

基于飞蛾扑火算法的关键节点挖掘方法*

许钦钧^{1,2,3}, 徐龙琴^{1,2,3†}, 刘双印^{1,2,3}, 赵兴华⁴

(1. 仲恺农业工程学院 信息科学与技术学院, 广州 510225; 2. 广东省高校智慧农业工程技术研究中心, 广州 510225; 3. 广州市农产品质量安全溯源信息技术重点实验室, 广州 510225; 4. 深圳信息职业技术学院 数字媒体学院, 广东 深圳 518172)

摘要: 关键节点挖掘在理解和控制复杂网络系统方面具有重要作用和巨大潜力。提出了一种基于飞蛾扑火优化算法的关键节点挖掘算法, 解决关键节点问题。该算法引入了反向学习等策略, 以提高解集的质量和加快收敛。同时, 设计了快速种群演化和复合高斯进化等方法, 以优化解集并增强解空间探索能力, 从而克服局部最优陷阱。在多个合成网络和真实网络数据集上进行的对比实验结果表明, 提出的算法相较于其他先进的对比算法具有更高的鲁棒性, 并验证了该算法部件的有效性。

关键词: 关键节点; 网络连通性; 群智能算法; 飞蛾扑火优化算法

中图分类号: TP393.02 **文献标志码:** A **文章编号:** 1001-3695(2023)09-023-2713-07

doi: 10.19734/j.issn.1001-3695.2023.02.0032

Enhanced moth-flame optimization algorithm for critical node detection

Xu Qinjun^{1,2,3}, Xu Longqin^{1,2,3†}, Liu Shuangyin^{1,2,3}, Zhao Xuehua⁴

(1. College of Information Science & Technology, Zhongkai University of Agriculture & Engineering, Guangzhou 510225, China; 2. Intelligent Agriculture Engineering Research Center of Guangdong Higher Education Institutes, Guangzhou 510225, China; 3. Guangzhou Key Laboratory of Agricultural Products Quality & Safety Traceability Information Technology, Guangzhou 510225, China; 4. School of Digital Media, Shenzhen Institute of Information Technology, Shenzhen Guangdong 518172, China)

Abstract: The detection of critical nodes plays an important role with significant potential in understanding and controlling complex network systems. This paper proposed critical node mining algorithm based on the moth-flame optimization to address the critical node problem. It introduced strategies such as opposition-based learning to improve the quality of the solution set and accelerate convergence. Additionally, it designed fast population evolution and hybrid Gaussian evolution methods to optimize the solution set and enhance the exploration capability of the solution space, overcoming local optima traps. Comparative experimental results conducted on multiple synthetic and real network datasets demonstrate that the proposed algorithm exhibits higher robustness compared to other advanced comparative algorithms. Furthermore, the effectiveness of the algorithm components is validated.

Key words: critical nodes; network connectivity; swarm intelligence algorithm; moth-flame optimization

近年来,由于关键节点挖掘在计算生物学^[1]、网络脆弱性评估^[2]、社交网络分析^[3]、传染病传播控制^[4]、精准营销等领域中的广泛应用潜力,该问题备受关注。例如,在流行病传播网络中,发现关键节点并采取适当措施能够大大减少疾病传播,这使得关键节点挖掘研究和调节复杂网络的功能具有重要意义。然而,经典关键节点挖掘问题是 NP-难问题^[5],这意味着其精确解法需要承担呈指数增长的计算复杂度,难以在大型网络中被接受。为了解决这个问题,本文提出了一种基于飞蛾扑火优化算法的关键节点挖掘算法(moth flame optimization for critical node problem, MFOCNP)。该算法包含一个快速种群进化(fast population evolution, FPE)机制,结合了相反学习(opposition-based learning, OBL)策略以挖掘具有高质量和多样性的初始种群;一种被称为飞蛾扑火优化算法(MFO)的群智能优化算法,通过探索新解决方案并克服局部最优陷阱实现优化;

以及一种复合高斯进化(HGE)机制,用于加强探索能力和加速收敛。为了验证本文方法的有效性,在合成网络和真实网络上进行了实验,并与其他一流算法进行了比较。实验结果表明,该算法的性能优于其他算法。

1 相关工作

1.1 关键节点问题进展

经典关键节点挖掘问题(critical node detection problem, CNDP)的定义是:对于一个有 $n = |V|$ 个点、 $m = |E|$ 条边的网络 $N = (V, E)$ 和一个正整数 K , CNDP问题求一个规模不大于 K 的子集 $S \subseteq V$,且删除它能最大限度地降低剩余网络 $N[V \setminus S]$ 的成对连通性(连接节点对的总和),求得的点集 S 就是网络关键节点的集合。

收稿日期: 2023-02-02; **修回日期:** 2023-04-04 **基金项目:** 国家自然科学基金资助项目(61871475); 广东省自然科学基金资助项目(2021A1515011994); 广州市重点研发计划资助项目(202103000033, 201903010043); 广东省科技计划资助项目(2020A1414050060, 2020B0202080002, 2016A020210122, 2015A040405014); 广东省普通高校创新团队项目(2021KCXTD019, 2020KCXTD040, 2022KCXTD057); 广东省普通高校特色创新项目(KA190578826); 梅州市科技计划资助项目(2021A0305010); 广州市增城区农村科技特派员资助项目(2021B42121631); 广东省教育科学规划课题(2020GXJK102, 2018GXJK072); 广东省研究生教育创新计划资助项目(2022XSLT056, 2022JGXM115)

作者简介: 许钦钧(1998-), 男, 湖北黄冈人, 硕士, 主要研究方向为复杂网络; 徐龙琴(1977-), 女(通信作者), 教授, 硕士, 主要研究方向为智能信息处理、农业物联网、数据挖掘(xlqw@126.com); 刘双印(1977-), 男, 教授, 博士, 主要研究方向为智能信息处理、物联网、大数据; 赵兴华(1977-), 男, 副教授, 博士, 主要研究方向为复杂网络、数据挖掘、大数据。

由于该问题的理论意义和实用潜力,文献中提出大量算法,它们主要分为精确方法和启发式方法。

精确方法主要利用整数线性规划(integer linear programming, ILP)将特定网络中的关键节点挖掘问题转换为可解的目标函数方程式。文献[5]提出了一个简化的函数方程和算法(HDCN)。文献[6]则提出了一种基于 Benders 分解的方法以精确识别随机网络上的关键节点。文献[7]使用分支切割算法解决一般网络上的关键节点挖掘,然而,由于该问题的 NP 难特性^[5],在一般网络和大型网络中,精确算法的计算成本较高。所以,启发式算法,特别是贪婪算法是一个性价比高且必要的替代选择,其能在有限时间内找出高质量的近似解,如 BCB^[8]、LRBG^[9]、KSBG^[10]等算法。以下介绍两种常见的贪婪算法程序。

第一种贪婪算法程序^[5]是对网络中每个节点进行计算,找出一个删除它后降低网络连通性最多的点,并将其从网络移入点集 S ;重复上述步骤直到点集大小达到预设要求。第二种程序是逐一移除有边的点并将其加入到点集中,直到网络中没有边存在。随后,贪心地将点集中的点逐一放回网络,使得每次放回的节点对网络增加的连通节点数最少,直到点集大小达到预设要求。这两种贪婪算法比较快速、高效,能够找出近似最优解,但结果缺乏准确性。为了改善贪婪算法的缺陷,研究者提出了很多其他启发式和元启发式算法。NSGACNP^[11]和复合启发式算法(Greedy3D/Greedy4D)^[12]使用贪婪算法作为基础,通过交换部分节点来改进解的质量。朱华等人^[13]通过增强贪婪算法,设计一种新的级联概率计算模型挖掘关键节点集。周丽娜等人^[14]利用超图中的邻接结构熵识别超网络中的关键节点。孙百兵等人^[15]根据网络社区结构提出一种节点重要性评估函数。刘子彤等人^[16]提出一种基于加权集体影响力模型,用于识别加权通信网络中的关键节点。此外,元启发式算法也提供了一些新颖的思路,如 Ventresca^[17]提出了退火算法(simulated annealing, SA)和基于增量学习的群体算法(incremental learning based on population, PBIL),并验证了其有效性。

1.2 飞蛾扑火优化算法简介

飞蛾扑火优化算法(moth-flame optimization algorithm, MFO)是一种基于横向定位和螺旋运动的群智能优化算法,它最初的灵感来源于飞蛾的行为模式。飞蛾以月亮为参照点,其飞行的方向与它跟月亮的连线保持固定角度,这种导航方法可以使飞蛾在夜间保持直线飞行。然而,人造光源会误导飞蛾绕着它飞行,甚至飞进光源,这就是“飞蛾扑火”现象(图1)。Mirjalili^[18]对飞蛾的行为模式进行了数学模拟,并提出了一种基于横向定位和螺旋运动的优化算法,即飞蛾扑火优化算法。该算法结合了飞蛾的行为模式和螺旋运动,具有较好的全局搜索能力和收敛速度。在近年来的群智能优化领域,它被认为是一种较为流行的优化算法。飞蛾的螺旋运动如图1所示。

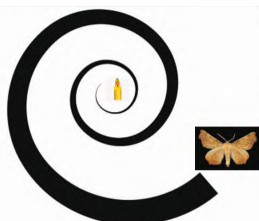


图1 飞蛾的螺旋运动
Fig.1 Spiral motion of moths

MFO 是一种基于群智能优化的算法,由于其鲁棒性和易

实现性,已被广泛应用于复杂问题的求解中。例如文献[19]利用 MFO 提取多晶太阳能电池二极管模型的最佳参数,且检验了该方法在求解类似问题时的优越性能;Zhao 等人^[20]将遗传算法(GA)、MFO 算法和滚曲机构(rolling mechanism)结合起来,开发了一种复合的用电量预测方法。

不过,MFO 存在容易陷入局部最优的问题。为了解决这个问题,一些研究人员通过结合其他算法或嵌入突变机制等方法来增强 MFO 的全局搜索和优化能力。例如 Barczak 等人^[21]结合随机游走萤火虫(Lévy-flight firefly)算法对 MFO 进行了增强;Hassanien 等人^[22]设计了一种用于番茄病害检测的改良 MFO。这些方法都试图在探索新解和挖掘旧解之间取得平衡,以便更好地克服 MFO 的局部最优问题。还有一些研究人员采用了其他方式来增强 MFO 的全局搜索和优化能力。

1.3 网络理论基础

定义1 复杂网络(complex network)。复杂网络即呈现高度复杂性的网络。用无向图 $N=(V, E)$ 表示,由点集 V 和无向边集 E 构成。图2展示了一个复杂网络的实例。

定义2 点的度数(degree)。其表示为点 i 的邻居节点的数量 $\text{degree}(i) = |Nb_i|$,其中点 i 的邻居节点集 $Nb_i = \{j \in V: (i, j) \in E\}$ 。

定义3 剩余网络(residual network)。对一个点集 $S \subseteq V$,剩余网络 $N[V \setminus S]$ 即为删除 S 后网络的导出子图 $N[V \setminus S]$ 。

$$N[V \setminus S] = N(V \setminus S, E(V \setminus S))$$

其中: $E(V \setminus S) = \{(i, j): i, j \in V \setminus S\}$ 。

定义4 连通节点对(connected nodes)和连通网络(connected network)。如果点 $i \in V$ 可以沿着网络中的一条路径到达另一点 $j \in V$,则 $[i, j]$ 是一个连通节点对。连通节点对越多,网络的成对连通性越好。如 $\forall i, j \in V$ 相互连通,则该网络为连通网络。

定义5 相反数(opposite number)。对于一个实数 $x \in [p, q]$,其对应的相反数 x' 的计算方法为 $x' = p + q - x$ 。例如, $d = 3$,它在 $[0, 4]$ 上的对应数字是 $d' = 1$ 。

定义6 反向学习^[23]。设 $f(x)$ 为目标函数,求极大值。 $y \in [p, q]$ 是一个候选解,而 y' 是它的相对数。计算 $f(y)$ 和 $f(y')$ 。如果 $f(y) > f(y')$,保留 y 的值;否则,保留 y' ,舍弃 y 。相反学习通过对比相反解的值,优化初始解。

定义7 关键节点挖掘问题。图3中有颜色标注的点比较重要,删除它们会破坏网络连通,这种节点被称为关键节点(critical node)。

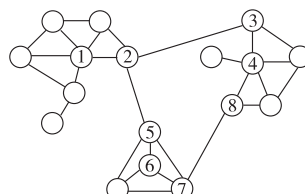


图2 复杂网络

Fig.2 Complex network

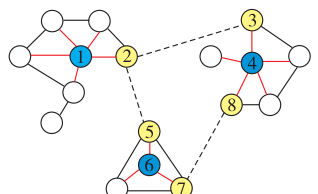


图3 关键节点挖掘

Fig.3 Critical node mining

对于一个无向网络 $N=(V, E)$ 和一个正整数 K ,经典的关键节点检测问题(CNDP)求一个有限的节点集 $S \subseteq V (|S| \leq K)$,使得去除 S 后剩余网络 $N[V \setminus S]$ 的连通节点对最少。其目标函数表示为

$$\arg \min f(S) = |\{[i, j]: i \text{ 和 } j \text{ 在 } N[V \setminus S] \text{ 中是连通的}\}| \quad (1)$$

其中: $i, j \in V \setminus S$ 。目标函数 $f(S)$ 值越小,表示删除 S 后剩余网络 $N[V \setminus S]$ 的连通节点对越少,则其成对连通性越差。

2 基于 MFO 的关键节点挖掘方法

2.1 种群表示方法

对于一个网络 $N = (V, E)$ 和给定的正整数 k , 任何不超过 k 个节点的子集 $S \subseteq V$ 都是关键节点挖掘问题的候选解。在算法中, 每个候选解都是种群中的一个个体。个体分为飞蛾和火焰, 其中飞蛾是搜索代理 (search agent), 而火焰存储当前最优解。个体的变量组 $m = (n_1, n_2, n_3, \dots, n_k)$ 代表了飞蛾或火焰的位置。飞蛾的搜索过程会改变个体的位置, 也改变了变量组的值。反过来, 个体的位置对应着一个节点集 $m = (n_1, n_2, n_3, \dots, n_k)$, 其中 $m \in V_k, |m| = k, n$ 只飞蛾的种群可以用矩阵表示。

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,k} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,k} \\ \vdots & \vdots & & \vdots \\ m_{n,1} & m_{n,2} & \cdots & m_{n,k} \end{bmatrix} \quad (2)$$

其中: k 是变量 (维度) 的数量, 它等于关键节点的数量; n 是飞蛾 (搜索代理) 的数量。每个飞蛾个体 (位置) 对应一个节点集 S , 因此飞蛾个体的适应度值对应着目标函数值, 即剩余网络 $N[V \setminus S]$ 的连通节点对。飞蛾种群的适应度值以数组形式表示为

$$FM = \begin{bmatrix} F_{m1} \\ F_{m2} \\ \vdots \\ F_{mn} \end{bmatrix} \quad (3)$$

同样, 火焰种群的位置用矩阵表示为

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \cdots & F_{1,k} \\ F_{2,1} & F_{2,2} & \cdots & F_{2,k} \\ \vdots & \vdots & & \vdots \\ F_{n,1} & F_{n,2} & \cdots & F_{n,k} \end{bmatrix} \quad (4)$$

其中: k 为关键节点的个数。火焰对应的适应度值为一个矩阵。

$$FF = \begin{bmatrix} FF_1 \\ FF_2 \\ \vdots \\ FF_n \end{bmatrix} \quad (5)$$

如上所述, 飞蛾和火焰的个体都是候选解, 但它们的功能和更新方法不同。飞蛾是探索新解的搜索代理, 而火焰存储了迄今为止最好的解, 用于挖掘旧解。火焰的位置随着飞蛾飞行的最佳位置而更新, 而飞蛾则通过在火焰周围螺旋飞行来改变它们的位置。这种螺旋运动机制在勘探新解 (exploration) 与挖掘旧解 (exploitation) 之间建立了较好的平衡。

2.2 MFOCNP 算法介绍

MFOCNP 可分为种群初始化和快速种群演化、螺旋搜索和复合高斯进化三个步骤。步骤 a) 该方法利用相反学习 (opposition-based learning) 和快速种群进化 (fast population evolution, FPE) 机制生成高质量的初始解; 步骤 b) 每个飞蛾被分配一个火焰且围绕其作螺旋运动, 并朝火焰逐渐收敛, 在这个过程中飞蛾探索更广阔的解空间并将当前最优的一批解的位置保存为火焰; 步骤 c) 飞蛾的每个新位置都使用复合高斯进化 (hybrid gaussian evolution, HGE) 来优化, 以克服局部最优的陷阱。步骤 c) 完成后转到步骤 b), 直到满足一定迭代次数。飞蛾的搜索有机组合多种有效机制, 实现了探索新解和挖掘旧解的平衡。算法 1 给出了该算法的伪代码。

算法 1 基于 MFO 的关键节点挖掘方法

输入: 网络 $N = (V, E)$; 正整数 k ; 最大迭代次数 i_{max} ; 搜索区间的下限和上限 (ub, lb); 飞蛾数量 n 。

输出: $flame$ 。

$M_0 \leftarrow \text{initialization}(V, k, ub, lb, n)$; // 初始化

$M_1 \leftarrow \text{FPE}(M_0)$; // 快速种群进化

$i \leftarrow 1$; // 迭代次数

while $i < i_{max}$ do

 update $flame_no$ using 式 (9); // 更新火焰数量 $flame_no$

 if $i == 1$ then

$flame = \text{sort}(flame_no, M_1)$;

 end

 else

$flame = \text{sort}(flame_no, M_{i-1}, M_i)$;

 end

 for $a \leftarrow 1$ to n do

 for $b \leftarrow 1$ to k do

 update z, t using 式 (7);

 calculate D using 式 (8); // 根据对应的火焰和公式更新

新 D *

 update $M(a, b)$ using 式 (6); // 飞蛾坐标更新

 end

 end

$M_i = \text{HGE}(M_i)$; // 复合高斯进化

$i \leftarrow i + 1$;

end

2.2.1 种群初始化与快速种群演化

第一个阶段, 搜索开始阶段的目的是初始化一个具有高质量和多样性的种群。

算法 2 种群初始化

输入: 搜索空间的下界和上界 (lb, ub); 种群大小 n ; 正整数 k 。

输出: 种群 M 。

for $i \leftarrow 1$ to n do

 for $j \leftarrow 1$ to k do

$M_{(i,j)} \leftarrow \text{round}((ub - lb) * \text{rand}() + lb)$; // 随机生成初

始解集 *

 end

end

首先按照算法 2 随机生成有 n 个个体的种群。搜索空间的下界为 1, 上界为网络节点数 $|V|$ 。算法中, 一旦变量超过上 (或下) 边界, 其位置将被重置为上 (或下) 边界的值。然后采用快速种群进化 (FPE) 方法作为算法 3 对种群进行改进。用相反学习 (OBL) 方法快速找到更好的解。OBL 机制从每个解中生成一个相反解, 并通过比较两者的适应度值来保留更好的解。最后, 使用快速种群进化机制对每个个体 m 进行操作: 从剩余节点集 $V \setminus m$ 中随机选择节点, 并将其移动到个体 m 中, 然后从个体 m 中计算并选择一个节点 v_i , 并将节点 v_i 从点集 m 移动到剩余网络中, 且节点 v_i 的移除使剩余网络的成对连通性下降最多。将新个体 m' 与原个体 m 的适应度值进行比较, 将值更高的个体保存为飞蛾的位置。上述过程重复 $I_{max} = 5$ 次。

算法 3 快速种群演化

输入: 种群 M , 最大迭代次数 I_{max} 。

输出: M 。

for $i \leftarrow 1$ to I_{max} do

 for each moth m of the population M do

$m' \leftarrow \text{OBL}(m)$; // 对种群 M 的每一个个体进行相反学习

 if $\text{fitness}(m') > \text{fitness}(m)$ then // 保存更适应的个体

$m \leftarrow m'$;

 end

$node \leftarrow \text{random}(V \setminus m)$;

$m' \leftarrow m \cup \{node\}$; // 随机选点, 并删除

```

 $v_i \leftarrow \arg \min v_x \in m \{f(m \setminus v_x) - f(m)\};$ 
 $m \leftarrow m \setminus v_i;$  //计算并删除最差的点
if fitness( $m'$ ) > fitness( $m$ ) then
     $m \leftarrow m';$ 
end
end
end

```

2.2.2 螺旋运动和高斯进化

第二阶段,获得了高质量的初始解集后,采用飞蛾扑火优化算法对解集进行迭代优化。飞蛾在相应的火焰周围螺旋飞行,利用螺旋运动探索解空间。在此过程中,飞蛾个体的位置变量改变的计算公式为

$$m_{p,q} = \text{round}(D_{p,q} \times e^{bt} \times \cos(2\pi t) + F_{j,q}) \quad (6)$$

其中: m_p 是第 p 个飞蛾; F_j 是它对应的火焰; $m_{p,q}$ 和 $F_{j,q}$ 分别是对应飞蛾与火焰的第 q 个变量; $D_{p,q}$ 是飞蛾与火焰在第 q 个变量上的几何距离; t 为 $[z, 1]$ 中的随机数; b 设为 1。参数 t 决定了飞蛾的下一个位置距离对应火焰的远近,其中 $t = z$ ($-2 \leq z \leq -1$) 会让飞蛾新位置离距离中心最近, $t = 1$ 会导致它飞向距离中心最远的位置(图4)。此外 t 的随机性使运动不确定,给种群的解带来多样性。当 z 减小时,飞蛾的运动在迭代中更接近火焰,这有助于更快的收敛。参数 z 随着迭代次数增加从 -1 减小到 -2 。

$$z = -1 + (-1) \times \frac{t_{\text{num}}}{t_{\text{max}}} \quad (7)$$

其中: t_{num} 是目前的迭代次数; t_{max} 是最大迭代次数。 D 为每个飞蛾与相应火焰之间的几何距离,其变量计算方法为

$$D_{p,q} = |F_{j,q} - m_{p,q}| \quad (8)$$

图4给出了飞蛾绕着相应火焰螺旋飞行的模型。有了这些机制,飞蛾可以探索和挖掘火焰周围广阔的解空间,如图4标记为 1, 2, 3, 4 的位置。当前的最佳解设置为火焰的坐标,而飞蛾围绕火焰位置移动。为了克服局部最优陷阱,每个飞蛾跟随不同的火焰,并按照式(6)移动。此外,为了收敛探索过程,本文将火焰的数量从 n 均匀地减少到 1。

$$\text{flame_no} = \text{round}(n - \frac{t_{\text{num}}}{t_{\text{max}}} \times (n - 1)) \quad (9)$$

其中: flame_no 为火焰的数量; n 为飞蛾的数量; t_{num} 为目前的迭代次数; t_{max} 为迭代的最大次数。逐渐减少火焰数量实现了探索新解与挖掘旧解之间的平衡。在螺旋运动中,火焰的位置随着目前最优的飞蛾位置更新,并根据适应度值进行排序。飞蛾按照顺序选择和跟随相应的火焰(例如第一只蛾子跟随第一个和最好的火焰,第二只跟随第二个火焰,依此类推;飞蛾多于火焰时,多出来的蛾子也都跟随第一个火焰)。为了进一步加快收敛速度和提高解的质量,本文实现了复合高斯进化(hybrid Gaussian evolution)方法作为算法4。飞蛾变量采用高斯突变(Gaussian mutation)的方法进行更新,突变公式如下:

$$f_{\text{gauss}}(0, \sigma^2)(\alpha) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\alpha^2}{2\sigma^2}} \quad (10)$$

其中: σ^2 为众飞蛾(搜索代理)变量的方差。高斯变异的计算方法为

$$m'_i = m_i \times (1 + G(\alpha)) \quad (11)$$

其中: $G(\alpha)$ 是式(10)中的高斯分布中的均匀分布随机数。如果新位置的适应度值优于原位置,则飞蛾的位置随之更新。

另外,为了提高解的多样性和克服局部最优造成的探索阻碍,使用了一种三步节点交换策略来更新解集。该策略包括重构、添加和移除三个阶段。设 $S(|S| = k)$ 为飞蛾 m_i 对应的节点集,则 $V \setminus S$ 为 S 对于网络节点集 V 的补集。在重构阶段,通过节点交换方法修改了解,其中随机将给定百分比 px 的节点从

S 移入剩余网络 $N[V \setminus S]$, 然后从 $N[V \setminus S]$ 随机移动 px 的节点回到 S 。被移除的节点数计算为

$$px = \text{round}(k \times x) \quad (12)$$

其中: k 是最大关键节点数; $x = 0.05$ 是调整参数。此外,从剩余网络 $N[V \setminus S]$ 随机移动一些节点到 S 。移动节点的数量计算为

$$pn = \text{round}(k \times pw) \quad (13)$$

其中: $pw = 0.2$ 是调整参数。在移除阶段,计算并选择一个节点加入剩余网络 $N[V \setminus S]$, 使得网络成对连通性增加最少。

算法4 复合高斯进化

输入:种群 M ;最大迭代次数 t_{max} 。

输出:种群 M 。

```

for  $i \leftarrow 1$  to  $t_{\text{max}}$  do
    for each moth  $m$  of the population  $M$  do
         $m' \leftarrow \text{GM}(m);$  //高斯变异
        if fitness( $m'$ ) > fitness( $m$ ) then
             $m \leftarrow m'$ 
        end
         $m' \leftarrow \text{reconstruction}(m);$  //重构解集
        if fitness( $m'$ ) > fitness( $m$ ) then
             $m \leftarrow m'$ 
        end
    end
end
end

```

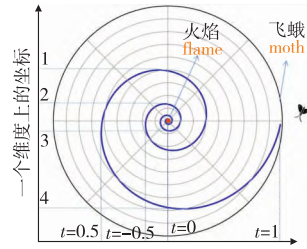


图4 螺旋运动的可能位置

Fig. 4 Possible positions of spiral motion

2.3 计算复杂度

为证明 MFOCNP 算法的效率,对算法复杂度进行分析。MFOCNP 算法主要有三个步骤:a)种群初始化和快速演化;b)飞蛾螺旋运动;c)复合高斯进化方法。第一阶段,相反学习和快速种群演化方法的复杂度分别不超过 $O(n \times |E|)$ 和 $O(k \times |E|)$, 其中 n 为蛾的数量, $|E|$ 为网络边的数量, k 为关键节点的数量。在第二阶段,火焰的更新和分配的极限为 $O(t_m \times n \times (|E| + n))$, 其中 t_m 是迭代的最大次数。螺旋运动的计算复杂度为 $O(t_m \times n \times k)$ 。最后,复合高斯进化操作的复杂度为 $O(t_m \times n \times |E| \times k)$, 因此, MFOCNP 算法的总计算复杂度为 $O(t_m \times n \times (|E| \times k + n))$, 跟传统算法相比低很多。

3 实验设计

为了评估 MFOCNP 算法的性能,在多个数据集上进行了实验。使用了多种对比算法,这些对比算法在各自的领域内具有较高的性能表现,能够提供一个有意义的比较标准。在实验中,分别使用了人工构造和真实的网络数据集。其中,人工数据集针对不同的网络拓扑特征进行构建,并加入了一定的噪声以模拟现实场景;真实数据集则来自于网络运营商和互联网公司的监测数据,包括传输数据的源地址、目的地址、时间戳等信息。为了更好地评估算法的效率,还对计算复杂度进行了分析。采用了事前分析法,通过对算法步骤的时间复杂度进行计算,得出了在不同数据规模下的算法复杂度及其增长趋势。这一分析为评估算法性能提供了理论支持。

实验运行在一台装有 AMD Ryzen 5 4600H 处理器,

3.00 GHz clock 和 16 GB RAM 的电脑上,系统运行 Windows,代码使用 Python 编写(源代码已发布在 <https://github.com/Axuqj/Enhanced-Moth-Flame-Optimization-Algorithm-for-Critical-Node-Detection.git> 上)。

3.1 对比算法

为了克服 CNP 这个 NP-难问题,许多启发式算法被提出。在验证 MFOCNP 算法的性能时,使用了一些常见的对比算法,并对它们的性能进行了比较。使用了简洁高效的关键节点挖掘算法(HDCN)^[7],利用贪婪策略初始化极大独立点集(MIS),并通过删除节点来达到目标点集大小。

基于介数中心性的贪婪算法(BCB)^[8]利用全局中心性来优化解集。复合启发式算法(Greedy3D)及其改进版本(Greedy4D)^[12]结合贪心算法和迭代局部搜索来解决局部最优解问题,而基于 K-Shell 的贪心算法(KSBG)^[10]则使用快速有效的节点中心性模型进行节点重要性评估,以提高算法性能。

GCNP 算法^[24]利用局部节点中心性和贪婪策略快速有效地找出近似最优解,而基于 LocalRank 的贪婪算法(LRBG)^[9]使用广泛应用的节点度量模型加速搜索最优解过程。

此外,还利用了一种基于非支配的多目标优化遗传算法 NSGA-II,该算法具有高效的排序算法和机制,无须事先输入参数^[15]。NSGA-II 算法在关键节点挖掘问题中被应用(即 NSGACNP),将该问题划分为剩余网络中连通图的数量和连通图之间的基方差两个独立的目标函数。

3.2 数据集

为了更好地验证算法的效果和鲁棒性,采用了各种复杂网络生成模型和真实网络数据集进行实验。

1) 合成网络的生成模型

为评估 MFOCNP 算法的效果,选择了五种经典的复杂网络生成模型来构建人工复杂网络数据集,并使用这些数据集进行实验。下面介绍这些网络生成模型:

a) Erdos-Renyi (ER) 模型^[25]。它是一种重要的网络生成模型,然而其合成的网络缺乏现实中常见的社区结构。参考文献后设定了参数^[25],并分别合成三组网络(每组包含 10 个相似网络,保存平均值),如表 1 所示。

表 1 ER 模型生成的网络

Tab. 1 Network generated by ER model

网络名称	节点数 n	连接概率 p	边数
ER235	235	0.012 6	346
ER465	465	0.006 1	651
ER940	940	0.003 1	1 388

b) Barabasi-Albert (BA) 模型^[26]。其生成的网络又被称为无标度网络,其中节点的度遵循幂律分布(指数分布)。参考文献后设定参数^[26],生成三组网络(每组包含 10 个相似网络,保存平均值),如表 2 所示。

表 2 BA 模型生成的网络

Tab. 2 Network generated by BA model

网络名称	最终节点数 n	初始节点数 m	边数
BA500	500	0.012 6	499
BA1000	1 000	0.006 1	999
BA2500	2 500	0.003 1	2 499

c) Watts-Strogatz (WS) 小世界模型^[27]。其生成的网络中多数节点几步内就能到达其他任一节点,所以被称为小世界网络。参考文献后设定参数^[27],生成三组网络,如表 3 所示。

2) 真实网络数据集

为更好地研究 MFOCNP 算法的性能,从文献和互联网选择真实复杂网络数据集用于实验,其参数如表 4 所示。其中

Electronic-circuits 网络被存储在 <https://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks>; Yeast 网络由 <https://www.weizmann.ac.il/mcb/UriAlon/download/collection-complex-networks> 提供; Polbooks 网络由 V. Krebs 在 <http://www.orgnet.com/> 上编译。

表 3 WS 模型生成的网络

Tab. 3 Network generated by WS model

网络名称	节点数 n	度数 d	连接概率 p	边数
WS250	250	10	0.3	1 250
WS500	500	6	0.3	1 500
WS1000	1 000	10	0.3	5 000

3.3 性能评价指标

采用不同性能指标对算法的性能进行综合验证,其中包括:

a) 成对连通性(pairwise connectivity)。该指标是关键节点问题的一个主要指标,它反映了剩余网络的连通程度和关键节点集的质量。具体来说,成对连通性指的是剩余网络 $N[V \setminus S]$ 中连通节点对的数量,该指标数值越小,意味着点集质量越高,算法性能越好。

b) 随机水平相对差值。将算法处理后剩余网络的成对连通性标记为 *solution*,将随机删除节点后的成对连通性标记为 *random*。采用随机水平相对差值来归一化,便于比较算法性能。随机选择策略是随机移除与算法相同数量的节点。 E_r 值越大,说明算法性能越好。相对差值 E_r 用百分数表示为

$$E_r = 100 \times \frac{\text{random} - \text{solution}}{\text{random}} \quad (14)$$

c) 性能概况指标(performance profiles)^[28]。对于算法 A,它在实例集合 T 上进行各项实验并在部分实例中提供了优秀解,性能概况指标 $p_A(n)$ 表示算法 A 提供了优秀解的实例数占总数的比例。其中,优秀解指在某实例中与最佳解的相对差值(相对差值表示为 $E = [A(I) - \text{best}(I)] / \text{best}(I)$ 不大于 2^{n-1} 的解,当 $n=0$ 时,优秀解即为某实例中出现的最佳解, $p_A(0)$ 表示算法 A 提供最优解的实例数占所有实例的比例。 $p_A(n)$ 表示为

$$p_A(n) = \frac{|\{I \in T: A(I) \leq 2^n \text{best}(I)\}|}{|T|} \quad (15)$$

其中: $I \in T$ 是 T 的一个实例; $A(I)$ 是剩余网络的连通节点对; $\text{best}(I)$ 是在实例 I 上出现的最佳解(成对连通性越小,解越好)。

表 4 真实网络

Tab. 4 Real network

网络名称	节点数 n	边数	网络名称	节点数 n	边数
Celegan ^[27]	297	2 359	Karate ^[30]	34	77
Electronic-circuits	512	819	Lesmis ^[31]	77	254
Yeast	688	1 079	Polbooks	105	441
email ^[29]	1 133	10 903			

4 实验结果与分析

本章实验主要在不同合成网络和真实网络数据集上,比较了本文算法与其他算法的有效性。采用了多种评价指标,包括连通节点对指标、随机水平相对差值和性能概况指标等,以深入分析和比较各算法的性能表现。

4.1 在人工网络数据集上的实验结果与分析

a) 连通节点对指标。表 5 中算法的结果是剩余网络的连通节点对指标,其值越小说明方法性能越好。

在表 5 中,第 1 列表示生成网络的类型和大小, k 为被移除的节点数,第 3~10 列表示所有算法的连通节点对指标,每个实验中的最优解用黑体标出。表 5 数据表示 MFOCNP 算法生成了最优的一批解。

b) 随机水平相对误差。将该算法与随机抽样策略进行比

较。随机选择策略中,随机选择与算法相同数量的节点作为解。在图 5 中,结果以百分比表示,数值越大,性能越好。在 BA 网络中,结果接近 100%,表明 MFOCNP 算法的性能是随机策略的近两倍。 x 轴表示相对误差, y 轴表示实验名称。从图 5 可以看出,性能最好的是 MFOCNP 算法。

表 5 三个合成网络中的实验结果(连通节点对)

Tab.5 Experimental results in three synthetic

networks(connected node pairs)

network	k	MFOCNP	GCNP	Greedy3D	Greedy4D	HDCN	KSBG	LRBG	BCB
BA500	70	295	346	313	313	467	437	467	467
BA1000	125	1524	2734	1938	1993	3467	3742	2764	2745
BA2500	200	5012	6476	8106	8419	10435	14375	10243	10457
BA5000	265	959407	1345287	1118785	1112685	1022754	994215	9522756	973417
ER235	50	195	2042	195	195	1878	1153	1141	
ER465	75	559	4185	559	559	1017	3404	2839	2635
ER940	100	3722	14839	3722	3722	12929	11131	10665	
ER2343	150	10196	11427	10196	10196	33096	35165	25847	25097
WS250	50	11037	13467	11694	11401	16452	11972	16110	16110
WS500	80	5629	11428	11981	11981	16294	19647	19412	24514
WS1000	140	114927	377852	318003	318003	401675	164524	175172	189185
WS1500	200	16294	17634	25467	25484	19548	18425	18649	19627

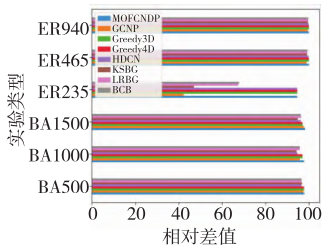


图 5 合成网络上的
随机水平相对误差

Fig.5 Random horizontal relative
error on synthetic networks

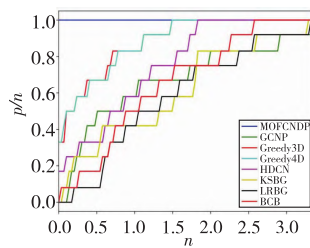


图 6 合成网络上的
性能概况指标

Fig.6 Performance overview
indicators on synthetic networks

c)性能概况指标。图 6 可以直观地比较各算法的性能表现。图 6 表示在一系列实验中,算法提供合格解的比率, n 越大,需要达到的优秀解的要求越高。曲线位置越靠近左上角,算法性能越好。从图 6 可以看出,MFOCNP 和 Greedy3D、Greedy4D 在 ER、BA 和 WS 网络中性能最好; $n=0$ 处,MFOCNP 在近 100% 的实例上提供了最佳解,充分说明了算法的鲁棒性, Greedy3D 和 Greedy4D 则表现出比其他算法更稳定的效果。

4.2 在真实网络数据集上的实验结果与分析

为进一步验证算法的性能,在真实网络上进行了实验。实验中采用不同的 K 值。

$$K = z|V| \quad z \in [0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3]$$

其中: $|V|$ 为节点总数; z 为关键节点所占比率。共生成 $8 \times 7 = 56$ 个关键节点挖掘问题实例。表 6 表明 MFOCNP 算法在大多数实验中提供了最佳解。

a)随机水平相对误差。 x 轴表示相对误差, y 轴表示网络类型。例如,lesmis1 表示 $z=0.01$ 的 lesmis 网络实例。

从图 7 中可以看出,MFOCNP 在大多数情况下比其他算法表现得更好。

b)性能概况指标(performance profiles)。从图 8 可以看出,MFOCNP 算法提供了一系列较好的解,几乎所有的最佳解都来自 MFOCNP 算法。

4.3 算法组件验证

将 MFOCNP 算法与 MFOCNP₀ 和 MFOCNP₁ 两个修改版本进行比较。两个版本分别去掉了两个关键组件。对于 MFOCNP₀ 算法,初始种群是随机生成的,且缺少了原算法所提出的 FPE 优化策略。对于 MFOCNP₁,其解更新过程相对简单,而 MFOCNP 的更新过程则通过 HGE 复合高斯进化策略,得到了很大的改进。

a)连通节点对指标。对比结果如表 7 所示。表中: z 为被移除节点的百分比; f_{best} 为最佳目标值; f_{avg} 为平均适应度值; t_{avg} 为找到最优解的平均时间。

b)随机水平相对误差。如图 9 所示,在大多数实验中,MFOCNP 算法的性能更好。并且 MFOCNP 的性能略好于 MFOCNP₁。数据结果说明两种算法组件在 MFOCNP 算法中都发挥着重要的作用。

表 6 在真实网络上的结果(连通节点对)

Tab.6 Results on real networks(connected node pairs)

network	k	MFOCNP	GCNP	Greedy3D	Greedy4D	HDCN	KSBG	LRBG	BCB
Celegan	3	821	1 511	840	840	1 448	854	1 456	1 498
	15	1 275	1357	1275	1 275	1 268	1 257	1 364	1 303
	30	638	1 163	1 095	335	1 260	1 230	1 216	1 162
	45	546	1 107	1 115	1 247	1 200	1 180	1 094	1 193
	60	524	1 052	942	1 146	1 096	1 012	1 055	1 027
	75	467	1 057	807	890	889	842	953	953
	90	354	851	863	811	800	712	861	917
	6	452	454	484	452	482	488	490	462
	26	446	495	767	734	714	500	483	471
	52	369	485	793	740	697	444	480	532
Electronic-circuits	77	378	475	832	755	735	410	443	504
	103	347	457	760	726	687	615	444	522
	128	349	432	670	634	611	507	413	462
	154	194	502	652	210	194	401	432	433
	7	1 649	1 807	1 770	1 908	1 807	1 715	1 766	1 718
	35	1 497	1 592	1 588	1 805	1 696	1 600	1 601	1 750
Yeast	69	1 294	1 407	1 658	1 766	1 686	2 885	1 527	1 652
	108	861	1 376	1 600	3 187	3 044	4 836	1 380	2 371
	138	547	1 192	2 750	3 560	3 467	4 225	1161	4 348
	172	349	1 101	2 458	3 354	3 287	4 172	1 099	4 249
	207	268	1 012	2 006	3 646	3 380	3 941	967	4 249
	12	4 164	4 701	4 553	4 592	4 325	4 320	4 406	4 531
email	57	3 497	4 328	5 090	4 661	4 536	4 393	4 638	4 421
	114	2 497	4 168	4 611	5 129	4 680	3 563	4 656	4 941
	170	1 971	4 223	4 020	4 619	4 287	3 163	4 386	4 148
	227	1 642	4 290	3 307	3 586	3 387	2 837	4 517	3 221
	284	843	4 148	3 023	3 106	2 855	2 299	4 452	2 859
	340	579	3 910	2 491	2 587	2 572	2 159	4 175	2 334
Karate	1	163	166	135	135	177	135	163	167
	2	127	173	127	127	158	134	171	156
	4	84	145	84	84	144	129	133	149
	6	82	130	95	144	138	112	114	121
	7	46	139	78	144	135	111	108	111
	9	27	119	68	139	130	88	85	77
Lesmis	11	6	107	53	114	113	78	67	77
	1	210	295	143	143	280	210	302	283
	4	150	243	232	215	203	213	232	150
	8	127	218	151	221	208	207	222	151
	12	116	158	153	152	151	168	153	128
	16	104	158	145	149	146	163	146	116
Polbooks	20	64	134	148	137	126	169	151	79
	24	47	141	149	138	130	123	138	67
	2	413	425	413	413	419	445	436	413
	6	251	261	393	351	323	415	311	411
	11	162	232	295	348	329	439	162	248
	16	104	204	227	279	265	275	128	243
Celegan1	21	123	237	131	194	186	266	138	245
	27	127	238	127	177	165	235	128	260
	32	94	177	136	127	124	116	179	168

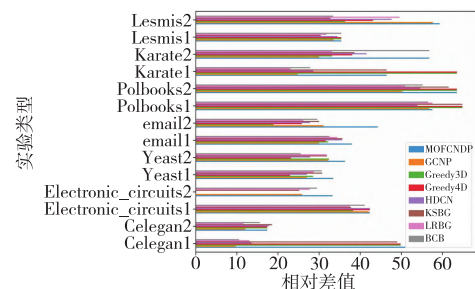


图 7 真实网络上的随机水平相对误差

Fig.7 Random horizontal relative error on real networks

表7 MFOCNP算法与其修改版本的结果比较(连通节点对)

Tab.7 Comparison results of MFOCNP algorithm and its modified version (connected node pairs)

name	z	MFOCNP			MFOCNP ₀			MFOCNP ₁		
		f_{best}	f_{avg}	t_{avg}	f_{best}	f_{avg}	t_{avg}	f_{best}	f_{avg}	t_{avg}
email	0.01	1 649	1 649	0	1 649	1 649	0	1 649	1 649	0
	0.05	1 497	1 497	1	1 544	1 642	0	1 603	1 638	0
	0.1	1 294	1 294	1	1 447	1 598	0	1 321	1 336	0
	0.15	861	861	13	917	1 001	12	921	930	11
	0.2	547	547	13	637	714	12	556	596	11
	0.25	349	349	0	382	451	0	351	368	0
	0.3	268	268	0	292	305	0	272	307	0
Polbooks	0.01	4 164	4 164	0	4 164	4 164	0	4388	4435	0
	0.05	3 497	3 497	0	3 619	4 659	0	3 573	3 638	0
	0.1	2 497	2 497	51	2 774	3 056	62	2 698	2 779	47
	0.15	1 971	1 971	0	2 065	2 144	0	2 001	2 256	0
	0.2	1 642	1 642	0	1 909	2 120	0	1 768	1 797	0
	0.25	843	843	0	997	1 104	0	868	877	0
	0.3	579	579	0	592	647	0	614	665	0
Karate	0.01	163	163	0	163	163	0	163	163	0
	0.05	127	127	0	127	127	0	127	127	0
	0.1	84	84	0	84	84	0	84	84	0
	0.15	82	82	0	82	82	0	82	82	0
	0.2	46	46	0	46	51	0	46	46	0
	0.25	27	27	0	29	31	0	27	27	0
	0.3	6	6	0	6	6	0	6	6	0
Lesmis	0.01	210	210	0	210	210	0	210	210	0
	0.05	150	150	0	150	150	0	150	150	0
	0.1	127	127	21	127	127	25	127	130	16
	0.15	116	116	0	116	116	0	116	116	0
	0.2	104	104	0	113	135	0	109	109	0
	0.25	64	64	28	82	82	33	64	70	22
	0.3	47	47	0	56	56	0	47	47	0

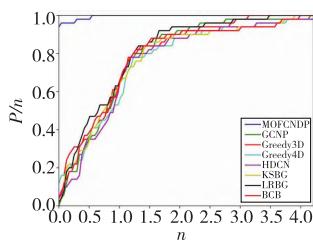


图8 性能概况指标

Fig.8 Performance overview indicators

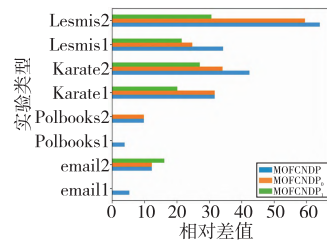


图9 随机水平相对误差

Fig.9 Random horizontal relative error

5 结束语

针对经典关键节点挖掘问题,提出一种基于飞蛾扑火优化的关键节点挖掘算法(MFOCNP)。该算法结合快速种群进化(FPE)机制来获得高质量的初始种群,采用飞蛾扑火优化算法(MFO)和复合高斯进化算法(HGE)来更新优化种群。为加快收敛速度,FPE机制采用相反学习机制(OBL)和快速种群演化策略来优化初始化。本文算法实现了利用飞蛾扑火优化算法和复合高斯进化算法探索更大的解空间,克服了局部最优陷阱。为了证明MFOCNP算法的性能,在各种合成网络和真实网络上进行了大量实验。计算结果表明,MFOCNP算法在关键节点挖掘问题上优于其他算法,并验证了算法各个组件的性能。如何将机器学习与关键节点挖掘相结合,从而解决动态的大规模的网络问题,是复杂网络中一个重大挑战,后期也将对这一问题进行深入探索。

参考文献:

- [1] Tomaino V, Arulselman A, Veltri P, *et al.* Studying connectivity properties in human protein-protein interaction network in cancer pathway [M]//Pardalos P, Xanthopoulos P, Zervakis M. Data Mining for Biomarker Discovery. Boston, MA: Springer, 2012: 187-197.
- [2] Dinh T N, Thai M T. Precise structural vulnerability assessment via mathematical programming [C]//Proc of Military Communications Conference. Piscataway, NJ: IEEE Press, 2011:1351-1356.

- [3] Borgatti S P. Identifying sets of key players in a network [C]//Proc of Managing Technologically Driven Organizations: The Human Side of Innovation and Change. Piscataway, NJ: IEEE Press, 2003: 127-131.
- [4] Ventresca M, Aleman D. A derandomized approximation algorithm for the critical node detection problem [J]. Computers and Operations Research, 2014, 43: 261-270.
- [5] Arulselman A, Commander C W, Eleftheriadou L, *et al.* Detecting critical nodes in sparse graphs [J]. Computers and Operations Research, 2009, 36(7): 2193-2200.
- [6] Hosteins P, Scatamacchia R. The stochastic critical node problem over trees [J]. Networks, 2020, 76(3): 381-401.
- [7] Summa M D, Grosso A, Locatelli M. Branch and cut algorithms for detecting critical nodes in undirected graphs [J]. Computational Optimization and Applications, 2012, 53: 649-680.
- [8] Freeman L C. A set of measures of centrality based on betweenness [J]. Sociometry, 1977, 40(1): 35-41.
- [9] Chen Duanbing, Lyu Linyuan, Shang Mingsheng, *et al.* Identifying influential nodes in complex networks [J]. Physica A: Statistical Mechanics and Its Applications, 2012, 391(4): 1777-1787.
- [10] Kitsak M, Gallos L K, Havlin S, *et al.* Identification of influential spreaders in complex networks [J]. Nature Physics, 2010, 6: 888-893.
- [11] Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 182-197.
- [12] Addis B, Aringhieri R, Grosso A, *et al.* Hybrid constructive heuristics for the critical node problem [J]. Annals of Operations Research, 2016, 238(1): 637-649.
- [13] 朱华, 潘侃, 王磊, 等. 基于DynamicRank的重要节点集挖掘算法 [J]. 重庆邮电大学学报: 自然科学版, 2022, 34(5): 869-876. (Zhu Hua, Pan Kan, Wang Lei, *et al.* Critical nodes mining algorithm based on DynamicRank [J]. Journal of Chongqing University of Posts and Telecommunications: Natural Science Edition, 2022, 34(5): 869-876.)
- [14] 周丽娜, 常笑, 胡枫. 利用邻接结构熵确定超网络关键节点 [J]. 计算机工程与应用, 2022, 58(8): 76-82. (Zhou Lina, Chang Xiao, Hu Feng. Using adjacent structure entropy to determine vital nodes of hypernetwork [J]. Computer Engineering and Applications, 2022, 58(8): 76-82.)
- [15] 孙百兵, 孙家政, 何泉, 等. 融入社区评估的节点重要性分析 [J]. 计算机工程与应用, 2023, 59(3): 226-233. (Sun Baibing, Sun Jiazheng, He Quan, *et al.* Node importance analysis integrated with community assessment [J]. Computer Engineering and Applications, 2023, 59(3): 226-233.)
- [16] 刘子彤, 王威, 丁国如, 等. 一种面向有权重通信网络的关键节点识别方法 [J]. 数据采集与处理, 2023, 38(1): 51-62. (Liu Zitong, Wang Wei, Ding Guoru, *et al.* A key node identification approach for weighted communication networks [J]. Journal of Data Acquisition and Processing, 2023, 38(1): 51-62.)
- [17] Ventresca M. Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem [J]. Computers & Operations Research, 2012, 39(11): 2763-2775.
- [18] Mirjalili S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm [J]. Knowledge-Based Systems, 2015, 89: 228-249.
- [19] Allam D, Younsi D A, Eteiba M B. Parameters extraction of the three diode model for the multi-crystalline solar cell/module using moth-flame optimization algorithm [J]. Energy Conversion and Management, 2016, 123: 535-548.
- [20] Zhao Huiru, Zhao Haoran, Guo Sen. Using GM(1,1) optimized by MFO with rolling mechanism to forecast the electricity consumption of Inner Mongolia [J]. Applied Sciences, 2016, 6(1): 20.

- emission dispatch[J]. *International Journal of Emerging Electric Power Systems*, 2014,15(2): 141-150.
- [13] Zhang Huifeng, Yue Dong, Xie Xiangpeng, *et al.* Multi-elite guide hybrid differential evolution with simulated annealing technique for dynamic economic emission dispatch[J]. *Applied Soft Computing*, 2015,34: 312-323.
- [14] Fakhim-Babaei A, Nazari-Heris M, Mohammadi-Ivatloo B, *et al.* Economic dispatch of large-scale integrated heat and power systems by application of a novel harmony search approach[J]. *Hybrid Energy System Models*, 2021,52(3): 25-49.
- [15] Yang Zhile, Li Kang, Niu Qun, *et al.* A self-learning TLBO based dynamic economic/environmental dispatch considering multiple plug-in electric vehicle loads[J]. *Journal of Modern Power Systems & Clean Energy*, 2014,2(4): 298-307.
- [16] Ma Haiping, Yang Zhile, You Pengcheng, *et al.* Multi-objective biogeography-based optimization for dynamic economic emission load dispatch considering plug-in electric vehicles charging[J]. *Energy*, 2017,135: 102-111.
- [17] Bahmanifirooz B, Farjah E, Niknam T. Multi-objective stochastic dynamic economic emission dispatch enhancement by fuzzy adaptive modified theta particle swarm optimization[J]. *Journal of Renewable & Sustainable Energy*, 2012,4(2): 013102-6.
- [18] Niknam T, Golestaneh F, Sadeghi M S. θ -multiobjective teaching learning-based optimization for dynamic economic emission dispatch[J]. *IEEE Systems Journal*, 2012,6(2): 341-352.
- [19] 武慧虹, 钱淑渠, 刘衍民, 等. 精英克隆局部搜索的多目标动态环境经济调度差分进化算法[J]. *山东大学学报: 工学版*, 2021,51(1):11-23. (Wu Huihong, Qian Shuqu, Liu Yanmin, *et al.* Multiobjective dynamic economic emission dispatch differential evolution algorithm based on elites cloning local search[J]. *Journal of Shandong University: Engineering Science*, 2021,51(1): 11-23.)
- [20] 武慧虹, 林好, 曾茜, 等. 自适应差分进化算法及对动态环境经济调度问题应用[J]. *计算机应用研究*, 2021,38(5):1443-1448,1454. (Wu Huihong, Lin Yu, Zeng Xi, *et al.* Adaptive differential evolutionary algorithm and its application of dynamic emission economic dispatch problem[J]. *Application Research of Computers*, 2021,38(5): 1443-1448,1454.)
- [21] Qian Shuqu, Wu Huihong, Xu Guofeng. An improved particle swarm optimization with clone selection principle for dynamic economic emission dispatch[J]. *Soft Computing*, 2020,24(20): 15249-15271.
- [22] Elaiw A M, Xia X, Shehata A M. Hybrid DE-SQP and hybrid PSO-SQP methods for solving dynamic economic emission dispatch problem with valve-point effects[J]. *Electric Power Systems Research*, 2013,103: 192-200.
- [23] Deb K. An efficient constraint handling method for genetic algorithms[J]. *Computer Methods in Applied Mechanics & Engineering*, 2000,186(24): 311-338.
- [24] De Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. *IEEE Trans on Evolutionary Computation*, 2002,6(3): 239-251.
- [25] Panigrahi B K, Yadav S R, Agrawal S, *et al.* A clonal algorithm to solve economic load dispatch[J]. *Electric Power Systems Research*, 2007,77(10): 1381-1389.
- [26] El-Sharkh M Y. Clonal selection algorithm for power generators maintenance scheduling[J]. *International Journal of Electrical Power & Energy Systems*, 2014,57(5): 73-78.
- [27] Aragon V S, Esquivel S C, Coello C C. An immune algorithm with power redistribution for solving economic dispatch problems[J]. *Information Sciences*, 2015,295: 609-632.
- [28] Deb K, Pratap A, Agarwal S, *et al.* A fast and elitist multiobjective genetic algorithm: NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2002,6(2): 182-197.
- [29] Zhu Yongsheng, Qiao Baihao, Dong Yan, *et al.* Multiobjective dynamic economic emission dispatch using evolutionary algorithm based on decomposition[J]. *IEEE Trans on Electrical and Electronic Engineering*, 2019,14(9): 1323-1333.
- [30] Kukkonen S, Deb K. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems[C]//Proc of IEEE International Conference on Evolutionary Computation. Piscataway, NJ: IEEE Press, 2006: 1179-1186.
- [31] Qian Shuqu, Ye Yongqiang, Jiang Bin, *et al.* Constrained multi-objective optimization algorithm based on immune system model[J]. *IEEE Trans on Cybernetics*, 2016,46(9): 2056-2069.
- [32] Mason K, Duggan J, Howley E. A multi-objective neural network trained with differential evolution for dynamic economic emission dispatch[J]. *International Journal of Electrical Power & Energy Systems*, 2018,100: 201-221.
- [33] Basu M. Particle swarm optimization based goal-attainment method for dynamic economic emission dispatch[J]. *Electric Power Components & Systems*, 2006,34(9): 1015-1025.
- [34] Wang Weiwei, Dai Shijuan, Zhao Wanzhong, *et al.* Multi-objective optimization of hexahedral pyramid crash box using MOEA/D-DAE algorithm[J]. *Applied Soft Computing*, 2022,118: 108481.
- [35] Zitzler E, Thiele L, Laumanns M, *et al.* Performance assessment of multiobjective optimizers: an analysis and review[J]. *IEEE Trans on Evolutionary Computation*, 2003,7(2): 117-132.
- [36] Basu M. Dynamic economic emission dispatch using evolutionary programming and fuzzy satisfying method[J]. *International Journal of Emerging Electric Power Systems*, 2007,8(4): 14-19.
- (上接第2719页)
- [21] Barczak T M, Oyler D C. A model of shield-strata interaction and its implications for active shield setting requirements, RI-9394[R/OL]. (1991-12-01). <https://www.osti.gov/biblio/5290205>.
- [22] Hassanien A E, Gaber T, Mokhtar U, *et al.* An improved moth flame optimization algorithm based on rough sets for tomato diseases detection[J]. *Computers and Electronics in Agriculture*, 2017,136: 86-96.
- [23] Tizhoosh H R. Opposition-based learning: a new scheme for machine intelligence[C]//Proc of International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce. Piscataway, NJ: IEEE Press, 2005: 695-701.
- [24] Zheng Wenping, Wu Zhikang, Yang Gui. A novel algorithm for identifying critical nodes in networks based on local centrality[J]. *Journal of Computer Research and Development*, 2019,56: 1872-1880.
- [25] Erdos P, Rényi A. On random graphs[J]. *Publications Mathematicae*, 1959, 6: 290-297.
- [26] Barabási A L, Albert R. Emergence of scaling in random networks[J]. *Science*, 1999,286(5439): 509-512.
- [27] Watts D J, Strogatz S H. Collective dynamics of 'small-world' networks[J]. *Nature*, 1998, 393: 440-442.
- [28] Dolan E D, Mor E J J. Benchmarking optimization software with performance profiles[J]. *Mathematical Programming*, 2002, 91: 201-213.
- [29] Guimera R, Danon L, Díaz-Guilera A, *et al.* Self-similar community structure in organisations[J]. *Physical Review E*, 2003, 68: 065103.
- [30] Zachary W W. An information flow model for conflict and fission in small groups[J]. *Journal of Anthropological Research*, 1977,33(4): 452-473.
- [31] Knuth D E. The Stanford GraphBase: a platform for combinatorial computing[M]. New York: ACM Press, 1993.