM 算法.

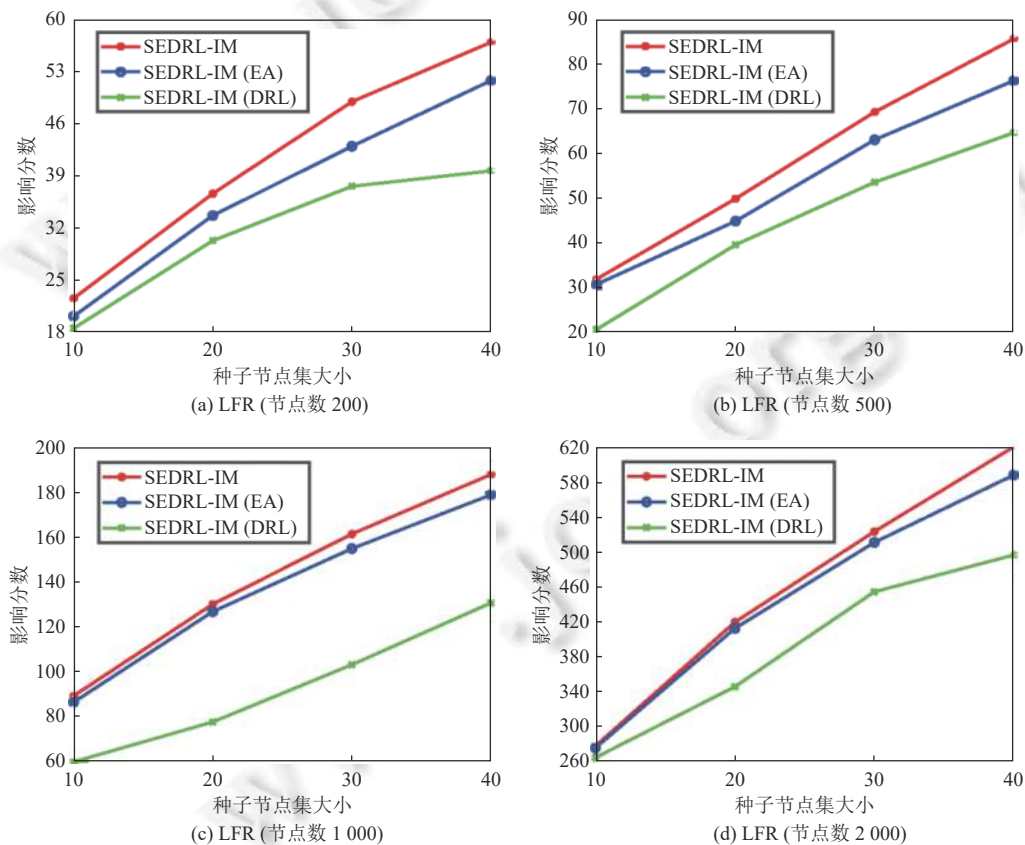


图 8 不同种子集大小下在基准网络上的消融实验性能

图 8(a)–图 8(d) 分别展示了 SEDRL-IM 算法在节点数为 200、500、1000 和 2000 的 LFR 网络上进行消融实验的测试结果; 从图 8 中可以看出, SEDRL-IM(DRL) 算法性能较差, 并与 SEDRL-IM 算法的性能差距随着数据集规模的增大而增大. SEDRL-IM(EA) 算法获得比 SEDRL-IM(DRL) 算法更好的性能表现. 然而, 其性能却低于 SEDRL-IM 算法. SEDRL-IM 算法通过结合 EA 模块的全局搜索和 DRL 模块的 n -step Q-learning 局部搜索技术, 取得了比 SEDRL-IM(EA) 算法高 5% 的性能.

7.2 真实网络数据集上的结果与分析

图 9 展示了所有算法在真实网络测试的总体性能排名. 如图 9 所示, SEDRL-IM 算法的平均性能排名 (1.04) 远靠前于 CMA-IM (3.79)、MA-IM (3.50)、Degree (5.54)、S2V-DQN (5.66)、PageRank (5.95)、SEDRL-IM(EA) (3.49)、SEDRL-IM(DRL) (6.95). 表 7 展示了所有算法在真实网络测试下的统计结果, 其结果显示: SEDRL-IM 算法在 28 个实例中有 25 个实例的性能优于其他算法. 上述结果充分验证了 SEDRL-IM 算法在求解真实符号网络 IM 问题上的优越性.

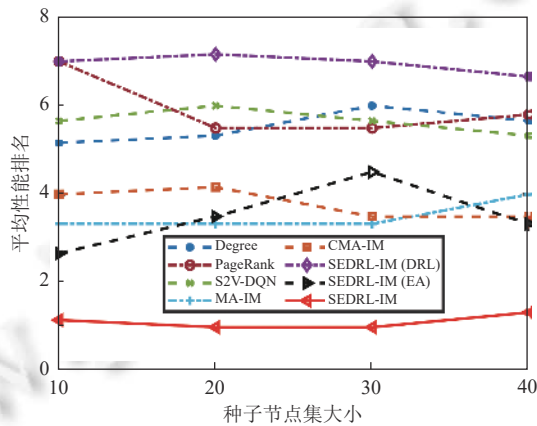


图 9 不同种子集大小下所有算法在真实网络上的平均性能排名

从表 7 还可以看出, PageRank 算法、S2V-DQN 算法和 MA-IM 算法对大规模网络解空间的搜索能力有限, 这导致上述算法无法在有限时间内搜索到较好解. 此外, Degree 算法在种子集大小为 10 的情况下展示了较好的性能, 但在其他种子集规模下性能较差. 这是因为真实网络相比于基准网络具有更为丰富的异构信息, Degree 算法可以有效获得较小规模的度信息, 但其搜索能力仅限于局部范围, 因此在大规模种子集上无法取得较好性能. 该现象同样出现在基于局部搜索的 S2V-DQN 算法中. 图 9 中的总体排名也进一步验证了基于局部搜索的 Degree 以及 S2V-DQN 算法的平均性能远低于其他算法.

表 7 和图 9 进一步说明了 SEDRL-IM (DRL) 算法缺乏对解空间的有效探索, 导致其在大多数真实网络中的性能较差, 甚至低于 Degree 算法与 S2V-DQN 算法. 此外, SEDRL-IM (EA) 算法通过遗传算法优化 DQN 的连续权重, 并通过锦标赛策略调整演化方向, 从而进行有效的全局探索并避免过度探索. 因此, 在大多数真实网络实验实例中, SEDRL-IM(EA) 算法的性能以及总体排名优于除 SEDRL-IM 算法外的其他算法. SEDRL-IM 算法通过结合 EA 模块和 DRL 模块, 更好地平衡对解空间的全局探索和局部搜索, 在性能表现上全面超过其他算法.

图 9 也展示了相比于 Degree 算法和 S2V-DQN 算法, CMA-IM 和 MA-IM 算法通过结合 EA 的全局搜索能力与贪心策略的局部搜索能力, 能在不同种子节点集规模下稳定地取得较好性能. 但是, 演化算法中全局搜索方向的不确定性也使得这类算法在较大的数据集上的实验结果标准差较大, 这验证了 CMA-IM 和 MA-IM 算法在较大规模网络上无法很好地稳定收敛 (如表 7 所示). 本文所提算法 SEDRL-IM 算法可通过 DNESBP 降维来充分捕获和学习真实网络的正负拓扑特性, 并结合 EA 算法的全局探索能力与 DRL 算法的局部搜索能力, 使得其性能在大多数真实网络上都优于 CMA-IM、MA-IM 算法和 S2V-DQN 算法. 此外, SEDRL-IM 算法在每次演化种群后通过 DRL 模块来反向更新 DQN 种群, 加速全局探索的收敛过程, 使得其性能标准差均低于基于种群搜索的 CMA-IM 和 MA-IM 算法. 这也验证了 SEDRL-IM 算法相比于其他基于演化思想的 IM 算法, 具有更好的收敛能力.

表 7 不同种子集大小下所有算法在真实网络上的实验数据结果

| 符号网络 | 节点数 | 种子集大小 | Degree | PageRank | S2V-DQN | MA-IM (multi) | CMA-IM | SEDRL-IM(EA) | SEDRL-IM(DRL) | SEDRL-IM |
|------------|------|-------|--------|--------------|--------------|---------------|---------------|--------------|---------------|--------------|
| War | 166 | k=10 | 69.42 | 15.26±0.3412 | 27.51±0.0265 | 59.46±0.6636 | 55.49±1.130 | 66.68±0.4335 | 33.26±0.1035 | 68.89±0.1941 |
| | | k=20 | 84.63 | 29.63±1.036 | 47.72±0.0212 | 87.18±0.4712 | 84.61±1.052 | 90.18±0.6013 | 55.43±0.1128 | 95.23±0.2815 |
| | | k=30 | 101.6 | 38.07±0.3936 | 67.18±0.0316 | 110.2±0.7666 | 107.15±0.9640 | 111.7±0.7553 | 76.21±0.1154 | 114.0±0.2717 |
| | | k=40 | 112.6 | 48.46±0.4732 | 67.78±0.0126 | 121.5±0.5232 | 127.1±1.357 | 118.0±0.7231 | 84.19±0.1228 | 124.0±0.5056 |
| EGFR | 329 | k=10 | 8.813 | 12.05±1.415 | 11.65±0.0310 | 14.76±1.393 | 12.56±1.621 | 13.69±1.252 | 10.64±0.3116 | 16.42±0.4434 |
| | | k=20 | 16.97 | 23.31±1.950 | 21.69±0.0444 | 26.17±1.253 | 22.85±3.381 | 24.04±1.164 | 20.90±0.2414 | 30.64±0.8757 |
| | | k=30 | 26.85 | 36.73±1.566 | 31.81±0.0728 | 41.09±1.178 | 33.67±2.732 | 35.10±1.065 | 31.05±0.2464 | 42.86±0.8813 |
| | | k=40 | 35.75 | 47.73±0.9132 | 41.31±0.0535 | 52.02±0.9565 | 42.86±2.6447 | 50.39±1.315 | 43.44±0.2347 | 52.85±0.6531 |
| Macrophage | 678 | k=10 | 11.25 | 11.70±1.1544 | 12.15±0.1125 | 15.99±1.881 | 12.45±2.484 | 14.04±1.712 | 10.43±0.4222 | 16.68±0.8608 |
| | | k=20 | 20.91 | 23.55±3.026 | 21.77±0.0846 | 24.47±1.836 | 24.26±3.571 | 22.53±1.921 | 20.11±0.4905 | 29.51±1.071 |
| | | k=30 | 30.60 | 35.80±0.9873 | 33.69±0.0765 | 40.71±1.596 | 34.82±1.056 | 33.54±1.511 | 30.11±0.4145 | 42.07±1.044 |
| | | k=40 | 42.58 | 47.44±1.264 | 56.48±0.1250 | 50.27±1.421 | 45.74±2.436 | 50.18±2.183 | 38.74±0.4652 | 54.93±1.076 |
| Yeast | 690 | k=10 | 25.08 | 13.93±2.386 | 20.44±0.0835 | 23.18±2.961 | 23.82±2.791 | 23.97±1.672 | 9.59±0.4924 | 27.69±1.042 |
| | | k=20 | 39.88 | 26.12±1.613 | 32.13±0.0733 | 34.85±2.623 | 39.49±3.885 | 34.98±1.778 | 18.97±0.5928 | 43.24±1.815 |
| | | k=30 | 52.18 | 41.14±2.873 | 43.49±0.0832 | 47.28±2.456 | 53.21±3.314 | 46.89±2.372 | 29.94±0.5215 | 56.74±1.046 |
| | | k=40 | 64.32 | 54.29±2.433 | 44.38±0.0903 | 58.63±3.082 | 66.28±2.906 | 59.15±2.952 | 39.68±0.5122 | 71.22±1.022 |
| E.coli | 1461 | k=10 | 38.67 | 29.02±2.557 | 33.75±0.3025 | 41.29±3.141 | 37.69±7.816 | 38.58±3.124 | 33.76±0.7931 | 51.49±1.131 |
| | | k=20 | 45.43 | 51.76±2.012 | 38.68±0.5812 | 58.32±3.103 | 38.85±9.313 | 57.44±2.746 | 37.76±0.5913 | 66.24±1.483 |
| | | k=30 | 51.98 | 66.35±3.984 | 53.24±0.3175 | 79.05±2.831 | 58.90±8.726 | 71.07±2.372 | 47.15±0.6844 | 81.96±1.504 |
| | | k=40 | 61.91 | 86.19±2.024 | 57.44±0.3733 | 88.04±2.901 | 77.44±8.251 | 83.93±2.548 | 54.18±0.8425 | 90.40±1.804 |
| Soc-alpha | 3783 | k=10 | 454.3 | 63.86±4.478 | 479.6±0.9915 | 345.8±12.90 | 464.6±14.43 | 483.5±10.67 | 233.72±2.382 | 496.6±5.786 |
| | | k=20 | 648.3 | 111.9±5.793 | 656.9±0.8265 | 425.1±12.59 | 665.5±17.01 | 677.6±8.732 | 364.40±1.916 | 690.2±6.340 |
| | | k=30 | 739.2 | 92.54±5.033 | 739.0±0.8022 | 480.7±16.16 | 763.3±19.01 | 743.2±11.50 | 490.45±2.420 | 776.7±7.651 |
| | | k=40 | 765.7 | 132.7±8.996 | 809.4±0.6203 | 543.8±11.58 | 826.8±16.15 | 832.9±14.15 | 632.36±1.991 | 884.3±5.822 |
| Wiki | 7115 | k=10 | 1219 | - | - | 319.2±18.95 | 1240±23.71 | 1282±21.11 | 482.2±3.221 | 1329±10.85 |
| | | k=20 | 1977 | - | - | 505.9±26.40 | 2051±26.28 | 1974±26.54 | 1850±3.157 | 2069±13.74 |
| | | k=30 | 2491 | - | - | 595.7±19.14 | 2609±28.16 | 2605±25.04 | 2393±3.048 | 2648±12.14 |
| | | k=40 | 2922 | - | - | 651.5±29.58 | 2886±31.51 | 3019±32.49 | 2751±3.303 | 3084±10.57 |

为了可视化比较各个算法的性能, 图 10 展示了所有算法在 War 网络上的种子节点可视化分布情况. 从图 10 中可以看出, Degree 算法和 S2V-DQN 算法选取的种子节点多为度高的节点, 而容易忽略节点之间边的正负特性, 导致这类算法陷入局部最优. CMA-IM 算法、MA-IM 算法和 PageRank 算法选取的节点则均匀分散在各个社区, 能更容易找到分布不同且最大限度传播正影响的种子节点. SEDRL-IM 算法不仅能利用 DRL 算法的贪心策略选取节点度高的节点, 还能利用 EA 算法的全局搜索特性去寻找各社区中潜力较高的节点. 综合而言, SEDRL-IM 算法对真实异构网络中 IM 问题的求解能力较优于其他 IM 算法.

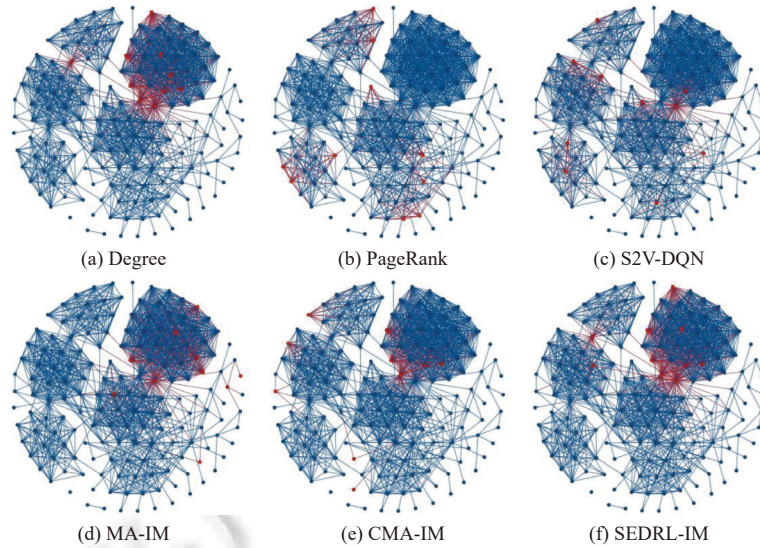


图 10 所有算法在 War 网络上的种子集可视化分布

图 11 展示了种子规模为 40 时各算法在不同网络数据集规模上的平均运行时间. 从图 11 可以看出, SEDRL-IM (DRL) 算法和 MA-IM 算法的运行时间最少. 这是因为 SEDRL-IM(DRL) 仅使用 DRL 的随机梯度下降法进行局部搜索, MA-IM 算法使用轮盘赌策略来加速全局搜索的收敛过程. 相比于 MA-IM 算法, 在每次迭代中, CMA-IM 算法需要进行社区划分, S2V-DQN 算法需要采用大量的经验数据来反向更新 DQN, 以及 PageRank 算法需要采用随机游走算法统计每个节点的 PR 值. 因此, 相比于 MA-IM 算法, CMA-IM、S2V-DQN、PageRank 算法所用时间较多. SEDRL-IM 算法在演化时采用锦标赛策略来加速全局探索, 且只针对当前最优的 DQN 个体采用随机梯度下降算法进行局部搜索. 因此, SEDRL-IM 算法的运行时间远远低于 S2V-DQN 算法、PageRank 算法和 CMA-IM 算法, 略高于 SEDRL-IM (EA) 算法. 上述时间对比分析验证了 SEDRL-IM 算法在求解 IM 问题的高效性.

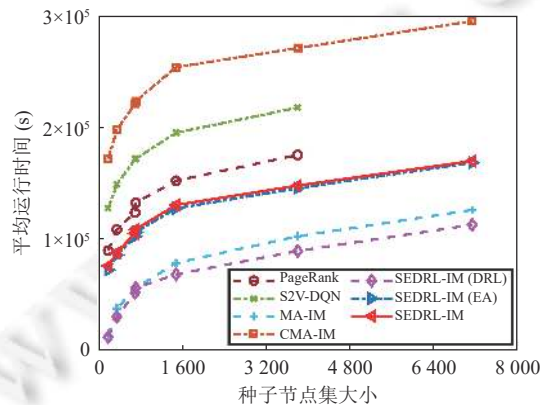


图 11 不同算法在种子规模为 $k=40$ 下在各个数据集上的平均对数运行时间

图 12 展示了不同算法分别在 E.coli、War 与 Wiki 网络上的两跳以内影响分数的迭代变化过程. 从实验结果可以看出, SEDRL-IM (EA) 和 S2V-DQN 这类局部搜索算法在大规模数据集上会过早收敛. CMA-IM 算法在大规模网络上会出现无法收敛的情况. 这是因为其在演化过程中缺乏有效策略来限制全局探索的方向. SEDRL-IM (EA) 算法通过采用锦标赛策略以及 MA-IM 算法采用轮盘赌策略, 从而分别实现了在 30 次以及 70 次迭代前后达到良好的收敛效果. 相比于上述算法, 本文所提 SEDRL-IM 算法实现了在 30 次迭代前后达到收敛. 此对比实验验证了 SEDRL-IM 算法在求解 IM 问题时的较优求解能力和稳健收敛性.

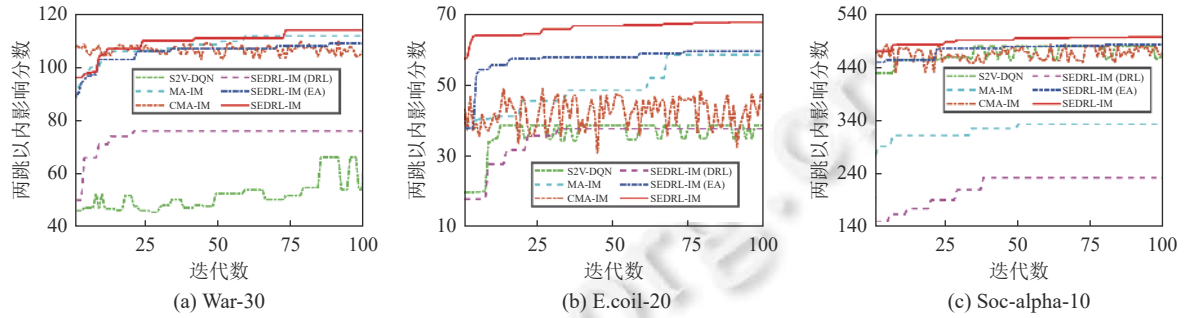


图 12 所有算法在 3 种网络上的两跳以内影响分数的迭代变化

7.3 自适应实验的结果与分析

表 8 对比了 SEDRL-IM 算法与该算法加入 5 种不同自适应策略后的实验结果. 为了验证这 6 种算法的统计性能, wilcoxon 秩和检验+/-/~被用来衡量不同算法在显著性水平为 5% 时的统计差异, 其中+, -与~分别表示自适应算法的性能优于、差于与无显著差异于 SEDRL-IM 算法. 此外, 为了进一步验证这 6 种算法的整体性能, 性能排名的统计方法也被用于对所有算法在所有网络数据集上进行对比排分, 其中性能最好的算法排名为 1. 具体的总体排名如图 13 所示.

表 8 自适应算法和 SEDRL-IM 算法在真实网络上的实验数据结果

| 符号网络 | 节点数 | 种子集大小 | SEDRL-IM | Adaptive_1 | Adaptive_2 | Adaptive_3 | Adaptive_4 | Adaptive_5 |
|------------|------|-------|--------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| War | 166 | k=10 | 68.89 | 69.91(+) | 69.89(+) | 63.78(-) | 67.54(-) | 69.42(+) |
| | | k=20 | 95.23 | 98.02(+) | 90.79(-) | 94.80(-) | 91.56(-) | 94.77(-) |
| | | k=30 | 114.0 | 110.1(-) | 113.4(-) | 113.3(-) | 112.1(-) | 111.9(-) |
| | | k=40 | 124.1 | 122.0(-) | 123.0(-) | 119.5(-) | 124.3(+) | 121.9(-) |
| EGFR | 329 | k=10 | 16.42 | 17.08(+) | 17.04(+) | 16.98(-) | 17.01(+) | 17.04(+) |
| | | k=20 | 30.64 | 30.08(-) | 30.19(-) | 30.06(-) | 29.95(-) | 29.98(-) |
| | | k=30 | 42.86 | 42.74(-) | 42.73(-) | 42.73(-) | 42.86(-) | 42.89(+) |
| | | k=40 | 52.85 | 55.05(+) | 55.03(+) | 54.71(+) | 55.04(+) | 55.03(+) |
| Macrophage | 678 | k=10 | 16.68 | 17.03(-) | 15.81(-) | 16.04(-) | 16.39(-) | 17.09(+) |
| | | k=20 | 29.51 | 30.72(+) | 30.49(+) | 29.39(-) | 30.59(+) | 30.62(+) |
| | | k=30 | 42.07 | 42.74(+) | 43.49(+) | 42.45(+) | 42.52(+) | 43.69(+) |
| | | k=40 | 54.93 | 55.35(+) | 56.46(+) | 54.75(-) | 56.55(+) | 54.81(-) |
| Yeast | 690 | k=10 | 27.69 | 27.50(-) | 27.51(-) | 27.51(-) | 27.34(-) | 27.65(+) |
| | | k=20 | 43.24 | 43.44(+) | 44.30(-) | 43.04(-) | 44.86(+) | 43.01(-) |
| | | k=30 | 56.74 | 58.66(+) | 56.67(-) | 57.40(+) | 56.93(-) | 56.65(-) |
| | | k=40 | 71.22 | 71.37(+) | 69.07(-) | 67.87(-) | 71.24(+) | 68.88(-) |
| E.coli | 1461 | k=10 | 51.49 | 50.81(-) | 50.96(-) | 50.96(+) | 50.88(-) | 50.96(-) |
| | | k=20 | 66.24 | 67.82(+) | 67.85(+) | 67.79(+) | 67.80(+) | 67.82(+) |
| | | k=30 | 81.96 | 81.57(+) | 80.97(-) | 80.96(-) | 80.97(-) | 80.97(-) |
| | | k=40 | 90.40 | 89.94(-) | 92.97(+) | 93.25(+) | 92.65(+) | 93.19(+) |

表 8 自适应算法和 SEDRL-IM 算法在真实网络上的实验数据结果 (续)

| 符号网络 | 节点数 | 种子集大小 | SEDRL-IM | Adaptive_1 | Adaptive_2 | Adaptive_3 | Adaptive_4 | Adaptive_5 |
|-----------|------|-------|-------------|-----------------|----------------|-----------------|-------------|-----------------|
| Soc-alpha | 3783 | k=10 | 496.7 | 503.2(+) | 497.2(~) | 497.3(+) | 500.6(+) | 501.4(+) |
| | | k=20 | 690.2 | 701.4(+) | 687.8(-) | 693.1(+) | 688.2(-) | 688.3(-) |
| | | k=30 | 776.7 | 781.6(+) | 788.7(+) | 780.4(+) | 782.3(+) | 788.8(+) |
| | | k=40 | 884.4 | 894.5(+) | 897.0(+) | 907.6(+) | 895.9(+) | 895.6(+) |
| Wiki | 7115 | k=10 | 1330 | 1336(+) | 1341(+) | 1318(-) | 1327(~) | 1331(~) |
| | | k=20 | 2070 | 2078(+) | 2067(-) | 2066(-) | 2067(~) | 2073(+) |
| | | k=30 | 2650 | 2660(+) | 2643(-) | 2641(-) | 2633(-) | 2650(-) |
| | | k=40 | 3085 | 3106(+) | 3083(+) | 3085(+) | 3079(-) | 3087(+) |
| best/all | - | - | 5/28 | 12/28 | 2/28 | 2/28 | 3/28 | 4/28 |
| +/-/~ | - | - | - | 20/7/1 | 12/13/3 | 11/17/0 | 13/12/3 | 15/11/2 |

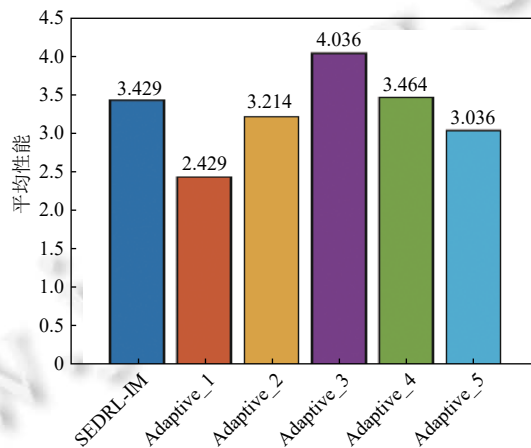


图 13 自适应算法和 SEDRL-IM 算法在真实网络上的平均性能排名

图 13 中排名结果表明, SEDRL-IM 算法的总体性能排名 (3.429) 要略差于 Adaptive_3 (4.036)、Adaptive_4 (3.464), 而高于 Adaptive_1 (2.429)、Adaptive_2 (3.214)、Adaptive_5 (3.036)。这表明自适应策略能帮助 SEDRL-IM 算法平衡局部搜索和全局探索。值得注意的是, 这 6 个算法之间的排名差异较小, 说明目前自适应策略对优化性能仅有微小的提升。其主要原因是 SEDRL-IM 算法中基于强化学习的局部搜索策略非常依赖演化过程中产生的四元组训练数据, 其局部搜索性能与演化搜索过程相关联。

表 8 中结果表明, Adaptive_1 算法在 28 个实例中有 20 个实例的性能超过 SEDRL-IM 算法。这是因为 Adaptive_1 算法在演化前期提高交叉概率, 通过高适应值种群的基因置换来保留优秀基因, 并在演化后期提高变异概率, 通过种群多样性来摆脱局部最优。但是 Adaptive_1 算法过度依赖于演化算法, 导致时间复杂度远高于 SEDRL-IM 算法。Adaptive_5 算法随着数据集规模的增大, 有 15 个实例的性能超过 SEDRL-IM 算法。这是因为 Adaptive_5 算法不仅通过自适应比例因子在早期尽可能保留优秀种群, 并且在后期使用随机策略来增强种群多样性。其他自适应算法都只能在将近一半的实例性上超过 SEDRL-IM 算法。这是因为这些自适应算法主要关注局部搜索策略的自适应而忽略演化算法的自适应。

8 总 结

随着社交应用的崛起以及影响最大化理论的广泛应用, 符号化社交网络的 IM 问题研究开始受到广泛关注。然而, 近年来提出的求解算法往往将 IM 问题建模为种子节点的离散组合选取优化问题, 导致这些求解算法具有过高的计算代价而无法适用于处理大规模的异构网络。为了解决上述问题, 本文受到 EDRL-IM 算法启发, 提出了

用于求解符号化网络 IM 问题的演化深度强化学习算法 (SEDRL-IM). 该算法首先将经典 IC 模型扩展到符号网络, 提出 SNIC 模型模拟了网络中正负影响的传播. 其次, 该算法通过引入 DQN 来选取种子节点集, 结合马尔可夫决策与影响分数评估, 将离散的种子节点组合选取问题转化成为更易优化的连续网络权重优化问题. 另外, 该算法将演化过程的个体视作策略, 结合演化算法中交叉变异的全局搜索特性以及强化学习中随机梯度下降的局部搜索特性, 实现对 DQN 权重优化问题解的高效搜索. 最后, 在 2 个基准网络和 7 个真实网络上的性能测试验证了 SEDRL-IM 算法在综合性能上优于其他 5 个基准 IM 算法, 并表明了该算法具备高效处理不同类型异构网络时的强大泛化能力, 能在有效性和高效性之间取得较好平衡.

本文提出的基于 SNIC 的影响传播模型和符号网络的 IM 优化问题与大型传染病如新型冠状病毒 (COVID-19) 的传播研究密切相关. 在未来的研究中, 我们计划将 SNIC 模型拓展到传染病的传播研究中, 并结合影响最大化研究对传染病进行阻断研究. 此外, 受相关性工作^[28-34,48,52,54-57]的启发, 我们将进一步研究本文提出的演化深度强化学习算法 (SEDRL-IM) 在影响力最大化问题上的泛化能力, 解决因网络结构特征异质导致本文算法在网络优化相关问题上泛化能力弱的问题. 最后, 我们将进一步研究演化强化学习算法在复杂网络优化问题上的理论分析.

References:

- [1] Liu DY, Li J, Yang B. Learning information diffusion network. *Journal of Jilin University (Science Edition)*, 2012, 50(4): 767-774 (in Chinese with English abstract). [doi: 10.13413/j.cnki.jdxblxb.2012.04.010]
- [2] Domingos P, Richardson M. Mining the network value of customers. In: *Proc. of the 7th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2001)*. San Francisco: ACM Press, 2001. 57-66. [doi: 10.1145/502512.502525]
- [3] Kempe D, Kleinberg J, Tardos É. Maximizing the spread of influence through a social network. In: *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2003)*. Washington: ACM Press, 2003. 137-146. [doi: 10.1145/956750.956769]
- [4] Gong MG, Wang SF, Zhang TL, Mao YS, Yang YL, Niu XD, Wu Y. Influence maximization method of complex network nodes based on deep autoencoder. CN, 108510059A. 2018-09-07 (in Chinese).
- [5] Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N. Cost-effective outbreak detection in networks. In: *Proc. of the 13th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2007)*. San Jose: ACM Press, 2007. 420-429. [doi: 10.1145/1281192.1281239]
- [6] Ohsaka N, Akiba T, Yoshida Y, Kawarabayashi KI. Fast and accurate influence maximization on large networks with pruned monte-carlo simulations. In: *Proc. of the 28th AAAI Conf. on Artificial Intelligence*. Québec: AAAI Press, 2014. 138-144.
- [7] Goyal A, Lu W, Lakshmanan LVS. CELF++: Optimizing the greedy algorithm for influence maximization in social networks. In: *Proc. of the 20th Int'l Conf. Companion on World Wide Web (WWW 2011)*. Hyderabad: ACM Press, 2011. 47-48. [doi: 10.1145/1963192.1963217]
- [8] Li GL, Chu YP, Feng JH, Xu YQ. Influence maximization on multiple social networks. *Chinese Journal of Computers*, 2016, 39(4): 643-656 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2016.00643]
- [9] Yi XS, Hu JL, Wang XW. Influence maximization algorithm based on the weak tie in social network. *Ruan Jian Xue Bao/Journal of Software*, 2016, 27: 1-11 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/16012.htm>
- [10] Chen W, Wang YJ, Yang SY. Efficient influence maximization in social networks. In: *Proc. of the 15th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2009)*. Paris: ACM Press, 2009. 199-208. [doi: 10.1145/1557019.1557047]
- [11] Morone F, Makse HA. Influence maximization in complex networks through optimal percolation. *Nature*, 2015, 524(7563): 65-68. [doi: 10.1038/nature14604]
- [12] Wang Y, Cong G, Song GJ, Xie KQ. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In: *Proc. of the 16th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining (KDD 2010)*. Washington: ACM Press, 2010. 1039-1048. [doi: 10.1145/1835804.1835935]
- [13] Han ZM, Chen Y, Liu W, Yuan BH, Li MQ, Duan DG. Research on node influence analysis in social networks. *Ruan Jian Xue Bao/Journal of Software*, 2017, 28(1): 84-104 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/5115.htm> [doi: 10.13328/j.cnki.jos.005115]
- [14] Li YH, Chen W, Wang YJ, Zhang ZL. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In: *Proc. of the 6th ACM Int'l Conf. on Web Search and Data Mining (KDD 2013)*. Rome: ACM Press, 2013. 657-666. [doi: 10.1145/2433396.2433478]

- [15] Li D, Xu ZM, Chakraborty N, Gupta A, Sycara K, Li S. Polarity related influence maximization in signed social networks. *PLoS One*, 2014, 9(7): e102199. [doi: [10.1371/journal.pone.0102199](https://doi.org/10.1371/journal.pone.0102199)]
- [16] Liu W, Chen X, Jeon B, Chen L, Chen BL. Influence maximization on signed networks under independent cascade model. *Applied Intelligence*, 2019, 49(3): 912–928. [doi: [10.1007/s10489-018-1303-2](https://doi.org/10.1007/s10489-018-1303-2)]
- [17] Tian JT, Wang YT, Feng XJ. A new hybrid algorithm for influence maximization in social networks. *Chinese Journal of Computers*, 2011, 34(10): 1956–1965 (in Chinese with English abstract). [doi: [10.3724/SP.J.1016.2011.01956](https://doi.org/10.3724/SP.J.1016.2011.01956)]
- [18] Liang WX, Shen CG, Li X, Nishide R, Piumarta I, Takada H. Influence maximization in signed social networks with opinion formation. *IEEE Access*, 2019, 7: 68837–68852. [doi: [10.1109/ACCESS.2019.2918810](https://doi.org/10.1109/ACCESS.2019.2918810)]
- [19] Li D, Wang CH, Zhang SP, Zhou GL, Chu DH, Wu C. Positive influence maximization in signed social networks based on simulated annealing. *Neurocomputing*, 2017, 260: 69–78. [doi: [10.1016/j.neucom.2017.03.003](https://doi.org/10.1016/j.neucom.2017.03.003)]
- [20] Gong MG, Yan JN, Shen B, Ma LJ, Cai Q. Influence maximization in social networks based on discrete particle swarm optimization. *Information Sciences*, 2016, 367–368: 600–614. [doi: [10.1016/j.ins.2016.07.012](https://doi.org/10.1016/j.ins.2016.07.012)]
- [21] Şimşek A, Kara R. Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks. *Expert Systems with Applications*, 2018, 114: 224–236. [doi: [10.1016/j.eswa.2018.07.038](https://doi.org/10.1016/j.eswa.2018.07.038)]
- [22] Yin XY, Hu X, Chen YJ, Yuan X, Li BC. Signed-PageRank: An efficient influence maximization framework for signed social networks. *IEEE Trans. on Knowledge and Data Engineering*, 2021, 33(5): 2208–2222. [doi: [10.1109/TKDE.2019.2947421](https://doi.org/10.1109/TKDE.2019.2947421)]
- [23] Dai HJ, Khalil EB, Zhang YY, Dilkina B, Song, L. Learning combinatorial optimization algorithms over graphs. In: *Proc. of the 31st Int'l Conf. on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017. 6351–6361.
- [24] Zhang JW, Lv S, Zhang ZH, Yu JY, Gong XY. Survey on deep reinforcement learning methods based on sample efficiency optimization. *Ruan Jian Xue Bao/Journal of Software*, 2022, 33(11): 4217–4238 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/6391.htm> [doi: [10.13328/j.cnki.jos.006391](https://doi.org/10.13328/j.cnki.jos.006391)]
- [25] Wang S, Liu J, Jin YC. Finding influential nodes in multiplex networks using a memetic algorithm. *IEEE Trans. on Cybernetics*, 2021, 51(2): 900–912. [doi: [10.1109/TCYB.2019.2917059](https://doi.org/10.1109/TCYB.2019.2917059)]
- [26] Jin D, Liu DY, Huang J, He DX, Wang XH. Global optimization algorithm based on immune algorithm and evolutionary diffusion optimization. *Journal of Jilin University (Engineering and Technology Edition)*, 2009, 39(1): 124–130 (in Chinese with English abstract). [doi: [10.13229/j.cnki.jdxbgxb2009.01.046](https://doi.org/10.13229/j.cnki.jdxbgxb2009.01.046)]
- [27] Gong MG, Song C, Duan C, Ma LJ, Shen B. An efficient memetic algorithm for influence maximization in social networks. *IEEE Computational Intelligence Magazine*, 2016, 11(3): 22–33. [doi: [10.1109/MCI.2016.2572538](https://doi.org/10.1109/MCI.2016.2572538)]
- [28] Cui XD, Zhang W, Tüske Z, Picheny M. Evolutionary stochastic gradient descent for optimization of deep neural networks. In: *Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems*. Montréal: Curran Associates Inc., 2018. 6051–6061.
- [29] Such FP, Madhavan V, Conti E, Lehman J, Stanley KO, Clune J. Deep neuroevolution: Genetic algorithms are a competitive alternative for training deep neural networks for reinforcement learning. *arXiv:1712.06567*, 2017.
- [30] Liu FY, Li ZN, Qian C. Self-guided evolution strategies with historical estimated gradients. In: *Proc. of the 29th Int'l Joint Conf. on Artificial Intelligence*. Yokohama: Unknown Publishers, 2021. 205.
- [31] Houthoofd R, Chen RY, Isola P, Stadie BC, Wolski F, Ho J, Abbeel P. Evolved policy gradients. In: *Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems*. Montréal: Curran Associates Inc., 2018. 5405–5414.
- [32] Khadka S, Tumer K. Evolution-guided policy gradient in reinforcement learning. In: *Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems*. Montréal: Curran Associates Inc., 2018. 1196–1208.
- [33] Khadka S, Majumdar S, Nassar T, Dwiel Z, Tumer E, Miret S, Liu YY, Tumer K. Collaborative evolutionary reinforcement learning. In: *Proc. of the 36th Int'l Conf. on Machine Learning*. Long Beach: PMLR, 2019. 3341–3350.
- [34] Wang YT, Xue K, Qian C. Evolutionary diversity optimization with clustering-based selection for reinforcement learning. In: *Proc. of the 10th Int'l Conf. on Learning Representations*. New Orleans: OpenReview, 2022.
- [35] Pourchot A, Sigaud O. CEM-RL: Combining evolutionary and gradient-based methods for policy search. In: *Proc. of the 7th Int'l Conf. on Learning Representations*. New Orleans: ICLR, 2019.
- [36] Chang S, Yang J, Choi J, Kwak N. Genetic-gated networks for deep reinforcement learning. In: *Proc. of the 32nd Int'l Conf. on Neural Information Processing Systems*. Montréal: Curran Associates Inc., 2018. 1754–1763.
- [37] Ma LJ, Shao ZY, Li XC, Lin QZ, Li JQ, Leung VCM, Nandi AK. Influence maximization in complex networks by using evolutionary deep reinforcement learning. *IEEE Trans. on Emerging Topics in Computational Intelligence*, 2022. [doi: [10.1109/TETCI.2021.3136643](https://doi.org/10.1109/TETCI.2021.3136643)]
- [38] Liu YY, Slotine JJ, Barabási AL. Controllability of complex networks. *Nature*, 2021, 473(7346): 167–173. [doi: [10.1038/nature10011](https://doi.org/10.1038/nature10011)]
- [39] Yang YL, Lu ZY, Li VOK, Xu K. Noncooperative information diffusion in online social networks under the independent cascade model.

- IEEE Trans. on Computational Social Systems, 2017, 4(3): 150–162. [doi: [10.1109/TCSS.2017.2719056](https://doi.org/10.1109/TCSS.2017.2719056)]
- [40] Riquelme F, Gonzalez-Cantergiani P, Molinero X, Serna M. Centrality measure in social networks based on linear threshold model. Knowledge-Based Systems, 2018, 140: 92–102. [doi: [10.1016/j.knosys.2017.10.029](https://doi.org/10.1016/j.knosys.2017.10.029)]
- [41] Chen W, Lu W, Zhang N. Time-critical influence maximization in social networks with time-delayed diffusion process. In: Proc. of the 26th AAAI Conf. on Artificial Intelligence. Ontario: AAAI Press, 2012. 592–598.
- [42] Lee JR, Chung CW. A fast approximation for influence maximization in large social networks. In: Proc. of the 23rd Int'l Conf. on World Wide Web (WWW 2014). Seoul: ACM Press, 2014. 1157–1162. [doi: [10.1145/2567948.2580063](https://doi.org/10.1145/2567948.2580063)]
- [43] Clauset A, Newman MEJ, Moore C. Finding community structure in very large networks. Physical Review E, 2004, 70(6): 066111. [doi: [10.1103/PhysRevE.70.066111](https://doi.org/10.1103/PhysRevE.70.066111)]
- [44] Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment, 2008: P10008. [doi: [10.1088/1742-5468/2008/10/p10008](https://doi.org/10.1088/1742-5468/2008/10/p10008)]
- [45] Gong MG, Cai Q, Chen XW, Ma LJ. Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. IEEE Trans. on Evolutionary Computation, 2014, 18(1): 82–97. [doi: [10.1109/TEVC.2013.2260862](https://doi.org/10.1109/TEVC.2013.2260862)]
- [46] Ma LJ, Gong MG, Liu J, Cai Q, Jiao LC. Multi-level learning based memetic algorithm for community detection. Applied Soft Computing, 2014, 19: 121–133. [doi: [10.1016/j.asoc.2014.02.003](https://doi.org/10.1016/j.asoc.2014.02.003)]
- [47] Ma LJ, Li JQ, Lin QZ, Gong MG, Coello CAC, Ming Z. Cost-aware robust control of signed networks by using a memetic algorithm. IEEE Trans. on Cybernetics, 2020, 50(10): 4430–4443. [doi: [10.1109/TCYB.2019.2932996](https://doi.org/10.1109/TCYB.2019.2932996)]
- [48] Sun JY, Liu X, Bäck T, Xu ZB. Learning adaptive differential evolution algorithm from optimization experiences by policy gradient. IEEE Trans. on Evolutionary Computation, 2021, 25(4): 666–680. [doi: [10.1109/TEVC.2021.3060811](https://doi.org/10.1109/TEVC.2021.3060811)]
- [49] Liu FY, Li ZN, Qian C. Self-guided evolution strategies with historical estimated gradients. In: Proc. of the 29th Int'l Conf. on Artificial Intelligence. Yokohama: IJCAI, 2020. 1474–1480.
- [50] Andrychowicz M, Denil M, Colmenarejo SG, Hoffman MW, Pfau D, Schaul T, Shillingford B, De Freitas N. Learning to learn by gradient descent by gradient descent. In: Proc. of the 30th Int'l Conf. on Neural Information Processing Systems. Barcelona: Curran Associates Inc., 2016. 3988–3996.
- [51] Mucha PJ, Richardson T, Macon K, Porter MA, Onnela JP. Community structure in time-dependent, multiscale, and multiplex networks. Science, 2010, 328(5980): 876–878. [doi: [10.1126/science.1184819](https://doi.org/10.1126/science.1184819)]
- [52] Salimans T, Ho J, Chen X, Sidor S, Sutskever I. Evolution strategies as a scalable alternative to reinforcement learning. arXiv:1703.03864, 2017.
- [53] Lin JY, Chen YP. Analysis on the collaboration between global search and local search in memetic computation. IEEE Trans. on Evolutionary Computation, 2011, 15(5): 608–623. [doi: [10.1109/TEVC.2011.2150754](https://doi.org/10.1109/TEVC.2011.2150754)]
- [54] Xue K, Qian C, Xu L, Fei XD. Evolutionary gradient descent for non-convex optimization. In: Proc. of the 30th Int'l Joint Conf. on Artificial Intelligence. Montreal: IJCAI, 2021. 3221–3227.
- [55] Qian H, Yu Y. Derivative-free reinforcement learning: A review. Frontiers of Computer Science, 2021, 15(6): 156336. [doi: [10.1007/s11704-020-0241-4](https://doi.org/10.1007/s11704-020-0241-4)]
- [56] Sigaud O. Combining evolution and deep reinforcement learning for policy search: A survey. arXiv:2203.14009, 2022.
- [57] Majid AY, Saaybi S, Van Rietbergen T, Francois-Lavet V, Prasad RV, Verhoeven C. Deep reinforcement learning versus evolution strategies: A comparative survey. arXiv:2110.01411, 2021.
- [58] Cui P, Wang X, Pei J, Zhu WW. A survey on network embedding. IEEE Trans on Knowledge and Data Engineering, 2019, 31(5): 833–852. [doi: [10.1109/TKDE.2018.2849727](https://doi.org/10.1109/TKDE.2018.2849727)]
- [59] Shen X, Chung FL. Deep network embedding for graph representation learning in signed networks. IEEE Trans. on Cybernetics, 2020, 50(4): 1556–1568. [doi: [10.1109/TCYB.2018.2871503](https://doi.org/10.1109/TCYB.2018.2871503)]
- [60] Facchetti G, Iacono G, Altafini C. Computing global structural balance in large-scale signed social networks. Proc. of the National Academy of Sciences of the United States of America, 2011, 108(52): 20953–20958. [doi: [10.1073/pnas.1109521108](https://doi.org/10.1073/pnas.1109521108)]
- [61] Han XN, Liu HP, Sun FC, Zhang XY. Active object detection with multistep action prediction using deep Q-network. IEEE Trans. on Industrial Informatics, 2019, 15(6): 3723–3731. [doi: [10.1109/TII.2019.2890849](https://doi.org/10.1109/TII.2019.2890849)]
- [62] Todorov E. Efficient computation of optimal actions. Proc. of the National Academy of Sciences of the United States of America, 2009, 106(28): 11478–11483. [doi: [10.1073/pnas.0710743106](https://doi.org/10.1073/pnas.0710743106)]
- [63] Zhu YH, Zhao DB. Online minimax Q network learning for two-player zero-sum Markov games. IEEE Trans. on Neural Networks and Learning Systems, 2022, 33(3): 1228–1241. [doi: [10.1109/TNNLS.2020.3041469](https://doi.org/10.1109/TNNLS.2020.3041469)]
- [64] Chen SL, Wu HZ, Han XL, Xiao L. Multi-step truncated Q learning algorithm. In: Proc. of the 2005 Int'l Conf. on Machine Learning and

- Cybernetics. Guangzhou: IEEE, 2005. 194–198. [doi: [10.1109/ICMLC.2005.1526943](https://doi.org/10.1109/ICMLC.2005.1526943)]
- [65] Girvan M, Newman MEJ. Community structure in social and biological networks. *Proc. of the National Academy of Sciences of the United States of America*, 2002, 99(12): 7821–7826. [doi: [10.1073/pnas.122653799](https://doi.org/10.1073/pnas.122653799)]
- [66] Lancichinetti A, Fortunato S. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 2009, 80(1): 016118. [doi: [10.1103/PhysRevE.80.016118](https://doi.org/10.1103/PhysRevE.80.016118)]
- [67] Ghosn F, Palmer G, Bremer SA. The MID3 data set, 1993–2001: Procedures, coding rules, and description. *Conflict Management and Peace Science*, 2004, 21(2): 133–154. [doi: [10.1080/07388940490463861](https://doi.org/10.1080/07388940490463861)]
- [68] Oda K, Matsuoka Y, Funahashi A, Kitano H. A comprehensive pathway map of epidermal growth factor receptor signaling. *Molecular Systems Biology*, 2005, 1(1): 2005.0010. [doi: [10.1038/msb4100014](https://doi.org/10.1038/msb4100014)]
- [69] Oda K, Kimura T, Matsuoka Y, Funahashi A, Muramatsu M, Kitano H. Molecular interaction map of a macrophage. *AfCS Research Reports*, 2004, 2(14): 1–12.
- [70] Milo R, Shen-Orr S, Itzkovitz S, Kashtan N, Chklovskii D, Alon U. Network motifs: Simple building blocks of complex networks. *Science*, 2002, 298(5594): 824–827. [doi: [10.1126/science.298.5594.824](https://doi.org/10.1126/science.298.5594.824)]
- [71] Ma LJ, Huang XP, Li JQ, Lin QZ, You ZH, Gong MG, Leung VCM. Privacy-preserving global structural balance computation in signed networks. *IEEE Trans. on Computational Social Systems*, 2020, 7(1): 164–177. [doi: [10.1109/TCSS.2019.2944002](https://doi.org/10.1109/TCSS.2019.2944002)]
- [72] Leskovec J, Huttenlocher D, Kleinberg J. Signed networks in social media. In: *Proc. of the 2010 SIGCHI Conf. Human Factors in Computing Systems*. Georgia: ACM Press, 2010. 1361–1370. [doi: [10.1145/1753326.1753532](https://doi.org/10.1145/1753326.1753532)]
- [73] Srinivas M, Patnaik LM. Adaptive probabilities of crossover and mutation in genetic algorithms. *IEEE Trans. on Systems, Man, and Cybernetics*, 1994, 24(4): 656–667. [doi: [10.1109/21.286385](https://doi.org/10.1109/21.286385)]
- [74] Liu C, Li B. Memetic algorithm with adaptive local search depth for large scale global optimization. In: *Proc. of the 2014 IEEE Congress on Evolutionary Computation (CEC)*. Beijing: IEEE, 2014. 82–88. [doi: [10.1109/CEC.2014.6900456](https://doi.org/10.1109/CEC.2014.6900456)]
- [75] Liu CJ, Wei JH, Qiao Y, Gao YX, Sun GJ. An adaptive stochastic search algorithm. In: *Proc. of the 2011 IEEE Int'l Conf. on Automation and Logistics (ICAL)*. Chongqing: IEEE, 2011. 154–158. [doi: [10.1109/ICAL.2011.6024702](https://doi.org/10.1109/ICAL.2011.6024702)]
- [76] Yun YS, Gen M, Moon C. Hybrid genetic algorithm with adaptive local search for precedence-constrained sequencing problems. In: *Proc. of the 40th Int'l Conf. on Computers & Industrial Engineering*. Awaji: IEEE, 2010. 1–6. [doi: [10.1109/ICCIE.2010.5668428](https://doi.org/10.1109/ICCIE.2010.5668428)]
- [77] Wang YP, Xin B, Chen J. An adaptive memetic algorithm for the joint allocation of heterogeneous stochastic resources. *IEEE Trans. on Cybernetics*, 2022, 52(11): 11526–11538. [doi: [10.1109/TCYB.2021.3087363](https://doi.org/10.1109/TCYB.2021.3087363)]

附中文参考文献:

- [1] 刘大有, 李晶, 杨博. 信息传播网络学习方法. *吉林大学学报(理学版)*, 2012, 50(4): 767–774. [doi: [10.13413/j.cnki.jdxblxb.2012.04.010](https://doi.org/10.13413/j.cnki.jdxblxb.2012.04.010)]
- [4] 公茂果, 王善峰, 张天乐, 毛贻顺, 杨月磊, 牛旭东, 武越. 基于深度自编码器的复杂网络节点影响力最大化方法. *中国*, 108510059A. 2018-09-07.
- [8] 李国良, 楚娅萍, 冯建华, 徐尧强. 多社交网络的影响力最大化分析. *计算机学报*, 2016, 39(4): 643–656. [doi: [10.11897/SP.J.1016.2016.00643](https://doi.org/10.11897/SP.J.1016.2016.00643)]
- [9] 易秀双, 胡金林, 王兴伟. 基于社交网络弱连接属性的影响力最大化算法. *软件学报*, 2016, 27: 1–11. <http://www.jos.org.cn/1000-9825/16012.htm>
- [13] 韩忠明, 陈炎, 刘雯, 原碧鸿, 李梦琪, 段大高. 社会网络节点影响力分析研究. *软件学报*, 2017, 28(1): 84–104. <http://www.jos.org.cn/1000-9825/5115.htm> [doi: [10.13328/j.cnki.jos.005115](https://doi.org/10.13328/j.cnki.jos.005115)]
- [17] 田家堂, 王轶彤, 冯小军. 一种新型的社会网络影响最大化算法. *计算机学报*, 2011, 34(10): 1956–1965. [doi: [10.3724/SP.J.1016.2011.01956](https://doi.org/10.3724/SP.J.1016.2011.01956)]
- [24] 张峻伟, 吕帅, 张正昊, 于佳玉, 龚晓宇. 基于样本效率优化的深度强化学习方法综述. *软件学报*, 2022, 33(11): 4217–4238. <http://www.jos.org.cn/1000-9825/6391.htm> [doi: [10.13328/j.cnki.jos.006391](https://doi.org/10.13328/j.cnki.jos.006391)]
- [26] 金弟, 刘大有, 黄晶, 何东晓, 王新华. 基于免疫和进化扩散算法的全局优化问题求解算法. *吉林大学学报(工学版)*, 2009, 39(1): 124–130. [doi: [10.13229/j.cnki.jdxgbxb2009.01.046](https://doi.org/10.13229/j.cnki.jdxgbxb2009.01.046)]



马里佳(1990—), 男, 博士, 助理教授, 主要研究领域为演化计算, 深度学习, 复杂网络结构与行为分析.



李坚强(1980—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为人工智能, 机器人, 物联网, 移动医疗.



洪华平(1998—), 男, 硕士生, 主要研究领域为深度学习, 复杂网络结构分析.



公茂果(1979—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为计算智能理论与方法, 网络信息感知与隐私保护, 雷达与遥感智能系统.



林秋镇(1984—), 男, 博士, 副教授, 主要研究领域为演化计算, 多目标优化.

www.jos.org.cn

www.jos.org.cn