

基于强化学习的最小全支配集问题求解框架

黄堪谦, 何伟骅*

广东工业大学数学与统计学院, 广东 广州

收稿日期: 2026年2月16日; 录用日期: 2026年3月9日; 发布日期: 2026年3月18日

摘要

最小全支配集(Minimum Total Dominating Set, MTDS)问题旨在在图中寻找一个最小顶点集合, 使得每个顶点至少与集合中的一个顶点相邻, 该问题在网络监测与故障检测等领域具有广泛应用。针对现有启发式方法在复杂图结构下性能受限的问题, 本文提出了一种基于深度强化学习的端到端求解方法。该方法采用自回归方式逐步生成候选解, 并利用消息传递神经网络刻画图结构中的局部依赖关系, 以指导节点选择决策。此外, 本文设计了一种新的奖励函数, 对解的构造过程进行约束, 从而保证结果的可行性与有效性。实验结果表明, 该方法在不同规模和结构的图上的解质量均优于多种经典启发式算法。

关键词

组合优化, 图神经网络, 深度强化学习, 全支配集

A Reinforcement Learning Framework for Minimize Total Dominating Set Problem

Kanqian Huang, Weihua He*

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: February 16, 2026; accepted: March 9, 2026; published: March 18, 2026

Abstract

Finding a Minimum Total Dominating Set (MTDS)—a minimal vertex set where every graph node has a neighbor in the set—is a critical task in network monitoring and fault detection. However, existing heuristic approaches often suffer from limited scalability on complex graph structures. To overcome this, this paper presents an end-to-end deep reinforcement learning solver. The proposed model operates in an autoregressive manner, utilizing a Message Passing Neural Network (MPNN) to encode local topological dependencies and guide decision-making. Additionally, we

*通讯作者。

design a constraint-aware reward function to regulate the solution construction process, ensuring both feasibility and effectiveness. Experimental results confirm that our method achieves superior solution quality compared to standard baselines across a wide range of graph sizes and configurations.

Keywords

Combinatorial Optimization, Graph Neural Networks, Deep Q-Learning, Total Dominating Set

Copyright © 2026 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

支配(Domination)作为图论中的基本概念之一, 长期以来在理论研究与工程应用中受到广泛关注[1]。该理论为刻画图中节点之间的覆盖关系提供了统一的数学工具, 并在多种实际问题中具有重要应用价值, 例如复杂生态系统建模[2]、编码理论中的结构优化[3], 以及电力网络[4]和通信监控系统[5]的设计等。关于支配理论的系统性研究及其典型应用, 可参考 Abu-Khzam 与 Haynes 等人综述与专著[6] [7]。

在通信基础设施、传感器网络和分布式计算平台等系统中, 系统结构通常可以抽象为图模型, 其中顶点表示系统中的实体, 边刻画实体之间的交互或通信关系。在此类系统中, 一个核心问题是在资源受限的条件下选取尽量少的关键节点, 以保证系统整体的可观测性、可控性或容错能力。典型应用包括通信网络中的传感器部署、分布式系统中的控制节点选择, 以及容错架构中的验证节点配置等。

为了对上述问题进行统一的数学刻画, 相关研究通常将其抽象为图论中的支配模型。给定图 $G=(V, E)$, 其中 V 表示节点集合, E 表示边集合, 若存在顶点子集 $S \subseteq V$, 使得图中任意顶点要么属于 S , 要么与 S 中至少一个顶点相邻, 则称 S 为一个支配集(Dominating Set) [8]。等价地, 支配集要求所有顶点的闭邻域覆盖整个顶点集。最小支配集问题旨在寻找基数最小的支配集, 该问题在图论及其相关应用领域中已被广泛研究。

然而, 在许多安全关键或对抗性环境中, 节点通常不具备自监测能力, 而必须依赖邻居节点进行监督。在此情形下, 引出了全支配集(Total Dominating Set, TDS)问题[9]。若顶点集合 S 满足图中任意顶点均至少与 S 中的一个顶点相邻, 则称 S 为一个全支配集。该定义排除了顶点对自身的支配, 因此仅当图中不存在孤立顶点时, 全支配集才存在。与经典支配问题相比, 全支配约束引入了更严格的结构要求, 使问题在组合性质和算法设计方面均更加复杂。全支配及其相关变体已发展成为图论中的一个重要研究分支, 其系统性综述可参见文献[10]-[13]。

1.1. 研究背景

相较于标准支配问题, 全支配问题引入了更为严格的约束, 即图中每个顶点必须由其邻域中的某个被选顶点所支配。该约束在局部连通层面隐含地要求一定程度的冗余覆盖, 因此在容错网络建模、通信调度等应用场景中具有更强的现实意义。然而, 这一额外约束也显著压缩了可行解空间, 使节点选择之间呈现出更强的依赖关系。同时, 单个节点的选取不仅影响其自身的支配状态, 还会改变多个邻居节点的可行性, 从而对后续决策产生连锁影响, 使解的构造过程具有明显的路径依赖性。

在这一背景下, 节点选择不再构成可分解的局部决策问题, 而是需要在全局层面刻画重叠邻域之间

的相互作用。这种结构性依赖破坏了问题的可分解性与局部最优性质, 使得可行解空间呈现出高度受限的组合结构, 从而增加了问题的求解难度; 事实上, MTDS 问题已被证明是 NP-难问题。

现有针对 MTDS 的求解方法主要依赖人工设计的启发式规则、近似算法或分布式策略。这类方法通常建立在特定图结构假设之上, 例如度数受限、平面性或图直径较小等, 在这些场景中能够取得较为理想的效果。然而, 在一般图结构下, 全支配约束所引入的局部依赖关系会显著增强决策之间的结构性耦合: 单个节点的加入或移除往往会同步改变多个邻居节点的可行状态, 从而破坏问题的局部可分解性, 使得这类启发式算法容易产生累积误差, 并过早陷入局部最优。

近年来, 已有工作开始尝试引入学习驱动方法解决图上的组合优化问题。图神经网络(Graph Neural Network, GNN) [14]通过邻域信息传播与聚合机制学习节点间的结构表示, 而强化学习(Reinforcement Learning, RL) [15]则可将组合优化过程形式化为序列决策问题, 使智能体能够通过与环境交互逐步构造解。然而, 将 RL 直接应用于 MTDS 问题仍面临以下两个挑战: 一方面, 满足全支配约束的中间状态在搜索空间中高度稀疏, 使得有效探索面临较大困难; 另一方面, 在不引入额外修复或回溯机制的前提下, 如何在策略层面显式建模并保证动作序列的可行性约束, 仍缺乏通用且有效的方法。

1.2. 相关工作

强化学习已被广泛应用于组合优化问题的求解, 其核心思想是将组合优化过程形式化为序列决策问题, 使智能体能够通过与环境交互逐步构造解, 从而在无需显式设计复杂规则的情况下学习有效的优化策略[16]。

Khalil 等人提出了将结构化图嵌入(Structure2Vec, S2V)与 Q 学习相结合的框架(S2V-DQN), 系统性地深度强化学习引入图上的组合优化问题。该方法通过对图结构进行嵌入表示, 在此基础上学习节点级动作价值函数, 并采用贪婪策略逐步生成解, 在最小顶点覆盖、最大割和旅行商问题等任务中验证了其在不同图规模上的可迁移性[17]。随后, Guo 等学者进一步提出了一种结合模仿学习辅助与密度初始化的深度强化学习模型, 通过将设施重定位任务建模为 MDP 并引入基于交换操作的 PPO 算法, 在求解效率与收敛速度方面均取得了显著提升[18]。

针对最小全支配集(MTDS)问题, 现有研究仍主要依托于近似算法与分布式计算框架展开。在特定图结构(如树图、平面图或度受限图)下, 已有工作给出了若干多项式时间算法或常数近似解法; 而在一般图情形中, Chlebík 和 Chlebíková、Zhu 等人的研究系统刻画了该问题在近似意义下的理论界限。尽管这些结果在理论分析上具有重要价值, 但往往依赖较强的结构假设, 难以在复杂异构网络中保持稳定的解质量[19] [20]。

在分布式计算环境中, 相关研究进一步考虑了通信复杂度与局部信息限制等因素。Alipour 等人针对不含特定短圈的平面图设计了可在常数轮通信内完成的分布式近似算法; Belhoul 等人则从自稳定计算角度出发, 研究了最小全支配集的分布式构造问题。然而, 这类方法通常侧重于保证“极小性”意义下的可行解, 或依赖较强的图结构假设, 难以确保解的全局最优性[21] [22]。

总体来看, 现有针对 MTDS 的求解方法多建立在启发式规则、局部搜索机制或线性规划松弛基础之上, 并在不同程度上受限于图结构的规则性假设。面对一般异构图或大规模复杂网络, 这些方法在泛化能力、可扩展性以及解质量稳定性方面仍存在明显不足。

从方法论角度而言, 神经组合优化(Neural Combinatorial Optimization, NCO) [23]逐渐成为组合优化领域[24]-[27]的重要研究方向, 也为缓解上述局限提供了新的思路。早期工作主要集中于旅行商问题、车辆路径规划等经典任务, 通过监督学习或强化学习直接学习构造策略; 随后, 端到端强化学习框架被提出, 以减少对人工启发式规则的依赖。然而, 尽管传统启发式策略凝聚了大量领域经验知识, 在面对结构复

杂、约束高度耦合的组合优化问题时, 这类基于经验的设计往往难以覆盖所有边缘情况, 从而导致求解结果陷入局部最优[28]。

近期, 一些基于图神经网络(Graph Neural Network, GNN)的强化学习方法被用于求解最大独立集、顶点覆盖等图结构优化问题。这类方法通常在策略层引入松弛或局部搜索操作以处理非法动作, 尽管在部分任务上取得了一定效果, 但在一般图结构下, 仍缺乏一种能够以端到端方式直接生成满足全支配约束的 MTDS 解的深度强化学习框架。另一方面, 将强化学习引入经典启发式框架、构建混合驱动的求解策略已成为新的研究趋势, 该类方法利用智能体的自主探索能力挖掘潜在优化规则, 在提升算法泛化能力的同时, 有效降低了计算开销[29]-[31]。

2. 论文方法

本节首先对最小全支配集(MTDS)问题进行形式化定义, 随后介绍本文提出的基于图的强化学习求解框架。具体而言, 我们说明了图结构的表示方法, 并系统地描述了强化学习模型的构建过程, 包括状态空间、动作空间、奖励函数以及策略的设定。此外, 还详细阐述了一种用于引导模型训练的密集奖励设计机制。整体方法框架如图 1 所示。

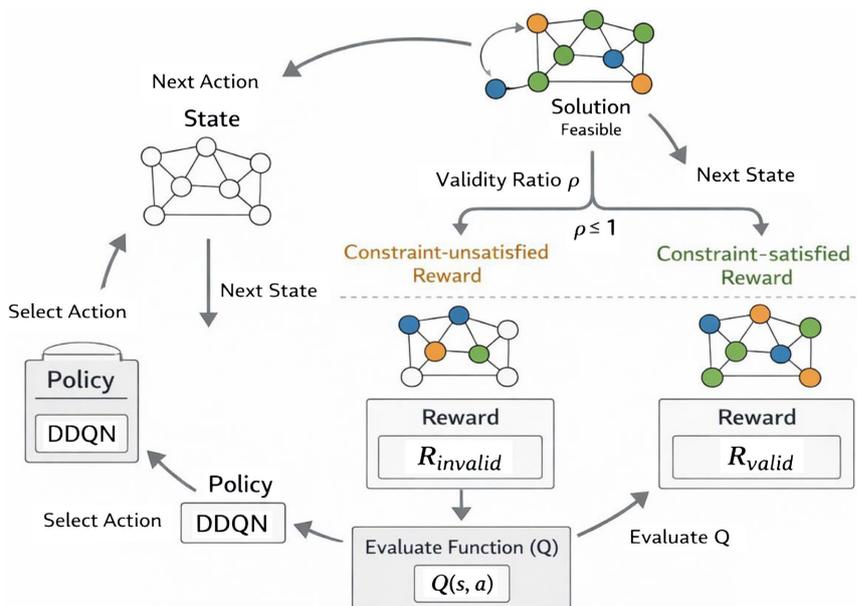


Figure 1. Flowchart of the reinforcement learning algorithm framework
图 1. 强化学习算法框架流程图

2.1. MTDS 问题定义

在一个无向图 $G=(V, E)$, 对于任意节点 $v \in V$, 记 $N(v)$ 为所有与 v 直接相连的邻居节点集合, 该集合不包含节点 v 本身。在该图中, 若一个节点子集 $S \subseteq V$ 满足对任意节点 $v \in V$, 其邻居集合中至少存在一个节点属于 S , 即 $\forall v \in V, N(v) \cap S \neq \emptyset$, 其中 $N(v) = \{u \in V | (v, u) \in E\}$, 则称 S 为一个全支配集(Total Dominating Set)。该定义要求每一个节点都必须通过邻接关系被支配, 因此节点不能通过自身满足支配条件, 同时集合 S 中的节点也必须被集合中的其他节点支配。

在所有满足上述条件的全支配集中, 节点数量最少的集合 S^* 被称为最小全支配集(Minimum Total Dominating Set, MTDS), 该问题的目标是在保证全支配约束成立的前提下, 使被选中节点的数量尽可能

小。

$$S^* = \operatorname{argmin}_{S \subseteq V} \{|S| : S \text{ is a total dominating set of } G\}.$$

2.2. 图形表示

虽然过去的研究常使用 DeepWalk、LINE 和 Struc2Vec 等方法来探索图嵌入,但在本文中,我们选择了一条不同的路径。为了有效建模图结构中的拓扑关系以及节点之间的长距离关系,本文采用消息传递神经网络(Message Passing Neural Network, MPNN)作为状态编码器,对当前图状态进行表示学习。

在时间步 t , 图的状态由节点特征矩阵表示。对于每个节点 $v \in V$, 其初始特征向量定义为:

$$h_v^{(0)} = \operatorname{Concat}(x_{t,v}, \operatorname{deg}(v), \mathbb{I}[v \in S_t], \rho_{\text{local}}(v))$$

其中, $x_{t,v} \in \{-1, 1\}$ 表示节点当前是否被选中, $\operatorname{deg}(v)$ 为归一化后的节点度数, $\mathbb{I}[v \in S_t]$ 为指示函数, 用于表示节点是否属于当前支配集 S_t , $\rho_{\text{local}}(v)$ 表示节点 v 的邻居中已有多少节点被选入支配集。这些特征综合刻画了节点的局部结构信息以及当前决策状态。

为了传播图结构信息并捕获高阶邻域关系,本文采用一个包含 K 层的 MPNN 进行迭代更新。第 k 层中,每个节点首先从其邻居节点聚合消息:

$$m_v^{(k)} = \sum_{u \in N(v)} \mathcal{M}^{(k)}(h_u^{(k-1)}, h_v^{(k-1)})$$

随后根据自身上一层表示和聚合消息更新节点嵌入:

$$h_v^{(k)} = \mathcal{U}^{(k)}(h_v^{(k-1)}, m_v^{(k)})$$

其中, $\mathcal{M}^{(k)}(\cdot)$ 和 $\mathcal{U}^{(k)}(\cdot)$ 分别表示第 k 层的消息函数和更新函数。本文采用求和作为聚合算子,以保留邻域规模相关的信息。经过 K 层传播后得到的节点表示 $h_v^{(k)}$ 被用于后续的 Q 值估计。

2.3. 强化学习

本文采用强化学习算法对 MTDS 问题进行求解。在明确给定状态空间、动作空间及奖励函数的基础上,选用双重深度 Q 网络(DoubleDeepQ-Network, DDQN)作为求解算法。

2.3.1. 强化学习建模

本文将 MTDS 问题建模为一个马尔可夫决策过程(MDP),其核心要素定义如下:

1) 状态 s_t : 状态由图的拓扑结构与节点属性共同构成,用以刻画当前解的整体配置。具体而言,状态包含节点的选择状态(是否属于当前集合 S_t)、邻域覆盖统计信息,以及若干反映局部结构变化的辅助特征,从而为决策过程提供必要的结构信息。

2) 动作 a_t : 动作定义为对某一顶点 $v \in V$ 的选择状态进行翻转。设二值变量 $x_v \in \{0, 1\}$ 表示节点 v 是否属于当前集合 S_t , 则执行动作 $a_t = v$ 等价于更新

$$x_v \leftarrow 1 - x_v$$

当节点状态发生变化时,其邻域 $N(v)$ 的相关覆盖信息将同步更新,并反映在后续状态表示中。

3) 奖励 r_t : 奖励函数用于刻画当前状态对全支配约束的满足程度及解规模的优劣。本文设计了一种与全支配结构一致的稠密奖励机制,使智能体在构造解的过程中能够逐步向可行区域收敛,并倾向于生成规模较小的解。奖励函数的具体形式将在第 2.4 节中给出。

4) 动作价值函数 $Q(s, a; \theta)$: 采用参数为 θ 的深度 Q 网络对动作价值函数进行近似,用于评估在状

态 s 下选择动作 a 的期望累计回报:

$$Q(s, a; \theta) \approx \mathbb{E} \left[\sum_{k=0}^T \gamma^k r_{t+k} \mid s_t = s, a_t = v \right]$$

网络输入为图的嵌入表示, 输出为针对每个顶点的 Q 值估计。

5) 策略 π : 在训练过程中, 采用 ϵ -greedy 策略进行动作选择, 即以概率 ϵ 随机选择动作进行探索, 以概率 $1-\epsilon$ 选择当前 Q 值最大的动作:

$$\pi(v|s) = \arg \max_{v' \in V'} Q(s, v')$$

2.3.2. DDQN

为了保障训练过程的稳定性并有效缓解标准 DQN 算法中常见的价值高估偏差, 本文采用了双重深度 Q 网络(Double Deep Q-Network, DDQN)架构。不同于传统的构造式方法, 我们将最小全支配集(MTDS)求解建模为一个支持状态回溯的可逆马尔可夫决策过程。在此框架下, 智能体的动作空间定义为对图上顶点状态的“翻转”操作(Flip Operation), 即允许将顶点加入解集中也允许从解集中移除。这种非单调的状态转移机制使得智能体能够在解空间中进行灵活的探索与利用, 从而以此跳出局部最优陷阱。在每一个时间步 t , 智能体基于当前图的全局状态与局部拓扑特征, 利用由 Q 函数导出的策略选择能够最大化长期回报的翻转动作, 而相应的目标值 y_t 则通过独立的目标网络 θ^- 进行计算:

$$y_t = r_t + \gamma Q(s_{t+1}, \arg \max_{a'} Q(s_{t+1}, a'; \theta); \theta^-)$$

其中, θ 表示在线 Q 网络的参数, 而 θ^- 则代表目标网络的参数。通过将动作的选择过程与评估过程进行解耦(decoupling), DDQN 有效地抑制了 Q 值的高估现象。

该 Q 网络的训练目标是 minimized 预测 Q 值与目标值之间的均方误差(MSE)损失:

$$\mathcal{L}(\theta) = \frac{1}{2} (Q(s_t, a_t; \theta) - y_t)^2$$

我们利用梯度下降算法对在线网络的参数进行迭代优化, 而针对目标网络, 则采用软更新策略来实现参数的平滑同步:

$$\theta^- \leftarrow \tau \theta + (1 - \tau) \theta^-$$

其中, $\tau \in (0, 1)$ 是用于控制参数更新速率的超参数。

2.4. 基于约束感知的稠密奖励机制

奖励函数的设计逻辑: 本文的奖励函数旨在引导智能体依次达成两个连续的优化目标: (i) 首先确保满足总支配集的约束条件, (ii) 在获得可行解的基础上, 进一步最小化解的规模。为了克服稀疏奖励可能引发的训练不稳定性, 同时将需要调节的超参数数量降至最低, 我们构建了一个简洁的密集奖励(dense reward)公式, 该公式仅由三个具有明确物理意义的系数控制。设 $S_t \subseteq V$ 为第 t 步时选定的顶点集合。我们引入两个归一化指标来刻画当前的解状态, 即合法性比率 ρ_t 和归一化规模 σ_t :

$$\rho_t = \frac{1}{|V|} \sum_{v \in V} \mathbb{I}[N(v) \cap S_t \neq \emptyset]$$

$$\sigma_t = \frac{|S_t|}{|V|}$$

其中, $N(v)$ 表示顶点 v 的开邻域(即不包含 v 自身的邻居集合)。当且仅当指标 ρ_t 达到 1 时, 当前的解集

才被视为一个合法的总支配集。基于此, 我们将“单步改进量(step-wise improvement)”定义如下:

$$\Delta\rho_t = \rho_{t+1} - \rho_t$$

$$\Delta\sigma_t = \sigma_t - \sigma_{t+1}$$

$\Delta\rho_t > 0$ 标志着解的可行性得到了增强(即更接近合法解), 而 $\Delta\sigma_t > 0$ 则意味着解集规模的有效缩减。

2.4.1. 可行性约束的满足

当解处于不可行区域(infeasible region)时, 优化的首要任务是最大化节点的覆盖范围。为此, 奖励函数被设计为一方面激励能够提升合法性比率的动作, 另一方面则根据当前状态与完全可行状态之间的距离(distance to feasibility)施加惩罚:

$$R_{\text{invalid}} = \lambda \cdot \Delta\rho_t - \lambda \cdot (1 - \rho_{t+1})$$

其中, $\lambda > 0$ 是一个调节系数, 用于设定约束满足任务的优先级。该数学形式利用 $\Delta\rho_t$ 的符号特性, 将针对建设性(即修复)动作与破坏性(即损伤)动作的反馈机制, 巧妙地统一在同一个框架之内。

2.4.2. 解集合优化

一旦达成有效性约束条件($\rho_t + 1 = 1$), 算法的目标便切换为解集规模的最小化。此时, 奖励机制将严格地仅针对那些成功减小集合大小的动作发放正向反馈:

$$R_{\text{valid}} = \mu \cdot \Delta\sigma_t$$

其中, 系数 $\mu > 0$ 决定了修剪冗余成分的力度。若动作导致解的大小不降反升, 随之产生的负 $\Delta\sigma_t$ 将直接转化为负向反馈。

2.4.3. 统一奖励函数

为了提升搜索效率并抑制潜在的策略震荡, 我们在每一个时间步都引入了一个微小的常数级时间惩罚 $c > 0$ 。综上所述, 第 t 步的最终统一奖励函数定义如下:

$$r_t = \begin{cases} R_{\text{invalid}} - c, & \text{if } \rho_{t+1} < 1 \\ R_{\text{valid}} - c, & \text{if } \rho_{t+1} = 1 \end{cases}$$

本文的奖励函数由三个标量系数 λ 、 μ 及 c 共同调控。在整个训练周期内, 我们对这些超参数采用了固定赋值策略。值得强调的是, 尽管参数本身保持恒定, 但得益于奖励函数的分层设计, 不同的奖励项会在状态空间的不同区域交替占据主导地位: 这一机制使得模型能够在决策初期优先致力于满足可行性约束, 而在一旦搜索到合法解后, 则自动将重心转向对冗余节点的有效剪枝。

3. 实验与结果分析

3.1. 实验生成

为保证所学习策略在不同图结构下的鲁棒性, 本文设计并实现了一个定制化的随机无孤立图生成器(random non-isolated graph generator)用于模型训练。与基于固定数据集的训练方式不同, 该生成器能够在训练过程中在线动态生成多样化的图实例, 从而显著提升训练数据的覆盖范围。针对最小全支配集问题的结构约束, 我们在图生成过程中引入了以下两项关键机制以保证实例的有效性:

- **动态密度机制:** 对于每一个训练样本, 连接概率并非固定设定, 而是在区间[0.2, 1.0]内进行均匀采样。该设计有助于避免模型过度适应特定稀疏或稠密模式, 从而促使智能体学习到对不同图密度具有良好适应性的通用决策策略。
- **连通性保障机制:** 鉴于最小全支配集问题要求图中不存在孤立节点, 生成器在构造图结构后会对所

有节点的度数进行校验。一旦检测到度数为 0 的节点, 系统将强制为其随机添加一条连接边。该机制确保所有生成实例在数学上均存在合法的全支配集解, 从而保证训练过程的可行性与稳定性。

在训练阶段, 我们仅使用节点规模为 $N=20$ 的小规模随机图进行模型优化。为了评估模型在未见拓扑结构上的零样本泛化能力(zero-shot generalization), 测试阶段额外构建了独立数据集, 其中包含高度规则化的结构图, 包括三角晶格图(Triangular Lattices)和网格图(Grid Graphs), 用于检验模型在强结构约束场景下的表现。

3.2. 实验设置

为验证所提出方法的有效性, 本文选取经典贪婪算法(Greedy baseline)作为对比基线。该算法在每一步迭代中选择能够覆盖最多当前未被支配节点的顶点, 属于解决支配类问题的标准启发式方法。

在评价指标方面, 我们主要关注解的规模, 即总支配集的节点数量 $|S|$ 。由于本文提出的深度强化学习智能体在所有测试实例上均能够稳定生成满足约束条件的可行解, 因此评估过程不涉及可行性统计, 而仅聚焦于解质量的对比。

在实现细节方面, 本文方法基于 PyTorch 框架实现。MPNN 编码器 $K=3$ 个消息传递层构成, 隐藏维度设为 64。训练阶段采用 Adam 优化器, 学习率为 1×10^{-4} , 总训练步数为 800,000。探索率 ϵ 在前 10,000 步内由 1.0 线性衰减至 0.05。经验回放池容量设为 100,000, 批次大小 32。所有实验均在 NVIDIA-ARTXA6000 GPU 上完成。

3.3. 结果分析

3.3.1. 随机图环境下的性能评估(ER 与 BA)

为了验证模型在不同图密度及拓扑结构下的鲁棒性, 我们在具有多样化参数配置的 Erdos-Renyi (ER) 和 Barabasi-Albert (BA)图上进行了实验测试。将 Gurobi 求解器得到的解作为最优解基准, 计算本文模型(Ours)与其的差距(Gap)。

在 ER 随机图中, 在节点规模较小($N=20$)时, 本文方法能够接近最优解(Gap 仅为 0%~12.5%)。随着图规模的增长($N=200$), 虽然与最优解的 Gap 有所扩大, 但相对于传统启发式算法仍保持显著优势。以 ER ($p=0.1, N=200$)为例, 虽然我们的解(26.0)与 Gurobi (15.0)存在差距, 但相比于 Greedy (85.0)和 GA (47.0), 我们的方法分别实现了 69.4%和 44.7%的性能提升。这表明在处理大规模稀疏图的长程依赖问题时, 基于 GNN 的智能体虽然难以达到理论最优, 但其决策能力远超传统的贪婪策略, 见表 1。

Table 1. Evaluation of ability on ER graphs

表 1. ER 图上的能力评估

p	Nodes	Ours	Gurobi	GA	Greedy	Gap (%)
0.1	20	9.0	8.0	8.0	10.0	12.50
	40	13.0	12.0	14.0	19.0	8.33
	80	14.0	12.0	24.0	30.0	16.67
	100	17.0	13.0	27.0	41.0	30.77
	200	26.0	15.0	47.0	85.0	73.33
<u>0.3</u>	20	3.0	3.0	3.0	4.0	0.00
	40	5.0	4.0	6.0	14.0	25.00

续表

	80	10.0	5.0	9.0	27.0	100.00
	100	13.0	5.0	12.0	39.0	160.00
	200	26.0	6.0	41.0	82.0	333.33

在 BA 无标度网络图中, 实验结果表明, 该模型对无标度拓扑具有较强的适应性。在小规模图上(N = 20, 40): 无论 m = 4 还是 m = 8, 模型均能多次达到最优解(Gap = 0%或接近 0%), 证明了其在低维空间的求解精确度。在大规模图上(N = 200): 面对复杂的连通结构, Greedy 算法的性能急剧下降(例如 m = 8 时解集大小达 82.0), 而我们的模型将解集控制在 29.0。尽管此时 Gap 达到 107.14%, 但相对于 Greedy 算法, 解集规模缩减了近 65%, 有力证明了 RL 策略能有效避免解集在大规模网络中的爆发式增长, 见表 2。

表中 Gap (差距)定义为

$$\frac{\text{Ours} - \text{gurobi}}{\text{gurobi}} \times 100\%$$

Table 2. Evaluation of ability on BA graphs

表 2. BA 图上的能力评估

m	Nodes	Ours	Gurobi	GA	Greedy	Gap (%)
4	20	2.0	2.0	2.0	5.0	0.00
	40	6.0	6.0	8.0	11.0	0.00
	80	16.0	11.0	22.0	33.0	45.45
	100	19.0	12.0	25.0	44.0	58.33
	200	45.0	25.0	75.0	91.0	80.00
8	20	2.0	2.0	2.0	7.0	0.00
	40	4.0	3.0	5.0	12.0	33.33
	80	12.0	7.0	13.0	31.0	71.43
	100	14.0	8.0	20.0	40.0	75.00
	200	29.0	14.0	58.0	82.0	107.14

3.3.2. 规则格点图上的零样本泛化

为了验证模型是否具有泛化能力, 我们将仅在 N = 20 随机图上训练完成的模型, 直接部署到节点规模从 20 到 200 不等的三角晶格和网格图上进行测试, 且未进行任何微调(fine-tuning)。实验结果有力地证实了该框架具备出色的零样本迁移性能。尽管训练数据仅局限于小型的随机拓扑, 但模型依然成功地识别并利用了晶格图内在的规则结构, 见表 3:

- 三角图(Triangular Graphs): 我们的模型在三角晶格结构上展现了卓越的寻优能力。在测试的 10 组不同规模中, 模型在 5 组(N = 20, 30, 40, 80, 120)中均达到了与 Gurobi 求解器完全一致的最优解(Gap = 0.00%)。特别是在 N = 120 的中等规模下, 传统的贪婪算法解集大小为 32.0, 而我们的模型精准地找到了理论最优解 28.0, 实现了 12.5%的性能提升。即便在难度最大的 N = 200 设置下, 模型依然保持了对贪婪算法的优势(50.0 vs 53.0), 证明了其在规则稠密图上的稳健性。
- 网格图(Grid Graphs): 模型在此类图上同样表现出极高的鲁棒性, 能够跨越训练分布的限制。在 N =

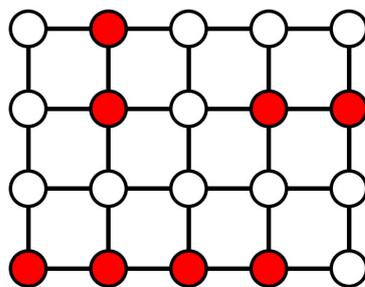
20 至 $N=100$ 的范围内, 模型在绝大多数案例(如 $N=63, 81, 100$)中均成功收敛至最优解。以 $N=100$ 为例, 与贪婪基线(34.0)相比, 我们的解集规模显著缩小至 30.0 (与 Gurobi 一致), 性能提升约 11.76%。这种随着图规模增大仍能频繁命中各类最优解的现象, 充分证明了 MPNN 编码器真正捕获了支配问题的局部拓扑法则, 而非仅仅是固态地记忆了全局的图模式。

本文模型的求解结果如图 2~5 所示, 其中图 2、图 3 为 Grid 图上的展示, 图 4、图 5 为 Tri 图上的展示。

Table 3. Evaluation of zero-shot generalization ability on regular grid graphs

表 3. 规则格点图上的零样本泛化能力评估

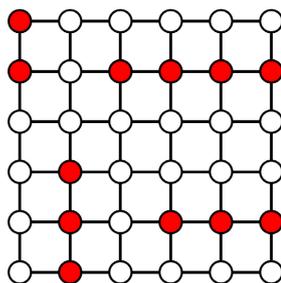
图类型	Nodes	Ours	Gurobi	GA	Greedy	Gap (%)
Grid	20	8.0	8.0	8.0	8.0	0.00
	36	12.0	12.0	13.0	12.0	0.00
	63	20.0	20.0	24.0	23.0	0.00
	72	24.0	22.0	33.0	24.0	9.09
	81	25.0	25.0	35.0	28.0	0.00
	100	30.0	30.0	46.0	34.0	0.00
	120	37.0	36.0	56.0	41.0	2.78
	128	40.0	38.0	60.0	42.0	5.26
Triangular	20	6.0	6.0	6.0	6.0	0.00
	30	8.0	8.0	9.0	9.0	0.00
	40	10.0	10.0	10.0	11.0	0.00
	50	12.0	12.0	17.0	14.0	9.09
	60	15.0	14.0	21.0	16.0	7.14
	80	19.0	19.0	26.0	21.0	0.00
	100	24.0	23.0	33.0	27.0	4.35
	120	28.0	28.0	47.0	32.0	0.00
	160	39.0	36.0	65.0	41.0	8.33
	200	50.0	44.0	87.0	53.0	13.64



20 VERTICES 4×5 GRID GRAPH

Figure 2. Solution of MTDS on grid graphs ($n=20$)

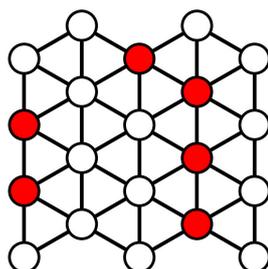
图 2. 网格图上的 MTDS 求解结果($n=20$)



**36 VERTICES
6×6 GRID GRAPH**

Figure 3. Solution of MTDS on grid graphs ($n = 36$)

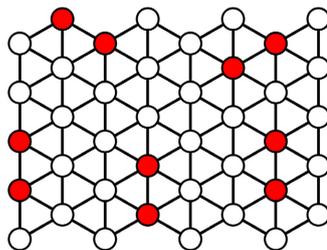
图 3. 网格图上的 MTDS 求解结果($n = 36$)



**20 VERTICES
TRIANGULAR LATTICE GRAPH**

Figure 4. Solution of MTDS on triangular lattice graphs ($n = 20$)

图 4. 三角格点图上的 MTDS 求解结果($n = 20$)



**40 VERTICES
TRIANGULAR LATTICE GRAPH**

Figure 5. Solution of MTDS on triangular lattice graphs ($n = 40$)

图 5. 三角格点图上的 MTDS 求解结果($n = 40$)

3.3.3. 计算开销分析

从算法复杂度角度分析, 基于 GNN 的 RL 在单次推理时间上相较于贪婪基线存在一定开销。这一现象符合预期, 因为贪婪算法仅涉及简单的节点度数统计, 而本文方法需要执行神经网络的前向传播计算, 主要开销来自于多层消息传递过程中的矩阵运算。

尽管如此, 我们认为该额外计算代价在组合优化场景中是可以接受的。一方面, 两种方法在时间复杂度上均保持多项式级别; 另一方面, 相比于毫秒级的推理时间差异, 解质量往往是更为关键的评价因素。实验结果显示, 在可接受的计算开销下, 所提出方法能够稳定地将解规模平均降低最高达 14.3%。

此外, 相较于精确求解器(如 Gurobi)所面临的指数级时间复杂度瓶颈, 本文方法在保证多项式复杂度的同时, 仍能够获得显著优于启发式基线的解质量, 从而在效率与性能之间提供了一种更具实践意义的折中方案。

4. 结论

本文针对最小全支配集问题, 提出了一种端到端的深度强化学习求解框架。通过将 MTDS 建模为序列决策过程, 并引入基于消息传递神经网络的状态表示以及约束感知的稠密奖励机制, 模型能够在无需后处理或启发式修复的情况下, 直接生成满足全支配约束的可行解。实验结果表明, 所提出的方法在不同规模和结构的图数据上均表现出良好的泛化能力, 并在解的规模上稳优于传统贪心策略。上述结果验证了该框架在处理强结构约束组合优化问题时的有效性与鲁棒性。未来工作将进一步考虑将该方法扩展至更大规模的图实例, 并探索其在其他具有复杂约束条件的图优化问题中的适用性与潜在优势。

代 码

[GitHub] (<https://github.com/Hyelow0/RL-Total-Dominating-main>)

参考文献

- [1] Ore, O. (1962) Theory of Graphs. American Mathematical Society Colloquium Publications.
- [2] Allesina, S. and Bodini, A. (2004) Who Dominates Whom in the Ecosystem? Energy Flow Bottlenecks and Cascading Extinctions. *Journal of Theoretical Biology*, **230**, 351-358. <https://doi.org/10.1016/j.jtbi.2004.05.009>
- [3] Chen, J., He, K., Du, R., Zheng, M., Xiang, Y. and Yuan, Q. (2015) Dominating Set and Network Coding-Based Routing in Wireless Mesh Networks. *IEEE Transactions on Parallel and Distributed Systems*, **26**, 423-433. <https://doi.org/10.1109/tpds.2013.303>
- [4] Haynes, T.W., Hedetniemi, S.M., Hedetniemi, S.T. and Henning, M.A. (2002) Domination in Graphs Applied to Electric Power Networks. *SIAM Journal on Discrete Mathematics*, **15**, 519-529. <https://doi.org/10.1137/s0895480100375831>
- [5] Jiang, P., Liu, J., Wu, F., Wang, J. and Xue, A. (2016) Node Deployment Algorithm for Underwater Sensor Networks Based on Connected Dominating Set. *Sensors*, **16**, Article 388. <https://doi.org/10.3390/s16030388>
- [6] Abu-Khzam, F.N. (2022) An Improved Exact Algorithm for Minimum Dominating Set in Chordal Graphs. *Information Processing Letters*, **174**, Article ID: 106206. <https://doi.org/10.1016/j.ipl.2021.106206>
- [7] Haynes, T.W., Hedetniemi, S.T. and Slater, P.J. (1998) Fundamentals of Domination in Graphs, Monogr. CRC Press.
- [8] Alzoubi, K.M., Wan, P. and Frieder, O. (2003) Maximal Independent Set, Weakly-Connected Dominating Set, and Induced Spanners in Wireless AD HOC Networks. *International Journal of Foundations of Computer Science*, **14**, 287-303. <https://doi.org/10.1142/s012905410300173x>
- [9] Cockayne, E.J., Dawes, R.M. and Hedetniemi, S.T. (1980) Total Domination in Graphs. *Networks*, **10**, 211-219. <https://doi.org/10.1002/net.3230100304>
- [10] Haynes, T.W., Hedetniemi, S.T. and Henning, M.A. (2020) Topics in Domination in Graphs. *Developments in Mathematics*, Vol. 64. Springer. <https://doi.org/10.1007/978-3-030-51117-3>
- [11] Haynes, T.W., Hedetniemi, S.T. and Henning, M.A. (2021) Structures of Domination in Graphs. *Developments in Mathematics*, Vol. 66. Springer. <https://doi.org/10.1007/978-3-030-58892-2>
- [12] Haynes, T.W., Hedetniemi, S.T. and Henning, M.A. (2023) Domination in Graphs: Core Concepts. *Springer Monographs in Mathematics*. Springer. <https://doi.org/10.1007/978-3-031-09496-5>
- [13] Henning, M.A. and Yeo, A. (2013) Total Domination in Graphs: Springer Monographs in Mathematics. Springer. <https://doi.org/10.1007/978-1-4614-6525-6>
- [14] Corso, G., Stark, H., Jegelka, S., Jaakkola, T. and Barzilay, R. (2024) Graph Neural Networks. *Nature Reviews Methods Primers*, **4**, Article No. 17. <https://doi.org/10.1038/s43586-024-00294-7>
- [15] Neftci, E.O. and Averbek, B.B. (2019) Reinforcement Learning in Artificial and Biological Systems. *Nature Machine Intelligence*, **1**, 133-143. <https://doi.org/10.1038/s42256-019-0025-4>
- [16] Kool, W., Van Hoof, H. and Welling, M. (2019) Attention, Learn to Solve Routing Problems! *ICLR 2019: International*

-
- Conference on Learning Representations*, New Orleans, 6-9 May 2019.
- [17] Khalil, E., Dai, H.J., Zhang, Y.Y., Dilkina, B. and Song, L. (2017) Learning Combinatorial Optimization Algorithms over Graphs. *Advances in Neural Information Processing Systems*, **30**, 6348-6358.
- [18] Guo, W., Xu, Y. and Jin, Y. (2023) Swap-Based Deep Reinforcement Learning for Facility Location Problems in Networks. arXiv: 2312.15658.
- [19] Chlebik, M. and Chlebiková, J. (2008) Approximation Hardness of Dominating Set Problems in Bounded Degree Graphs. *Information and Computation*, **206**, 1264-1275. <https://doi.org/10.1016/j.ic.2008.07.003>
- [20] Zhu, J. (2009) Approximation for Minimum Total Dominating Set. *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human*, Seoul, 24-26 November 2009, 119-124. <https://doi.org/10.1145/1655925.1655948>
- [21] Alipour, S., Futuhi, E. and Karimi, S. (2020) On Distributed Algorithms for Minimum Dominating Set Problem, from Theory to Application. arXiv: 2012.04883.
- [22] Belhou, Y., Yahiaoui, S. and Kheddouci, H. (2014) Efficient Self-Stabilizing Algorithms for Minimal Total K-Dominating Sets in Graphs. *Information Processing Letters*, **114**, 339-343. <https://doi.org/10.1016/j.ipl.2014.02.002>
- [23] Bello, I., Pham, H., Le, Q.V., et al. (2016) Neural Combinatorial Optimization with Reinforcement Learning. arXiv: 1611.09940.
- [24] Applegate, D.L., Bixby, R.E., Chvátal, V. and Cook, W.J. (2006) *The Traveling Salesman Problem: A Computational Study*. Princeton University Press.
- [25] Perboli, G. and Rosano, M. (2019) Parcel Delivery in Urban Areas: Opportunities and Threats for the Mix of Traditional and Green Business Models. *Transportation Research Part C: Emerging Technologies*, **99**, 19-36. <https://doi.org/10.1016/j.trc.2019.01.006>
- [26] Li, Y., Chu, F., Feng, C., Chu, C. and Zhou, M. (2019) Integrated Production Inventory Routing Planning for Intelligent Food Logistics Systems. *IEEE Transactions on Intelligent Transportation Systems*, **20**, 867-878. <https://doi.org/10.1109/tits.2018.2835145>
- [27] Brouer, B.D., Alvarez, J.F., Plum, C.E.M., Pisinger, D. and Sigurd, M.M. (2014) A Base Integer Programming Model and Benchmark Suite for Liner-Shipping Network Design. *Transportation Science*, **48**, 281-312. <https://doi.org/10.1287/trsc.2013.0471>
- [28] Golden, B., Bodin, L., Doyle, T. and Stewart, W. (1980) Approximate Traveling Salesman Algorithms. *Operations Research*, **28**, 694-711. <https://doi.org/10.1287/opre.28.3.694>
- [29] Chen, M., Liu, S. and He, W. (2024) Learn to Solve Dominating Set Problem with GNN and Reinforcement Learning. *Applied Mathematics and Computation*, **474**, Article ID: 128717. <https://doi.org/10.1016/j.amc.2024.128717>
- [30] Wang, Q. and Tang, C. (2021) Deep Reinforcement Learning for Transportation Network Combinatorial Optimization: A Survey. *Knowledge-Based Systems*, **233**, Article ID: 107526. <https://doi.org/10.1016/j.knosys.2021.107526>
- [31] Wang, D. (2023) Reinforcement Learning for Combinatorial Optimization. In: Wang, J., Ed., *Encyclopedia of Data Science and Machine Learning*, IGI Global Scientific Publishing, 2857-2871. <https://doi.org/10.4018/978-1-7998-9220-5.ch170>