arXiv:2304.03998v1 [cs.NE] 8 Apr 2023

演化强化学习环境以最小化学习者可实现的奖励：在强化活动目录系统中的应用

预印本

Diksha Goel、Aneta Neumann、Frank Neumann、Hung Nguyen、Mingyu Guo

阿德莱德大学计算机与数学科学学院，澳大利亚 {diksha.goel, aneta.neumann, frank.neumann,hung.nguyen,mingyu.guo}@adelaide.edu.au

2023年4月11日

摘要

我们研究可配置环境中一名攻击者与一名防御者之间的斯塔克尔伯格博弈。防御者选择特定的环境配置。攻击者观察该配置，并通过针对所观察环境训练的强化学习（RL）发起攻击。防御者的目标是找到能使攻击者可实现奖励最低的环境。我们采用进化多样性优化（EDO）生成多样化的环境种群用于训练。可实现奖励明显较高的环境会被淘汰，并由新的后代取代，以避免浪费训练时间。多样性不仅能提高训练质量，也很适合我们的强化学习场景：强化学习智能体往往逐步提升，因此，早期稍逊一筹的环境之后可能会变得更好。我们通过聚焦于一个具体应用——活动目录（AD），展示了所提方法的有效性。活动目录是 Windows 域网络的默认安全管理系统。活动目录环境描述了一个攻击图，其中节点代表计算机、账户等，边代表访问权限。攻击者旨在找到最佳攻击路径，以抵达最高权限节点。防御者可以通过移除数量有限的边（撤销访问权限）来改变该图。与现有方法相比，我们的方法能够生成更好的防御方案，且可扩展性更强。

关键词 ***rds*** 活动目录、强化学习、进化多样性优化、攻击图

1 引言

在对抗环境中，攻击者和防御者相互对抗。攻击者旨在设计一种能够成功实施攻击的技术，而防御者的目标是保护系统免遭入侵。在对抗环境中，攻击者和防御者的策略与动作相互依赖，并彼此影响。本文研究可配置环境设置中的攻击者-防御者斯塔克尔伯格博弈，其中防御者（领导者）尝试各种环境配置来保护系统。相比之下，攻击者（跟随者）观察环境配置，并使用强化学习（强化学习）设计攻击策略，以最大化其奖励。防御者旨在找到一种环境配置，使攻击者可获得的奖励最小。我们考虑一个具体的应用场景“强化活动目录系统”，以详细讨论该问题。

活动目录是微软为*Windows* 域网络开发的目录服务，旨在管理和保护网络资源，例如用户账户、计算机、打印机等。活动目录被视为 Windows 域网络的默认安全管理系统 [1]，由于被大多数大型公司广泛采用，因此已成为网络攻击者的热门目标。微软报告称，《财富》1000强企业中有 90%

使用活动目录。根据企业管理协会的一项研究，受访组织中有 [2021], 50% 自 2019 年以来遭受过针对活动目录的攻击。

活动目录攻击图是工业从业者和现实世界攻击者广泛使用的攻击图模型，用于对活动目录环境中的潜在攻击场景进行建模和分析。活动目录的结构描述了攻击图，其中节点代表账户/计算机等，有向边（i，j）表示攻击者可以通过已知漏洞或现有访问权限，从节点 i 获取对节点 j 的访问权限。BLOODHOUND 是一种广泛使用的工具，可用于调查活动目录图，并识别活动目录图结构中的各种攻击路径。BLOODHOUND 模拟身份雪球攻击，攻击者从低权限账户（通过网络钓鱼攻击获得访问权限）开始，随后转向其他节点，目标是到达权限最高的账户——域管理员（DA）。BLOODHOUND 使用戴克斯特拉最短路径算法来确定从入口节点到域管理员的路径。鉴于组织活动目录的敏感性质，以及针对活动目录的大量网络攻击，安全专业人员正在设计各种解决方案来保护活动目录。BLOODHOUND 的设计源于一篇学术论文 [2], , 论文作者设计了一种启发式方法，有选择地阻断攻击图中的部分边，以断开图并阻止攻击者到达域管理员。值得注意的是，在活动目录环境中，边阻断通过撤销访问权限或实施监控来实现，以防止攻击者到达域管理员。

本文研究了活动目录图上的斯塔克尔伯格博弈，其中一名攻击者与一名防御者对抗。攻击者旨在制定策略，最大限度地提高成功到达域管理员的概率；防御者则旨在制定策略，尽可能降低攻击者的成功率。 我们模型中的攻击者具有策略性，在实施攻击时采取主动方式。活动目录图中的每条边都关联着一个失败率、检测率和成功率。攻击者从某个入口节点开始攻击，并尝试遍历一条边以到达新节点。如果攻击者未能遍历该边，就会尝试另一条边，直到被检测到、尝试完所有可能路径，或到达域管理员。如果攻击者此前未能通过某条边，就不会再次尝试该边。具有策略性的攻击者维护一个称为检查点的集合，其中包含攻击者可以用作起点或继续攻击的节点。初始时，检查点仅包含入口节点。攻击者会尝试各种策略以到达域管理员，并保存相关信息，例如未能通过和成功通过的边。成功通过的边的终点会被加入检查点；在攻击过程中，攻击者可以随时尝试从检查点中的节点出发、尚未探索过的边。最优攻击策略是在不被检测到的前提下，最大限度地提高攻击者到达域管理员的概率。值得注意的是，设计复杂而精确的攻击者策略至关重要，因为只有准确掌握攻击者策略，才能制定有效的防御。此外，防御者的目标是制定防御策略，以尽可能降低攻击者的成功率。 防御者通过将边的失败率从原始值提高到 100%，来阻断一组 k 边。在已有研究中，Goel 等人 [3] 研究了与我们相同的活动目录模型。他们训练神经网络来解决攻击者问题，并设计了基于 EDO 的策略来解决防御者问题。与之不同的是，我们旨在提出一种可扩展性更强、能更准确地近似攻击者策略并生成优于 [3]的防御方案的方法。

我们的贡献。 本文旨在设计防御者策略，以阻断一组边，目标是最小化攻击者成功到达域管理员的成功概率。 针对攻击者问题，我们提出一种基于强化学习（***RL***） 的策略，以最大化攻击者成功到达域管理员的成功概率（最大化攻击者的可实现奖励）。我们提出一种基于评论家网络辅助进化多样性优化（***C-EDO***） 的防御策略，以寻找能够最小化攻击者成功率的防御计划配置。攻击者最大化其成功到达域管理员的成功概率这一问题可以建模为马尔可夫决策过程（MDP）。我们提出基于强化学习的策略来近似攻击者问题，并使用近端策略优化 强化学习算法——一种基于演员-评论家的方法 ——来训练攻击者策略。强化学习智能体通过建议动作，同时与“多个环境”交互，目标是最大化总体奖励。我们提出一种基于评论家网络辅助进化多样性优化的策略来解决防御者问题。C-EDO 生成大量环境配置（防御方案）。我们的方法使用经过训练的强化学习评论家网络来估计环境配置的适应度。防御者持续监控强化学习训练过程，并在固定时间间隔后，使用经过训练的评论家网络评估当前配置，并使用 C-EDO 生成更优配置。攻击者和防御者并行地相互对抗。

总体而言，防御者的 C-EDO 会生成大量高质量、多样化的环境配置，值得用于强化学习智能体的学习，以训练出更好的攻击策略。我们使用基于演员-评论家的算法训练攻击者策略，因此天然拥有一个评论家网络，可以合理估计状态值，并可用作防御者 C-EDO 的适应度函数。这样一来，这两个问题相互关联，解决策略也相辅相成。我们进行了广泛的实验分析，并将结果与现有方法进行了比较[3]。结果表明：1）所提出的方法具有可扩展性，可应用于 r4000[1] AD 图，而现有方法无法扩展至此规模；2）所提出的方法效果显著。对于 r1000 AD 图，所提出方法中的最佳防御，其成功概率平均为 41.01%，显著低于其他方法；

[1] r4000 AD图是一个包含4000台计算机的AD图 ters。

3）基于 ProposedRL 的策略能更准确地近似攻击者的问题。对于 r1000 AD 图，使用我们的攻击策略后，攻击者的平均成功率提高了 0.63%；4）所提出的方法能够生成更好的防御。对于 r1000 AD 图，在我们提出的防御下，攻击者的平均成功率降低了 1.76%。

由于我们使用强化学习来近似攻击者的问题，因此我们提出的方法取得了比 [3] 更好的结果。NNDP 攻击者策略 [3] 一次针对一个防御计划训练模型，因此会遗忘之前的计划。这样，它不断学习并遗忘这些计划。然而，我们的方法让基于强化学习的攻击者策略一次针对多个防御计划进行训练，因此能够学习共享经验，并取得更好的性能。对于强化学习智能体而言，多样化的环境配置仅在“开局阶段”有所不同，而不同环境中的“残局阶段”或“中局阶段”可能是相似的。在并行训练中可以利用后续阶段的相似性，此时智能体会同时针对多个环境进行训练并获得共享经验，从而实现更快的收敛并提升性能。此外，NNDP 方法是基于值迭代的强化学习算法，而我们的方法是基于策略迭代的强化学习算法。通常，基于策略迭代的算法比基于值迭代的算法收敛更快 [4], ，这也是我们的方法性能更优的另一个原因。我们的实验结果进一步支持了这一观点。

2 相关工作

活动目录。Goel 等人 [3] 研究了与我们相同的攻击者和防御者之间在活动目录图上的斯塔克尔伯格博弈模型。作者使用神经网络解决攻击者问题，并使用EDO 解决防御者问题。然而，他们的方法无法扩展到大型活动目录图。我们提出的方法能够扩展到大型活动目录图，并且可以更有效地近似攻击者问题。Guo等人 [5] 研究了一种不同的活动目录图防御模型，在该模型中，防御者旨在最大化攻击者的期望路径长度。作者提出了固定参数算法和基于图卷积神经网络的方法，以发现需要被阻断的边。Guo 等人 [6] 提出了用于防御AD攻击图的多种可扩展算法。然而，他们考虑的问题设置与我们不同。作者提出了基于树分解的动态规划和基于强化学习的方法，这些方法能够扩展到大型活动目录图。Ngo等人 [7] 设计了用于在活动目录图动态变化时将蜜罐部署到计算机节点上的近似最优策略；然而，作者研究的是识别需要阻断的节点，而我们关注的是阻断边。Zhang等人 [8] 提出了一种可扩展的双重预言机算法，并将其解决方案与各种行业解决方案进行了比较；作者考虑的问题设置与我们不同。

进化多样性优化。 进化多样性优化旨在发现高质量且差异最大的解。Ulrich 等人 [9] 首次研究了 EDO，此后它在进化计算领域受到了广泛关注。Huang 等人 [10] 设计了一种协同进化方法，以提升竞争性群优化器的搜索能力和收敛性能。Bossek 等人 [11] 研究了 EDO 在计算最小生成树问题的一组多样化解方面的应用。Do 等人 [12]研究了 EDO 在排列问题中的应用。Nikfarjam 等人 [13] 设计了一种基于熵的进化方法，用于计算旅行商问题的多样化解。

3 问题描述

活动目录图可以表示为有向图 $G = (V, E)$，其中 V 是节点集， E 是边集。活动目录中的最高权限账户称为域管理员（DA）。本文研究一名防御者与一名攻击者之间的双人斯塔克尔伯格博弈，以防御活动目录图。共有 s 个入口节点。攻击者可以从其中一个入口节点出发，并旨在制定策略，最大化成功到达 DA 的成功概率。防御者则试图阻断一组边，以尽量降低攻击者的成功率。防御者的预算有限，只能阻断 k 条边。并非所有边都可阻断；因此，攻击者只能阻断“可阻断”边。边阻断需要付出成本和精力（审计访问日志）才能安全地移除一条边；因此，我们无法移除过多边，并假设预算有限。在我们的模型中，活动目录图中的每条边都有检测率、失败率和成功率。检测率 $p_{d(e)}$ 表示如果攻击者被发现，攻击会立即终止。失败率$p_{f(e)}$ 表示如果攻击者失败，就无法成功通过边 e（攻击者可能需要输入正确密码才能通过该边）。在这种情况下，攻击不会终止，攻击者可以尝试尚未探索的边，从其他检查点继续攻击。成功率$p_{s(e)}$ 表示攻击者成功通过边 e 的可能性，计算方式为 $p_{s(e)} = 1 - p_{f(e)} - p_{d(e)}$。策略型攻击者从一个入口节点出发，尝试尚未探索的边，以到达 DA。在攻击过程中的任意时刻，检查点都表示攻击者控制的节点集合；如果攻击者未能通过某条边，便可从这些节点继续攻击。Goel 等人 [3] 证明，计算防御者和攻击者的

最优策略（及其价值）是 #P-困难。因此，为近似求解攻击者问题，我们设计了一种基于强化学习的策略，使智能体能够同时从多个环境配置（防御方案）中学习，从而加快训练过程。我们提出一种评论家网络辅助进化多样性优化策略，以解决防御者问题。

4 提出的方法

本节介绍我们提出的用于防御活动目录图的方法。我们首先讨论所提出的预处理过程，该过程将原始 AD图转换为较小的图。然后，我们讨论基于强化学习的攻击者策略，以及基于评论家网络辅助进化多样性优化的防御者策略。最后，我们介绍整体的攻击者—防御者方法。

4.1 AD图预处理

我们首先利用 AD图的结构特征对其进行预处理，以获得较小的图。在 AD图中，分裂节点是指具有多条发出边的节点。入口节点是攻击者可以从中发起攻击的起始点。分裂节点集合和入口节点集合分别记为 SPLIT 和 ENTRY。

定义 1. 非分裂路径（***NSP***）。非分裂路径 *NSP* (i, j) 是从节点 i 到 j的一条路径，其中 j 是节点 i唯一的后继节点；随后迭代地移至 j的唯一后继节点，直至遇到 *DA* 或分裂节点 *[6]*。

$$NSP = \{NSP(i,j) | \, i \in \textsc{Split} \cup \textsc{Entry}, j \in \textsc{Successors}(i)\}$$

只有当NSP 上至少有一条边是可阻断的时，我们才称该 NSP 是可阻断的。

定义 2。值得阻断的边 ***(bw)***。在 NSP上距离节点 i 最远的任何可阻断边都称为值得阻断的边 $bw(i, j)$。值得阻断的边集合定义如下：

$$BW = \{bw(i,j) | \, i \in \textsc{Split} \cup \textsc{Entry}, j \in \textsc{Successors}(i)\}$$

值得阻断的边可由两个 NSP 共享。我们仅花费了一个预算单位来阻断 $NSP(i, j)$。如果原始图包含 n 个节点和 m 条边；经过预处理后，得到的图由 $(|\textsc{Entry}| + |\textsc{Split}| + 1)$ 个节点和 $|NSP|$ 条边组成。

4.2 攻击者策略：强化学习

攻击者的目标是设计一种策略，最大限度地提高其成功到达域管理员的成功概率。我们将攻击者的问题描述为一个马尔可夫决策过程，并提出一种基于强化学习的策略来解决攻击者的问题。我们提出的基于强化学习的攻击者策略使用 近端策略优化 ***(PPO)*** 算法，这是一种基于演员-评论家的方法，用于训练强化学习智能体。我们同时针对多个环境配置实例并行训练我们的强化学习智能体，并且每个环境都包含一个来自防御者的防御计划。本节将详细讨论我们提出的基于强化学习的攻击者策略。

4.2.1 环境

我们将攻击者设计策略以最大化成功到达域管理员的概率这一问题建模为马尔可夫决策过程。我们将攻击者的马尔可夫决策过程 $M = (S,\ A,\ R,\ T)$ 称为环境，其中 S 表示状态空间，A 表示动作空间，R 是奖励函数，T 表示转移函数。下文将讨论该环境的描述。

状态空间（***S***）：状态空间 S 是一个有限的攻击者状态集合，状态“s”是一个向量，其长度等于 AD图中的 NSP 数量。状态 s 中的每个坐标代表一个 NSP，且每个 NSP 的状态为“S”、“F”或“?”。攻击者状态 s表示为：

$$\text{Attacker state s} = \underbrace{< S, F, ?, \ldots\ldots, F, ?, S, ? >}_{\text{Length = Number of NSPs}} \tag{1}$$

其中，状态“S”表示攻击者尝试后成功到达 NSP 的另一端，“F”表示攻击者虽已尝试，但未能到达另一端，而“?”表示攻击者尚未尝试该 NSP。

给定一个状态 s，攻击者会探索一个状态为“?”的 NSP；尝试后，该 NSP 的状态会变为“S”或“F” [2]，攻击者也会到达一个新状态。攻击者每次继续探索一个尚未探索的 NSP，从而到达新状态，直到攻击者到达域管理员或被检测到。在攻击过程中的任意时刻 t，攻击者的当前状态 s_t 都充当知识库，并传达以下信息：攻击者已控制的 NSP 集合（状态为“S”的 NSP）、攻击者尝试但未成功的 NSP（状态为“F”的 NSP），以及攻击者将来可以尝试的 NSP（状态为“?”的 NSP）。攻击者有两种终止状态：1）攻击者到达域管理员时，攻击成功并终止；2）攻击者无法到达域管理员时，可能是因为被检测到，或已尝试所有可能的 NSP，且没有其他选项可供探索；在这种情况下，攻击失败并结束。

动作空间 ***(A)***：状态 s的动作空间 A 是该状态可用的动作集，即状态 s中成功的 NSP 所发出的 NSP。攻击者的动作空间是离散的。动作 a 是状态 s 的动作空间中的一个 NSP，表示攻击者尝试通过该 NSP 到达域管理员。

奖励函数 ***(R)***：奖励 $r(s,\ a)$ 对于状态 s 执行动作 a，如果攻击者能够在未被检测到的情况下到达域管理员，则为 1。否则，奖励为 0。

转移函数***(T)***：对于给定的状态 s 和动作 a，转移函数对状态 s执行动作 a，并可能产生一组未来状态。每个未来状态都与其转移概率相关联，并选择其中一个状态作为下一个状态（按其转移概率加权）。

4.2.2 策略训练

我们提出利用 近端策略优化（*PPO*）强化学习算法训练攻击者策略，以最大化攻击者的成功率。PPO 遵循一种 演员-评论家方法，该方法利用基于策略和基于价值的方法的优势，同时消除其缺点。在此方法中，策略网络和价值网络相互促进、共同改进。在我们的方法中，我们训练两个网络：演员网络和评论家网络。演员网络也称为策略网络，以当前状态 s 作为输入，并输出将在 s 上执行的动作 a。演员网络根据评论家网络所指示的方向更新策略参数。评论家网络也称为价值网络，以状态作为输入，并输出状态的价值。对于攻击者问题，状态的价值就是攻击者的成功率。

我们的强化学习智能体使用基于演员-评论家的 PPO 算法，通过与环境交互来训练攻击者策略（每个环境都与一种配置，即防御计划相关联）。强化学习智能体不具备任何关于环境的先验知识。我们不是让强化学习智能体针对单一环境进行训练，而是让智能体针对多个并行环境进行训练。每个环境都以攻击者的初始状态初始化（将防御计划转换为攻击者的初始状态）。以下过程在所有环境中同时执行。在一个回合的每个时间步 t，强化学习智能体都会接收状态 s_t。经过训练的演员网络会针对当前状态 s_t 发出要执行的动作 a_t，并将动作 a_t 发送至环境。环境在状态 s_t 下执行动作 a_t，并到达新状态 s_{t+1}（遵循转移函数），同时获得奖励 r_{t+1}（遵循奖励函数）。该过程会持续进行，直到到达基准状态，即攻击者成为域管理员或被检测到。这样，我们就获得了一个以基准状态为终点的状态、动作和奖励序列。所设计的策略旨在最大化一个回合内获得的总奖励。最终奖励为攻击者的成功率。通过这种方式，攻击者设计出基于强化学习的策略，以最大化其可实现奖励（成功率）。

4.3 防御者策略：评论家网络辅助的进化多样性优化

防御者的目标是阻断 k 值得阻断的边，以降低攻击者成功抵达 k 域管理员的成功概率。我们提出一种基于评论家网络辅助进化多样性优化（***C-EDO***）的防御策略，用于计算高质量且多样的环境配置（防御计划）。我们的目标是识别有价值的环境，即对应于潜在有效防御的环境配置。我们的核心思路是让强化学习智能体与环境配置对抗；如果经过一段时间的训练后，该配置被证明不利于防御者（即攻击者针对该配置的成功率很高），我们就舍弃该环境配置。我们不会在这种环境上浪费计算资源投入，而是将有限的计算资源分配给其他更具挑战性的环境。环境的高质量和多样性有助于提高攻击者建模的准确性。训练后的强化学习评论家网络作为防御者的 C-EDO 适应度函数。对于每个环境配置，适应度函数都用于评估该环境，即计算攻击者针对该配置（防御计划）的成功概率。防御者只阻断值得阻断的边来生成环境配置。防御者的环境配置/防御计划可表示为：

$$\text{Environment configuration } = < N, B, \ldots, B, N, B > \tag{2}$$

[2]如果已到达目的地，或联合值得阻断的边失效，其他一些 NSP 的状态也可能发生变化。

Alg算法 1基于评论家网络辅助进化多样性优化的防御者策略

1：使用 μ环境配置初始化种群 P 2：从 P中均匀随机选择个体 p'，并通过变异或交叉生成后代 p'_{new} 3：如果 $(OPT-0.1) \leq fitness(p'_{new}) \leq (OPT+0.1)$，则将 p'_{new}添加到 P4：如果 $|P| = \mu + 1$，则从 P 中移除对多样性贡献最小的个体 r，即使 $SortedDiver(C(bw)\backslash p_r)$ 最小的个体 5：重复步骤 2 至 4，直至满足终止条件

其中，“B”表示已阻断边，“N”表示未阻断边。值得注意的是，配置（防御计划）的长度等于 AD图中值得阻断的边的数量。

算法 1 概述了防御者的策略。该过程首先按照公式 (2) 生成任意种群 P 个 μ 配置。在每个配置中，Bs 的总数为 k（防御者预算）。为了创建新的后代（环境配置），我们对从种群 P 中随机选取的配置 p' 执行变异或交叉操作，两种操作各有 0.5 的概率。我们从均值为 1 的泊松分布中随机选择一个变量 x。对于变异，我们从 P 中随机选择一个个体 p'，并将 $x\,Bs$ 翻转为 Ns，将 $x\,Ns$ 翻转为 Bs。对于交叉，我们从 P 中随机选择两个个体 p' 和 p''，并寻找满足以下条件的 x 个索引：在这些索引处，p' 为 N，且 p'' 为 B。接着，对于这些索引，我们将 p' 中的 Ns 改为 Bs，并将 p'' 中的 Bs 改为 Ns。类似地，我们寻找满足以下条件的 x 个索引：在这些索引处，p' 为 B，且 p'' 为 N；然后将 p' 中的 Bs 改为 Ns，并将 p'' 中的 Ns 改为 Bs。变异和交叉操作确保环境配置中恰好有 k 条边被阻断。只有当新创建的后代的适应度分数处于 $(OPT \pm 0.1)$ 范围内时，我们才会将其加入种群；否则，即使该后代有助于提升多样性，也会被拒绝。如果新后代被加入种群，我们的目标是最大化已阻断边的多样性，并移除对多样性贡献最小的个体。

最大化被阻断边的多样性。 我们将多样性度量定义为“所有值得阻断的边都被同等程度地阻断”。我们提出的多样性度量旨在最大化环境配置中被阻断边的多样性。它会计算每条值得阻断的边在配置种群中被阻断的频率，目标是使这些频率相等。通过变异或交叉生成新的后代时，只有当其适应度分数接近最佳适应度分数时，才会将其加入种群，并淘汰对多样性贡献最小的个体。假设 P 中有 μ 个配置（我们将这些配置称为个体）。每个个体 p_i 可描述为：

$$p_i = \big((B/N, bw_1), (B/N, bw_2), ..., (B/N, bw_{|BW|})\big)$$

其中，B/N 表示值得阻断的边的状态；“B”表示已阻断，“N”表示未阻断，且 $i \in \{1, ..., \mu\}$。然后，我们计算值得阻断的边计数向量 $C(bw)$，其定义为“对于每条值得阻断的边bw_j，$j \in \{1, ..., |BW|\}$，已阻断该 bw_j边的个体数量”。

$$C(bw) = (c(bw_1), c(bw_2), ..., c(bw_{|BW|}))$$

其中，$c(bw_1)$表示在 μ 个个体中阻断了 bw_1 条边的个体总数。对于每个个体 p_i，我们随后确定向量 $Diver(C(bw)\backslash p_i)$，该向量计算移除个体 p_i 后的种群多样性，如下所示：

$$Diver(C(bw)\backslash p_i) = C(bw) - p_i$$

我们的目标是最大化被阻断边的多样性；因此，我们计算 $SortedDiver(C(bw)\backslash p_i)$如下：

$$SortedDiver(C(bw)\backslash p_i) = sort\Big(Diver(C(bw)\backslash p_i)\Big)$$

为了最大化被阻断边的多样性，我们的目标是按字典序最小化 $SortedDiver(C(bw)\backslash p_i)$，其中排序按降序进行。具有最小 $SortedDiver(C(bw)\backslash p_l)$的个体 l，就是移除后能够最大化多样性的个体。因此，如果移除个体 l 能够最大化多样性，且其适应度分数与最优值相差不大，则将其从种群中移除。特殊情况下，如果新生成的后代具有最佳适应度分数，我们会将其加入 P（尽管其多样性最差），并淘汰适应度分数最低的个体。通过这一流程，防御者能够创建多样化的环境配置。图1展示了种群中最大化被阻断边多样性的示例。

4.4 攻击者-防御者总体方法

防御者使用 C-EDO 生成高质量且多样化的环境配置，每种配置都包含值得攻击者策略学习的防御计划。攻击者使用基于行动者-评论家的强化学习算法来训练其

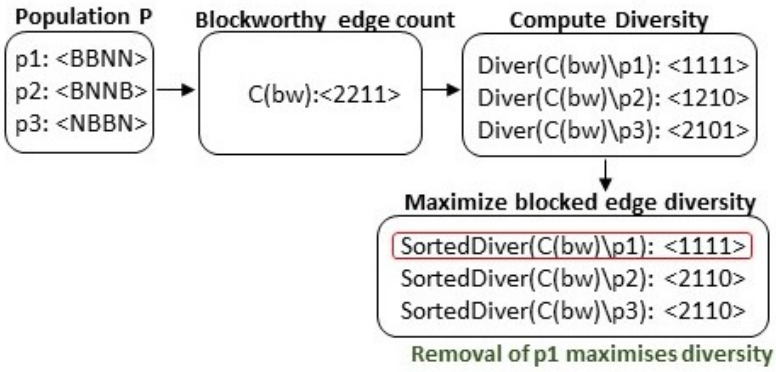

图 1：最大化已阻断边的多样性的示例。

策略，以应对防御者的环境配置。攻击者训练得到的强化学习评论家网络作为防御者 C-EDO 的适应度函数。攻击者首先将环境配置中的防御计划（式（2））转换为攻击者的初始状态[3]（式（1））。随后，强化学习智能体根据训练好的策略执行动作，并行地与各个环境交互和学习。环境执行这些动作，并返回新的状态和奖励。策略的训练质量取决于智能体在一个回合中获得的总奖励。起初，策略尚未经过训练，所建议的动作可能只能获得较低奖励；随着训练进行，奖励（即攻击者的成功率）会提高。通过这种方式，强化学习智能体训练策略，以最大化奖励。

防御者进程持续监控强化学习训练过程，并在每个固定时间间隔后，使用训练好的评论家网络评估当前的环境集合。防御者丢弃那些有利于攻击者的环境，并将其替换为更优的环境。这样，在每个回合开始时，强化学习智能体都会检查与每个环境对应的防御计划配置是否发生变化。如果发生变化，攻击者便使用新的防御配置初始化环境，并开始针对该环境训练智能体。通过这种方式，攻击者和防御者并行对抗：攻击者的强化学习进程持续学习，而防御者进程则评估当前的环境配置并生成更优配置。C-EDO 的多样性特征有助于强化学习避免陷入局部最优。强化学习智能体仅在早期阶段经历环境配置之间的差异，而在后期阶段，各环境中的配置趋于相似。我们利用了这种相似性，在多个环境上并行训练智能体，因此智能体获得了共享经验，从而实现更快的收敛和更高的性能。总体而言，随着防御者利用 C-EDO 生成更优的环境配置，训练后的攻击者强化学习策略也不断改进。随着攻击者的评论家网络得到训练，防御者策略会生成更优的环境。通过这种方式，这两个进程相互促进，共同提升性能。图 2 展示了我们提出的整体方法。

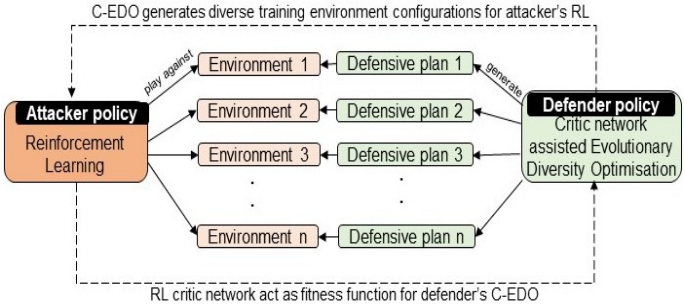

图 2：提出的攻击者—防御者整体方法。

[3]在攻击者的状态中，将防御者环境配置里对应于已阻断的值得阻断的边的 NSP 状态从“？”更改为“F”，以获得攻击者的初始状态。

5 实验结果

我们在一台配备 Intel Gold 6148/6248 处理器的高性能集群服务器上进行了实验，每次试验使用 1 个中央处理器和 20 个核心。我们使用 OpenAI Gym [14] 实现强化学习环境，并使用 TIANSHOU 库 [15]中的近端策略优化算法训练强化学习模型。成功率/成功概率表示在防御者已阻断某些边的情况下，攻击者在未被检测到的情况下到达域管理员的概率。

5.1 数据集

真实世界的活动目录图高度敏感；因此，我们使用 BLOODHOUND 团队的合成图生成器 DBCREATOR 来生成合成活动目录图。我们生成了四种规模的活动目录图，即 r500、r1000、r2000 和 r4000，其中 500、1000、2000 和 4000 表示活动目录图中的计算机数量。r500 AD 图包含 1493 个节点和 3456 条边；r1000 AD 图包含 2996 个节点和 8814 条边；r2000 AD 图包含 5997 个节点和 18795 条边；r4000 AD 图包含 12001 个节点和 45780 条边。我们的实验仅考虑 BLOODHOUND 中默认存在的 3 种特定边：HASSESSION、MEMBEROF 和 ADMINTO。为设置攻击者的起始节点，我们首先找到距离域管理员较远的 40 个节点，然后从中任意选取 20 个节点作为起始节点。每条边 e 可阻断的概率为 = e 与域管理员之间的最小跳数 任意 e 与域管理员之间的最大跳数。这样，距离域管理员最远的边（即不太重要的边）更有可能被阻断。在大型活动目录图上执行计算颇具挑战；因此，我们对其进行预处理，以得到一个更小的图。预处理包括将所有域管理员合并为 1 个域管理员，以及移除与攻击者无关的节点和边（域管理员的出边、所有指向入口节点的入边，以及没有任何入边的节点）。我们还将一个 NSP（定义 1）视为一条边。此外，为了研究边 e 的失败率 $p_{f(e)}$ 和检测率 $p_{d(e)}$ 与攻击者成功概率之间的关系，我们采用了三种不同的分布：独立分布、正相关和负相关。在独立分布中，我们将$p_{d(e)}$ 和 $p_{f(e)}$ 的值设为：

$$p_{d(e)}, p_{f(e)} = \text{Independent uniform (0, 0.2)}$$

在正相关中，我们将$p_{d(e)}$ 和 $p_{f(e)}$ 的值设为：

$$p_{d(e)}, p_{f(e)} = \mathcal{N}(\mu, \Sigma)$$

其中

$$\mu = [0.1, 0.1] \text{ and } \Sigma = [[0.05^2, 0.5 \times 0.05^2], [0.5 \times 0.05^2, 0.05^2]]$$

这里，$\mathcal{N}$ 表示多元正态分布，μ 表示均值，Σ 表示协方差矩阵。在负相关中，我们将$p_{d(e)}$ 和$p_{f(e)}$ 的值设为：

$$p_{d(e)}, p_{f(e)} = \mathcal{N}(\mu, \Sigma)$$

其中

$$\mu = [0.1, 0.1] \text{ and } \Sigma = [[0.05^2, -0.5 \times 0.05^2], [-0.5 \times 0.05^2, 0.05^2]]$$

5.2 训练参数

强化学习： 我们使用一个简单的多层感知机神经网络来实现演员网络和评论家网络。对于 r500 和 r1000 活动目录图，隐藏层大小为 128；对于 r2000 和 r4000活动目录图，隐藏层大小为 256。使用Adam优化器、学习率 $1e^{-4}$ 和批量大小为800 个状态来训练参数。对于 PPO 专用超参数，我们使用原始论文 [16]中指定的标准超参数。我们创建了 20 个环境。对于实验设置 1，我们在 20 个环境上并行训练强化学习策略，总计训练 700轮（对于r2000和r4000活动目录图则为 1200 轮）。每训练 20轮后，防御者都会评估并重置环境。当满足终止条件（训练轮次数）时，防御者选择攻击者成功率最低的防御计划，作为其最佳环境配置。对于实验设置2和 3，我们在 20个环境上并行训练强化学习策略 150轮。随后，在最佳环境配置上模拟训练好的攻击者策略5000 个回合，5000 个回合的平均奖励即为攻击者的成功率。

评论家网络辅助的 进化多样性优化：防御者最多只能阻断 5 条边。在 20000 次迭代中 s，防御者创建一个 20 种环境配置（防御）的评估。

5.3 实验设置1

在此实验设置中，我们评估整体所提方法的有效性。 .

表1：不同攻击者-防御者策略下攻击者成功概率的比较（数字越小表示性能越好）。结果表明，所提出的C-EDO 防御策略带来了最佳防御。此外，我们的强化学习+C-EDO策略可扩展到 r4000 个图，而 NNDP-EDO [3] 未能扩展到此规模。

图策略		成功概率 独立正向、负向、平均				时间（小时） 独立正 负		
r1000	强化学习+C-EDO（所提方法）	**40.16%**	**41.36%**	**41.51%**	**41.01%**	43.82	44.73	47.27
	强化学习+进化计算	41.69%	48.45%	42.89%	44.34%	44.43	43.88	48.43
	强化学习+贪心	56.45%	47.86%	47.88%	50.73%	43.32	44.95	47.38
r2000	强化学习+C-EDO（所提出的）	**25.09%**	**32.45%**	**30.44%**	**29.33%**	112.74	118.58	119.53
	强化学习+进化计算	28.13%	35.51%	37.28%	33.64%	114.22	112.57	118.51
	强化学习+贪心	33.43%	34.30%	40.06%	35.93%	113.84	116.43	113.76
r4000	强化学习+C-EDO（本文提出）	**22.02%**	**17.29%**	**21.81%**	**20.37%**	127.54	121.11	137.73
	强化学习+进化计算	22.78%	21.87%	24.71%	23.12%	132.55	120.67	135.31
	强化学习+贪心	25.16%	24.18%	22.34%	23.89%	125.61	120.73	127.29

5.3.1 基线

我们将基于强化学习的攻击者策略与各种防御者策略相结合，以比较所提出的防御策略的有效性。

• 强化学习+***C-EDO***（所提出的方法）：在此方法中，强化学习被用作攻击者策略，而 C-EDO 被用作防御者策略。在 C-EDO 中，防御者会舍弃对多样性贡献最小的环境配置。
• 强化学习+进化计算：在强化学习+进化计算方法中，强化学习被用作攻击者策略，而进化计算（进化计算）被用作防御者策略。在进化计算中，适应度分数最低的配置会被舍弃。
• 强化学习+贪心：在强化学习+贪心方法中，强化学习被用作攻击者策略。防御者使用贪心技术生成环境配置。防御者通过贪心地阻断一条能最大限度降低攻击者成功率的边。通过这种方式，防御者迭代地发现 k 需要阻断的边。

5.3.2 结果

我们在 $r1000$、$r2000$ 和 $r4000$ 活动目录图上进行实验。首先，我们在由防御者策略（即 C-EDO、进化计算和贪心）生成的环境配置（防御计划）上训练基于强化学习的攻击者策略。随后，我们针对防御者的最佳环境配置测试攻击者策略的有效性。我们通过在最佳环境中模拟攻击者策略运行 5000 个回合，报告平均奖励（成功率）。对于每个活动目录图，我们使用从 0 到 4 的 5 个随机种子进行实验，并报告这 5 个随机种子上的平均成功率。表 1报告了 r1000、r2000 和 r4000 活动目录图的实验结果。对于 r1000 活动目录图，表中结果显示，在独立分布下，采用基于 C-EDO 的防御策略时，攻击者的成功概率为 40.16%。相比之下，在基于进化计算的防御和贪心防御下，攻击者的成功概率分别增加到 41.69% 和 56.45%。对于 r2000 活动目录图，我们的结果显示，在独立分布下，当防御者采用基于 C-EDO 的策略时，攻击者的成功概率最低，即为 25.09%；在基于进化计算的防御和贪心防御下，成功率分别增加到 28.13% 和 33.43%。我们提出的方法可扩展至 r4000 活动目录图。r4000 活动目录图的结果显示，在正相关条件下，采用基于 C-EDO 的防御时，攻击者成功率最低，即为 17.29%；然而，在基于进化计算的防御和贪心防御下，成功率分别增加到 21.87% 和 24.18%。

总的来说，我们的结果表明，对于全部三个活动目录图，即 r1000、r2000 和 r4000，平均而言，基于 C-EDO 的防御是最佳防御，攻击者的成功率最低。此外，基于进化计算的防御优于贪心防御。值得注意的是，*NNDP-EDO* 方法 *[3]*无法扩展到 *r4000* 这类大型活动目录图；然而，我们提出的强化学习+*C-EDO* 方法可以扩展到 *r4000* 图。

5.4 实验设置 2

在此实验设置中，我们评估所提出的基于强化学习的攻击者策略的有效性。

* 表示在采用通用参数设置时，强化学习策略的结果略差于基线。因此，我们将强化学习策略的训练轮次设为 300，而非 150。只要给予足够的时间，强化学习策略就会优于基线。

表2：不同攻击者策略下攻击者成功概率的比较（数值越大表示性能越好）。结果表明，所提出的基于强化学习的攻击者策略比NNDP策略更准确地近似攻击者的问题 [3]。

图	策略	独立	正	负	平均
r500	强化学习（提出的方法）	**88.01%**	**86.58%**	**89.37%**	**87.98%**
	NNDP	87.57%	86.08%	89.28%	87.64%
r1000	强化学习（提出的方法）	**54.99%**	48.21%*	**52.69%**	**51.96%**
	NNDP	53.52%	**48.32%**	52.15%	51.33%
r2000	强化学习（所提方法）	**45.28%***	**56.41%**	**42.43%***	**48.04%**
	NNDP	45.11%	56.29%	42.39%	47.93%

表3：不同攻击者-防御者策略下，攻击者攻破最佳防御的成功概率（数值越小，性能越好）。结果表明，强化学习+C-EDO 策略比 NNDP-EDO [3]能生成更好的防御。

图	策略	独立	正	负	平均
r1000	强化学习中 C-EDO 的最佳防御+（提出的方法）	**40.16%**	**41.36%**	**41.51%**	**41.01%**
	NNDP-EDO 的最佳防御	42.02%	44.76%	41.53%*	42.77%
r2000	最佳防御 来自 强化学习+C-EDO (提出)	**25.09%**	32.45%	**30.44%**	**29.32%**
	NNDP-EDO 的最佳防御	30.31%	**30.17%***	30.85%	30.44%

5.4.1 基线

我们将所提出的基于强化学习的攻击者策略与基于神经网络的动态规划（NNDP）攻击者策略 [3] 进行比较。在 NNDP 方法中，作者训练了神经网络来解决攻击者的问题。

5.4.2 结果

我们的基线 NNDP 方法仅适用于 r2000 图；因此，我们在 $r500$、$r1000$ 和 $r2000$ 活动目录图上进行实验。我们为每个活动目录图随机生成 10 种环境配置。首先，我们在 10 个环境中训练基于 NNDP 的攻击者策略，共训练 2000 轮，并进行 5000 次蒙特卡洛模拟，以计算攻击者在每个环境中的成功率。随后，我们在同一组 10 个环境中训练所提出的基于强化学习的攻击者策略，训练 150 轮，然后通过 5000 个回合评估训练好的策略，以计算攻击者的成功概率。我们在表2中报告了攻击者在 10 个环境中的平均成功概率。对于给定的环境配置，产生更高成功率的攻击者策略表明，该策略能够比其他策略更准确地逼近攻击者的问题。表2显示，对于 r500 图，强化学习策略下攻击者的平均成功率为 87.98%，略高于基于 NNDP 的策略。对于 r1000 图，攻击者的平均成功概率为 51.96%，同样高于 NNDP 策略。我们的结果表明，与基于 NNDP 的策略相比，在我们提出的基于强化学习的策略下，攻击者成功率更高，这意味着强化学习策略比 NNDP 策略更能有效地应对防御。

5.5 实验设置 3

在此实验设置中，我们确定了所提出的基于 C-EDO 的防御措施的有效性。

5.5.1 基线

我们将所提出方法的最终环境配置（防御计划）与 NNDP-EDO 方法的最终配置进行比较 [3]。

5.5.2 结果

我们使用5个随机种子（0至4）运行强化学习+C-EDO和NNDP-EDO方法，以获得防御者的最佳环境。我们在两种方法各自最优的环境配置上（使用5个随机种子）训练强化学习攻击者策略，共训练150个训练轮次。随后，我们对训练好的策略进行5000个回合的评估，以计算攻击者成功率。结果见表3。若基于强化学习的攻击者策略在某环境配置下取得较低的成功率，则该配置被视为最佳环境配置。表3的结果显示，攻击者成功率的平均值

在r1000 AD图上，采用NNDP-EDO的最佳配置时为42.77%。然而，采用强化学习+C-EDO的最佳配置时，攻击者成功率为41.01%，比前一种方法低1.76%。同样，对于r2000 AD图，采用基于强化学习+C-EDO的防御计划时，攻击者成功率最低。结果表明，我们的方法强化学习+C-EDO能够生成更优的环境配置，并降低攻击者的成功率。

5.6 讨论

我们的第一个实验设置证明，所提出的基于多样性的C-EDO防御策略优于基于EC的防御和贪心防御。结果还表明，基于EC的防御优于贪心防御。我们的第二个实验设置证明，所提出的基于强化学习的攻击者策略比NNDP策略更准确地近似攻击者的问题。对于相同的环境配置，基于强化学习的攻击者策略比NNDP策略取得更高的成功率。我们的第三个实验设置证明，所提出的RL+C-EDO方法生成的环境配置优于NNDP-EDO，进而降低了攻击者的成功率。通过这种方式，我们所提出的方法比现有方法更能实现攻击者和防御者的目标。值得注意的是，我们所提出的方法可扩展至大规模 $r4000$ 活动目录图，而NNDP-EDO方法无法扩展至此规模。总之，与现有方法相比，我们所提出的方法效果显著、可扩展性更强，能够更准确地近似攻击者的问题，并生成更优的防御方案。

6 结论

我们研究了可配置环境中的斯塔克尔伯格博弈模型，其中攻击者的目标是制定策略，以最大化其可实现奖励。防御者则试图确定使攻击者可获得的奖励最低的环境配置。我们提出了一种基于强化学习的方法来解决攻击者问题，并提出了一种基于评论家网络辅助进化多样性优化的策略来解决防御者问题。我们针对众多环境同时训练攻击者策略。我们利用训练好的强化学习评论家网络评估环境配置的适应度。实验结果表明，所提出的方法十分有效，并且可扩展至大规模活动目录攻击图。

致谢

本研究得到澳大利亚研究理事会的资助，资助项目编号为DP190103894和FT200100536；同时使用了阿德莱德大学提供的Phoenix高性能计算服务的超级计算资源。

参考文献

[1] John Dias。《微软活动目录（AD）设计指南》。技术报告，劳伦斯利弗莫尔国家实验室（LLNL），美国加利福尼亚州利弗莫尔，2002 年。[2] John Dunagan、Alice X Zheng 和 Daniel R Simon。Heat-ray：利用机器学习、组合优化和攻击图对抗身份雪球攻击。载于*ACM SIGOPS* 第 *22* 届操作系统原理研讨会论文集，第 305–320 页，2009 年。[3] Diksha Goel、Max Hector Ward-Graham、Aneta Neumann、Frank Neumann、Hung Nguyen 和 Mingyu Guo。结合基于神经网络的动态规划与进化多样性优化来防御活动目录。载于遗传与进化计算会议论文集，GECCO ’22，第 1191–1199 页，2022 年。[4] Leslie Pack Kaelbling、Michael L Littman 和 Andrew W Moore。强化学习：综述。人工智能研究杂志，4：237–285，1996 年。[5] Mingyu Guo、Jialiang Li、Aneta Neumann、Frank Neumann 和 Hung Nguyen。用于防御活动目录风格攻击图的实用固定参数算法。载于*AAAI*人工智能会议论文集，第 36 卷，第 9360–9367 页，2022 年。[6] Mingyu Guo、Max Ward、Aneta Neumann、Frank Neumann 和 Hung Nguyen。用于防御活动目录风格攻击图的可扩展边阻断算法。载于*AAAI*人工智能会议论文集（*2023*），2023 年。[7] Quang Huy Ngo、Mingyu Guo 和 Hung Nguyen。动态大型活动目录网络中蜜罐部署的近最优策略。载于第 *22* 届自主智能体与多智能体系统国际会议（*AAMAS*），2023 年。扩展摘要。

[8] Yumeng Zhang、Max Ward、Mingyu Guo 和 Hung Nguyen。用于强化大型活动目录系统的可扩展双重预言机算法。载于第 *18* 届 *ACM*亚洲计算机与通信安全会议（*ACM ASIACCS*），澳大利亚墨尔本，*2023* 年，2023 年。[9] Tamara Ulrich、Johannes Bader 和 Eckart Zitzler。将决策空间多样性融入基于超体积的多目标搜索。载于第 *12* 届遗传与进化计算年度会议论文集，第 455–462 页，2010 年。[10] Chen Huang、Xiangbing Zhou、Xiaojuan Ran、Yi Liu、Wuquan Deng 和 Wu Deng。采用三阶段策略的大规模复杂优化问题协同进化竞争群优化器。*Information Sciences*，619：2–18，2023 年。[11] Jakob Bossek 和 Frank Neumann。进化多样性优化与最小生成树问题。载于遗传与进化计算会议论文集，第 198–206 页，2021 年。[12] Anh Do、Mingyu Guo、Aneta Neumann 和 Frank Neumann。排列问题进化多样性优化的分析。*ACM*进化学习汇刊，2（3）：1–27，2022 年。[13] Adel Nikfarjam、Jakob Bossek、Aneta Neumann 和 Frank Neumann。面向旅行商问题的基于熵的进化多样性优化。*arXiv*预印本 *arXiv:2104.13538*，2021 年。[14] Greg Brockman、Vicki Cheung、Ludwig Pettersson、Jonas Schneider、John Schulman、Jie Tang 和 Wojciech Zaremba。OpenAI Gym。*arXiv*预印本 *arXiv:1606.01540*，2016 年。[15] Jiayi Weng、Huayu Chen、Dong Yan、Kaichao You、Alexis Duburcq、Minghao Zhang、Hang Su 和 Jun Zhu。天授：高度模块化的深度强化学习库。*arXiv*预印本 *arXiv:2107.14171*，2021 年。[16] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。近端策略优化算法。*arXiv*预印本 *arXiv:1707.06347*，2017 年。