



正线性约束组合优化问题的非自回归学习求解

汪润中^{1,2}, 郇洋^{1,2}, 严骏驰^{3,1,2*}, 杨小康²

1. 上海交通大学计算机科学与工程系, 上海 200240

2. 上海交通大学人工智能教育部重点实验室, 上海 200240

3. 上海交通大学人工智能学院, 上海 200030

* 通信作者. E-mail: yanjunchi@situ.edu.cn

收稿日期: 2023-09-17; 修回日期: 2024-03-08; 接受日期: 2024-06-17; 网络出版日期: 2024-09-30

国家自然科学基金委重大研究计划重点项目 (批准号: 92370201)、国家自然科学基金委优秀青年基金项目 (批准号: 62222607)、国家自然科学基金委专项项目 (批准号: 72342023) 和上海市市级科技重大专项项目 (批准号: 2021SHZDZX0102) 资助

摘要 组合优化问题的求解是计算机科学、应用数学等学科共同研究的基础性问题. 其固有的计算复杂性为精确求解带来了挑战. 而采用深度神经网络进行求解已经成为一个前沿的研究方向. 本文设计了一种能够求解正线性约束组合优化问题的非自回归式神经网络. 本文方法的优势在于, 正线性约束代表了一大类组合优化问题, 突破了现有非自回归网络的通用性瓶颈; 与目前常用的自回归网络相比, 非自回归网络具有高效性、排列不变性等优势; 在神经网络框架中, 本文采用的离线无监督学习对标注的需求低, 无需求解最优解进行监督训练; 本文提出的在线可微分搜索方法显著提升了神经网络求解器的泛化能力. 本文在设施布局、最大集合覆盖、旅行商问题等代表性的组合优化问题中验证了非自回归求解器的有效性. 特别是在综合考虑求解效率和求解效果时, 非自回归网络求解器持平甚至超越了 SCIP, Gurobi 等开源或者商用的主流传统求解软件.

关键词 组合优化, 深度学习, 非自回归网络, 图神经网络, 梯度优化

1 引言

组合优化是一类具有离散决策空间的非凸优化问题的总称, 因其决策空间通常随着参数量的增加“指数爆炸”而得名. 作为计算机科学、应用数学、管理学等共同关注的问题, 组合优化问题的研究具有悠久的历史, 例如数学家 Euler^[1] 在 18 世纪开创了对图论问题的研究, Birkhoff^[2], von Neumann^[3], Dantzig^[4] 等现代计算机理论和应用数学的先驱在 20 世纪 40~50 年代系统地研究了组合优化问题. 目前, 多数研究者们已经达成共识: 除了少数“简单”的问题外, 多数组合优化问题具有 NP 难的性质, 即无法在多项式时间内精确求解. 不过, 在理论研究之外, 随着算力的发展、机器学习等技术的成熟, 研究者们依然可以提升现有求解算法的性能边界, 使得该方向的研究历久弥新.

引用格式: 汪润中, 郇洋, 严骏驰, 等. 正线性约束组合优化问题的非自回归学习求解. 中国科学: 信息科学, 2024, 54: 2368-2384, doi: 10.1360/SSI-2023-0269
Wang R Z, Li Y, Yan J C, et al. Learning to solve combinatorial optimization under positive linear constraints via non-autoregressive neural networks (in Chinese). Sci Sin Inform, 2024, 54: 2368-2384, doi: 10.1360/SSI-2023-0269

特别地,图灵奖得主、蒙特利尔大学 (Université de Montréal) 的 Bengio [5] 指出,拟合和求解特定分布下的组合优化问题往往比开发一个通用求解器更为容易.与通常运行在 CPU 上的传统求解算法相比 [6,7],运行在 GPU 上的神经网络算法可以充分发挥其算力优势.此外,由于神经网络可由数据驱动,擅长拟合数据分布,神经网络具有作为组合优化求解器的巨大潜力.然而,在实践层面,如何设计一个性能优秀的神经网络求解器依然是一个开放性问题 [8,9].其中的主要困难是如何让神经网络的输出满足特定的离散约束.一种常见的处理方法是构造自回归式 (autoregressive) 的神经网络求解器,将单个解的输出拆分为多步决策,在每一步通过规则控制可行的动作,使得最终解落在原问题的可行域中 [10,11].这种自回归式网络的通用性使其在大量组合优化问题的求解中有了广泛应用,如文献 [12~14] 等;此外,这种多步决策过程天然地可以建模成马尔可夫 (Markov) 决策过程,进而可以用强化学习算法进行训练,无需最优解作为监督信号.由于具备上述优点,自回归式范式仍然是目前研究的主流.然而,过长的决策序列可能会造成误差累积,也容易在训练阶段产生动作空间过大、奖励信号稀疏等问题 [15].此外,序列化求解过程通常更难建模问题和解空间内在的排列不变性、对称性等性质.

而本文则诉诸非自回归 (non-autoregressive) 网络架构作为神经组合优化求解器.非自回归网络的特点是只需一次前向计算即可获得完整的决策变量,规避了现有自回归网络中误差累积、奖励信号稀疏、排列不变性难以刻画等问题.此外,非自回归网络是一个成熟架构,其速度、精度已在计算机视觉等领域得到了广泛验证 [16~18].最新的理论研究也阐明了神经网络 (特别是通过随机特征破坏了对称性的图神经网络) 能够学习以混合整数规划为代表的组合优化问题 [19].即便存在理论优势,非自回归网络尚未得到充分研究的原因在于其技术难点:神经网络的输出通常是无约束的,但组合问题通常要求解网络的输出满足特定的约束 (如回路/匹配关系等).若设计神经网络在训练过程中自动地学习约束,在实际中会造成泛化性不足 [20];而使用一般的投影算法将输出投影到可行域,则会导致梯度被截断,无法端到端训练.

现有的非自回归式神经求解器 [20,21] 的主要缺陷是通用性不足,其原因是在非自回归网络的框架下可处理的约束有限:文献 [21] 的惩罚项技术缺乏泛化性,而文献 [20] 的 Sinkhorn 网络层只能处理基数约束.为提升非自回归式网络的通用性,需要一种可微分且通用的约束网络层.为此,本文引入了文献 [22] 提出的线性可满足网络 (linear satisfiability network, LinSATNet),其特点是能够显式地将任意神经网络输出投影到正线性约束 (positive linear constraints) 构成的可行域¹⁾:

$$\mathbf{Ax} \leq \mathbf{b}, \mathbf{Cx} \geq \mathbf{d}, \mathbf{Ex} = \mathbf{f}, \quad \text{where } \mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{d}, \mathbf{E}, \mathbf{f} \geq 0, \mathbf{x} \in [0, 1]^l. \quad (1)$$

这一形式可以通用地表示常见的针对决策变量的约束.例如,对于背包问题,设 $x_i = 0$ 或 1 分别代表选择不选择第 i 项物品, w_i 代表第 i 项物品的重量, m 代表重量上限,背包问题的约束可写作 $\mathbf{w}^T \mathbf{x} \leq m$,可由式 (1) 中 “ $\mathbf{Ax} \leq \mathbf{b}$ ” 一项表示.实际上,本文引入的 LinSAT 网络层显著地提升了非自回归网络的通用性,使得非自回归式网络 (在原理上) 可以处理任意具有正线性约束的组合优化问题,包括路径、匹配、排列、覆盖、布局等典型的组合优化问题及其变种.基于本文的神经网络设计框架,非自回归式网络的通用性可以和目前最常用的自回归式网络相当.

如图 1 所示,本文提出了一种非自回归网络范式,用于高效地求解组合优化问题.从问题参数、约束条件等出发,首先通过图建模,将问题转化为神经网络可处理的图结构形式 [11].随后采用图神经网络

1) 更一般的非正线性约束超过了本文范围,且属于未解决的开放性问题.事实上,正线性约束已经可以覆盖大量典型图论与组合问题.例如,路径问题、匹配问题、投资组合优化问题及其变种 (文献 [22] 已论证),及本文实验中论证的集合覆盖问题、设施布局问题等.

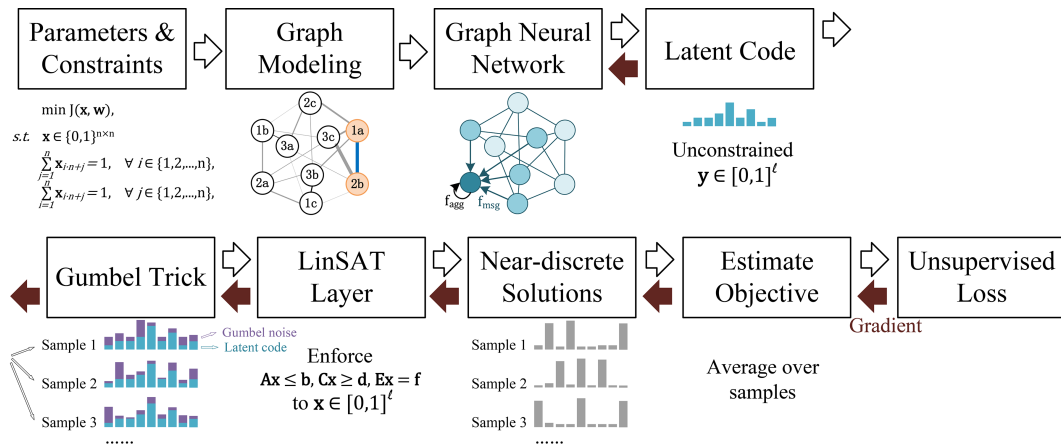


图 1 (网络版彩图) 非自回归神经网络求解组合优化问题之范式 (训练阶段)

Figure 1 (Color online) Non-autoregressive neural network paradigm proposed to solve combinatorial optimization problems (in training)

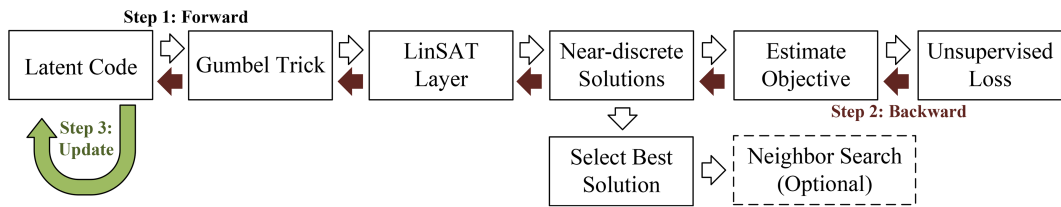


图 2 (网络版彩图) 非自回归神经网络求解器中基于梯度的搜索步骤 (推理阶段)

Figure 2 (Color online) Gradient-based optimization steps in non-autoregressive neural network combinatorial optimization solvers (inference)

络模型^[23, 24], 输出的隐变量与决策变量维度相同. 随后采用 Gumbel 技巧^[25], 并通过 LinSAT 层^[22]引入式 (1) 中的正线性约束, 将隐变量投影到多个准离散的可行解 (多个解来自于多次的 Gumbel 采样). 其中, 添加 Gumbel 噪声可以使网络输出更加离散^[20] (同时通过重参数化技巧^[25]保留梯度), 本文将这一技术推广到了任意正线性约束的组合优化问题. 随后, 通过可微分的方式依次计算每个采样的目标函数. 由于所有模块都支持梯度反传, 此时平均的目标函数值可直接视作无监督的损失函数 (假设是最小化问题), 在训练阶段进行高效训练. 在此非自回归网络的范式下, 隐空间内的每一个点都对应了一组可行解的分布, 神经网络的预测可视为隐空间中的初始点. 在推理阶段, 如图 2 所示, 针对具体问题, 可基于梯度在连续隐空间进行在线搜索, 以获得更好的解.

采用本文提出的非自回归网络求解组合优化问题具有以下优势:

- 本文的方法具有通用性. 本文采用了 LinSAT 技术使组合优化求解网络的输出能够满足特定的正线性约束, 因此能处理所有具有正线性约束的组合优化问题. 实际上, 这已经覆盖了大量经典的组合优化问题及其变种, 且显著地拓宽了非自回归式求解网络原本极有限的应用范围.

- 非自回归网络具有高效性、排列不变性等优势. 设决策变量的维度是 l , 现有的自回归式网络通常需要 $\mathcal{O}(l)$ 次神经网络推理才可得到一个解, 而非自回归网络只需要 $\mathcal{O}(1)$ 次. 更高的并行度可带来更高的性能上限. 此外, 排列不变性 (即输入变量随机排列后输出不变) 是组合优化问题的内在性质, 自回归式网络难以建模这种重要性质, 但非自回归式网络天然可以.

- 离线的无监督学习降低了对标注的需求. 在组合优化中若要进行传统的监督学习, 则需要事先求解大量 NP 难问题的近似解乃至最优解作为监督信息. 在训练阶段, 由于本文的非自回归网络完全可微分, 批处理内目标函数的均值可作为无监督的损失函数, 无需精确求解大量 NP 难问题.

- 在线的可微分搜索提升了泛化能力. 本文提出, 在推理阶段可以通过梯度在隐空间搜索以获得更好的解, 如图 2 所示. 组合优化方向最新的研究工作表明, 采样、搜索等步骤对神经网络求解器性能的贡献不容忽视^[26,27]. 可微分的非自回归网络求解器天然地支持梯度优化, 可以在连续的隐空间搜索更优质的解; 此外, 还可复用最新的深度学习框架, 使可微分搜索更加高效.

为了验证非自回归神经网络的求解性能, 本文在(有/无上限的)设施布局、最大集合覆盖、旅行商问题等代表性的组合优化问题上实现了神经网络求解器. 旅行商问题的实验验证了本文框架下设计的神经网络求解器与当下其他神经网络求解器性能持平甚至更好. 在另外两类问题的实验中, 在给定相同的求解时间条件下比较求解效果, 本文设计的神经网络求解器持平甚至超越了 SCIP^[28], Gurobi^[29]等专业求解软件. 本文的训练、测试代码开源在 <https://github.com/Thinklab-SJTU/NAR-CO-Solver>.

2 相关研究工作

本文遵循文献[30], 将神经组合优化求解器分为自回归式、非自回归式. 本节将简要讨论这两类方法, 读者可自行参阅文献[30,31]. 另有一类工作聚焦于融合神经网络与现有的求解算法^[32~35]. 这类融合方法通常需要针对特定求解器定制, 不如其他两类方法灵活, 故本文不展开讨论.

组合优化问题的自回归学习求解. 自回归式 (autoregressive) 的神经网络广泛应用于时序学习, 其中 $(t+1)$ 时刻的输出依赖于 t 时刻. 在组合优化领域, 自回归式的神经网络求解器以序列的形式逐步构造一个完整解. 因此, 对于决策变量维度为 l 的问题, 自回归式的神经网络共需要输出 $\mathcal{O}(l)$ 次. 自回归网络的优势在于可在序列的每一步人为地限制动作空间, 剔除不符合约束的解. 这也是现代神经网络首次应用于组合优化求解时所采用的范式^[10]. 随后, 戴涵俊等^[11]发现这种多步决策天然地可以写成一个马尔可夫决策过程, 进而采用强化学习算法进行训练^[36]. 得益于其普适性, 这种“自回归式网络 + 强化学习”的范式成为了研究的主流, 它被广泛应用于求解车间作业调度^[37]、计算任务调度^[13]、二次指派^[38]、装箱^[39]等组合优化问题. 本文认为, 自回归式的网络容易在长决策序列中累积误差, 丧失的排列不变性会制约网络的求解性能和泛化能力, 同时动作空间过大、奖励信号稀疏等挑战也可能影响强化学习的性能.

组合优化问题的非自回归学习求解. 很多组合优化问题本身设定往往是离线的, 与时序无关. 因此, 在其他机器学习任务中被广泛采用的非自回归网络似乎是一种更自然的选择. 早在 20 世纪 80 年代, 研究者就提出了采用 Hopfield 网络求解旅行商问题^[40]. 该工作利用神经网络和梯度直接针对特定问题进行搜索, 一定程度上也印证了本文方法支持在线搜索的优势. 而近期的工作^[21]提出了一种比自回归网络更高效的非自回归网络, 其中约束被刻画为训练阶段的惩罚项. 然而, 这种惩罚项的做法很难严格保证约束, 且在训练集上学到的约束信息无法泛化到测试集^[20]. 文献[20]进一步推导了不同的网络设计方式所对应约束违反量的理论上界, 并进行了实验验证. 实验结果证实, 约束违反量越高, 神经网络求解器的性能越差. 本文认为, 支持在线搜索是非自回归式神经网络的一大优势, 它允许神经网络求解器在测试阶段通过梯度方向寻找更好的解. 邱睿中等^[26]的研究工作恰好印证了梯度搜索在求解组合优化问题时的优势. 不过, 他们在约束处理阶段采用了方差更高的 REINFORCE 技术^[41], 与本文采用的 Gumbel 重参数化与 LinSAT 的组合有所不同.

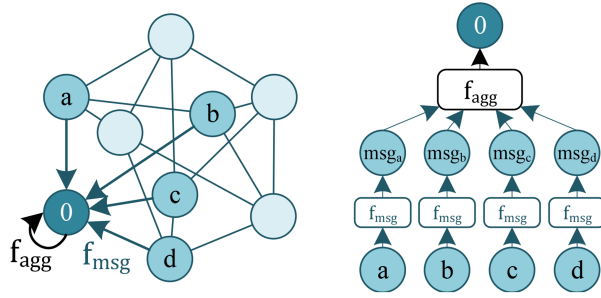


图 3 (网络版彩图) 图神经网络的一般框架

Figure 3 (Color online) General framework of graph neural networks

3 非自回归式神经组合优化求解器

3.1 问题形式

本文提出的非自回归网络框架可以求解具有如下正线性约束的二元组合优化问题:

$$\min_{\mathbf{x}} J(\mathbf{x}, \mathbf{w}), \quad (2a)$$

$$\text{s.t. } \mathbf{Ax} \leq \mathbf{b}, \mathbf{Cx} \geq \mathbf{d}, \mathbf{Ex} = \mathbf{f}, \mathbf{x} \in \{0, 1\}^l, \quad (2b)$$

其中, \mathbf{x} 是决策变量, 又称为问题的解; \mathbf{w} 是问题参数; $J(\mathbf{x}, \mathbf{w})$ 是优化的目标函数 (若为最大化问题, 取反即可); 约束条件中 $\mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{d}, \mathbf{E}, \mathbf{f}$ 中的元素必须大于等于 0, 三组约束条件不必同时存在. 具体而言, 组合优化问题的输入是 \mathbf{w} 和约束条件, 输出是在限定时间内 (尽可能) 最小化 $f(\mathbf{x}, \mathbf{w})$ 的解 \mathbf{x} .

3.2 框架组件

本小节将逐个介绍图 1 所示的非自回归组合优化求解框架中的组件.

图建模. 神经网络难以直接处理公式 (2) 中数学形式的输入, 一种通用的解决方案是将问题参数进行图建模. 针对不同问题的图建模方法各有不同, 但存在一条最优实践: 决策变量需等价于在图上进行点分类或边分类. 针对单个图上的组合优化问题 (例如图割、节点覆盖、旅行商等), 图建模的方法较为直观 [11, 32, 42]; 若同时存在两个图 (例如图匹配), 则可以采用“伴随图”的形式构建融合两个图结构的大图 [43]; 对于布尔可满足性问题, 也存在对应的图建模方法. 例如将合取范式等价于二分图 [44]; 对于一般的矩阵形式 (如线性整数规划), 目前常见的处理方式是将“ $\mathbf{Ax} \leq \mathbf{b}$ ”等约束条件和线性的问题参数一起构成一个二分图 [35]. 综上所述, 对多数组合优化问题而言, 均存在其某种合理的图建模方式. 读者可参阅上述文献, 完成对特定问题的图建模.

图神经网络. 图上的点分类或边分类问题天然地可以采用图神经网络进行处理 [24]. 图神经网络的变种很多, 用于点分类的图神经网络可以采用如图 3 所示的一般框架概括 [45]: 在图神经网络的每一层, 当节点 0 需要更新时, 只需考虑其邻居节点 a, b, c, d. 对于每个邻居节点, 由一个消息函数 (f_{msg} , 通常是多层神经网络) 将其当前的节点特征变换得到消息 (例如 msg_a). 随后, 通过聚合函数 (f_{agg} , 可以是求平均、求和或注意力机制 [46]) 将所有邻居节点的消息更新到当前节点. 每一层图神经网络中 $f_{\text{msg}}, f_{\text{agg}}$ 的权重在所有节点之间共享. $f_{\text{msg}}, f_{\text{agg}}$ 的不同选择即对应了不同的图神经网络, 常用的包括 GCN [24], GraphSage [47], GIN [48] 等. 对于边分类问题, 可以将特征经邻居节点传递到边, 或在偶图上进行节点分类.

隐变量. 以点分类建模的问题为例, 设决策变量的维度是 l , 使用 Sigmoid 激活函数的图神经网络将输出一个维度同样为 l 的 $[0, 1]$ 向量, 本文称之为隐变量. l 维的实数空间构成了一个隐空间, 其中每个点都对应了一组可行解. 这一对应关系由下文中介绍的 Gumbel 重参数化和 LinSAT 约束层得到. 特别注意的是, 隐空间是一个连续域; 通过在隐空间上搜索、优化, 即可寻找更优质的决策变量. 隐变量的取值范围限制在 $[0, 1]$ 可避免隐空间过大, 造成搜索步骤的挑战.

Gumbel 重参数化. 该步骤的目的是将隐变量看作可行解的一个分布, 通过在该分布上进行离散的采样以获得可行解. 但是, 离散采样的步骤是不可导的, 因此本文使用了近似离散采样常用的 Gumbel 重参数化技巧^[25,49,50]. 给定 Gumbel 分布 $g_\sigma(u) = -\sigma \log(-\log(u))$, 其中 σ 控制方差、 u 采样自 $(0, 1)$ 均匀分布, 将其作用于图神经网络的输出 $\mathbf{y} \in \mathbb{R}^l$ 上,

$$\tilde{\mathbf{y}} = [y_1 + g_\sigma(u)_1, y_2 + g_\sigma(u)_2, \dots, y_l + g_\sigma(u)_l]^T, \quad (3)$$

其中所有 $g_\sigma(u)_i$ 独立同分布. 为了更好地刻画可行解分布, 需基于式 (3) 重复采样. 采样可通过批处理在 GPU 上并行. 采样次数越多, 对分布的估计更准确, 但开销更大.

LinSAT 约束层. Gumbel 重参数化后, 需要将 $\tilde{\mathbf{y}}$ 投影到约束空间. 为了使框架尽可能通用, 本文采用了目前精确梯度方法中能处理约束种类最多的 LinSAT 层^[22]. 不失一般性, 针对采样 $\tilde{\mathbf{y}}$,

$$\mathbf{x} = \text{LinSAT}(\tilde{\mathbf{y}}, \mathbf{A}, \mathbf{b}, \mathbf{C}, \mathbf{d}, \mathbf{E}, \mathbf{f}), \quad (4)$$

得到的向量 $\mathbf{x} \in [0, 1]^l$ 即为符合约束的准离散解 (即数值接近离散, 但本质上仍为连续且具有梯度的解). 一方面, 为了保证网络的可微分性, 我们不能简单地将网络的输出离散化; 另一方面, 为了使后续的目标函数估计尽可能准确, 我们又希望网络输出尽可能接近离散. 本文采取的 Gumbel 重参数化技巧保证了神经网络的输出在不截断梯度的前提下是准离散的, Gumbel 分布的系数 σ 越大输出越接近离散 (代价是迭代轮数增加). LinSAT 层保证了网络输出满足给定的问题约束.

LinSAT 的原理基于最优传输理论. 最优传输研究在给定距离矩阵下, 如何以最小的代价将边缘分布 \mathbf{u} 变换到另一个边缘分布 \mathbf{v} . 通过合理地构造边缘分布, 我们发现单条线性约束的投影过程可以形式化成一个最优传输问题. 例如, 对于约束 $\mathbf{a}^T \mathbf{x} \leq b$ (变量维度为 l), 一个满足如下分布的最优传输矩阵 Γ 即为一个满足该约束的投影 (δ 均为虚拟变量):

$$\Gamma = \begin{bmatrix} \boxed{x_1 \ x_2 \ \cdots \ x_l} & \delta_0 \\ \delta_1 & \delta_2 \ \cdots \ \delta_l \ \delta_{l+1} \end{bmatrix}; \quad \mathbf{u} = \underbrace{[a_1 \ a_2 \ \cdots \ a_l \ b]}_{\substack{l \text{ 维变量} \\ +1 \text{ 维虚拟变量}}}, \quad \mathbf{v} = \begin{bmatrix} b \\ \sum_{i=1}^l a_i \end{bmatrix}, \quad (5)$$

其中最优传输保证了 Γ 的行和等于 \mathbf{v} , 列和等于 \mathbf{u} . 对于另外两种正线性约束 $\mathbf{c}^T \mathbf{x} \geq d$ 和 $\mathbf{e}^T \mathbf{x} = f$, 同理亦可构造最优传输形式. 因为最优传输问题存在一种近似程度可控的可微分求解算法^[51], 它与神经网络天然契合. 接下来的挑战在于如何同时处理多组约束. 我们发现, 最优传输算法可以扩展到同时处理多组边缘分布, 并保持可微性和收敛性, 这就是 LinSAT 算法. 换言之, 对于具有多条正线性约束的组合优化问题, 可以求解一个具有多组边缘分布的最优传输问题, 即调用 LinSAT, 它的输出 \mathbf{x} 是输入向量 $\tilde{\mathbf{y}}$ 在约束范围内的投影. LinSAT 的详细原理和收敛性证明可参见文献^[22].

目标函数估计. 本文所述的非自回归式神经网络的优势之一是可以端到端地估计目标函数, 将其作为损失函数指导网络的训练 (假设是最小化问题):

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}^l, \mathbf{I}^{l \times l})} [J(f(\mathbf{u}, \mathbf{y}_\theta), \mathbf{w})], \quad (6)$$

其中 \mathbf{y}_θ 是图神经网络的输出, $f(\cdot)$ 代表了 Gumbel 重参数化 (u 采样自 l 维独立同分布的标准正态分布) 和 LinSAT 层 (无参数). 式 (6) 计算的是 Gumbel 分布下的平均目标函数, 因为 Gumbel 重参数化和 LinSAT 层共同保证了其输出是准离散、满足约束的解, 这一估计比直接使用神经网络输出进行估计更精确 (3.3.2 小节将对此详细讨论). 在得到 $\mathbf{x} = f(u, \mathbf{y}_\theta)$ 后, $J(\mathbf{x}, \mathbf{w})$ 的计算就相对直接. 例如, 对于线性目标函数 $J(\mathbf{x}, \mathbf{w}) = \mathbf{x}^T \mathbf{w}$, 直接计算即可. 但是, 部分组合优化问题的目标函数并不直观, 可能嵌套了 \min, \max 操作, 这应当在实践中予以避免. 例如, 第 4.1 小节考虑的设施布局问题中, 实际的目标函数是 $\sum_{j=1}^m \min(\{\Delta_{i,j} | \forall \mathbf{x}_i = 1\})$. 实验发现在端到端神经网络中使用 \min 算子会截断梯度, 造成效果不佳. 若将 \min 替换为温度为 β 的 softmin 操作, 当 β 设为合适的值时, 求解效果得到了显著提升. 本文总结了目标函数估计的最佳实践: 计算步骤需尽可能地保留梯度, 并将 \min, \max 算子替换为 $\text{softmin}, \text{softmax}$ 等柔性版本.

离线的无监督预训练. 可微分的非自回归网络支持无监督的预训练. 在训练阶段, 式 (6) 所示的目标函数均值可直接作为无监督的损失函数. 训练的前向传播如图 1 的白色箭头所示, 反向传播如图 1 的棕色箭头所示. 通过在训练阶段最小化该损失函数、更新图神经网络的权重, 图神经网络学习了如何将原始问题的图建模映射到一个较好的隐变量.

在线的梯度搜索和推理. 神经网络的单次输出通常无法获得足够满意的解, 一系列研究展示了基于神经网络输出进行优化和搜索的必要性 [21, 26, 27, 52~54]. 本文提出的非自回归网络范式可直接适配经典的集束搜索、蒙特卡洛 (Monte Carlo) 树搜索等方法; 更重要的是, 框架的可微分特性在隐空间内提供了优化目标函数所指向的梯度方向, 框架的可并行特性使得不论是前向计算还是梯度传播都可以在 GPU 上完成. 在测试阶段, 固定网络参数 θ , 式 (6) 可改写成

$$\mathcal{L}(\mathbf{y}) = \mathbb{E}_{u \sim \mathcal{N}(\mathbf{0}^l, \mathbf{I}^{l \times l})} [J(f(u, \mathbf{y}), \mathbf{w})]. \quad (7)$$

此时梯度反向传播不再更新网络参数 θ , 而是更新隐变量 \mathbf{y} . 这可以类比成一个新的神经网络训练问题, 其本质是利用框架的可微分特性通过梯度进行优化和搜索. 如图 2 所示, 针对单个问题实例, 可以通过这一机制搜索更好的隐变量. 相比于在完整数据集上的训练过程, 在单个问题实例上进行优化效率更高、收敛更快. 此外, 根据具体的问题背景知识, 还可设计专用的邻域搜索算法, 在每一轮搜索的最优解周围进行快速搜索. 实验表明, 与离线训练后的神经网络配合, 这种在线梯度搜索方法的效率可以比肩乃至超越 SCIP/Gurobi 等专业求解软件.

3.3 进一步讨论

3.3.1 图神经网络对组合优化问题的表达能力

图神经网络在提出伊始, 处理的是社交网络等“常见”的图结构. 不过, 目前对图神经网络表达力的理论研究多从以图同构为代表的图论组合问题出发. 主流观点认为, 图神经网络的表达力与 Weisfeiler-Lehman (WL) 图同构测试相当 [48, 55]. 虽然图同构问题也是一类组合优化问题, 但需要注意的是, WL 测试只是一种近似算法, 是图不同构的一种充分非必要条件. 图神经网络是否具有表达任意组合优化问题的能力依然是一个待研究的课题. 不过, 机器学习组合优化领域的实践经验说明图神经网络可以胜任这一角色 [11, 21, 35]. 并且, 最近的理论研究展示了正面的结果. 陈子昂等先后在线性规划问题 [56] 和混合整数规划问题 [19] 上证明了图神经网络在理论上具有足够的表达能力来预测问题的可行性、最优目标函数和最优解. 因此, 本文选择图神经网络作为主要的神经网络; 当然, 本文的设计并不依赖于具体的网络结构, 本框架亦可迁移到其他表达力更强的神经网络上.

3.3.2 Gumbel 重参数化技巧的两方面优势

本文采用了 Gumbel 重参数化技巧^[25]实现(近)离散空间的微分采样,具有两方面优势。

首先,具有 Gumbel 重参数化的组合优化求解网络在求梯度时更加精确。组合优化问题的一大特征是决策变量的可行域是非连续的,而这与神经网络的连续性质冲突。非连续的可行域为端到端求梯度带来了挑战:若神经网络的输出是离散的决策变量,此时输入-输出映射是一个分段函数^[57]。分段函数在跳变时,其梯度是冲激函数,它无法为神经网络提供有意义的信息。将约束的离散部分完全舍弃则可以保留梯度,但此时训练阶段考虑的问题会退化为一个松弛后的连续优化问题。很多时候,这是一个更简单的凸优化问题,和我们实际考虑的组合优化具有本质不同,在凸优化问题上训练的神经网络通常无法泛化到更难的组合优化问题。因此,引入 Gumbel 重参数化可以在保留梯度的前提下使得神经网络的输出尽可能地接近离散域,更契合实际的组合优化问题。这种情况下,通过 Gumbel 重参数化技巧求得的梯度更加精确。

其次,具有随机性的 Gumbel 重参数化方法实现了推理阶段的并行采样。在本文中,每个隐变量都对应了一组离散解的分布。在推理阶段, Gumbel 重参数化实际上在这个分布中执行了随机采样。特别地,更精确的梯度使得优化过程中隐变量的更新速度更快, Gumbel 重参数化的随机性使网络有更大概率找到更好的解。更重要的是, Gumbel 重参数化的采样过程是可并行的,通过批处理操作即可在 GPU 上批量执行,其效率比传统的采样、搜索算法更高^[43,58]。

3.3.3 对高效深度学习框架的复用

深度学习近年的成功很大程度上得益于 GPU 算力的发展以及成熟的开源深度学习框架(特别是深度学习框架高性能的底层实现)。本文提出的非自回归网络求解框架取得成功的内在原因之一即是复用了高效率的深度学习框架。凭借深度学习框架对 GPU 的良好支持和高效率底层代码, Gumbel 采样、多个准离散解的目标函数估计等操作可以在 GPU 上并行地执行。此外,深度学习框架也可以看作一种高效率的大规模梯度优化算法。本文复用了深度学习框架中自动微分的功能,特别是推理阶段通过自动求导为隐变量更新计算梯度,实现了在线的可微分搜索。

3.3.4 从元学习视角看待组合优化求解

邱睿中等^[26]首次提出了以元学习的视角(准确地说是模型无关的元学习, Model-Agnostic Meta-Learning^[59])看待组合优化求解:预训练阶段,神经网络会拟合到一个对完整数据集有效的初始权重;在具体的组合优化问题上,可以进一步更新神经网络权重,得到更好的解。本文提出的框架亦可采用类似的元学习视角理解,不过元学习的对象从神经网络权重转换成了更轻量化的隐变量:预训练阶段,神经网络的输出会拟合到一个对完整数据集有效的初始隐变量;在具体的组合优化问题上,通过梯度直接更新隐变量,得到更好的解。

4 实验和分析

本文以设施布局、集合覆盖和旅行商问题为例,与现有神经网络方法^{[20,21]2)}、专业求解软件^[28,29]进行了对比,验证了非自回归网络求解方法的有效性。其中,进行设施布局与集合覆盖问题实验的主

2) 在设施布局与集合覆盖问题中,本文比较的神经网络方法均为非自回归式网络,在本文的数据集上均可在单 GPU、一小时内训练完成。考虑到自回归式网络显著更高的训练代价(通常以天计),本文未与它们进行直接比较,而是比较了在实际中更常用的专业求解软件。

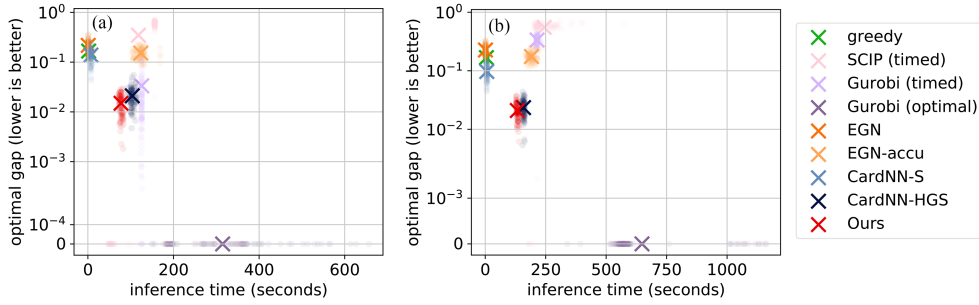


图 4 (网络版彩图) 在无服务能力上限的设施布局问题上, 比较求解器相对最优解的差距

Figure 4 (Color online) Comparison on the optimal gap of the solvers on the uncapped facility location problem. (a) FLP ($k = 30, m = 500$); (b) FLP ($k = 50, m = 800$)

要目的是将非自回归神经网络的求解效果直接对比当前的默认解决方案——整数规划求解器; 旅行商问题的深度学习求解方法研究较充分^[11,26,60,61], 旅行商问题实验的主要目的是比较本文的神经网络 (特别是结合了 LinSAT 层的非自回归网络) 与其他最先进的神经网络方法. 设施布局和集合覆盖实验在一台配备 i7-9700K CPU, 16 GB 内存, RTX 2080Ti GPU 的工作站上完成, 旅行商问题实验在另一台配备 AMD 3970X CPU, 32 GB 内存, RTX 3090 GPU 的工作站上完成.

4.1 设施布局问题

本文考虑如下的设施布局问题 (facility location problem, FLP): 给定共 m 个地点, 我们要选择其中 k 个地点新建设施, 其中每个设施将 (在服务能力上限之内) 服务距离其最近的其他地点. 我们的优化目标是通过选择最优的 k 个地点新建设施, 最小化每个地点到其对应设施的距离之和. 特别地, 对于不考虑服务能力上限的简化形式, 其优化形式是

$$\min_{\mathbf{x}} \sum_{j=1}^m \min(\{\Delta_{i,j} \mid \forall x_i = 1\}), \quad \text{s.t.} \quad \mathbf{x} \in \{0, 1\}^m, \quad \sum_{i=1}^m x_i \leq k, \quad (8)$$

其中 \mathbf{x} 是决策变量, 问题参数 $\Delta_{i,j}$ 代表地点 i, j 之间的距离. 对于上述式 (8), 可在本文提出的非自回归网络范式下求解. 具体实现方式为

- 图建模步骤基于地点的坐标建图, 其中每个地点视为节点, 距离小于分布区域直径 2% 的点之间定义连边, 设施布局问题的求解等价于在这个图上进行节点分类;
- 图神经网络采用样条卷积网络^[62], 共 3 层, 隐层维度是 16, 样条核数目是 5;
- Gumbel 重参数化和 LinSAT 约束层能处理式 (8) 中正线性约束, 进而将输出投影到接近离散的可行域;

• 目标函数估计需要给定 \mathbf{x} 计算式 (8). 由于式 (8) 中的 \min 算子会在大部分元素处截断梯度, 本文采用了更平滑的 softmin 算子 (即将向量取负后输入 softmax) 替代 \min , 以保证足够的可微性. 在实现上, 本文将 $\min(\{\Delta_{i,j} \mid \forall x_i = 1\})$ 替换为 $\text{softmax}(-\beta \Delta \circ \mathbf{x})$. 其中 \circ 代表逐元素乘, β 是温度参数.

- (推理阶段的) 邻域搜索采用了 k -中值 (k -median) 聚类, 在批中最好解的基础上进行快速搜索.

本文遵循文献 [20] 的实验设计, 在单位面积的二维平面上随机生成了 m 个坐标, 构成训练/测试集. 所有神经网络方法在训练阶段没有最优解相关信息. 实验中, 本文方法与贪心算法 (greedy)、专业求解软件 (SCIP^[28], Gurobi^[29], 图 4(a) 和 (b) 的求解限时分别是 120, 200 s)、神经网络求解方法 (EGN^[21], CardNN^[22]) 进行了比较, 结果如图 4 所示. 图 4 在求解耗时、求解性能两个维度比较了不同的求解

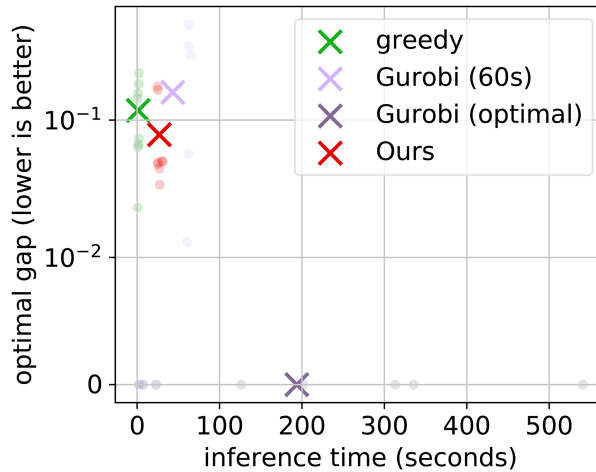


图5 (网络版彩图) 在真实数据上的泛化实验

Figure 5 (Color online) Generalization study in real-world dataset

方法, 其中求解性能由相对最优解的差距衡量 (越小越好). 其中, 最优解由无限求解时间的 Gurobi 软件通过分支定界法求解并证明其最优性, 相对最优解的差距定义为

$$\text{gap} = \frac{\text{目标函数} - \text{最优目标函数}}{\text{目标函数}}.$$

测试集中的每一个样例都是图中的一个圆点, 单个方法平均性能由叉号表示. 本文提出的非自回归网络求解方法 (红色点) 位于求解精度与求解耗时的帕累托最优 (Pareto optimal) 上, 即若要获得比本文更好的性能, 则必须花费更多的求解时间. 特别地, 本文方法在求解耗时和求解性能两方面均超过了目前最好的商业求解软件 Gurobi.

为了测试神经网络求解器在未见分布上的泛化能力, 本文构造了一个真实数据集: 将上海、纽约、伦敦、首尔所有星巴克门店的实际经纬度作为输入, 分别计算欧氏距离、曼哈顿距离, 求解对应的设施布局问题. 在实验中, 神经网络在仿真数据上训练, 在真实数据上直接测试. 图5的结果表明神经网络具有较强的泛化能力, 在训练过程中未见到的真实数据上, 其平均求解性能优于 Gurobi. 表1和2则面向本文的主要模块进行了消融实验, 一方面说明了在线梯度搜索和推理的重要性, 另一方面展示了设施布局问题上的 softmin 算子和超参数选择对神经网络求解性能的重要性.

具有服务能力上限的设施布局问题. 为进一步展示本文求解方法实用性, 本文还考虑了服务能力上限, 即每个新建设施只能服务有限的需求. 此时优化形式可写作:

$$\min_{\mathbf{x}} \min_{\mathbf{P}} \sum_{i=1}^m \sum_{j=1}^m \Delta_{i,j} P_{i,j}, \quad \text{s.t.} \quad \mathbf{x} \in \{0, 1\}^m, \sum_{i=1}^m \mathbf{x}_i \leq k, \sum_{i=1}^m \mathbf{x}_i \geq \sum_{i=1}^m d_i, \quad (9a)$$

$$\mathbf{P} \in \mathbb{R}_+^{m \times m}, \sum_{i=1}^m P_{i,j} = d_j, \sum_{j=1}^m P_{i,j} \leq \mathbf{x}_i. \quad (9b)$$

变量 $P_{i,j}$ 代表了设施 i 将服务多少来自地点 j 的需求. 在实现非自回归网络求解器时, 大部分网络模块可从求解式 (8) 的网络中直接借用. 其中的改动来自于式 (9) 的约束条件: 决策变量 \mathbf{x} 需满足服务能力上限, 同时要保证提供的服务总量大于总需求.

表 1 在无服务能力上限设施布局问题 ($k = 30, m = 500$) 上进行的消融实验: 比较不同 β 取值下, 求解器相对最优解的差距

Table 1 Ablation experiment on the optimal gap of the solvers with different β values on the uncapacitated facility location problem ($k = 30, m = 500$)^{a)}

	$\beta = 10$	$\beta = 20$	$\beta = 50$	$\beta = 100$	$\beta = 200$
Optimal gap	0.0711	0.0501	0.0149	0.0428	0.0606

a) Best performing entries are in bold.

表 2 固定 β , 研究其他参数对性能的影响

Table 2 Ablation study with fixed β ^{a)}

Gradient search	min/softmin	Gap
✓	min	0.1772
×	softmin ($\beta = 50$)	0.1753
✓	softmin ($\beta = 50$)	0.0149

a) Best performing entries are in bold.

表 3 具有服务能力上限的设施布局问题的结果

Table 3 Results on the capacitated facility location problem^{a)}

Objective score (\downarrow)	Gurobi	Ours
Time limit = 30 s	2.6454 \pm 0.1018	2.5026\pm0.0824
Time limit = 100 s	2.4524\pm0.0756	2.4797 \pm 0.0715

a) Best performing entries are in bold.

• LinSAT 约束层中, 离散决策变量 \mathbf{x} 的约束有所更新. 更新后的约束仍然属于正线性约束, 只需加上新的约束, 就可用 LinSAT 层处理 [63]3).

• 目标函数估计变得更复杂, 这是因为存在服务上限, 每个地点不能简单地选择前往离它最近的设施. 观察式 (9) 发现, 当决策变量的离散部分 \mathbf{x} 由网络预测后, 剩余部分是一个线性规划问题, 更具体而言是最优传输问题. 虽然直接求解线性规划问题理论上可导 [64], 但在本文的框架下, 需要批量求解同一个批次中的多个线性规划问题, 时间开销过大. 本文采用了可 GPU 并行的近似方法求解最优传输问题 [63], 并通过文献 [65] 技术进行求导, 相比自动微分求导速度更快、更节省显存.

实验采用了与图 4(a) 相同的数据分布, 每个设施最多可满足 50 个地点的需求. 实验比较了图 4 中表现最好的传统求解软件 Gurobi, 结果如表 3 所示. 在求解时间较短时 (约 30 s), 本文的非自回归网络求解方法比 Gurobi 更好; 在求解更长时间后 (约 100 s), Gurobi 的性能超过了本文方法, 但二者非常接近. 在这个问题上, 由于约束条件改变, 神经网络求解方法 CardNN [20] 不再适用; EGN [21] 和 SCIP [28] 方法由于在图 4 中性能不及最好的神经网络和传统算法, 简洁起见未列表 3.

作为总结, 在设施布局问题上, 本文的非自回归网络算法在适用范围和求解精度上超过了现有的 (非自回归) 神经网络求解方法, 求解性能与最先进的专业求解软件 Gurobi 相当. 在部分问题设定下, 非自回归网络的性能甚至可以超过 Gurobi.

3) 该问题同时出现了两组决策变量 \mathbf{x}, \mathbf{P} . 在本文的框架下, 神经网络只需预测离散的决策变量 \mathbf{x} , 因为 \mathbf{P} 可以在确定 \mathbf{x} 后通过一个多项式时间算法 (此处是最优传输算法 [63]) 得到.

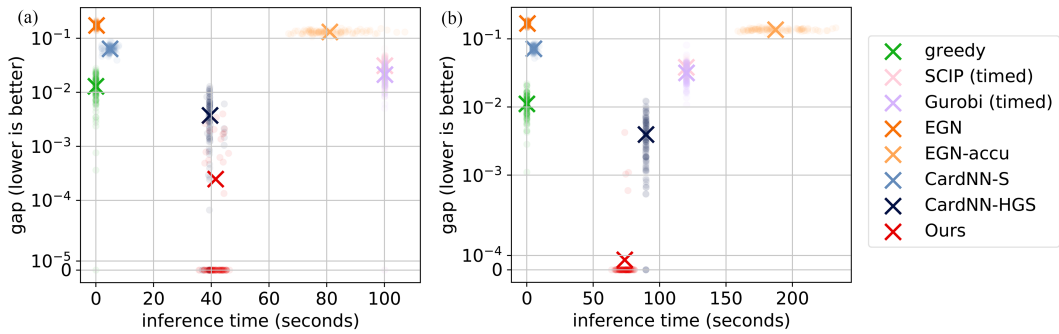


图 6 (网络版彩图) 最大集合覆盖问题上, 比较求解器相对已知最好解的差距

Figure 6 (Color online) Comparison on the gap with respect to the best known incumbent solution on the max-set covering problem. (a) $k = 50, m = 500, n = 1000$; (b) $k = 100, m = 1000, n = 2000$

4.2 最大集合覆盖问题

本文考虑如下的最大集合覆盖问题 (max-set covering problem, MCP): 给定 m 个集合, n 个物品, 每个集合会覆盖全集的若干个物品. 我们要选择其中 k 个集合, 使得被这 k 个集合覆盖到的物品具有最高的总价值. 最大集合覆盖问题的典型案例来自社交网络: 集合代表 m 个意见领袖 (即“网红”), n 个物品指代共 n 个用户. 广告商可通过求解最大集合覆盖问题, 最大化有限预算下广告的影响力. 具体而言, 最大集合覆盖问题的数学形式是

$$\max_{\mathbf{x}} \sum_{j=1}^n \left(\mathbb{I} \left(\sum_{i=1}^m \mathbf{x}_i \mathbf{A}_{ij} \right) \cdot \mathbf{v}_j \right), \quad \text{s.t.} \quad \mathbf{x} \in \{0, 1\}^m, \sum_{i=1}^m \mathbf{x}_i \leq k, \quad (10)$$

其中, 参数 $\mathbf{v} \in \mathbb{R}_+^n$ 代表每个物品的价值, $\mathbf{A}_{i,j} = 1$ 代表集合 i 覆盖了物品 j , 否则 $\mathbf{A}_{i,j} = 0$. $\mathbb{I}(\cdot)$ 为指示函数, 如果内部元素 > 1 则输出 1, 否则 0. 对于该问题, 本文非自回归网络的具体实现方式如下.

- 图建模步骤利用了集合与物品之间的对应关系, 构造了一个二分图. 若集合 i 覆盖了物品 j , 则存在一条边, 即 $\mathbf{A}_{i,j} = 1$. 求解最大集合覆盖问题等价于对二分图的“集合”一侧进行节点分类.

- 神经网络采用了在二分图上常用的 GraphSage 网络^[47], 共有 3 层. 其中从“集合”到“物品”以及反方向的消息传递分别是两个独立的网络, 网络的隐层宽度是 16.

- Gumbel 重参数化和 LinSAT 约束层能处理式 (10) 中正线性约束, 将网络输出投影到接近离散的可行域.

- 目标函数估计步骤中, 与第 4.1 小节处理设施布局问题的流程相比, 式 (10) 更加直观: 在神经网络输出 \mathbf{x} 的值后, 可直接通过式 (10) 计算目标函数.

- (推理阶段的) 邻域搜索在最大集合覆盖问题中未采用, 但在线梯度搜索依然起作用.

本文采用文献 [20] 的设定, 生成两组不同规模的、与 ORLIB^[66] 分布相近的训练和测试数据. 实验对比了贪心算法 (greedy)、专业求解软件 (SCIP^[28], Gurobi^[29]), 图 6(a) 和 (b) 的求解限时分别是 100 和 120 s)、神经网络求解方法 (EGN^[21], CardNN^[22]), 实验结果如图 6 所示, 图中包含了相对差距

$$\text{gap} = \frac{\text{已知最优目标} - \text{目标函数}}{\text{已知最优目标}}$$

与求解耗时的关系, 数据点越靠近左下角代表性能越好. 在最大集合覆盖问题中, 即便是性能最好的 Gurobi 求解器也无法在 24 小时内返回单个问题的最优解, 因此在此处采用相对于当前已知最优解的

差距作为性能指标 (而非“相对全局最优解的差距”). 测试集中每个样例都是图中的一个圆点, 单个方法的平均性能由叉号表示. 在最大集合覆盖问题中, 本文的非自回归网络求解方法 (红色) 位于精度与耗时的帕累托最优曲线上. 本文方法在耗时低于 Gurobi (紫色) 的前提下, 求解精度依然显著高于 Gurobi. 此外, 本文方法在大量测试样例上找到了当前所有方法中的已知最优解.

4.3 旅行商问题

本文考虑如下的旅行商问题 (traveling salesman problem, TSP): 给定 m 个城市的坐标, 一位旅行商需要遍历所有城市并回到原地, 使经过的总路程最短. 旅行商问题的数学形式是

$$\min_{\mathbf{X}} \frac{1}{2} \text{tr}(\mathbf{D}^T \mathbf{X}), \quad \text{s.t.} \quad \mathbf{X} \in \{0, 1\}^{m \times m}, \forall j: \sum_{i=1}^m \mathbf{X}_{i,j} = 2, \forall i: \sum_{j=1}^m \mathbf{X}_{i,j} = 2, \mathbf{X} \in \mathcal{H}, \quad (11)$$

其中 $\mathbf{D} \in \mathbb{R}_+^{m \times m}$ 代表任意两城市之间的距离, $\mathbf{X}_{i,j} = 1$ 代表旅行商依次经过了城市 i, j , $\mathbf{X} \in \mathcal{H}$ 代表 \mathbf{X} 为对称方阵形式表征的哈密顿回路 (Hamiltonian cycle). 本文采用的非自回归网络实现如下:

- 图建模步骤将城市坐标作为节点特征, 两城市间距离边特征, 构造了全连接的图. 对于规模较大的 TSP-500 任务, 采用 k 近邻方法 ($k = 50$) 将图稀疏化.

- 图神经网络采用各向异性的消息传播网络 (anisotropic graph neural networks) [67], 区别于经典图神经网络 [45] 仅编码节点特征, 其同时编码节点和边特征, 消息传递在节点和边之间进行.

- LinSAT 约束层无法直接处理哈密顿回路约束, 但是其松弛形式 $\forall j: \sum_{i=1}^m \mathbf{X}_{i,j} = 2, \forall i: \sum_{j=1}^m \mathbf{X}_{i,j} = 2$ 是正线性约束, 本文采用 LinSAT 处理这一松弛后的约束. Gumbel 重参数化与现行的旅行商问题求解框架冲突, 本实验中未考虑.

- 神经网络离线训练步骤中, 本文为了和其他求解旅行商问题的深度学习方法公平对比, 采用了有监督训练 [60, 61], 未采用本文其他实验中无监督形式的目标函数估计.

- (推理阶段的) 邻域搜索设计上, 本文采用了在旅行商问题中广泛应用的贪心输出结合 2Opt 搜索和蒙特卡洛树搜索 (Monte Carlo tree search, MCTS) 策略.

如表 4 [26, 60, 61] 所示, 在 $m = 50, 100, 500$ 的旅行商数据集上验证了本文方法的有效性. 与现有工作保持一致, 问题样例的城市坐标在二维 $[0, 1]$ 均匀分布上随机生成, 计算两两城市之间的欧氏距离. TSP-50, 100 共有 1280 个测试样例, TSP-500 共有 128 个, 表 4 汇报的求解时间是在完整测试集上的总耗时. 在综合考虑求解性能和求解耗时的情况下, 本文的非自回归方法可以与现有更复杂的神经网络求解器 [26, 60, 61] 性能持平, 甚至更好. 在所有组合优化问题中, 旅行商问题是机器学习研究最早、最深入的问题之一 [10, 11]. 该实验的主要目的是将本文提出的神经网络 (特别是通过 LinSAT 层施加约束的非自回归网络) 与现有神经网络方法进行对比. 其中, 与本文方法相比, 表 4 中其他的基线方法均采用了更复杂的深度学习架构, 例如文献 [26] 采用了元强化学习、文献 [60, 61] 采用了扩散模型. 表 4 还对比了无 LinSAT 版本的神经网络. 无 LinSAT 版本与 MCTS 配合时, 在小规模问题上有一定优势; 但是当城市数目增加到 500 时, LinSAT 为求解效果带来了显著提升. 这表明了在神经网络中显式地考虑约束 (即使是松弛后的约束) 可以获得更大的模型容量, 否则必然会有一部分额外的模型参数被用于学习约束. 最后需要注意的是, 传统 LKH3 算法的求解精度仍然优于现有的神经网络方法 (不论是否是自回归方法). 这一现象并不意外, 尤其是考虑到 LKH 算法包已经针对旅行商问题进行了数十年的开发和调优; 但是在更多的组合优化问题中, 旅行商问题只是特例, 其他问题通常缺乏类似 LKH3 的金标准, Gurobi 才是目前默认的通用求解器. 基于此, 表 4 同样以 LKH3 作为基准方法.

表 4 在旅行商问题上, 本文的神经网络求解器与其他神经网络求解器持平甚至更优

Table 4 Our solver performs on-par or even better than other neural networks on the traveling salesman problem

Solver type		TSP-50			TSP-100			TSP-500		
Neural network	Search strategy	Obj (\downarrow)	Gap (%) (\downarrow)	Time (\downarrow)	Obj (\downarrow)	Gap (%) (\downarrow)	Time (\downarrow)	Obj (\downarrow)	Gap (%) (\downarrow)	Time (\downarrow)
LKH3	(max_trials = 500 5k 50k)	5.688	0.00	3 min 12 s	7.756	0.00	33 min 18 s	16.548	0.00	56 min 12 s
DIMES ^[26]	Active Search+Sampling	5.735	0.82	1 h 44 min	7.918	2.09	3 h 19 min	17.646	6.64	9 h 42 min
DIFUSCO ^[60]	Greedy+2Opt	5.694	0.11	8 min 24 s	7.776	0.26	8 min 44 s	16.800	1.52	2 min 34 s
T2T ^[61]	Greedy+2Opt	5.689	0.02	25 min 12 s	7.761	0.07	25 min 58 s	16.695	0.89	6 min 3 s
GNN	Greedy+2Opt	5.694	0.12	15 s	7.806	0.65	20 s	16.899	2.12	15 s
GNN+LinSAT (ours)	Greedy+2Opt	5.691	0.07	18 s	7.777	0.27	25 s	16.746	1.19	18 s
GNN	Greedy+MCTS	5.689	0.02	15 s	7.767	0.01	29 s	16.706	0.95	1 min 4 s
GNN+LinSAT (ours)	Greedy+MCTS	5.689	0.03	27 s	7.771	0.20	2 min 8 s	16.646	0.59	1 min 8 s

5 结论和展望

神经网络的蓬勃发展与深度学习框架、GPU 芯片等“基础设施”相辅相成. 考虑到组合优化问题的基础性和重要性, 利用神经网络, 特别是高效的深度学习框架和 GPU 芯片的算力求解组合优化已经成为了一个前沿的研究方向. 本文首先总结了现有的自回归式神经网络求解器, 这种序列决策的方式可以使得约束更容易被满足, 且可以用强化学习算法进行训练. 然而, 在面对大规模问题时, 自回归范式面临误差累积、奖励稀疏、效率降低、排列不变性缺失等挑战. 而可微分约束网络、Gumbel 重参数化、无监督学习等技术日趋成熟, 将有望推动非自回归网络得到进一步发展.

面向通用非自回归网络组合优化求解器这一目标, 本文提出了一种求解范式, 包括原始问题的图建模、图神经网络、隐变量、Gumbel 重参数化、LinSAT 约束层、(可微分) 目标函数估计等模块, 能够处理所有正线性约束的组合优化问题, 具有较强的通用性. 本文框架可以采用离线无监督预训练, 对标注的需求量降低, 更具实操性; 本文还设计了在线梯度搜索阶段, 提升了网络的泛化能力. 本文在(有/无上限的) 设施布局问题、最大集合覆盖问题和旅行商问题上分别实现了非自回归网络求解器. 在实验中, 这一方法的适用范围和求解性能均超过了现有非自回归网络. 其中在旅行商问题上, 具有 LinSAT 层的非自回归网络与其他更复杂的神经网络性能相当; 在设施布局、集合覆盖问题上, 其本文方法的性能与 Gurobi 等专业求解软件趋近, 在特定问题上甚至比 Gurobi 更快、更精确.

我们认为, 神经网络求解器 (特别是本文所讨论的非自回归网络求解器) 的优势在于提供了一种自主可控的技术路径, 在特定分布的数据集上, 其最终效果有望持平甚至超越以 Gurobi 为代表的传统求解器. 一方面, Gurobi 的易用性与高性能源自其高度的封装和内部调优. 对求解器的使用者而言, Gurobi 软件近乎是一个黑盒, 很难进一步优化. 另一方面, Gurobi 作为商业软件的性质以及不菲的授权费用并不适合所有场景, 而开源的 SCIP 的性能还是相对逊色. 与之相比, 本文的非自回归网络求解器完全基于开源软件搭建, 算法设计、性能优化更加透明和可控. 此外, 在实际应用中的数据分布通常就是特定的、有限的, 今日的业务数据相比昨日并不会有明显的区别, 因此, 在实际应用中损失一定通用性往往是可接受的. 最后, 本文实验并未充分考虑神经网络设计空间、定制邻域搜索算法、Python 及时编译技术等, 作者相信, 相比本文汇报的实验结果, 神经网络求解器的性能还有进一步提升的空间. 在提升通用性、提升计算效率、探索更多典型应用等方向, 非自回归网络求解组合优化的研究方兴未艾.

参考文献

- 1 Euler L. *Solutio problematis ad geometriam situs pertinentis*. *Commentarii academiae scientiarum Petropolitanae*, 1741, 8: 128–140

- 2 Birkhoff G. Tres observaciones sobre el algebra lineal. *Univ Nac Tucuman Rev Ser A*, 1946, 5: 147–154
- 3 von Neumann J. A Certain Zero-sum Two-person Game Equivalent to the Optimal Assignment Problem. Princeton: Princeton University Press, 1953
- 4 Dantzig G B. Application of the Simplex Method to A Transportation Problem. New York: John Wiley & Sons, Inc., 1951
- 5 Bengio Y, Lodi A, Prouvost A. Machine learning for combinatorial optimization: a methodological tour d’horizon. *Eur J Oper Res*, 2021, 290: 405–421
- 6 Wang F D. Functional representation for graph matching. Dissertation for Ph.D. Degree. Wuhan: Wuhan University, 2021 [王福东. 基于泛函表示的图匹配研究. 博士学位论文. 武汉: 武汉大学, 2021]
- 7 Zhou Y L, Wang J H, Huang C. Estimation of distribution-discrete particle swarm optimization algorithm for permutation-based problems. *Acta Electron Sin*, 2014, 3: 561–571 [周雅兰, 王甲海, 黄聪. 求解排列问题的分布估计离散粒子群优化算法. *电子学报*, 2014, 3: 561–571]
- 8 Guo T D, Han C Y, Tang S Q. Machine Learning Methods for Combinatorial Optimization. Beijing: Science Press, 2019 [郭田德, 韩丛英, 唐思琦. 组合优化问题的机器学习求解方法. 北京: 科学出版社, 2019]
- 9 Guo T D, Han C Y. From numerical optimization method to learning optimization method. *Oper Res Trans*, 2019, 4: 1–12 [郭田德, 韩丛英. 从数值最优化方法到学习最优化方法. *运筹学学报*, 2019, 4: 1–12]
- 10 Vinyals O, Fortunato M, Jaitly N. Pointer networks. In: *Proceedings of Neural Information Processing Systems*, 2015. 2692–2700
- 11 Khalil E, Dai H, Zhang Y, et al. Learning combinatorial optimization algorithms over graphs. In: *Proceedings of Neural Information Processing Systems*, 2017. 6351–6361
- 12 Liu C, Wang R, Jiang Z, et al. Revocable deep reinforcement learning with affinity regularization for outlier-robust graph matching. In: *Proceedings of International Conference on Learning Representations*, 2023
- 13 Mao H, Schwarzkopf M, Venkatakrisnan S B, et al. Learning scheduling algorithms for data processing clusters. In: *Proceedings of the ACM Special Interest Group on Data Communication*, 2019. 270–288
- 14 Lu H, Zhang X, Yang S. A learning-based iterative method for solving vehicle routing problems. In: *Proceedings of International Conference on Learning Representations*, 2019
- 15 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018
- 16 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: *Proceedings of Computer Vision and Pattern Recognition*, 2016. 770–778
- 17 Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection. In: *Proceedings of Computer Vision and Pattern Recognition*, 2016. 779–788
- 18 Zhang S, Wen L, Lei Z, et al. RefineDet++: single-shot refinement neural network for object detection. *IEEE Trans Circuits Syst Video Technol*, 2020, 31: 674–687
- 19 Chen Z, Liu J, Wang X, et al. On representing mixed-integer linear programs by graph neural networks. In: *Proceedings of International Conference on Learning Representations*, 2023
- 20 Wang R, Shen L, Chen Y, et al. One-shot neural combinatorial optimization solvers: theoretical and empirical notes on the cardinality-constrained case. In: *Proceedings of International Conference on Learning Representations*, 2023
- 21 Karalias N, Loukas A. Erdos goes neural: an unsupervised learning framework for combinatorial optimization on graphs. In: *Proceedings of Neural Information Processing Systems*, 2020
- 22 Wang R, Zhang Y, Guo Z, et al. LinSATNet: the positive linear satisfiability neural networks. In: *Proceedings of International Conference on Machine Learning*, 2023
- 23 Scarselli F, Gori M, Ah C T M, et al. The graph neural network model. *IEEE Trans Neural Netw*, 2009, 20: 61–80
- 24 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In: *Proceedings of International Conference on Learning Representations*, 2017
- 25 Jang E, Gu S, Poole B. Categorical reparameterization with Gumbel-Softmax. In: *Proceedings of International Conference on Learning Representations*, 2017
- 26 Qiu R, Sun Z, Yang Y. DIMES: a differentiable meta solver for combinatorial optimization problems. 2022. ArXiv:2210.04123
- 27 Sun H, Goshvadi K, Nova A, et al. Revisiting sampling for combinatorial optimization. In: *Proceedings of International Conference on Machine Learning*, 2023
- 28 Gamrath G, Anderson D, Bestuzheva K, et al. The SCIP Optimization Suite 7.0. Technical Report, 2020
- 29 Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, Version 11.0, 2021
- 30 Peng Y, Choi B, Xu J. Graph learning for combinatorial optimization: a survey of state-of-the-art. *Data Sci Eng*, 2021, 6: 119–141

- 31 Yan J, Yang S, Hancock E R. Learning for graph matching and related combinatorial optimization problems. In: Proceedings of International Joint Conferences on Artificial Intelligence, 2020. 4988–4996
- 32 Wang R, Hua Z, Liu G, et al. A bi-level framework for learning to solve combinatorial optimization on graphs. In: Proceedings of Neural Information Processing Systems, 2021
- 33 Wang R, Zhang T, Yu T, et al. Combinatorial learning of graph edit distance via dynamic embedding. In: Proceedings of Computer Vision and Pattern Recognition, 2021
- 34 Nair V, Bartunov S, Gimeno F, et al. Solving mixed integer programs using neural networks. 2020. ArXiv:2012.13349
- 35 Gasse M, Chételat D, Ferroni N, et al. Exact combinatorial optimization with graph convolutional neural networks. In: Proceedings of Neural Information Processing Systems, 2019
- 36 Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with deep reinforcement learning. 2013. ArXiv:1312.5602
- 37 Zhang C, Song W, Cao Z, et al. Learning to dispatch for job shop scheduling via deep reinforcement learning. In: Proceedings of Neural Information Processing Systems, 2020
- 38 Liu C, Wang R, Jiang Z, et al. Deep reinforcement learning of graph matching. 2020. ArXiv:2012.08950
- 39 Duan L, Hu H, Qian Y, et al. A multi-task selected learning approach for solving 3D flexible bin packing problem. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, 2019. 1386–1394
- 40 Hopfield J J, Tank D W. “Neural” computation of decisions in optimization problems. *Biol Cybern*, 1985, 52: 141–152
- 41 Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach Learn*, 1992, 8: 229–256
- 42 Kool W, van Hoof H, Welling M. Attention, learn to solve routing problems! In: Proceedings of International Conference on Learning Representations, 2019. 1–25
- 43 Wang R, Yan J, Yang X. Neural graph matching network: learning Lawler’s quadratic assignment problem with extension to hypergraph and multiple-graph matching. *IEEE Trans Pattern Anal Mach Intell*, 2022, 44: 5261–5279
- 44 You J, Wu H, Barrett C, et al. G2SAT: learning to generate SAT formulas. In: Proceedings of Neural Information Processing Systems, 2019
- 45 You J, Ying Z, Leskovec J. Design space for graph neural networks. In: Proceedings of Neural Information Processing Systems, 2020. 17009–17021
- 46 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Neural Information Processing Systems, 2017
- 47 Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of Neural Information Processing Systems, 2017
- 48 Xu K, Hu W, Leskovec J, et al. How powerful are graph neural networks? In: Proceedings of International Conference on Learning Representations, 2019
- 49 Mena G, Belanger D, Linderman S, et al. Learning latent permutations with Gumbel-Sinkhorn networks. In: Proceedings of International Conference on Learning Representations, 2018
- 50 Grover A, Wang E, Zweig A, et al. Stochastic optimization of sorting networks via continuous relaxations. In: Proceedings of International Conference on Learning Representations, 2019
- 51 Sinkhorn R, Rangarajan A. A relationship between arbitrary positive matrices and doubly stochastic Matrices. *Ann Math Statist*, 1964, 35: 876–879
- 52 Choo J, Kwon Y D, Kim J, et al. Simulation-guided beam search for neural combinatorial optimization. In: Proceedings of Neural Information Processing Systems, 2022. 8760–8772
- 53 Joshi C K, Laurent T, Bresson X. An efficient graph convolutional network technique for the travelling salesman problem. 2019. ArXiv:1906.01227
- 54 Abbas A, Swoboda P. DOGE-train: discrete optimization on GPU with end-to-end training. 2022. ArXiv:2205.11638
- 55 Weisfeiler B, Leman A. The reduction of a graph to canonical form and the algebra which appears therein. *Nauchno-Technicheskaya Informatsia*, 1968, 2: 12–16
- 56 Chen Z, Liu J, Wang X, et al. On representing linear programs by graph neural networks. In: Proceedings of International Conference on Learning Representations, 2023
- 57 Pogancić M V, Paulus A, Musil V, et al. Differentiation of blackbox combinatorial solvers. In: Proceedings of International Conference on Learning Representations, 2019
- 58 Ning X. A cross-entropy approach to the permutation-based combinatorial optimization problems. Dissertation for Ph.D. Degree. Beijing: Tsinghua University, 2019 [宁秀. 置换型组合优化问题的交互熵算法研究. 博士学位论文. 北京: 清华大学, 2019]
- 59 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of

- International Conference on Machine Learning, 2017. 1126–1135
- 60 Sun Z, Yang Y. DIFUSCO: graph-based diffusion solvers for combinatorial optimization. In: Proceedings of Neural Information Processing Systems, 2023
- 61 Li Y, Guo J, Wang R, et al. T2T: from distribution learning in training to gradient search in testing for combinatorial optimization. In: Proceedings of Neural Information Processing Systems, 2023
- 62 Fey M, Eric Lenssen J, Weichert F, et al. SplineCNN: fast geometric deep learning with continuous B-spline kernels. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. 869–877
- 63 Cuturi M. Sinkhorn distances: lightspeed computation of optimal transport. In: Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013. 2292–2300
- 64 Amos B, Kolter J Z. OptNet: differentiable optimization as a layer in neural networks. In: Proceedings of the 34th International Conference on Machine Learning, 2017. 136–145
- 65 Gould S, Hartley R, Campbell D J. Deep declarative networks. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 3988–4004
- 66 Beasley J E. OR-Library: distributing test problems by electronic mail. J Oper Res Soc, 1990, 41: 1069–1072
- 67 Wang X, Ding X, Tung K H A, et al. An efficient graph indexing method. In: Proceedings of IEEE 28th International Conference on Data Engineering, 2012. 210–221

Learning to solve combinatorial optimization under positive linear constraints via non-autoregressive neural networks

Runzhong WANG^{1,2}, Yang LI^{1,2}, Junchi YAN^{3,1,2*} & Xiaokang YANG²

1. Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China;

2. Key Laboratory of Artificial Intelligence, Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China;

3. School of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai 200030, China

* Corresponding author. E-mail: yanjunchi@sjtu.edu.cn

Abstract Combinatorial optimization (CO) is the fundamental problem at the intersection of computer science, applied mathematics, etc. The inherent hardness in CO problems brings up a challenge for solving CO exactly, making deep-neural-network-based solvers a research frontier. In this paper, we design a family of non-autoregressive neural networks to solve CO problems under positive linear constraints with the following merits. First, the positive linear constraint covers a wide range of CO problems, indicating that our approach breaks the generality bottleneck of existing non-autoregressive networks. Second, compared to existing autoregressive neural network solvers, our non-autoregressive networks have the advantages of higher efficiency and preserving permutation invariance. Third, our offline unsupervised learning has a lower demand on high-quality labels, getting rid of the demand of optimal labels in supervised learning. Fourth, our online differentiable search method significantly improves the generalizability of our neural network solver to unseen problems. We validate the effectiveness of this framework in solving representative CO problems including facility location, max-set covering, and traveling salesman problem. Our non-autoregressive neural solvers are competitive and can be even superior to state-of-the-art solvers such as SCIP and Gurobi, especially when both efficiency and efficacy are considered.

Keywords combinatorial optimization, deep learning, non-autoregressive neural network, graph neural network, gradient-based optimization