



组合优化问题的机器学习求解方法

献给韩继业教授 90 寿辰

郭田德, 李安琪, 韩丛英*

中国科学院大学数学科学学院, 北京 100049

E-mail: tdguo@ucas.ac.cn, lianqi20@mails.ucas.ac.cn, hancy@ucas.ac.cn

收稿日期: 2024-05-30; 接受日期: 2024-09-22; 网络出版日期: 2025-01-17; * 通信作者

国家自然科学基金重大项目课题 (批准号: 11991022) 资助项目

摘要 最优化是一个非常经典的研究领域, 许多工程、管理等领域的问题本质上都是最优化问题. 目前, 以深度学习、强化学习为代表的机器学习方法发展迅速, 在诸多学科领域都带来了关键性的促进和发展. 最优化是机器学习的重要支撑学科之一, 机器学习归根到底都要转化成求解一个最优化问题, 这属于科学智能 (science for AI (artificial intelligence)) 的研究内容. 反过来, 人工智能技术 (包括机器学习) 已经对科学的发展产生了重大影响, 并且将形成新的科学研究范式—智能科学 (AI for science). 最优化问题的人工智能求解方法就是利用人工智能的理论方法设计最优化问题快速高效的求解算法. 本文总结了组合优化机器学习求解方法最近几年的研究进展, 深入分析了组合优化问题求解的难点及其人工智能求解方法的瓶颈问题, 结合图像处理、博弈论和强化学习等热门领域的研究现状, 给出了一些重要的研究方向.

关键词 组合优化 机器学习 最优化

MSC (2020) 主题分类 90C27

1 引言

连续优化问题是目标函数和可行域均连续的优化问题, 由于其本身连续的性质, 相比于组合优化问题而言具有更完备的理论保证 (参见文献 [22, 29, 52]). 组合优化是应用数学和组合领域的新兴学科, 它由线性规划、算法理论、组合数学发展而来, 目的是解决离散结构的最优化问题, 它是解决信息论、控制论、管理学、生物学、分子物理学、大规模集成电路设计、图像处理、模式识别、计算机科学等 (参见文献 [54, 61, 71]) 诸多学科中一些核心问题的重要工具, 并在工程技术、经济、军事等诸多方面有着极为重要的应用. 探讨组合优化问题的各种算法以及快速求解、设计专门的求解器、研究计算复杂性理论一直是最优化理论、计算机科学等相关学科领域备受关注的研究热点. 组合优化问题, 特别是大规模的组合优化问题, 其快速求解具有重要的理论意义和实际应用价值.

英文引用格式: Guo T D, Li A Q, Han C Y. Machine learning method for combinatorial optimization problems (in Chinese). Sci Sin Math, 2025, 55: 451–480, doi: 10.1360/SSM-2024-0180

求解连续优化的传统迭代方法多数以梯度法为基础, 解析与数值计算相结合, 是一种基于固定模式的最优化迭代求解方法. 算法的迭代过程实质上是对迭代点进行非线性变换的过程, 非线性变换通过方向和步长来实现 (参见文献 [28]). 对于最优化问题的每一个实例, 都需要从头到尾执行整个算法, 计算复杂度是固定的. 一旦算法被程序实现, 算法的效率 (计算精度和复杂度) 就被固定. 线性规划是连续优化中非常重要的一个问题, 值得注意的是, 它既可以视为连续优化问题在连续的可行域内沿下降方向迭代搜索求解 (如内点算法), 也可以表征为组合优化问题, 从初始解出发沿着多面体的顶点离散地搜索最优解顶点 (如单纯形算法). 因此, 本文同样将线性规划视为一个组合优化问题, 介绍其基于机器学习的求解方法. 单纯形算法是在求解器中最广泛使用的算法, 尤其是对于中小规模线性规划问题, 这是因为单纯形算法的最坏情形虽然是指数算法, 但最坏情形发生的概率很小, 平均性能优于目前线性规划问题的多项式时间算法. 单纯形算法的思想很简单, 本质上就是基于初始可行解进行进基变量和出基变量的转换, 最终沿着可行域多面体的边转移至最优解. 进基变量和出基变量的转换过程就是转轴规则的执行过程, 也是单纯形算法的核心. Dantzig 规则、Bland 规则和 Steepest-edge 规则是比较经典且常用的转轴规则. 单纯形算法转轴规则本质上是可行解对应的基之间某一列的变换, 对应到可行域多面体上就是从初始解顶点出发, 每次转轴对应相邻顶点间的一步转移, 直至到达最优解顶点.

求解组合优化问题的传统方法主要包括精确算法、近似算法、近似方案和启发式算法等 (参见文献 [44, 75]). 精确算法可以保证找到问题的最优解, 但往往需要对解空间的有效区域进行搜索, 因此精确算法只适用于求解小规模问题. 当问题规模较大时, 虽然理论上可以找到问题的最优解, 但是由于计算量太大对时间的消耗很长, 难以在实际问题中应用. 然而当问题规模较大时, 求出问题的精确解往往是不现实的. 但如果问题对精度的要求不过分严格, 就可以用“精度换时间”的方式, 利用近似算法在多项式的时间内找到问题的近似解. 近似算法目前已越来越多地应用于现实应用中的组合优化问题求解. 不同于一味追求时效性的启发式算法, 近似算法在追求时间效率的同时, 还会要求找到解的质量. 近似算法通常有时间和近似比两种度量指标, 时间度量了算法的时间效率; 近似比度量了找到的近似解与最优解之间的差距, 近似比越接近 1 算法越精确, 近似比为 1 的算法就是精确算法. 通常说的改进近似算法多指设计比现有算法近似比更接近 1 的算法. 这样的改进是否存在极限? 或者说, 怎样的算法才能真正令人满意? 一类近似性能“最好”的近似算法, 称为近似方案. 近似方案分为多项式时间近似方案 (polynomial time approximation scheme, PATS) 和完全多项式时间近似方案 (full polynomial time approximation scheme, FPATS). 与之不同, 启发式算法无法准确描述找到的解与最优解的近似程度, 甚至无法确保找到的解的可行性与最优性. 但是由于其简单、直观、快速、易于实现、解质量优良的特性, 启发式算法特别受实际工作者的青睐. 启发式算法是一种基于直观理解或经验构造的算法, 可以在可接受的时间和存储空间内求出问题的解, 但是解的质量难以保证. 目前的启发式算法以模仿自然现象为主, 从一些物理和生物现象中受启发而来, 基于一定的搜索机制, 同时具备全局优化能力. 使用较为广泛的有遗传算法、模拟退火算法、蚁群算法、禁忌搜索算法、粒子群算法和人工神经网络算法等. 这类智能优化算法不是专门为某个具体的问题设计的, 而是一种求解框架, 可以根据不同的问题做某些调整, 从而适用于不同类型的问题. 这种特性保障了其实践的通用性, 可以快速处理大规模数据并得到满意的解, 更适用于工业问题求解.

尽管如此, 上述组合优化问题的传统方法设计仍存在很多问题: (1) 设计难度大, 需要复杂的专家经验, 求解限制多; (2) 一般而言, 传统方法都是基于问题而设计, 不同问题需要设计不同算法, 从而产生了很多研究分支, 如整数规划、最小支撑树、最短路、最大流、最小费用流、最大匹配、背包问题、装箱问题、旅行商问题、选址问题、Steiner 树问题 (Steiner tree problem, STP) 等; (3) 由于实例的独

立性, 无法利用其中部分的实例为其他实例的算法设计提供有效的信息. 但是, 在实际应用中, 一方面, 同一问题的不同实例, 往往会保留相同的离散优化结构, 而只在具体数值上有所差异, 问题实例之间存在固有的相似性; 另一方面, 同一领域中的不同问题, 虽然建成的模型各异, 但也具有内在的相似性, 如度约束最小生成树问题 (degree-constrained minimum spanning tree problem, DCMSTP)、最小路由成本生成树问题 (minimum routing cost spanning tree problem, MRCSTP)、STP 等, 可以归结为某种框架下的一类生成树问题. 传统的某一类具体算法没有系统地利用这些性质, 更不可能同时直接求解这些问题的所有实例.

近年来机器学习的兴起, 给组合优化问题的求解带来了新的思路 (参见文献 [1, 13]). 机器学习是一门多领域交叉的学科, 涉及概率论、统计学、最优化、逼近论等多门学科, 其核心是“学习”, 顾名思义即让机器具备学习的能力. 机器学习研究利用计算机模仿人类的行为, 对经验行为的历史数据进行分析 and 挖掘, 从而学习出一般的规律替代人类的行为. 机器学习技术是人工智能领域的核心, 广泛应用于人工智能的各个领域中, 如语音识别、目标检测、风格迁移、自然语言处理等. 机器学习一般有 3 个步骤. 首先, 根据问题选择恰当的网络模型以及使得模型性能最优的度量, 然后基于训练数据确定算法的参数, 最后将训练好的模型应用于测试数据集, 当在测试集上表现出一致良好的性能时, 一般认为模型训练得较为成功.

深度学习是机器学习领域内的一个研究方向, 是基于深度神经网络进行学习的一种方法 (参见文献 [58, 70]). 因为机器学习容易陷入局部最优解和梯度消失等问题, 在深度学习出现之前缺乏对深度神经网络的充分训练, 导致传统的机器学习方法性能不佳. 深度学习之所以称为“深度”, 是因为深度学习是含有多个隐藏层的多层学习模型, 可以通过组合低层次的特征抽象出更高层的特征表示, 以发现和提取数据的分布式特征. 大规模硬件加速设备的出现和性能的不断提升, 更进一步地推动了深度学习的普及性. 深度学习可以是监督、半监督和无监督的. 目前通用的深度学习架构有深度神经网络、深度信念网络、递归神经网络、卷积神经网络等, 已经被广泛应用于生物信息学、自然语言处理、药物设计、医疗图像处理等各个领域, 有的方面可以与专家的性能相当, 有的方面甚至可以优于专家处理的效果. 深度学习使得机器更进一步地模仿人类的活动, 解决了很多复杂的模式识别新难题, 大大促进了人工智能的发展.

强化学习 (reinforcement learning, RL) 也是机器学习领域的一个重要研究内容, 它是一个序列决策过程, 旨在通过智能体与环境的交互和环境的反馈来调整智能体的动作, 从而使智能体实现预设的目标 (参见文献 [47, 55, 56]). 与监督学习不同, 强化学习没有带专家标签的训练数据, 最终目标实现与否也不是瞬时可以输出的, 需要经过很长时间才能观察到完整轨迹的奖励, 并且强化学习采取的动作是可以改变数据分布的 (监督学习的输出不会改变数据的分布). 强化学习模型通常会有状态、动作、状态转移、奖励这些基本元素. 状态描述了智能体当前时刻的过程信息, 如姿态、求解状态、位置等. 动作是智能体当前可以采取的决策, 通常用动作集描述智能体可以采取的动作的集合. 状态转移描述了当前状态在执行某个动作后下一个状态如何改变. 奖励是智能体在当前状态执行动作后环境的反馈, 通常我们用回报来描述累计奖励的总和, 这也是强化学习需要优化的目标. 常用的强化学习算法有 Monte Carlo 法、时间差分法、Sarsa 算法、Q 学习 (Q-learning)、Reinforce 算法、Actor-Critic 算法、信赖域策略优化算法、确定性策略梯度算法等. 随着深度学习的发展, 深度强化学习的方法也逐渐出现在大家的视野中 (参见文献 [20]), 深度强化学习有效地将具有学习决策能力的强化学习与具有感知表达能力的深度学习结合起来, 简言之就是用深度神经网络逼近值函数或者策略函数. 深度强化学习对人工智能的发展具有里程碑的意义, 也进一步推动了组合优化问题的求解.

郭田德等^[24]提出了机器学习前端学习和后端学习的二元分类模式. 前端学习可以视为特征再表

达过程, 简单来说就是对输入的原始数据进行特征提取的学习. 而后端学习是以前端学习的输出结果为输入进行分类或识别等后端任务的学习. 前端学习和后端学习可以独立进行, 在任务中可以利用前端学习得到原始数据的再表达, 然后基于一种具体的后端学习模式, 对前端学习的结果进行分类识别等特征匹配过程, 即前后端组合学习. 而前后端融合学习则与之不同, 它以原始数据作为输入, 利用统一的学习模型, 直接输出分类或者识别结果, 将前端学习和后端学习融合为一个整体. 顾名思义, 前后端组合学习是分别独立地进行前端学习和后端学习过程, 而前后端融合学习是将其视为一个整体, 直接进行从原始数据到任务目标的映射过程. 后面我们将以此为脉络, 梳理组合优化机器学习求解方法的进展.

机器学习是一门多领域交叉的学科, 其中最优化理论和算法是机器学习的支柱学科, 一直在机器学习理论和技术中发挥着重要作用. 反过来, 人工智能技术又为求解最优化问题提供了新的思路. 2019年, 郭田德和韩丛英^[23] 借鉴人工智能解决问题的理论方法, 提出了学习最优化方法的概念. 学习最优化方法分为线下训练和线上运行两个阶段, 对于最优化问题的每一个实例, 不需要从头到尾执行整个算法, 而只需要根据训练得到的参数进行简单的运算得到求解结果, 计算复杂度低, 线上运行时间短; 算法具有自学习功能, 算法被程序实现后, 算法的效率 (计算精度和复杂度) 不是固定的, 随着训练的不断深入, 线上运行的效果会越来越好. 学习最优化方法更好地利用在同一应用领域内问题实例之间存在的固有相似性, 实现了线下学习挖掘问题的本质信息, 线上自动更新求解策略. 同年, 郭田德等^[25] 在《组合优化机器学习方法》一书中, 对组合优化的机器学习方法从研究的必要性、设计思想、已有的研究方法、问题迁移和复杂性分析等多个角度论述了机器学习求解组合优化问题需要解决的一些根本性内容, 还给出了一些具体组合优化问题人工智能求解的初步研究结果. 在该书中, 他们分析了组合优化问题的特性, 以及机器学习方法求解组合优化问题目前的局限和未来有价值的研究方向. 2020年以来, 利用人工智能方法求解组合优化问题这一方向, 在国内外引起了越来越多学者的关注, 已经出现的一些利用深度学习求解组合优化问题的方法, 包括利用循环神经网络 (recurrent neural network, RNN) 结合有监督的训练方式求解凸包、Delaunay 三角剖分、二部图匹配等问题, 利用强化学习, 扩大求解 NP (non-deterministic polynomial)- 困难问题的规模, 求解旅行商问题 (traveling salesman problem, TSP) 和背包问题等. 这些已有的机器学习的方法, 大多只针对同一类问题、相同约束的不同数据, 通过训练数据自动发现特征, 进行求解. 这些方法属于学习类方法, 虽然还存在一些问题, 但也取得了一定的效果, 说明了学习类方法的可行性. 由于机器学习方法种类繁多, 同时考虑到目前组合优化问题的主流方法, 下面从前后端组合学习和前后端融合学习的视角展开介绍组合优化问题的机器学习求解方法 (如图 1 所示), 分析制约组合优化机器学习方法发展的瓶颈问题, 并给出未来可以尝试的研究方向.

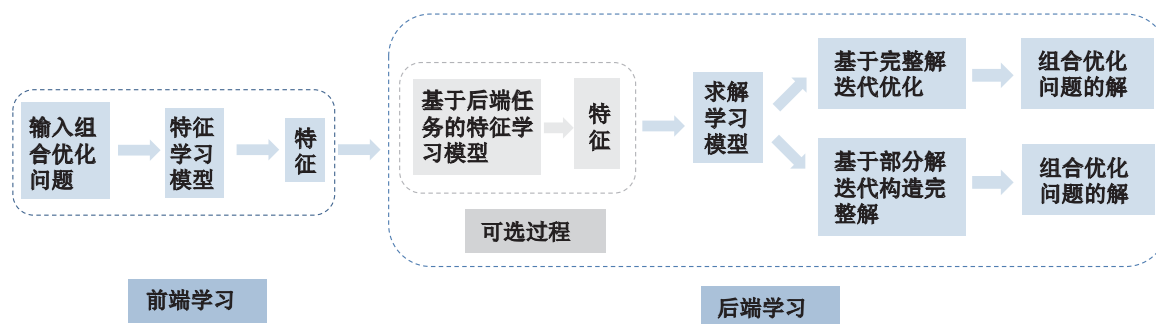


图 1 (网络版彩图) 优化问题的机器学习求解方法脉络图

2 组合优化机器学习框架

我们为基于机器学习求解组合优化问题的方法提取出了统一的学习框架, 如图 1 所示. 学习的首要问题是空间的描述, 即数据的再表达问题, 即前端学习或“特征学习”, 也称为数据再表达的学习 (参见文献 [24]). 前端学习的目的是发现特征, 在有目标类标签 (监督) 或者没有目标类标签 (无监督) 的情形下, 通过学习得到数据的特征. 组合优化机器学习求解方法的前端学习, 主要目的是发现组合优化问题实例的特征. 后端学习, 也称之为求解学习, 学习的输入是前端学习输出的特征, 学习的目的是用于求解后端任务, 在有目标类标签 (有监督) 或者没有目标类标签 (无监督) 的情形下, 通过学习得到后端任务的解. 组合优化机器学习求解方法的后端学习是求解组合优化问题的实例. 前后端组合学习是前端学习和后端学习分别独立进行, 取一种前端学习模式得到数据的一种再表达, 即特征; 然后取一种后端学习的模式, 根据前端学习得到的特征对后端任务进行求解, 没有统一的损失函数. 前后端融合学习是使用统一设计的学习模型, 将前端学习和后端学习融合为一体, 使用统一的损失函数. 该框架从输入的组合优化问题开始, 依赖前端学习的方法提取出问题的特征, 作为后端学习任务的输入. 这一过程可以基于人工提取特征的方式, 也可以通过神经网络得到问题特征的编码. 对于后端学习过程, 有的方法可能会有基于后端任务的特征提取模型, 来进一步得到特征嵌入表示. 在此之后的求解学习模型阶段, 可以按照解的可行性, 划分为基于完整解迭代优化和基于部分解迭代构造完整解两个框架, 最终得到问题的解.

3 前后端组合学习

前后端组合学习的方法, 具体分为前端学习和后端学习两部分, 在前端学习中我们按照人工提取特征和基于神经网络的特征提取两个部分, 在后端学习中我们按照基于完整解迭代优化的思路 and 基于部分解迭代构造完整解两类方法论述. 对于前后端融合的方法, 考虑到整个过程都是一个整体, 也同样划分为基于完整解寻找最优解和基于部分解构造完整解两类方法.

3.1 前端学习方法

3.1.1 人工提取特征

混合整数线性规划 (mixed integer linear programming, MILP) 是一类非常重要也难于求解的组合优化问题, 很多组合优化问题都可以化为混合整数线性规划问题. 分枝定界法是求解混合整数线性规划较为常用的一类方法, 在求解过程中往往将初始问题视为根节点, 依赖变量选择策略固定分数变量从而产生子节点, 再利用节点选择策略, 选择要进一步分枝的子节点, 逐步构造出尽可能小的分枝定界树并找到最优解. 但是分枝定界树随着求解过程的进行是动态变化的, 因此问题的特征也是需要随着求解过程的进行, 不断融合全局信息, 捕捉当前的求解状态信息. 对此, 目前主流的研究方法往往是基于专家构造特征的方式进行特征提取, 这属于前端学习的范畴. 然后再将提取到的特征, 输入到神经网络中进行分枝变量的选择或者候选分枝节点的选取.

最早的研究是 2014 年 Marcos Alvarez 等^[58]对 0-1 整数线性规划的研究, 从目标函数系数、约束矩阵内部、约束矩阵与右端项和目标函数系数的关系三个角度入手, 给出了一组人为构造的特征用于捕捉当前的求解状态. 基于极端随机树 (ExtraTrees) 方法, 利用监督学习来模仿强分枝变量选择的策略. 在此特征构造的基础上, Marcos Alvarez 等^[59]引入了可靠性检验的思想, 同时进行了技术上的改

进, 在分枝定界整个过程中使用线性回归的方法, 相对于文献 [58] 中的方法, 在较短时间内, 取得了性能上的提升. 与此同时, Khalil 等 [35] 针对传统的强分枝策略耗时较长且缺乏对输入实例的自适应性的问题, 借鉴了信息检索领域的排序方法, 将变量选择问题转化为学习排序问题, 此方法相对于竞争方法取得了明显优势, 尤其是在运行时间方面, 对于中等、困难的问题, 可以比其他方法的速度快 15% 至 20%.

2019 年, Gasse 等 [21] 提出了一种里程碑式的方法, 将分枝定界树的节点状态编码为具有节点和边特征的二部图 (如图 2 所示), 从而使得图神经网络 (graph neural network, GNN) 的应用成为了可能. 这种二部图结构包含变量节点和约束节点两类节点, 变量节点的维数与变量的个数一致, 变量节点记录一组与变量相关的特征; 同样地, 约束节点的维数与约束个数一致, 记录着约束节点的一组特征. 当且仅当当前变量在某个约束中时, 对应的变量节点与约束节点之间才会添加一条边进行连接, 而边对应着约束矩阵系数相关的特征. Gasse 等 [21] 采用图卷积神经网络 (graph convolution neural networks, GCNN) 与监督学习结合的方法, 将变量选择策略参数化为图卷积神经网络, 以两个交错的半卷积执行单个图卷积, 通过最小化交叉熵损失来近似强分枝策略. 在 4 个 NP- 困难问题的基准集上测试表明, 该方法不但优于先前的基于机器学习的方法, 而且还具有较强的泛化性. 在这种二部图特征表示的基础上, 后续涌现出了很多的改进工作. 鉴于 Gasse 等 [21] 的方法计算成本过高, Gupta 等 [26] 只在分枝定界树的根节点使用图神经网络模型, 在剩余节点使用多层感知机完成快速预测, 提出了一种混合分枝的方法, 在 CPU (central processing unit) 机器上能够逼近 GNN-GPU (graphics processing unit) 的性能. Nair 等 [62] 构建了神经潜水和神经分枝两个基于神经网络的组件, 用于基本的混合整数线性规划求解器. 神经潜水学习深度神经网络的目的是为整数变量生成局部赋值, 而神经分枝则学习分枝定界中的变量选择策略, 尽可能地用一棵小的分枝定界树来控制目标值的优化间隙 (gap). 在大规模实际应用数据集和 MIPLIB 公开数据集上, 求解器的性能取得了非常大的改进. 不同于以往二部图的构造方式, Ding 等 [16] 为 0-1 整数规划问题设计了一个三部图的构造方法, 以图的形式表示 0-1 整数规划问题, 此外在图卷积网络的基础上还添加了注意力权重, 利用图注意力网络 (graph attention network, GAN) 采用非权值共享的方式, 强化三部图边所蕴含的系数特征. 类似地, Xu 等 [89] 解决了非周期性 2D 平铺问题 (non-periodic 2D tiling problem), 包括使用一种或多种类型的平铺作为输入覆盖任意 2D 形状. 该问题被建模为一个图问题, 并引入了一种新的图神经网络架构, 称为 TilingGNN. 这种方法可以有效地覆盖超过 2,000 个瓷砖的各种 2D 形状. Nowak 等 [66] 以有监督的方式训练图神经网络预测二次指派问题 (quadratic assignment problem, QAP) 的解. 为此, 他们将二次指派问题实例表示为两个邻接矩阵, 并使用两个对应的图作为图神经网络的输入. 针对二次指派问题, Li 和 Han [50] 首先将

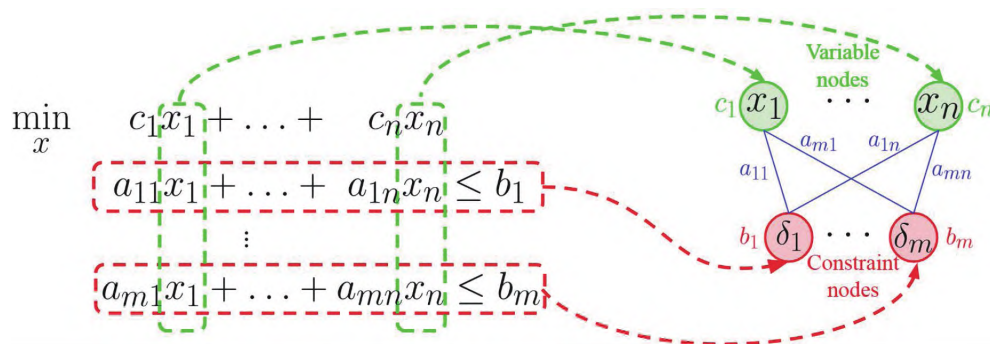


图 2 (网络版彩图) MILP 的二部图结构 [62]

其表述为图上的节点选择问题, 并构造了对应的强化学习模型, 结合图神经网络与 Transformer 设计了基于 Actor-Critic 的求解算法 ACQAP (actor-critic for QAP).

对于混合整数线性规划问题, 除了变量选择策略之外, 节点选择策略、割平面算法等也是非常经典的研究问题, 这类问题的研究中同样以人工设计的特征作为前端学习的方式, 将特征提取的结果作为输入馈入到不同的后端学习方法中. 对于节点选择策略, 最早的研究是 2014 年 He 等^[27] 应用模仿学习技术学习自适应的节点搜索顺序. He 等^[27] 使用模仿学习, 基于 DAgger (dataset aggregation) 算法自动学习启发式方法, 避免了大多数领域专家进行反复实验的调整, 同时也巧妙地避免了专家打分这个复杂的过程. 在多个数据集上实验表明, 此方法与先进求解器 (SCIP (solving constraint integer programs) 和 Gurobi) 相比, 可以在相对短的时间内找到具有更好目标值的解. Song 等^[76] 针对混合整数线性规划分枝定界框架中的节点选择策略, 基于传统模仿学习的 DAgger 框架^[27], 设计了自模仿策略和自模仿泛化算法, 从而可以有效地从错误中学习, 并且可以应用于比原始有监督训练集更大的问题实例中, 从而有效地将自模仿学习作为一种迁移学习方式. 同样值得注意的是 Khalil 等^[36] 的工作, 引入了一种节点处的评分方式, 为每个节点固定下来的变量的可信程度进行评分, 用所有固定变量可信程度评分的总和来代表整个节点的可信程度, 最终基于评分来选择节点. 除此之外, Kabbaj 和 El Afia^[31] 提出了一种变量选择和节点选择“二合一”策略, 在构造的特征集中, 变量选择策略的特征是节点选择策略特征与其余提出特征的线性组合, 从而可以同时处理上述策略, 然后两次应用基于支持向量机的回归技术, 分别学习节点选择策略、变量选择策略, 最终选择得分最高的节点和变量进行选择 and 分枝. 实验中就此方法与 SCIP 的默认策略进行了对比, 结果表明, 此方法有效地减少了求解时间, 并且在测试集 10 个实例上获得了与原方法几乎相同的对偶边界, 并且在三个实例上, 获得了更好的结果. 对于割平面算法, Paulus 等^[67] 提出了一种新的网络结构 NeuralCut, 在三部图上使用图神经网络, 用于对前视专家进行模仿学习, 并在各种基准测试集中显示了对标准割平面选择策略的改进. 针对快速找到混合整数线性规划最优解的启发式算法, Khalil 等^[34] 首次研究了在分枝定界过程中何时何处运行何种启发式算法以优化求解器的总体性能, 将确定一系列启发式方法在分枝定界树上何时何处运行的问题, 转化为针对单个启发式算法, 确定其运行的节点集的问题, 最终使用逻辑回归的方法学习一个可以预测启发式算法是否执行的二进制分类器. 相对于最先进的混合整数线性规划求解器的原始性能, 在标准实例上, 此方法将性能提高了 6%, 而在一系列困难实例上, 此方法将性能提高了 60%.

与混合整数线性规划的研究思路类似, 线性规划单纯形算法也涌现出了一些基于机器学习的研究方法. Adham 等^[1] 在前端学习阶段给出了两种人工设计的特征提取方式, 第一种特征是基于变量约束图和系数的统计信息, 第二种特征设计基于截断的奇异值分解 (truncated singular value decomposition) 实现了维数约减. 在这两种特征构造的基础上, 基于监督学习的方法研究了自适应于实例的转轴规则, 基于提升 (boost) 树和神经网络从 5 个经典转轴规则中学习最适用于当前待求解实例的转轴规则, 相比于这 5 种传统转轴规则, 学习的转轴规则使得单纯形法转轴迭代次数有了一定的减少, 但是方法的泛化性有待进一步比较和研究. Chen 等^[13] 从变量节点二部图出发, 证明了对于线性规划问题, 一定存在一个图神经网络可以估计问题的可行性、问题的最优解以及问题最优解下的目标值, 并将其推广到了混合整数线性规划问题中 (参见文献 [14]). 然而上述基于监督学习的方法往往以几种经典的传统的转轴规则为标签, 学习到的转轴规则的性能完全受限于监督信号, 这种思路无法为一般的线性规划实例找到最优转轴路径. Li 等^[44] 恰好以此为切入点, 针对一般的线性规划实例^[46] 将单纯形转化为了一种伪树结构, 从而使 Monte Carlo 树搜索方法适用于单纯形法, 同时避免转轴过程中的基循环 (如图 3 所示), 最终设计出了单纯形算法的最优转轴规则. 与这个思路类似, Liu 等^[53] 基于预先找到的最优解和 Steepest-edge 规则的检验数, 提出了两个转轴专家规则, 在一系列实验中可以生成比经典

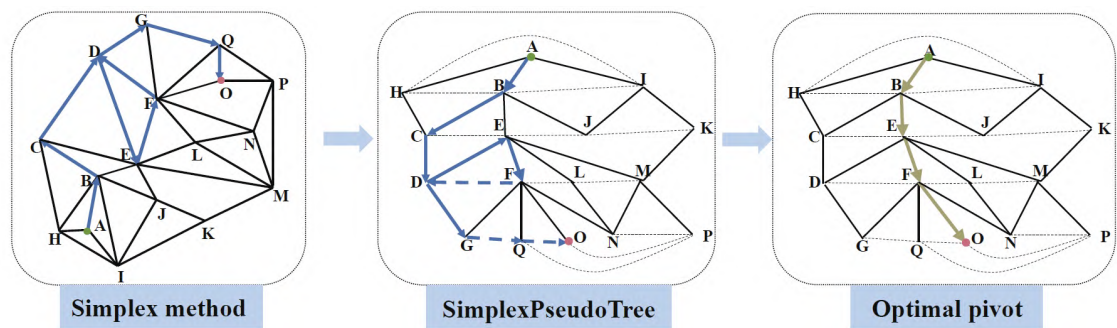


图 3 (网络版彩图) 单纯形算法的伪树结构^[44]

转轴规则更短的转轴路径. 后续, Liu 等^[53] 在 Klee-Minty 立方体的变体上证明了这两个专家转轴规则所对应的转轴路径长度上界为变量个数 n . 与 Monte Carlo 树搜索 (Monte Carlo tree search rule, MCTS) 规则相比, 时间上有了一定的提升, 但是无法保证转轴规则的最优性, 转轴路径长度的界相较于 MCTS 规则也更松弛一些.

除此之外, 0-1 整数规划问题也是混合整数线性规划的特殊情形, 布尔满足性问题 (Boolean satisfiability problem, SAT) 就可以建模为 0-1 整数规划问题. 在其求解过程中既要保持搜索的移动性, 又需要保证解的质量, 这构成了一个探索 - 利用的困境. 近年来, 使用 Monte Carlo 树搜索 (MCTS) 方法在一定程度上解决了这个问题. 在此基础上, Keszocze 等^[32] 将布尔满足性问题表述为树搜索结构, 通过可视化生成的搜索树, 进一步分析了通过添加 CDCL (conflict-driven clause learning) 扩展 Monte Carlo 树搜索算法的行为, 并在分析的基础上引入了多种启发式方法, 以提高学习子句的质量. 通过使用 MCTS 求解器作为预处理器, 可以利用学习到的子句来提高现有求解器的性能. 沿着这种思路, Chen 等^[12] 用退火的模式来编码布尔满足性问题, 将量子退火和 Monte Carlo 树搜索算法结合起来设计了 QZero 算法. 在 3-SAT 实例上实验发现, 即使在退火时间短的情形下, 也能高效地完成退火过程. 考虑到现有研究针对不同类型的布尔满足性问题提供了不同的算法, 缺乏一个通用的框架. Li 等^[45] 构建了 4 类布尔满足性问题统一的 0-1 整数规划表示 (如图 4 所示), 并在此基础上构造了适合布尔满足性问题的强化学习模型, 最终提出了一种统一求解框架 DCSAT, 用于解决 Max-SAT、加权 Max-SAT、PMS (partial Max-SAT) 和 WPMS (weighted partial Max-SAT) 等不同类型的布尔满足性问题. 不同于基于 Monte Carlo 树搜索的算法设计思想, Kurin 等^[41] 为布尔满足性问题提出了一个分枝启发式算法 Graph-Q-SAT, 它使用基于值的强化学习训练, 用二部图结构表示布尔满足性问题实例,

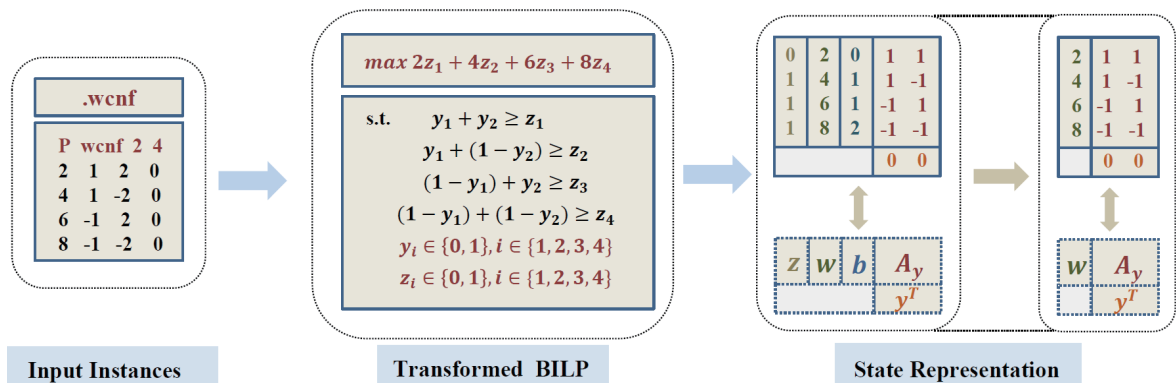


图 4 (网络版彩图) 四类 SAT 问题的统一表示^[45]

利用图神经网络进行函数逼近. Kurin 等^[41] 使用强化学习接口与 MiniSat 求解器训练 Graph-Q-SAT, 可以将求解所需要的迭代次数减少一半甚至更多. 此外, Graph-Q-SAT 还可以推广到不满足的布尔满足性问题实例, 以及比它训练的变量多 5 倍的实例上. 类似地, Selsam 等^[73] 将布尔满足性问题编码为一个无向图, 每个文字有一个节点, 每个子句有一个节点, 每个文字和它出现的每个子句之间有一条边, 每对互补文字之间有不同类型的边. 最终提出了一个消息传递神经网络 NeuroSAT, 它是一个端到端求解布尔满足性问题的图神经网络. 后续, 为了改进最先进的技术, Selsam 等^[72] 还训练了一个简化的 NeuroSAT 架构, 基于此改进的求解器获得了明显的性能提升. Sun 等^[78] 提出了另一种改进, 并推测 NeuroSAT 的局限性主要是由于学习通用 SAT 求解器的困难, 因此通过训练另一个分类器 (称为 NeuroGIST) 来评估这一假设, 实验结果表明, 在高度结构化的问题上, 新模型能够比原始的 NeuroSAT 表现得更好. 受最近图像领域的启发, Duan 等^[19] 从“对比学习”的视角重新审视了有监督的 NeuroSAT, 通过标签保留扩展生成每个 (未标记的) SAT 实例的多个视图, 然后通过最大化同一实例的不同视图表示之间的一致性 (正对), 同时最小化不同实例表示之间的一致性 (负对), 训练 NeuroSAT 的图神经网络编码器. 与完全监督的 NeuroSAT 方法相比, 这些表征可以用更少的标记数据进行微调, 以产生同样准确的满意度预测, 这种表征更容易转移到来自未知领域的更大问题. Yolcu 等^[92] 选择合取范式 (conjunctive normal form, CNF) 的因子图表示, 如图 5 所示, 得到了一个具有两个节点类型 (变量和子句) 和两个边类型 (正极性和负极性) 的无向二部图. 在随机局部搜索算法中加入图神经网络作为变量选择启发式算法, 考虑不同类别的布尔可满足性问题, 并对每个类别学习专门的启发式方法, 提出了一种基于深度强化学习从零开始学习的布尔满足性问题求解启发式算法. Cameron 等^[7] 将有 n 个子句和 m 个变量的原始布尔满足性问题编码为 $n \times m \times 2$ 维的稀疏张量输入到预测模型, 这种学习方法取得了更好的性能, 并且只需要线性时间来评估. Kumar 等^[40] 提出了一种从上下文示例中学习 Max-SAT 问题的新模式, 引入了硬约束和软约束学习 (hard and soft constraint learning, HASSLE), 利用语法指导的合成技术, 将学习编码为优化问题, 并使用高效求解器进行求解. 然而这只适用于特定类型的布尔满足性问题, 缺乏一个通用的框架 (参见文献 [33]).

3.1.2 基于神经网络的特征提取

对于最小化单纯形算法的转轴路径长度, 除了寻求最优的转轴规则, 还可以从初始解出发, 寻求尽可能接近最优解的初始解. 对于单纯形算法, 初始解对于转轴路径的长度至关重要, 一个好的初始解, 可以显著减小单纯形迭代次数, 毕竟最好的初始解就是最优解. Li 等^[49] 针对寻找初始解的问题, 将线性规划编码为变量节点二部图的结构, 并放入图神经网络中进行特征提取, 然后通过聚类的方法将经过图神经网络得到节点嵌入进行分块划分, 作为前端学习的最终结果 (如图 6 所示). 然后将其作为输入, 利用指针网络 (pointer network, Ptr-Net) 对线性规划实例的列变量进行重新排序, 利用强化学习的方

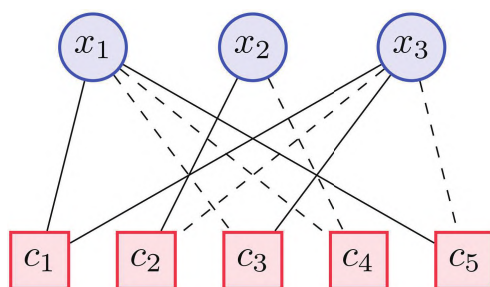


图 5 (网络版彩图) CNF 的因子图表示^[92]

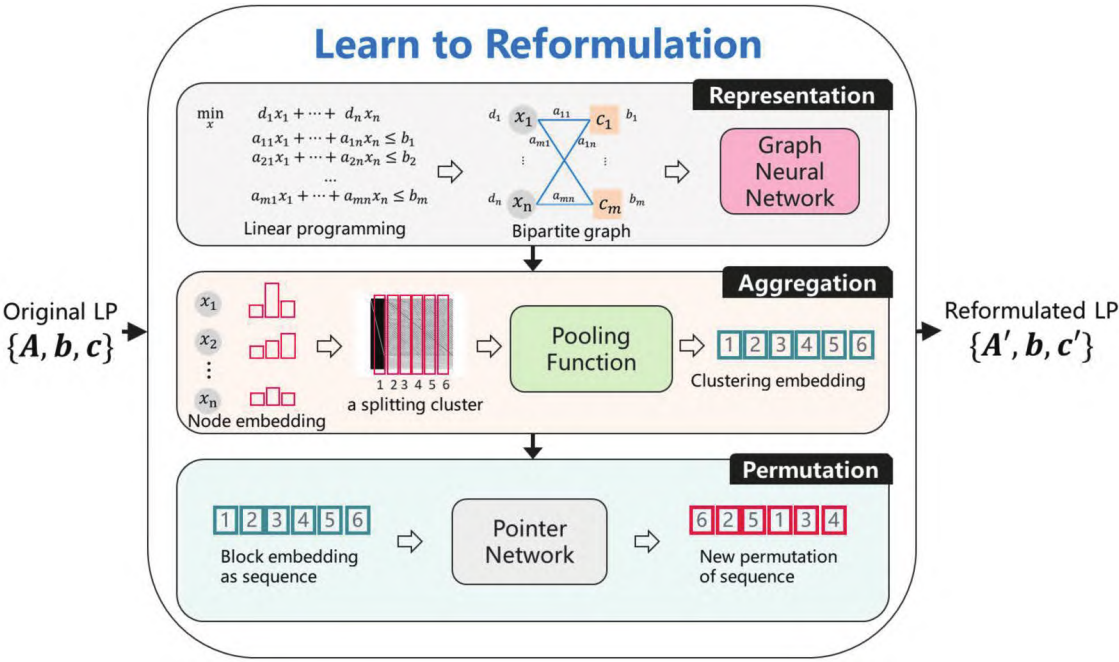


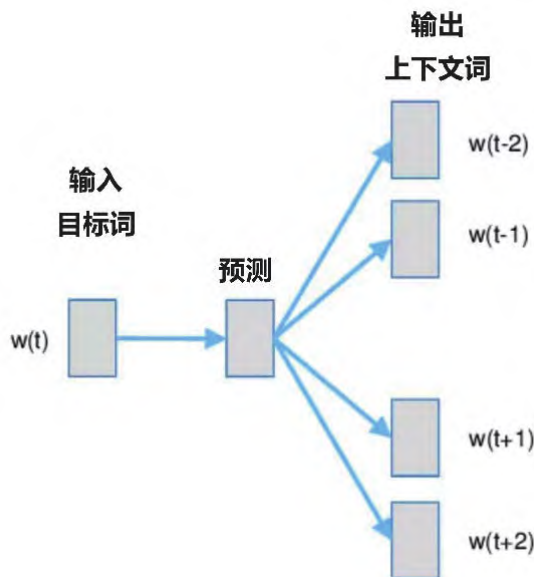
图 6 (网络版彩图) 线性规划的分块编码模式 [49]

法训练策略网络,使得重新排序后的初始可行基距离最优基的距离更近(所需要的转轴次数更少)。他们在 3 个混合整数线性规划问题集(通过放松其整数约束来得到线性规划实例)上验证了此重新排序方法的有效性,即可以用更少的转轴次数来求解线性规划实例。对于布尔满足性问题,Amizadeh 等^[5]提出了在有向无环图上的学习问题,将电路可满足性问题(circuit satisfiability problem)表述为一个有向无环图,基于一个丰富的嵌入架构用于编码问题结构,以及一个可微分的训练过程用于模拟强化学习,提出了一个可以学习求解电路可满足性问题的神经网络框架。实验结果表明,与 NeuroSAT 方法^[73]相比取得了更好的泛化性能。Lodi 等^[55]提出了一种将组合优化问题的求解与机器学习相结合的方法,首先使用监督学习预测在当前解下候选设施点集合内需要改变的设施的比例,接着给问题添加一个关于该比例的线性约束,然后利用传统求解器来得到高质量解。

广义跳字模型(generalized skip-gram)^[60]是图嵌入的图表示学习的一种方法,它来自于自然语言处理中的 Word2Vec 的跳字模型(skip-gram),对于给定的输入单词,预测其上下文。跳字模型是一种常见的神经网络模型,用于自然语言处理任务中的单词表示学习。它的作用是在给定一个单词时,预测它周围的单词。跳字模型的输入是一个中心单词,输出是该单词周围的若干个单词。跳字模型的核心思想是,用神经网络将每个单词映射成高维空间的向量,以更好的方式来抓住单词之间的语义关系。在这种情形下,模型的目标是使得对于每个中心词,其预测的上下文单词能够最好地反映其周围的实际单词,如图 7 所示。Word2Vec 是从大量文本语料中以无监督的方式学习语义知识的一种模型,它其实是通过学习文本,使用词向量的方式表征词的语义信息,即通过一个嵌入空间使得语义上相似的单词在该空间内距离很近。嵌入其实是一个映射,将单词从原先所属的空间映射到新的多维空间中,也就是将原先词所在空间嵌入到一个新的空间中去。

广义跳字模型是从自然语言处理中用于嵌入单词的跳字模型扩展而来的。广义跳字模型依赖于节点 u 的邻域 N_u 来学习 u 的嵌入向量 h_u ,目标是 N_u 中节点以 u 为条件的最大似然估计:

$$\max P(v_1, v_2, \dots, v_{|N_u|} | u), \quad v_i \in N_u. \quad (3.1)$$

图 7 (网络版彩图) Skip-gram 跳字模型^[60]

假设条件独立, 则

$$P(v_1, v_2, \dots, v_{|\mathcal{N}_u|} | u) = \prod_{v_i \in \mathcal{N}_u} P(v_i | h_u),$$

其中

$$P(v_i | h_u) = \frac{h_{v_i}^T h_u}{\sum_{v \in G} h_v^T h_u}.$$

我们希望优化

$$\max \sum_{v_i \in \mathcal{N}_u} \log P(v_i | h_u) = \max \sum_{v_i \in \mathcal{N}_u} \log \frac{h_{v_i}^T h_u}{\sum_{v \in G} h_v^T h_u}. \quad (3.2)$$

由于计算公式 (3.2) 的分母是非常耗时的, 因此已经提出了许多优化技术. 负采样是最著名的技术之一 (参见文献 [96]). 具体地, u 的邻域 \mathcal{N}_u 中的节点被视为 u 的正样本, 不在 \mathcal{N}_u 中的节点被认为是 u 的负样本. 然后, 可以如下实现公式 (3.2) 中的最大似然估计:

$$\max \log \sigma(h_v^T h_u) + \sum_{i=1}^K \mathbb{E}_{\bar{v} \sim p_n} \log \sigma(-h_{\bar{v}}^T h_u), \quad (3.3)$$

其中 v 是 u 的一个正样本, \bar{v} 是一个负样本, p_n 是负样本的概率分布, $\bar{v} \sim p_n$ 表示从概率分布 p_n 中对节点进行采样, k 是负样本数, σ 是 S 形激活函数, \mathbb{E} 是期望值.

图嵌入的图表示学习另外一个代表性方法是自编码方法 (autoencoder). 自编码模型与跳字模型类似, 唯一的区别在于设置编码器和解码器时, 解码器的输出尽量与输入接近, 而跳字模型的输出是输入的上下文. 自编码方法由一个编码器和一个解码器组成. 对于与图有关的组合优化问题, 编码器将图中的节点编码为 d 维嵌入向量, 解码器使用节点嵌入 (如指针网络) 来预测组合优化问题的解. 编码器为 $\text{enc} : \mathbb{R}^{d'} \rightarrow \mathbb{R}^d$, 解码器为 $\text{dec} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$, 损失函数为

$$\mathcal{L} = \sum_{(u,v) \in \mathcal{D}} \text{dist}(\text{dec}(h_u, h_v), \text{sim}(u, v)), \quad (3.4)$$

其中, $\text{sim} : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ 是一种相似性度量指标, 在实际应用时有多种取法, 可以设为余弦相似性、向量内积、Euclid 距离等. 目标函数是将重建损失降至最低:

$$\mathcal{L} = \sum_{u \in G} \|\text{dec}(h_u), x_u\|_2^2. \quad (3.5)$$

编码器和解码器可以通过不同类型的神经网络来实现, 如多层感知机 (multilayer perceptron, MLP) 或循环神经网络.

Du 等^[18] 设计了一种基于新型图神经网络和深度强化学习的新型模型 Vulcan, 将高维图结构数据 (即路径变化信息) 转换为低维向量表示. 第一步是对输入图进行预处理以获得初始顶点权重. 以顶点 v_2 为例, 编码器网络集成当前顶点 v_2 状态和权重信息, 生成 p 维隐向量表示; 接下来, 处理器网络捕获向量之间的变化, 并将它们拼接到 p 维向量后面; 最后, 解码器网络考虑整个图的状态和当前顶点 v_2 , 生成顶点的值 Q . 对于给定的 Steiner 树问题实例, Vulcan 对其路径相关信息进行编码, 并将编码后的图发送给 DDQN (double deep Q network) 来进行求解. 由于 Steiner 树问题是非结构化数据, 顶点的值取决于许多因素, 如顶点度、图结构、端点位置等. Du 等^[18] 希望在图中添加新节点时, 评估函数 Q 能有效结合当前状态的上下文信息并计算出值, 为此设计了一个编码器 - 处理器 - 解码器架构, 如图 8 所示 (颜色条表示具有不同维数的特征, 红色顶点是终端). 除了 Steiner 树问题之外, Vulcan 还可以通过将 NP- 困难问题 (如布尔满足性问题、最小顶点覆盖问题) 转化为 Steiner 树问题来进行求解. Cappart 等^[8] 提出通过奖励塑造来解决难以满足的问题 (如最大独立集问题和最大割问题), 引入的奖励信号有两个具体的层次分明的目标: 找到一个可行的完整的解, 在可行解中找到最小化目标函数的解. 一旦没有可用的行动, 解的构建就会停止, 这对应于一个不可行的部分解, 因此得到的每一个完整解都能保证是可行的. 这是第一篇应用机器学习直接改进由组合优化问题的通用边界机制获得的松弛界的论文.

3.2 后端学习方法

前后端组合学习的后端学习分为基于完整解迭代优化方法和基于部分解迭代构造完整解两类方法.

3.2.1 基于完整解迭代优化

针对旅行商问题, Zheng 等^[94] 使用 Q 学习得到的参数值代替了 LKH (Lin-Kernighan-Helsgaun) 算法中的 α ; 随后 Zheng 等^[95] 又将上述算法作为一个改进算子, 设计了一种遗传算法, 并在大小至多为 85,900 的旅行商问题实例中做了实验, 验证了提出的 RHGA (reinforced hybrid genetic algorithm)

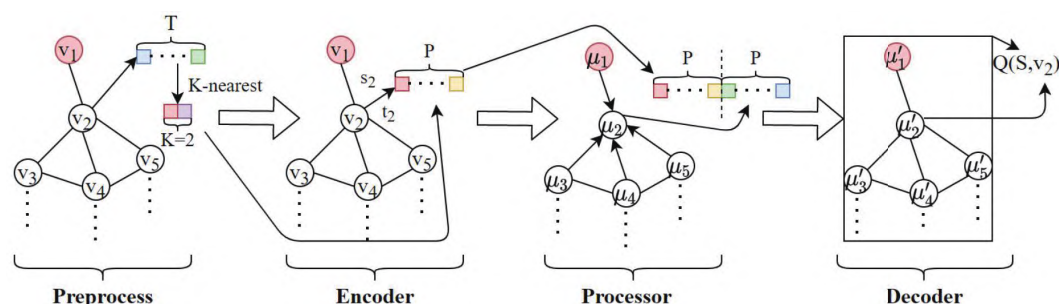


图 8 (网络版彩图) STP 问题的图嵌入模型^[18]

具有优异的性能. Wang 等^[85] 开辟了一个博弈视角下的求解路径离散优化问题的学习方法, 此方法不但大大提升了求解方法的泛化性能, 而且可以作为一个单独的求解框架与其他算法相结合, 从而增强学习类求解算法的泛化性能. 该方法利用二人零和博弈框架来训练求解器和数据生成器, 博弈框架会大量输出求解器好的决策响应, 混合这些响应输出一个组合模型用最少的探测来对抗生成器, 从而获得不同旅行商问题任务上的泛化性能, 如图 9 所示. 该方法在不同类型和规模的旅行商问题上进行实验, 取得了优于其他深度学习方法的性能, 而且不容易产生过拟合, 在公开数据集 TSPLIB 上, 该方法优化间隙超过了基线模型获得 12% 的提升.

对于其他路径优化问题, Chen 等^[11] 设计了一种大邻域搜索 (large neighborhood search, LNS) 破坏算子, 即利用图卷积神经网络对车辆路径问题 (vehicle routing problem, VRP) 可行解编码, 利用近端策略优化 (proximal policy optimization, PPO) 算法训练得到一个网络, 其输出为一个顶点和一个参数, 以该参数一定比例为半径, 抹去该点近邻的其他顶点及其自身所连边, 再通过启发式算法来重新结合各个分离的部分从而得到一个新的可行解. Kim 和 Park^[38] 提出了一种新的分层问题求解策略 LCP (learning collaborative policies), 使用两个深度强化学习策略: 播种器和修正器. 播种器生成尽可能多样化的候选解, 同时致力于探索整个组合动作空间 (即分配动作序列), 实验中使用熵正则化奖励来训练播种器的策略, 以鼓励播种器找到不同的解. 另外, 修正器修改由播种器生成的每个候选解, 将整个轨迹划分为子路线, 同时修正每个子路线以使其行进距离最小. 大量的实验表明, 所提出的双策略协作方法比单策略深度强化学习框架在各种 NP- 困难路径优化问题 (包括旅行商问题、有容量限制的车辆路径问题等) 上都有改进.

对于线性规划的单纯形算法, Li 等^[44] 旨在能够为利用机器学习求解一般的线性规划实例^[46] 找到最优的转轴规则. Li 等^[44] 的思路是将单纯形转化为一种伪树结构, 从而使 Monte Carlo 树搜索算法适用于单纯形法, 在此基础上设计出单纯形算法的最优转轴规则. Li 等^[44] 以单纯形表为基础, 提出了两种动作空间定义和两个奖励函数定义, 构造了 4 种强化学习模型. 最终, 基于 Monte Carlo 树搜索算法提出了 MCTS 规则, 为线性规划单纯形算法找到了最优的转轴规则, 如图 10 所示. 基于 Wiener-Khinchin 大数定律, Li 等^[44] 更进一步地证明了在单纯形算法的最坏情形下, 即当可行域的顶点数为 C_n^m 时, MCTS 规则可以生成所有最优转轴路径, 并且转轴路径长度为 $\min(m, n - m)$, 其中 m 表示约束个数, n 表示变量个数. 类似地, Li 等用 0-1 整数规划模型给出了四类布尔满足性问题统一的表示形式, 并在此基础上提出了恰当的强化学习模型, 将 Monte Carlo 树搜索方法应用于布尔满

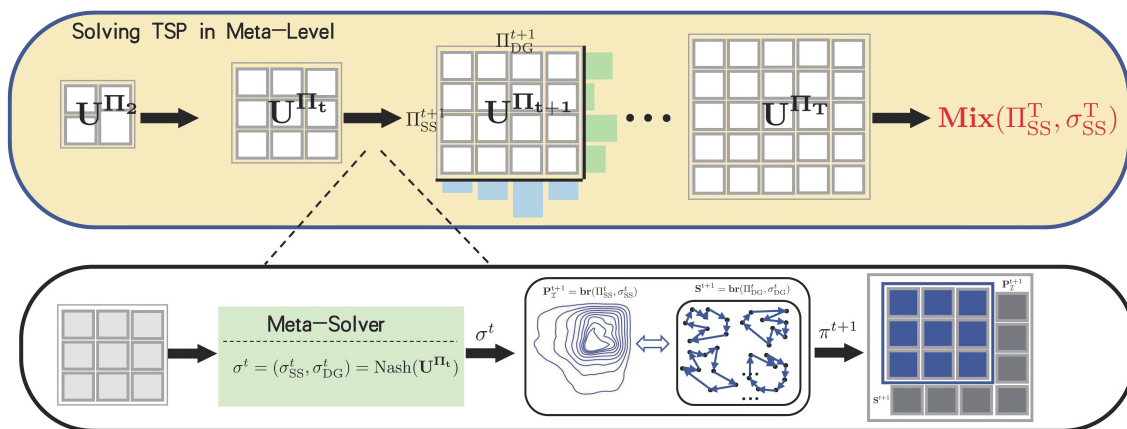


图 9 (网络版彩图) 基于二人零和博弈的 TSP 求解框架^[85]

足性问题, 最终提出了一种统一求解框架 DCSAT^[45], 用于解决 Max-SAT、加权 Max-SAT、PMS 和 WPMS 等不同类型的布尔满足性问题. 基于 Wiener-Khinchin 大数定律, 证明了 DCSAT 可以找到所有的最优布尔分配. 通过实验验证, 该范式可以减少不必要的搜索空间, 从而找到问题的最优布尔分配, 此外 DCSAT 还可以为布尔满足性问题的监督学习方法提供不同的标签.

基于完整解迭代优化中也有一些工作是通过使用分类技术来预测输入图中节点的类标签. 一般而言, 对于图上的组合优化问题, 这一类中的大多数工作都使用分类技术来预测输入图中节点的类标签. 对于图 G , 预测结果是 $|V| \times K$ 的矩阵 Y , 其中 K 是类的数量. 矩阵 Y 的第 u 行 y_u 是节点 u 的预测结果, 其中 $y_u[i]$ 是 u 属于第 i 类的概率, $1 \leq i \leq K$. 例如, 对于最小顶点覆盖问题, 分类是二进制的 (即 $K = 2$), 并且 $\{u \mid y_u[1] > y_u[0]\}$ 是预测的解. 对于图划分问题, K 是划分部分的数量, 节点 u 被分类为具有最大预测概率的那一部分. 有一些工作可以预测输入图的分值. 例如, 对于图相似性问题, 预测两个图之间的相似性得分.

对于图划分问题, GAP (generalizable approximate graph partitioning) 方法^[64] 是一种计算图的平衡划分方法, 由图嵌入模块和图划分模块组成, 图嵌入模块使用图神经网络模型来确定输入图的嵌入, 图划分模块使用多层感知机来预测节点的划分. GAP 的体系结构如图 11 所示. 该方法在小图上训练的 GAP 可以泛化到较大测试的图上. 损失函数包括标准化的划分尺寸和划分的平衡度. GAP 的损失函数有两个组成部分. 第一部分是最小化分区的标准化切割尺寸:

$$\mathcal{L}_1 = \sum_{i=1}^K \frac{\text{cut}(V_i, \bar{V}_i)}{\text{vol}(V_i)}, \quad (3.6)$$

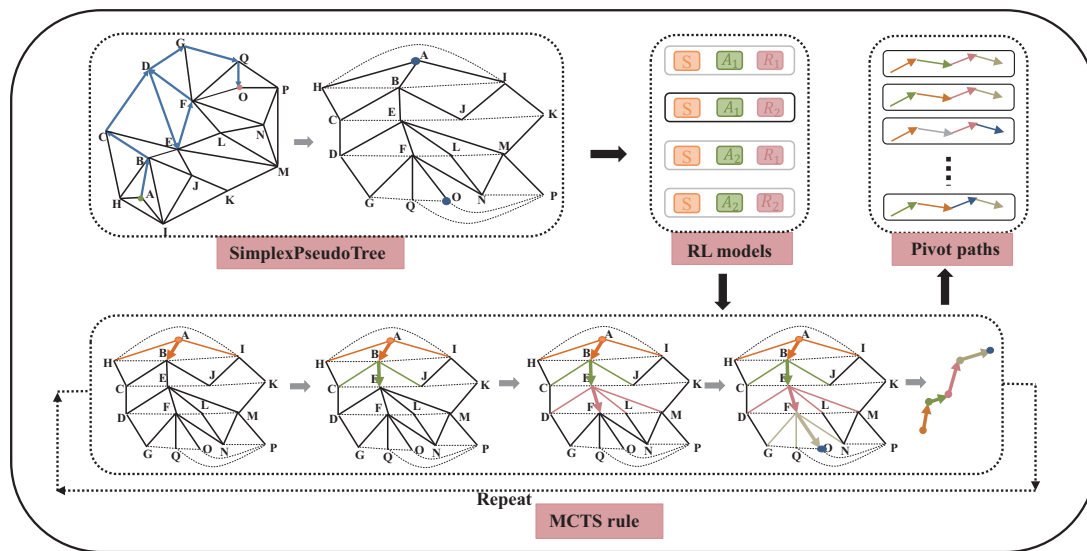


图 10 (网络版彩图) MCTS 规则整体框架^[44]

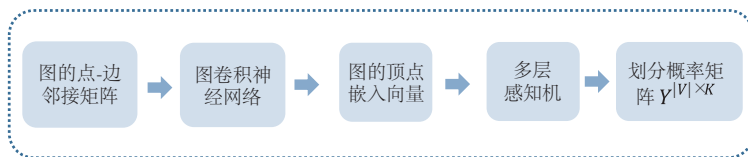


图 11 (网络版彩图) GAP 流程图^[64]

其中 \bar{V}_i 表示不在 V_i 中的节点, $\text{cut}(\bar{V}_i, V_i)$ 表示两端分别在 V_i 和 \bar{V}_i 的边的数量, $\text{vol}(V_i)$ 表示 V_i 中节点的总度数. 损失函数的第二个部分是最小化与平衡分区的距离:

$$\mathcal{L}_2 = \sum_{i=1}^K \sum_{u \in V} \left(Y[u, i] - \frac{|V|}{K} \right)^2, \quad (3.7)$$

其中 $Y[u, i]$ 表示节点 u 划分到 V_i 中的概率, $\frac{|V|}{K}$ 是均衡划分一个部分的平均大小. GAP 的目标函数为

$$\min \mathcal{L} = \min \alpha_1 \mathcal{L}_1 + \alpha_2 \mathcal{L}_2 = \min \alpha_1 \sum_{i=1}^K \frac{\text{cut}(V_i, \bar{V}_i)}{\text{vol}(V_i)} + \alpha_2 \sum_{i=1}^K \sum_{u \in V} \left(Y[u, i] - \frac{|V|}{K} \right)^2, \quad (3.8)$$

其中, $0 \leq \alpha_1 \leq 1$, $0 \leq \alpha_2 \leq 1$, $\alpha_1 + \alpha_2 = 1$.

3.2.2 基于部分解迭代构造完整解

该学习方法迭代地扩展部分解, 在每次迭代中, 都会将解的一部分 (如一个节点/边或者一个物体) 添加到部分解中. 大多数现有的工作使用基于序列模型的方法和基于强化学习的方法来迭代扩展部分解. 基于序列模型的方法一般为监督型, 而基于强化学习的方法一般是无监督型的.

Joshi 等^[30] 提出以监督的方式使用残差门控图卷积网络来求解旅行商问题, 该模型不输出有效的环游, 而是输出每条边属于环游的概率, 最后的回路使用贪婪解码或波束搜索计算. 为了增强监督学习方法的泛化能力, Fu 等^[20] 尝试 (以监督的方式) 训练一个小规模模型, 可重复用于为任意大规模的旅行商问题实例构建热图, 然后将热图输入到强化学习方法中, 以指导搜索高质量解, 如图 12 所示. 具体而言, 先通过离线学习的方式, 基于图采样从原始图中提取若干个子图并使用预训练模型为每个子图构建对应的子热图; 然后尝试将所有的子热图合并成一个完整的热图; 最后基于在线学习的方式, 使用 Monte Carlo 树搜索方法在合并热图中存储的信息的指导下搜索高质量的解. 大量实例 (多达 10,000 个顶点) 的实验结果表明, 这种新方法明显优于现有的基于机器学习的其他算法, 并且显著提高了训练模型的泛化能力. 对于更广泛的路径优化问题, Kool 等^[39] 将学习到的启发式算法与动态规划相结合, 利用深度神经网络派生的策略对动态规划状态空间进行优先级排序和限制, 提出了深度策略动态规划 (deep policy dynamic programming, DPDP) 算法. 在旅行商问题、车辆路径问题和带

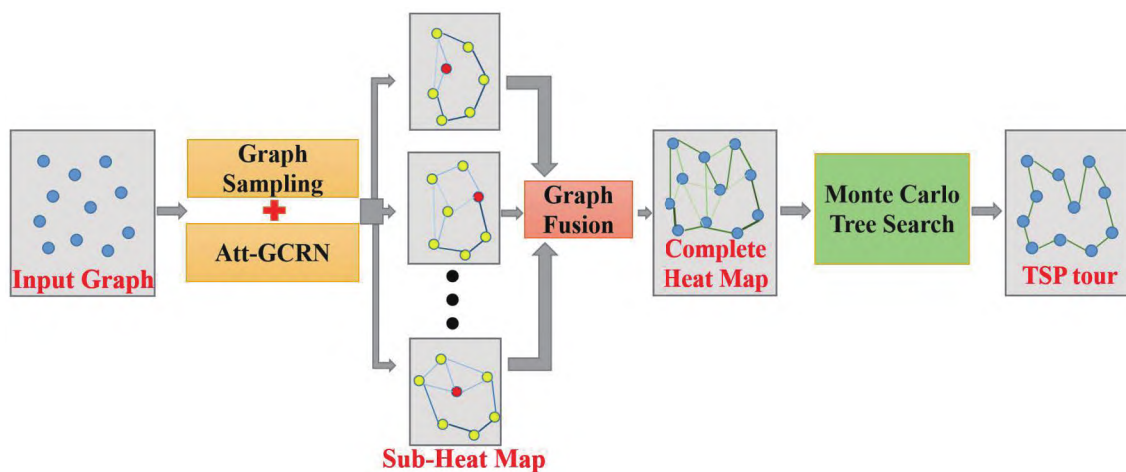


图 12 (网络版彩图) 基于子热图合并的旅行商问题求解框架^[20]

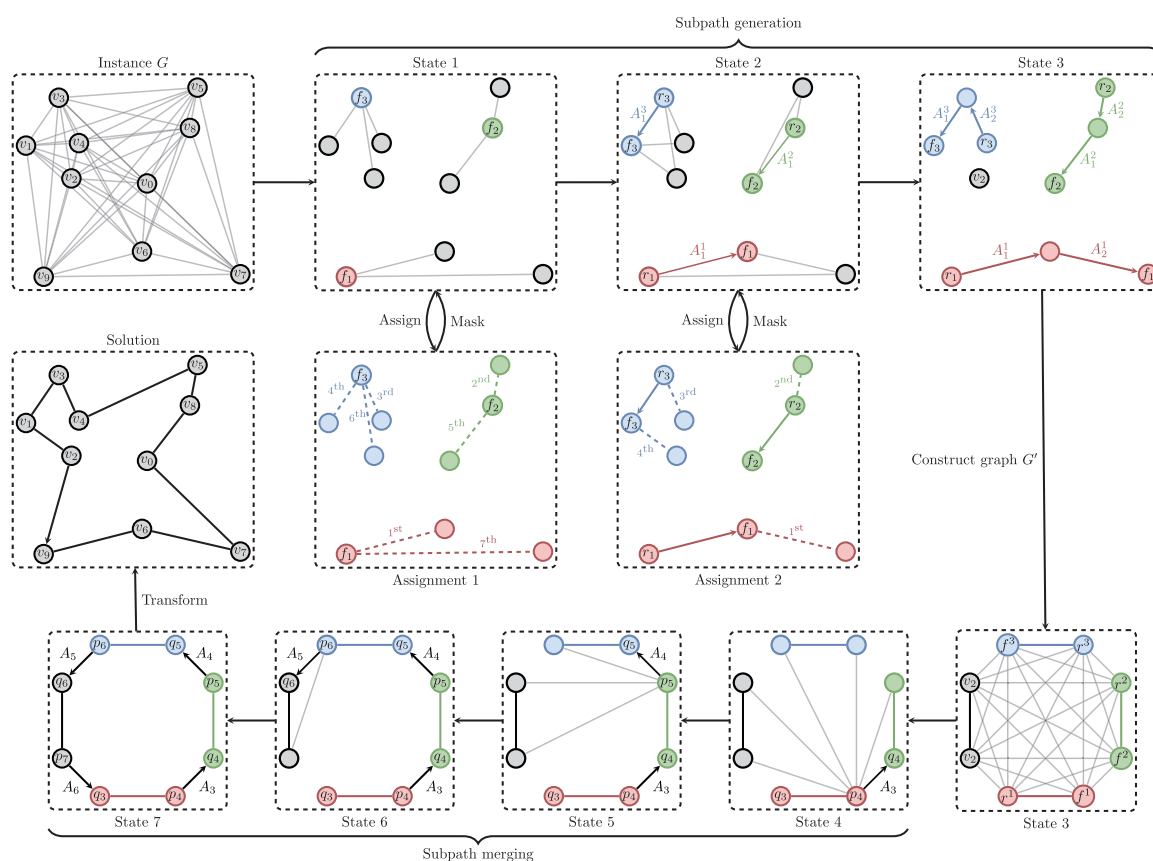
时间窗的 TSP 问题 (TSP with time windows, TSPTW) 上的评估表明, DPDP 算法提高了动态规划算法的性能, 使其可以与 LKH 等强大的算法竞争, 同时也优于大多数其他基于神经网络的方法. 此外, Wang 等^[86] 基于 Alphago Zero 提出了一个新的框架 (称为 OmegaZero), 不需要专家经验或标签数据而是通过自我对弈进行训练. 学习过程分为两个阶段, 第一阶段使用图注意力网络和门控循环单元 (gated recurrent unit, GRU) 来学习节点表示和记忆历史轨迹, 第二阶段基于 Monte Carlo 树搜索和深度强化学习来搜索解空间并训练模型. 针对无人机旅行商问题 (traveling salesman problem with drone, TSP-D), Nguyen 等^[65] 提出了一种新的基于 Monte Carlo 树搜索算法的求解思路, 通过在决策空间中随机取样并基于最终结果构建搜索树来寻找最优决策. 在 TSPLIB 公开数据集上测试, 实验表明该方法为 23 个实例提供了新的最优解, 求解质量平均提高了 12%.

Cappart 等^[9] 提出了一种基于深度强化学习和约束规划的混合方法来求解组合优化问题, 通过动态规划的范式将两者结合起来. Chalumeau 等^[10] 进一步扩展了这一思想, 提出了一种新的约束规划求解器 SeaPearl, 它支持机器学习例程, 以便使用强化学习来学习分枝决策. 为此, 他们将约束规划表示为一个简单的无向三部图, 这种表示的优点是对任何组合问题都具有通用性, 并已被 Song 等^[77] 用于学习变量排序启发式算法. 然而, 目前的挑战是生成的图的大小可能令人望而却步, 使训练更加烦琐.

多智能体的思路也被越来越多地应用到旅行商问题的求解中, Alipour 和 Razavi^[4] 于 2015 年率先提出了一种基于多智能体强化学习 (multiagent reinforcement learning, MARL) 的对称旅行商问题求解算法. 在多智能体系统中, 每个智能体都是一个具有个人记忆和行为的自主个体, 以此来构造环游. 然后将每个智能体构造的环游通过两元素优化 (2-optimization, 2-opt) 局部搜索启发式算法进行提升, 以在合理的时间内达到最优或接近最优解. Yang 和 Szeto^[90] 提出了一种基于多智能体系统的求解 N 个城市固定资源下旅行商问题的方法. 将城市划分为 m 组, 并分配单个智能体使用旅行商问题求解器搜索覆盖分配组中所有城市的最短路径. 后续, Shi 等^[74] 也基于多智能体强化学习的思想, 提出了一种利用合作多智能体强化学习来求解旅行商问题的新方法 CARSS (cooperative attention-guided reinforcement subpath synthesis), 如图 13 所示. CARSS 将旅行商问题的求解过程分解为两个不同但相互协同的步骤: “子路径生成” 和 “子路径合并”. 前者采用协作式多智能体强化学习框架, 利用多个智能体迭代生成子路径. 后者将生成的子路径逐渐合并形成一个完整的循环. 实验表明, CARSS 与单智能体方法相比, 可以容纳近 2.5 倍大的训练图, 并显示出可以扩展到更大规模的潜力. 此外, 对于 1,000 个顶点的旅行商问题实例, CARSS 大大减少了测试时间并提升了解的质量. 除了旅行商问题, 具有软时间窗的多车辆路径问题 (multi-vehicle routing problem with soft time windows, MVRPSTW) 传统的求解方法都是基于启发式规则的, 随着当前物流需求的快速增长, 传统的方法陷入了计算效率和求解质量之间无法权衡的困境. 为了有效地解决这一问题, Zhang 等^[93] 提出了多智能体注意力模型 (multi-agent attention model, MAAM), 将车辆路径问题视为车辆行程生成过程, 提出了一种具有关注层的编码器 - 解码器框架.

除了旅行商等路径优化问题, Kiarostami 等^[37] 对求解谜题展开了研究, 探讨了 Monte Carlo 树搜索算法在解决谜题中的适用性, 根据谜题的特征提出了一种通用的分类方法, 将每个谜题作为一系列基于数学公式的满足函数传递给算法, Monte Carlo 树搜索算法逐步解决一个难题, 直到函数完全满足提出的分类方式. Li 等^[51] 提出了一种监督学习框架用来研究最大独立集 (maximal independent set, MIS) 问题, 使用图卷积网络来估计图中每个顶点是否为最优解的一部分的可能性. 每个训练实例都与一个最优解标签相关联, 然后训练图卷积网络输出预测解. 在测试时, 这些预测被传递到树搜索和局部搜索中, 再将它们转换为可行的、潜在的高质量解. Li 等^[51] 以图卷积网络作为中心组件, 通过

训练来估计图中每个顶点是否为最优解的一部分的可能性, 在最大独立集、最小顶点覆盖 (minimum vertex cover, MVC)、最大团 (maximal clique, MC)、布尔满足性问题 4 个典型 NP- 困难问题上进行了评估, 实验结果表明该方法在一些 NP- 困难问题上与最先进的启发式算法相当. Afshar 等^[2] 将状态聚合步骤用于提取特征和构造状态, 基于 A2C (advantage actor critic) 算法训练策略, 提出了一种基于深度强化学习的构造型背包问题求解方法. 另外, Wang 等^[84] 提出了求解无容量约束的 P 中值问题 (P-median) 的多头交流图注意力的图神经网络模型, 采用 Reinforce 算法训练模型, 使用部分解添加策略. 此模型对求解无容量限制的 P 中值问题有好的性能, 实验结果表明在可接受的求解精度下, 此方法对于大规模问题时间减少有很大的优势. 许多图上的组合优化问题都可以被建模为 Markov 决策过程 (Markov decision process, MDP), 利用强化学习近似求解. Shi 等^[75] 对这类问题进行了统一, 在形式上可以定义为 (\mathcal{J}, f, μ) , 其中 \mathcal{J} 是图实例的集合, $f(G)$ 是图 $G = (V(G), E(G), w)$ 在 \mathcal{J} 中所有可行解的集合, $\mu(G, H)$ 表示 H 在 $f(G)$ 中与问题目标有关的度量. 以最小化问题为例, Shi 等^[75] 给出了其 Markov 决策过程的具体定义. 就整体任务而言, Shi 等^[75] 的目标是获得一个期望意义下回报最大化的策略, 该模型是一个求解图上组合优化问题的一般化模型, 求解得到的策略包含了所有该类问题的构造型启发式方法, 如求解最小生成树问题的 Prim 算法、求解旅行商问题的最近邻算法等都可以被视为该模型所得到的某种策略的贪婪解码方式. 考虑到上述定义的 Markov 决策过程的状态和动作空间极大, Shi 等^[75] 提出了 NeuroPrim 算法, 针对生成树问题的特性使用 Prim 算法的技巧来大大减小动作和状态空间, 从而可以通过强化学习高效地近似求解, 如图 14 所示. NeuroPrim 算法通

图 13 (网络版彩图) CARSS 求解框架^[74]

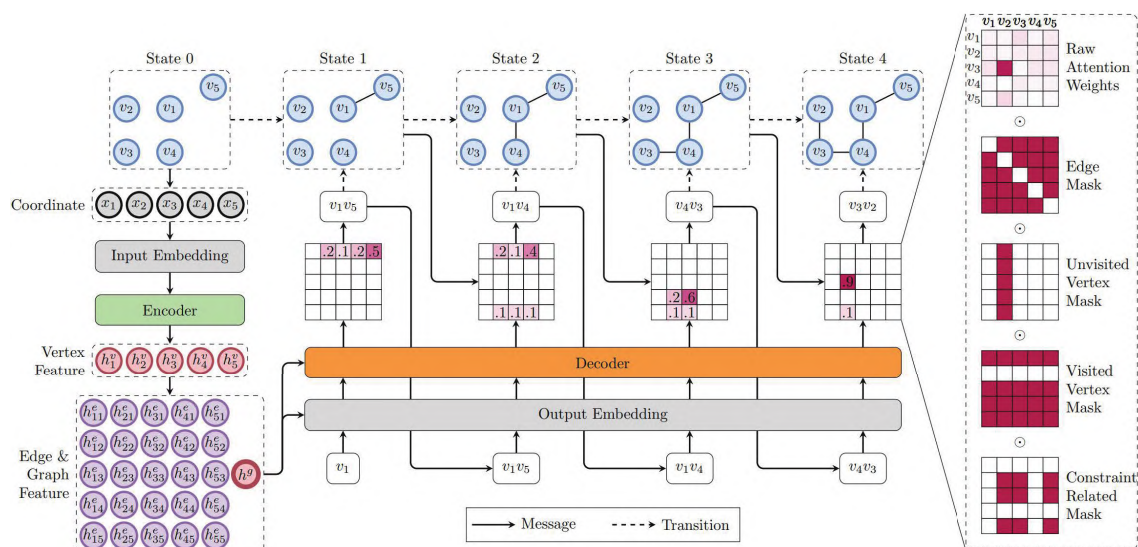


图 14 (网络版彩图) NeuroPrim 算法求解框架^[75]

过一个自注意力编码器来聚合顶点、图和边的特征作为输入; 通过自回归 (autoregressive) 的解码器得到参数化的策略, 利用过去时刻的动作状态信息来获得当前时刻的边选择概率; 最后基于带有 POMO 基准的 Reinforce 算法训练, 利用 Adam 优化器进行参数更新. 针对度约束最小生成树问题、最小路由成本生成树问题和图上 Steiner 树问题 (Steiner tree problem in graphs, STPG) 这三种不同的问题进行求解, 实验发现求得的解与基准解的间隙仅有 0.1%, 此模型保持了良好的泛化性. Ahn 等^[3] 引入了延迟 Markov 决策过程 (deferred Markov decision process) 的概念, 提出了一种新的深度强化学习方法来增强对大规模图的适用性. 在最大独立集问题上验证了在有限的时间预算下, 该方法可以在具有数百万个顶点的大规模图上优于传统的 MIS 求解器. 然而, 该方法仅限于局部可分解的问题, 即可行性约束和目标可以通过局部连通变量分解的问题.

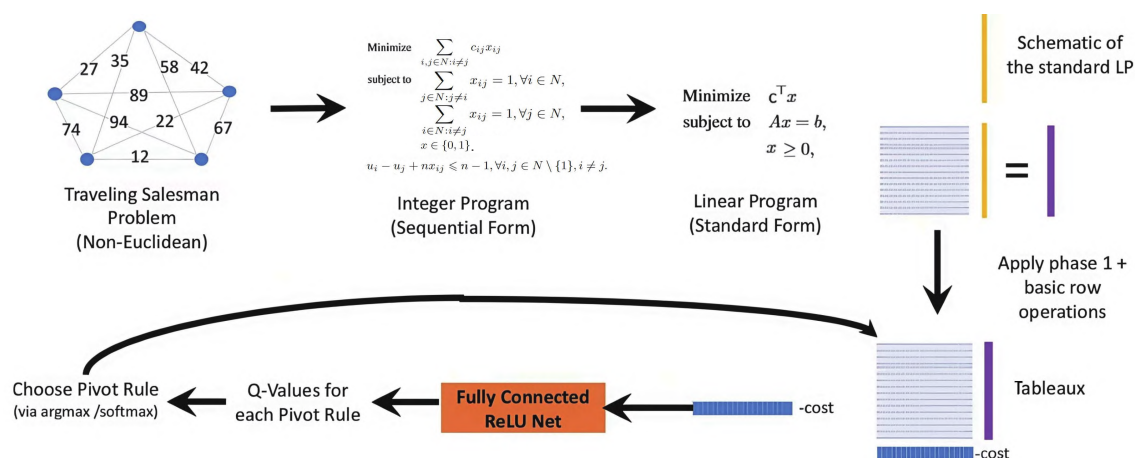
4 前后端融合学习

前后端融合学习同样分为基于完整解迭代优化和基于部分解迭代构造完整解两类方法.

4.1 基于完整解迭代优化

基于完整解迭代优化方法又分为神经网络学习和深度强化学习两类方法. 对于线性规划端到端的转轴规则, 最早的研究 DeepSimplex^[79] 以单纯形表为特征馈入后端学习的网络中, 基于监督学习和强化学习的方式分别进行了实现, 使得每次迭代可以从 Dantzig 规则和 Steepest-edge 规则中自适应地选择适用于当前求解状态的转轴规则, 如图 15 所示. 实验发现无监督的 Q 学习获得了与有监督相近的结果, 证明了 Q 学习在此问题上的潜力. 但是 DeepSimplex 在实验中并没有使用线性规划的标准集, 而是在 5 个节点的旅行商问题上进行实验, 通过加速 LP 松弛的求解进而加速旅行商问题的求解, 实验规模和最优标签的构造容易受到问题规模的限制, 这也是该方法可以进一步提升的一个切入点.

对于旅行商问题, 在研究初期大多采用的是监督型的学习方法. 2015 年, Vinyals 等^[83] 使用一个序列到序列的“指针网络”来映射二维点, 最终得到一个总长度较小的环游. 这是基于序列模型这一

图 15 (网络版彩图) DeepSimplex 方法框架^[79]

类方法中的一项开创性工作. 指针网络的编码器是一个以图 G 的节点为输入并输出 G 的嵌入的循环神经网络, 其中节点的顺序是随机选择的. 指针网络的解码器也是一个循环神经网络. 在每个时间步长中, 解码器计算输入节点上的注意力, 并选择具有最大注意力权重的输入节点作为输出. 指针网络使用监督学习进行训练, 因此需要近似最优解作为标签; 当旅行商问题实例难以求解而无法添加标签时, 这可能是一个限制因素. 与之不同的是, Prates 等^[68] 采用了图神经网络的方法, 在边与顶点间进行多次迭代通信, 以有监督的方式将图神经网络训练成一种有效的消息传递算法, 考虑了多至 105 个城市的小规模实例. Lemos 等^[43] 进一步扩展了这一思想, 设计了一个简单的图神经网络架构来进行图着色问题的决策, 实验发现这种模型在随机实例的训练上达到了很高的精度, 并且能够泛化到与训练时不同的图分布. 但是这种方法的一个重要缺陷是模型可能生成不可行的解 (违反一些约束), 需要设计适当机制来进行处理. Deudon 等^[15] 用城市坐标作为输入, 使用评论家 (critic) 网络来计算环游长度的基线, 通过强化学习训练神经网络预测城市排列的分布, 用 2-opt 启发式算法进一步改进了求解方法, 扩展了求解旅行商问题的神经组合优化框架. 实验发现, 该框架的性能通常与高性能启发式算法 (ORTools) 一样好, 并且当配备了一个简单的 2-opt 启发式算法时, 该框架可以优于这种启发式算法, 并在二维 Euclid 图上获得接近最优的结果. Li 等^[48] 介绍了一种新的深度学习方法来近似求解覆盖旅行商问题 (covering salesman problems, CSP), 以给定覆盖旅行商问题的城市位置作为输入, 使用无监督的强化学习方法进行训练, 设计一个深度神经网络模型来直接输出解. Li 等^[48] 在模型中采用多头注意力机制捕捉结构模式, 并设计一个动态嵌入来处理问题的动态模式, 一旦模型被训练, 就可以推广到各种类型的覆盖旅行商任务 (不同的大小和拓扑). 实验发现, 它比传统的启发式求解器运行速度快 20 倍以上. 这种方法与传统求解器相比, 非常适合实践中大多数具有挑战性的大规模快速决策的任务.

除旅行商问题之外, 其他组合问题也吸引了越来越多研究人员的兴趣. 自 2016 年起, 受 Milan 等^[61] 基于全连接神经网络求解图匹配问题的启发, Qin 等^[69] 和 Tang 等^[82] 从图像处理中的组合优化问题切入, 提出了基于机器学习的点集匹配算法^[80] 和图匹配算法^[81]. Tang 等^[80] 为点集匹配问题提出了一个端到端的网络模型, 即多指向型网络 (多指向型网络借助多标签分类的想法改进了指向型网络), 可以在每个时刻同时输入输出序列中多个元素, 如图 16 所示. 实验表明, 这种端到端的多指向型网络模型可以有效地处理大幅度平移、相似性和刚性等变换, 并且可以推广到 Delaunay 三角剖分等问题. 此外, Nazari 等^[63] 提出了一个端到端的框架来解决车辆路径问题, 应用策略梯度算法对随机

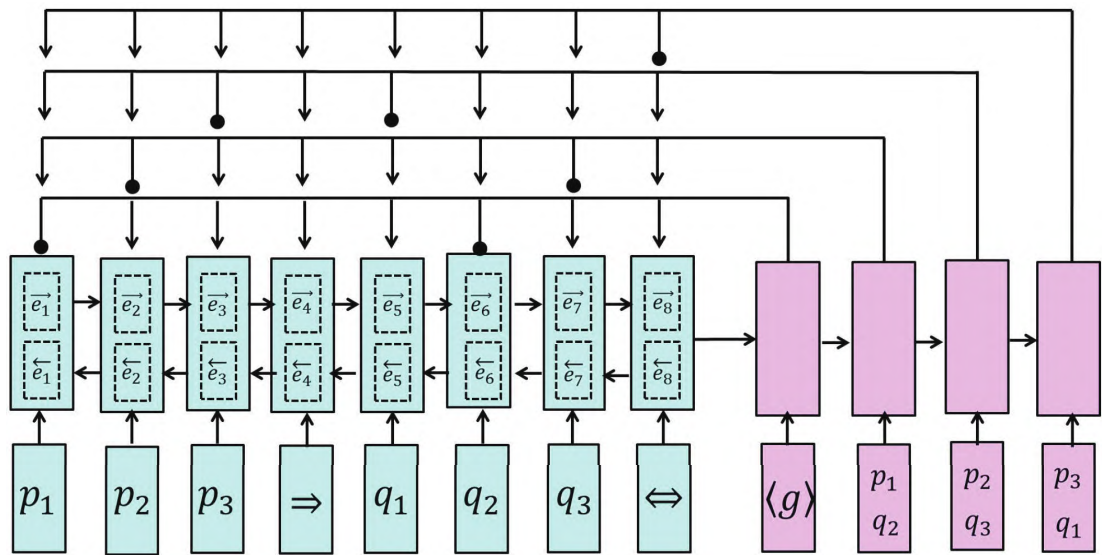


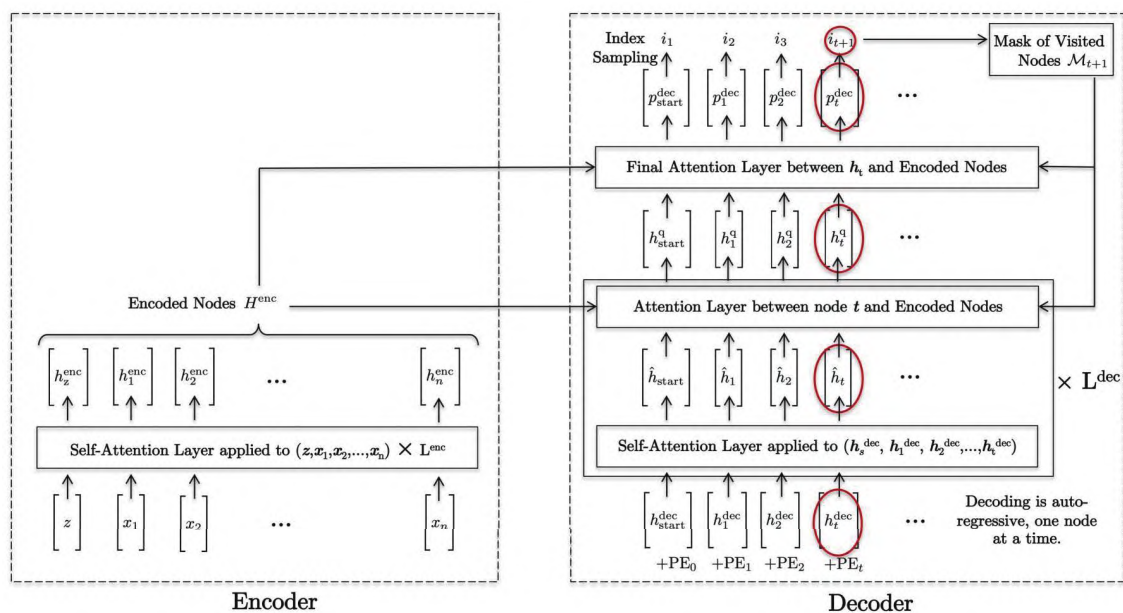
图 16 (网络版彩图) 多指向型网络框架 [80]

策略网络参数进行训练, 训练后的模型可以实时产生一系列连续动作的解, 而无需为每个新问题实例重新训练. 该模型仅通过观察奖励信号并遵循可行性规则, 就可以为大量类似规模的问题找到近似的最优解. 不同于构造式启发式算法, Wu 等 [87] 提出了一个深度强化学习框架, 来学习路径优化问题的改进启发式算法. Wu 等 [87] 设计了一个基于自注意力的策略参数化深度神经网络来指导下一个解的选择, 该架构可以结合大量常用的局部交换算子, 通过基于某个局部算子迭代执行邻域搜索来改进初始解, 朝着提高解质量的方向改进. 该方法应用于两个重要的路径优化问题: 旅行商问题和有容量限制的车辆路径问题 (capacitated vehicle routing problem, CVRP), 实验表明, 此方法优于最先进的基于深度学习的方法, 且比传统的启发式方法更有效, 还可以很好地推广到不同的问题规模和初始解, 甚至是真实世界的数据集. 0-1 背包 (0-1 Knapsack, 0-1 KP) 问题也是一个重要且众所周知的 NP- 困难问题. Gu 和 Hao [22] 提出了一种数据驱动的方法, 利用循环神经网络求解 0-1 背包问题. 目标函数各变量的系数和 0-1 背包问题的约束作为指针网络编码器的输入, 然后利用样本最优解给出的监督信号来训练网络模型. 实验结果表明, 用该算法训练的网络可以快速得到 0-1 背包问题的近似解.

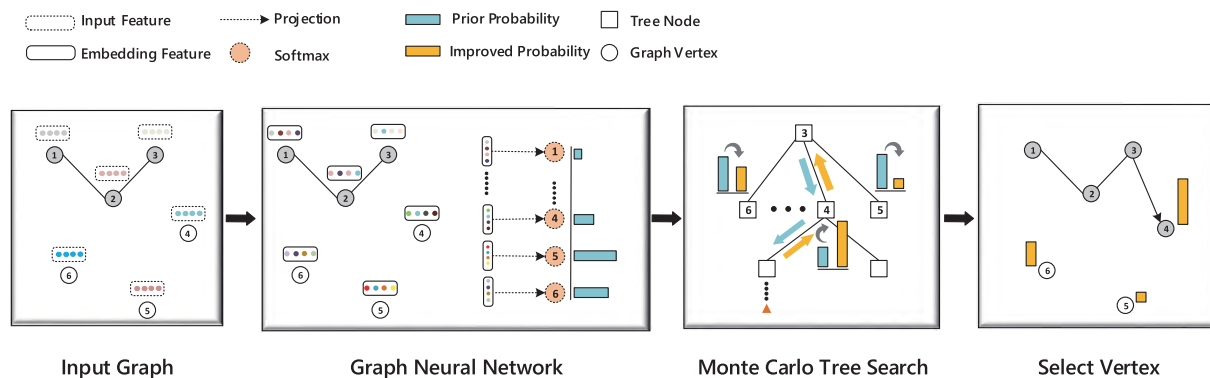
4.2 基于部分解迭代构造完整解

基于部分解迭代构造完整解的方法以监督学习和强化学习方法为主. 为了研究旅行商问题, Ma 等 [57] 引入了图指针网络 (graph pointer network, GPN), 基于强化学习训练分层图指针网络来求解旅行商问题. 图指针网络建立在指针网络的基础上, 通过在输入上引入图嵌入层来捕获节点之间的关系. 分层图指针网络的每一层都设计有单独的奖励函数, 从而实现稳定的训练, 学习到一个分层策略来找到约束下的最优城市排列. 在小规模 TSP50/100 问题上训练的图指针网络可以很好地推广到更大规模的 TSP500/1000 问题上, 并且具有更短的行程和更快的计算时间. Bresson 和 Laurent [6] 将最初为自然语言处理开发的 Transformer 架构纳入到旅行商问题的求解中 (图 17), 通过强化学习训练, 使用波束搜索进行解码. 实验表明, TSP50 和 TSP100 的最优间隙分别达到了 0.004% 和 0.39%.

基于部分解迭代构造完整解的研究方法中, 基于强化学习的求解方法也是一个非常具有代表性的研究思路. 对于基于强化学习的方法, 当迭代扩展部分解时, 每次迭代都会通过选择下一个节点来优

图 17 (网络版彩图) Transformer 架构^[6]

化最终解. 这种顺序决策过程可以建模为 Markov 决策过程, 并通过强化学习来求解. 由于迭代扩展组合优化问题的部分解本质上是一个顺序决策过程, 因此主要使用强化学习来扩展部分解, 由部分解和输入图一起确定强化学习的状态, 将可以添加到部分解中的节点或边作为动作. 强化学习可以学习最优策略来找到部分解的最优节点或边. Xing 和 Tu^[88] 采用将图神经网络与 Monte Carlo 树搜索相结合的思路, 基于贪婪算法框架通过连续添加顶点来得到一个有希望的环游, 训练图神经网络给出每步选择一个顶点的先验概率, 融合先验概率与探索的反馈信息提供更可靠的策略, 如图 18 所示. 在没有太多启发式设计的情形下, 该方法在旅行商问题的求解上, 优于最近提出的其他基于学习的方法. Kwon 等^[42] 旨在利用组合优化问题解表示形式中的对称性, 提出了多最优解的策略优化 (policy optimization with multiple optima, POMO) 方法. 经验表明, POMO 的低方差基线使得强化学习训练快速稳定, 并且比以往的方法更能抵抗局部极小值. 在旅行商问题、有容量限制的车辆路径问题和 0-1 背包问题上, 实验发现基于 POMO 的求解器比最近基于学习的启发式算法的性能都有了显著的提高. Khalil 等^[33] 结合图神经网络和深度 Q 学习提出了 S2V-DQN (structure2vec deep Q-network), 可用于求解

图 18 (网络版彩图) 图神经网络与 Monte Carlo 树搜索相结合的求解框架^[88]

MVC、TSP 和最大割问题 (maximum cut, MaxCut) 问题. 具体而言, 给定图 G , 设 U 表示当前的部分解, 并且 $\bar{U} = V \setminus U$, MVC 的强化学习任务可以表述如下:

(1) 图 G 和其部分解 U 定义的状态 $s = f_{\text{state}}(G, U)$, 如果 U 是一个顶点覆盖, 则状态为最终状态;

(2) 一个动作 a_v 就是增加一个点 $v \in \bar{U}$ 到 U 中;

(3) 状态转移函数 $T(f_{\text{state}}(G, U), a_v) = f_{\text{state}}(G, U \cup \{v\})$;

(4) 一个动作的奖励 $R(s, a_v) = -1$, 以便最小化顶点覆盖;

(5) 状态 s 的表示可以通过使用图神经网络嵌入 G 和 U 来计算, 具体地, 使用 v 的嵌入 h_v 来表示动作 a_v , 利用公式 (4.1)–(4.3) 计算 (s, a_v) 在多层感知机中的表示,

$$f_{\text{state}}(G, U) = \sum_v h_v^L, \quad (4.1)$$

$$h_u^L = \text{ReLU}\left(\theta_1 x_u + \theta_2 \sum_{v \in N_u} h_v^{L-1} + \theta_3 \sum_{v \in N_u} \text{ReLU}(\theta_4 w_{u,v})\right), \quad (4.2)$$

$$Q(s, a_v) = \theta_5 \text{ReLU}\left(\text{Concat}\left(\theta_6 \sum_{u \in V} h_u^L, \theta_7 h_v^L\right)\right). \quad (4.3)$$

深度 Q 学习用于优化参数, 在多层感知机和图神经网络被训练之后, 它们可以被泛化以计算其他图上的计算.

取送货问题 (pickup and delivery problem, PDP) 是车辆路径问题的一种变体, Zong 等^[97] 提出了一种新的基于多智能体强化学习的框架来解决协同取货问题, 如图 19 所示. 为了衡量不同节点的依赖关系, Zong 等^[97] 设计了一个配对上下文嵌入, 并利用协作式多智能体解码器, 基于新的协同 A2C 算法来训练集成模型. 在随机生成的数据集和真实世界的数据集上, 实验表明该方法比其他基准至少高出 1.64%, 并且在解推理过程中显示出了计算速度的明显提升.

除此之外, 针对数字背包问题, Ding 等^[17] 设计了高泛化性的结构化多头交流机制的注意力算法, 此模型大大提高了求解问题的泛化性能以注意力模型为基础. 具体而言, 该模型采用编码 - 解码的框架, 在编码阶段以物品的重量、价值序列为输入, 基于注意力模型得到输入序列的特征嵌入, 在解码时基于自注意力机制依次给出背包内物品的序号. 数值实验显示, 训练实例可以从 10^2 级别泛化到 10^3 , 而且解的精度几乎与 Gurobi 算法相同, 求解速度提高了接近 5 倍.

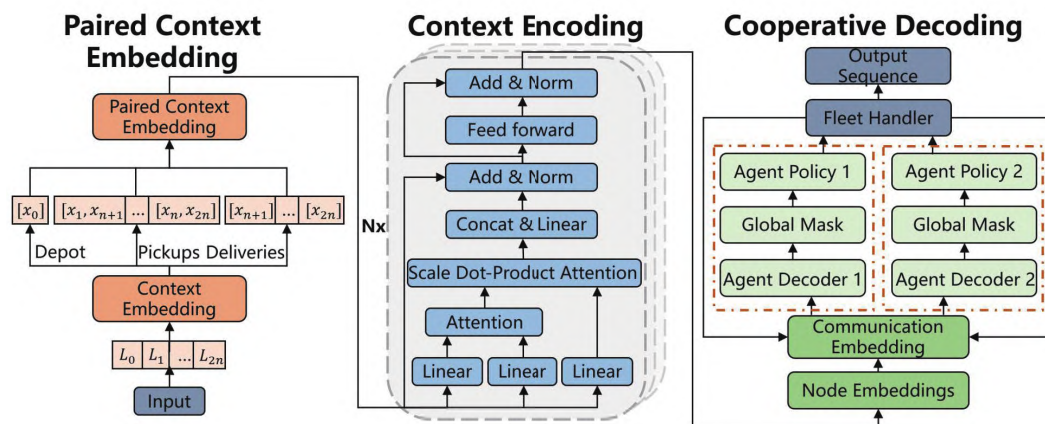


图 19 (网络版彩图) 协同取货问题的多智能体强化学习求解框架^[97]

5 总结和展望

机器学习的进一步发展为组合优化问题提供了崭新的求解思路, 表 1 对其中的主要方法进行了更直观的总结, 从中可以看出机器学习为组合优化问题的求解赋予了新的生命活力. 基于机器学习的求解方法虽然思想上源于传统算法, 却显著地提升了传统算法的性能. 两者相辅相成, 缺一不可. 我们未来应该将更多先进的学习方法纳入到组合优化问题的求解框架当中, 进一步实现求解的智能化. 对此, 首先分析组合优化问题目前研究的难点以及限制其发展的瓶颈问题, 然后就一些可能的发展趋势进行探讨.

5.1 组合优化问题的难点及人工智能方法的瓶颈

研究组合优化问题的人工智能求解方法, 需要更多地分析组合优化问题自身的特性, 针对这些特性定制与之契合的求解算法. 组合优化问题通常具有明确的目标函数和约束条件, 可以用约束条件判别当前解的可行性, 利用最优化目标函数的思想设计累计奖赏, 保证训练过程中解质量的稳步提升. 计算代价高昂也是组合优化问题的一个特性. 大规模组合优化问题求解往往需要很长的运行时间以及大量的存储空间, 在通常的计算机上实现困难, 但是这些问题在工程和科学领域应用又十分广泛, 实践的需要推动科研工作者们寻求更高效的解决方式. 离散求解空间庞大是组合优化问题另一显著的特性, 尤其是对于 NP- 困难问题, 求解空间往往呈指数级增长, 枚举遍历的时间代价现实中往往难以接受. 针对这一问题, AlphaGo 利用棋子位置和历史位置特征构建估值网络, 基于 Monte Carlo 树搜索算法大幅约减了搜索宽度和深度. 受此启发, 我们也可以构造恰当的数学模型, 基于问题全局上下界、求解状态等特性, 辅以适当的搜索策略, 进行搜索空间的约减, 在不失求解精度的情形下加速求解时间. 标注困难也是影响组合优化发展的一个难题. 以目前的计算水平, 我们很难为大规模组合优化问题提供大量精确解作为样本标签来学习网络参数. 但是 AlphaGo Zero 的成功也进一步表明了即便是在没有专家经验指导的情形下, 深度强化学习方法仍然可以帮助智能体完成复杂任务, 甚至跳出监督学习的局部最优解, 从而超越专家水平. 这也给了我们在少量标签或者无标签的情形下, 提升组合优化求解算法性能的新思路.

尽管人工智能方法在各个领域都取得了非凡的成就, 但是由于组合优化问题自身复杂的特性, 基于人工智能的组合优化求解方法还存在很多亟待解决的瓶颈问题. 首先, 最重要的是算法的执行时间, 当求解大规模问题时, 由于搜索空间巨大, 如果耗费的求解时间难以接受, 则这样的算法显然无意义. 求解时间长是限制算法实际应用性能的主要问题之一. 求解性能与网络的复杂度也是影响基于机器学习的组合优化方法改进的因素之一. 随着计算能力的提升和求解精度的精益求精, 为组合优化问题设计的深度神经网络结构也变得愈发复杂, 如何平衡求解性能和网络复杂性从而实现用较小的计算代价找到高质量的解成为一个难题. 此外, 机器学习泛化性的提升也是一个极其复杂的问题, 理想的泛化性不只局限于问题规模上的泛化, 不同实例分布乃至不同组合优化问题上的泛化性也是有待研究的问题之一. 当基于机器学习的方法可以广泛迁移到大规模问题场景中, 乃至作为一个通用框架迁移至不同类型组合优化问题中时, 通用意义上的人工智能时代可能会到来. 可解释性差是目前机器学习方法的一个共有问题, 尽管组合优化问题的理论研究不少, 但基于机器学习方法的理论背景仍有欠缺, 导致对现有的学习方法无法准确地描述找到的解与最优解之间的距离和算法的收敛性等. 因此有必要对组合优化问题的人工智能求解方法开展理论研究, 利用运筹学、统计学、算法复杂度理论等多种学科背景知识, 进一步使模型的可解释性变为可能.

表 1 文献整理框架图

方法	组合优化问题	前后端组合/融合	前端学习模型	基于任务的特征提取模型	后端学习模型
ExtraTrees [58]	BILP	组合	人工构造特征	-	ExtraTrees
GCN [21]	MILP	组合	人工 - 二部图嵌入	GCN	GCN
GCN + MLP [26]	MILP	组合	人工 - 二部图嵌入	GCN	GCN + MLP
GCN [62]	MILP	组合	人工 - 二部图嵌入	GCN	GCN
GAT [16]	MILP	组合	人工 - 三部图嵌入	GAT	GAT
GNN [66]	QAP	组合	人工 - 邻接矩阵表示	GNN	GNN
Actor-Critic [50]	ACQAP	组合	人工 - 关联图	GCN + Transformer	GCN + Transformer
GCN [36]	MILP	组合	人工构造特征	GCN	GCN
MLP [1]	LP	组合	人工构造特征 + SVD	MLP	MLP
GCN [13]	LP	组合	人工构造特征	GCN	GCN
MCTS [44]	LP	组合	人工 - 伪树结构	-	MCTS
GNN + Pointer network [49]	LP	组合	-	GNN	Pointer network
MCTS [32]	SAT	组合	人工 - 树搜索结构	-	MCTS
MCTS [45]	SAT	组合	人工 - BILP	-	MCTS
GNN [41]	SAT	组合	人工 - 二部图嵌入	GNN	GNN
GNN [73]	SAT	组合	人工 - 无向图嵌入	GNN	GNN
GNN [92]	SAT	组合	人工 - 因子图表示	GNN	GNN
HASSLE [40]	SAT	组合	人工 - 表示为优化问题	-	HASSLE
GNN + DDQN [18]	STP	组合	-	GNN	GNN + DDQN
元博弈 + PSRO [85]	TSP	组合	-	元博弈 + PSRO	元博弈 + PSRO
GCN + Autoregressive [30]	TSP	组合	-	GCN	GCN + Autoregressive
MCTS + AttGCRN [20]	TSP	组合	-	AttGCRN	MCTS + AttGCRN
Q-learning [4]	TSP	组合	-	Q-learning	Q-learning
GCN + PPO [11]	VRP	组合	-	GCN + PPO	GCN + PPO

(续表)

方法	组合优化问题	前后端组合/融合	前端学习模型	基于任务的特征提取模型	后端学习模型
MAAM [93]	VRP	组合	-	MAAM	MAAM
LCP [38]	TSP, PCTSP, CVRP	组合	-	LCP	LCP
DPDP + GNN [39]	TSP, VRP, TSPTW	组合	-	GNN	DPDP + GNN
GNN [51]	MIS, MVC, MC, SAT	组合	-	GNN	GNN
GAT + Reinforce [84]	P-median	组合	-	GAT	GAT + Reinforce
Transformer + Reinforce [75]	DCMST, MRCST, STP	组合	-	Transformer	Transformer + Reinforce
FCNN + Q-learning [79]	LP	融合	-	FCNN	FCNN + Q-learning
GNN [68]	TSP	融合	-	GNN	GNN
Encoder-Decoder + Reinforce [15]	TSP	融合	PCA	Encoder-Decoder	Encoder-Decoder + Reinforce
GPN [57]	TSP	融合	-	Pointer network	Pointer network
Transformer [6]	TSP	融合	-	Transformer	Transformer
GNN + MCTS [88]	TSP	融合	-	GNN	GNN + MCTS
Actor-Critic [63]	TSP, CVRP	融合	-	Actor-Critic	Actor-Critic
Attention model [42]	TSP, CVRP, 0-1 KP	融合	-	Attention model	Attention model
GNN [43]	GCP	融合	-	GNN	GNN
Encoder-Decoder + Reinforce [48]	CSP	融合	-	Encoder-Decoder	Encoder-Decoder + Reinforce
Multi-pointer network [80]	Point matching	融合	-	Multi-pointer network	Multi-pointer network
Encoder-Decoder [17]	KP	融合	-	Encoder-Decoder	Encoder-Decoder
Pointer network [22]	0-1 KP	融合	-	Pointer network	Pointer network
Encoder-Decoder + A2C [97]	PDP	融合	-	Encoder-Decoder	Encoder-Decoder + A2C

5.2 未来有价值的研究方向

5.2.1 基于多智能体合作机制的组合优化问题求解方法研究

目前多数应用于组合优化的强化学习方法通常仅采用单智能体强化学习, 通过设定智能体奖励使其对应于优化目标, 可以逐渐训练策略直到收敛于优化问题的解. 然而, 采用单智能体强化学习方法具有较多缺点, 其训练时间长、收敛效果差且难以发现和利用问题的结构特征. 另外, 由于优化问题会涉及维数、约束等的变化, 如果直接采用多智能体系统求解, 则可能会由于智能体个数的变化从而使模型难以收敛. 因此可以尝试通过多智能体合作机制设计模型用于求解组合优化问题.

5.2.2 基于博弈策略的组合优化问题求解方法研究

泛化性是目前求解组合优化问题的人工智能算法面临的瓶颈. 从一个特定问题数据集上训练得到的求解模型很难推广到一般问题. 这是由于基于样本的训练模型通常是过拟合的, 在应用于其他问题时很难处理训练样本所不具有的特征. 而如果采用较为一般的训练集, 受限于神经网络的表达能力, 模型的收敛性难以保证, 可以通过研究构建博弈模型, 将求解组合优化的人工智能算法在其上运行, 从而提高求解模型的泛化能力.

5.2.3 基于生成方法的组合优化问题求解方法研究

现有的求解组合优化问题的人工智能算法不管是基于部分解迭代的构造方式还是基于完整解的迭代的构造方式都需要昂贵的算力, 而且还限制了获取问题多模态的能力. 借鉴视觉领域风格迁移、零样本学习等图像生成方法的思想, 结合近年来的扩散概率模型与生成对抗模型构造求解组合优化问题的生成式求解方法是一个新的研究方向.

5.2.4 基于大语言模型的组合优化问题求解方法研究

大语言模型 (large language model, LLM) 的兴起对各个行业产生了重大影响, 同时也为组合优化问题求解提供了崭新的思路 (参见文献 [91]). 鉴于大语言模型利用大规模语料数据进行预训练的特点, 可将其自然地用于机器学习求解方法的预训练, 从而加速整个训练过程. 除此之外, 大语言模型还可以用来为现有的求解方法寻找初始解, 从而有效地集成到基于完整解迭代优化的框架中. 将大语言模型与最前沿的方法深度融合将会为组合优化问题的人工智能算法提供新的研究点.

总之, 最优问题智能求解算法设计的路还很长, 需要更多的数学、统计、计算机等领域工作者们的共同努力, 才有可能最终实现优化问题“求解智能化”的目标.

致谢 作者由衷地感谢审稿专家的支持, 使得本文能够进一步完善并增色.

参考文献

- 1 Adham I, De Loera J, Zhang Z. (Machine) Learning to improve the empirical performance of discrete algorithms. arXiv:2109.14271, 2021
- 2 Afshar R R, Zhang Y, Firat M, et al. A state aggregation approach for solving knapsack problem with deep reinforcement learning. In: Proceedings of the Asian Conference on Machine Learning. Bangkok: PMLR, 2020, 81–96
- 3 Ahn S, Seo Y, Shin J. Learning what to defer for maximum independent sets. In: Proceedings of the International Conference on Machine Learning. Vienna: PMLR, 2020, 134–144
- 4 Alipour M M, Razavi S N. A new multiagent reinforcement learning algorithm to solve the symmetric traveling salesman problem. Multiagent Grid Systems, 2015, 11: 107–119

- 5 Amizadeh S, Matushevych S, Weimer M. Learning to solve circuit-sat: An unsupervised differentiable approach. In: Proceedings of the International Conference on Learning Representations. New Orleans: ICLR, 2018, <https://openreview.net/forum?id=BJxgz2R9t7>
- 6 Bresson X, Laurent T. The transformer network for the traveling salesman problem. arXiv:2103.03012, 2021
- 7 Cameron C, Chen R, Hartford J, et al. Predicting propositional satisfiability via end-to-end learning. AAAI, 2020, 34: 3324–3331
- 8 Cappart Q, Goutier E, Bergman D, et al. Improving optimization bounds using machine learning: Decision diagrams meet deep reinforcement learning. AAAI, 2019, 33: 1443–1451
- 9 Cappart Q, Moisan T, Rousseau L M, et al. Combining reinforcement learning and constraint programming for combinatorial optimization. AAAI, 2021, 35: 3677–3687
- 10 Chalumeau F, Coulon I, Cappart Q, et al. SeaPearl: A constraint programming solver guided by reinforcement learning. In: Proceedings of the 18th Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Vienna: Springer, 2021, 392–409
- 11 Chen M, Gao L, Chen Q, et al. Dynamic partial removal: A neural network heuristic for large neighborhood search. arXiv:2005.09330, 2020
- 12 Chen Y Q, Chen Y, Lee C K, et al. Optimizing quantum annealing schedules with Monte Carlo tree search enhanced with neural networks. Nat Mach Intell, 2022, 4: 269–278
- 13 Chen Z, Liu J, Wang X, et al. On representing linear programs by graph neural networks. arXiv:2209.12288, 2022
- 14 Chen Z, Liu J, Wang X, et al. On representing mixed-integer linear programs by graph neural networks. arXiv:2210.10759, 2022
- 15 Deudon M, Cournut P, Lacoste A, et al. Learning heuristics for the tsp by policy gradient. In: Proceedings of the 15th Integration Conference on Constraint Programming, Artificial Intelligence, and Operations Research. Delft: Springer, 2018, 170–181
- 16 Ding J Y, Zhang C, Shen L, et al. Accelerating primal solution findings for mixed integer programs based on solution prediction. AAAI, 2020, 34: 1452–1459
- 17 Ding M, Han C, Guo T. High generalization performance structured self-attention model for knapsack problem. Discrete Math Algorithms Appl, 2021, 13: 2150076
- 18 Du H, Yan Z, Xiang Q, et al. Vulcan: Solving the Steiner tree problem with graph neural networks and deep reinforcement learning. arXiv:2111.10810, 2021
- 19 Duan H, Vaezipoor P, Paulus M B, et al. Augment with care: Contrastive learning for combinatorial problems. In: Proceedings of the International Conference on Machine Learning. Baltimore: PMLR, 2022, 5627–5642
- 20 Fu Z H, Qiu K B, Zha H. Generalize a small pre-trained model to arbitrarily large TSP instances. AAAI, 2021, 35: 7474–7482
- 21 Gasse M, Chételat D, Ferroni N, et al. Exact combinatorial optimization with graph convolutional neural networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Red Hook: Curran Associates, 2019, 15580–15592
- 22 Gu S, Hao T. A pointer network based deep learning algorithm for 0-1 knapsack problem. In: Proceedings of the 2018 10th International Conference on Advanced Computational Intelligence (ICACI). New York: IEEE, 2018, 473–477
- 23 Guo T D, Han C Y. From numerical optimization methods to learning optimization methods (in Chinese). J Oper Res, 2019, 23: 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, 23: 1–12]
- 24 Guo T D, Han C Y, Li M Q. The theory, model and algorithm of front and back end fusion learning for layer by layer data rerepresentation (in Chinese). Sci China Inf Sci, 2019, 49: 739–759 [郭田德, 韩丛英, 李明强. 逐层数据再表达的前后端融合学习的理论及其模型和算法. 中国科学: 信息科学, 2019, 49: 739–759]
- 25 Guo T D, Han C Y, Tang S Q. Machine Learning Methods for Combinatorial Optimization (in Chinese). Beijing: Science Press, 2019 [郭田德, 韩丛英, 唐思琦. 组合优化机器学习方法. 北京: 科学出版社, 2019]
- 26 Gupta P, Gasse M, Khalil E, et al. Hybrid models for learning to branch. In: Advances in Neural Information Processing Systems, vol. 33. La Jolla: NIPS, 2020, 18087–18097
- 27 He H, Daume H III, Eisner J. Learning to search in branch-and-bound algorithms. In: Advances in Neural Information Processing Systems. La Jolla: NIPS, 2014, 3293–3301
- 28 Hu J, Guo T, Zhao T. A faster stochastic alternating direction method for large scale convex composite problems. Appl Intell, 2022, 52: 14233–14245
- 29 Hu J, Han C, Guo T, et al. On inexact stochastic splitting methods for a class of nonconvex composite optimization problems with relative error. Optim Methods Softw, 2023, 38: 1–33
- 30 Joshi C K, Laurent T, Bresson X. An efficient graph convolutional network technique for the travelling salesman problem. arXiv:1906.01227, 2019

- 31 Kabbaj M M, El Afia A. Towards learning integral strategy of branch and bound. In: Proceedings of the 2016 5th International Conference on Multimedia Computing and Systems (ICMCS). New York: IEEE, 2016, 621–626
- 32 Keszocze O, Schmitz K, Schloeter J, et al. Improving SAT solving using Monte Carlo tree search-based clause learning. In: Advanced Boolean Techniques. Cham: Springer, 2020, 107–133
- 33 Khalil E, Dai H, Zhang Y, et al. Learning combinatorial optimization algorithms over graphs. In: Advances in Neural Information Processing Systems. La Jolla: NIPS, 2017, 6348–6358
- 34 Khalil E, Dilkina B, Nemhauser G, et al. Learning to run heuristics in tree search. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence. Melbourne: Morgan Kaufmann, 2017, 659–666
- 35 Khalil E, Le Bodic P, Song L, et al. Learning to branch in mixed integer programming. AAAI, 2016, 30: 724–731
- 36 Khalil E, Morris C, Lodi A. MIP-GNN: A data-driven framework for guiding combinatorial solvers. AAAI, 2022, 36: 10219–10227
- 37 Kiarostami M S, Daneshvaramoli M, Khalaj Monfared S, et al. On using Monte-Carlo tree search to solve puzzles. In: Proceedings of the 7th International Conference on Computer Technology Applications. Vienna: ACM, 2021, 18–26
- 38 Kim M, Park J, Kim J. Learning collaborative policies to solve NP-hard routing problems. In: Advances in Neural Information Processing Systems, vol. 34. La Jolla: NIPS, 2021, 10418–10430
- 39 Kool W, van Hoof H, Gromicho J, et al. Deep policy dynamic programming for vehicle routing problems. In: Proceedings of the International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research. Cham: Springer, 2022, 190–213
- 40 Kumar M, Kolb S, Teso S, et al. Learning MAX-SAT from contextual examples for combinatorial optimisation. Artif Intell, 2023, 314: 103794
- 41 Kurin V, Godil S, Whiteson S, et al. Can Q-learning with graph networks learn a generalizable branching heuristic for a SAT solver? In: Advances in Neural Information Processing Systems, vol. 33. La Jolla: NIPS, 2020, 9608–9621
- 42 Kwon Y D, Choo J, Kim B, et al. Pomo: Policy optimization with multiple optima for reinforcement learning. In: Advances in Neural Information Processing Systems, vol. 33. La Jolla: NIPS, 2020, 21188–21198
- 43 Lemos H, Prates M, Avelar P, et al. Graph colouring meets deep learning: Effective graph neural network models for combinatorial problems. In: Proceedings of the 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI). Portland: IEEE, 2019, 879–885
- 44 Li A, Guo T, Han C, et al. Optimal pivot path of the simplex method for linear programming based on reinforcement learning. Sci China Math, 2022, 67: 1263–1286
- 45 Li A, Han C, Guo T, et al. General method for solving four types of SAT problems. arXiv:2312.16423, 2023
- 46 Li A, Han C, Guo T, et al. Generating linear programming instances with controllable rank and condition number. Comput Oper Res, 2024, 162: 106471
- 47 Li H R, Zhang Z C, Luo W, et al. Towards optimal adversarial robust Q-learning with Bellman infinity-error. In: Proceedings of the 41st International Conference on Machine Learning (ICML'24). Vienna: JMLR, 2025, 29324–29372
- 48 Li K, Zhang T, Wang R, et al. Deep reinforcement learning for combinatorial optimization: Covering salesman problems. IEEE Trans Cybern, 2021, 52: 13142–13155
- 49 Li X, Qu Q, Zhu F, et al. Learning to reformulate for linear programming. arXiv:2201.06216, 2022
- 50 Li X Y, Han C Y. Solving quadratic assignment problem based on actor-critic framework (in Chinese). J Univ Chin Acad Sci, 2019, 41: 275–284 [李雪源, 韩丛英. Actor-critic 框架下的二次指派问题求解方法. 中国科学院大学学报(中英文), 2024, 41: 275–284]
- 51 Li Z, Chen Q, Koltun V. Combinatorial optimization with graph convolutional networks and guided tree search. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. La Jolla: NIPS, 2018, 537–546
- 52 Liao S, Han C, Guo T, et al. Subspace Newton method for sparse group ℓ_0 optimization problem. J Global Optim, 2024, 90: 93–125
- 53 Liu T, Pu S, Ge D, et al. Learning to pivot as a smart expert. AAAI, 2024, 38: 8073–8081
- 54 Liu Z, Li M, Han C, et al. STDNet: Rethinking disentanglement learning with information theory. IEEE Trans Neural Networks Learn Systems, 2023, 35: 10407–10421
- 55 Lodi A, Mossina L, Rachelson E. Learning to handle parameter perturbations in combinatorial optimization: An application to facility location. EURO J Transp Logistics, 2020, 9: 100023
- 56 Luo W, Li H, Zhang Z, et al. SAMBO-RL: Shifts-aware model-based offline reinforcement learning. arXiv:2408.12830, 2024
- 57 Ma Q, Ge S, He D, et al. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. arXiv:1911.04936, 2019
- 58 Marcos Alvarez A, Louveaux Q, Wehenkel L. A supervised machine learning approach to variable branching in branch-

- and-bound. <https://api.semanticscholar.org/CorpusID:55071488>, 2014
- 59 Marcos Alvarez A, Wehenkel L, Louveaux Q. Online learning for strong branching approximation in branch-and-bound. <https://optimization-online.org/wp-content/uploads/2016/01/5304.pdf>, 2016
 - 60 McCormick C. Word2vec tutorial-the skip-gram model. <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model>, 2016
 - 61 Milan A, Rezaatofghi S, Garg R, et al. Data-driven approximations to NP-hard problems. AAAI, 2017, 31: 1453–1459
 - 62 Nair V, Bartunov S, Gimeno F, et al. Solving mixed integer programs using neural networks. arXiv:2012.13349, 2020
 - 63 Nazari M, Oroojlooy A, Snyder L, et al. Reinforcement learning for solving the vehicle routing problem. ACM Trans Intell Syst Tech, 2018, 15: 1–19
 - 64 Nazi A, Hang W, Goldie A, et al. Gap: Generalizable approximate graph partitioning framework. arXiv:1903.00614, 2019
 - 65 Nguyen M A, Sano K, Tran V T. A Monte Carlo tree search for traveling salesman problem with drone. Asian Transp Studies, 2020, 6: 100028
 - 66 Nowak A, Villar S, Bandeira A S, et al. Revised note on learning quadratic assignment with graph neural networks. In: 2018 IEEE Data Science Workshop (DSW). Lausanne: IEEE, 2018, 1–5
 - 67 Paulus M B, Zarpellon G, Krause A, et al. Learning to cut by looking ahead: Cutting plane selection via imitation learning. In: International Conference on Machine Learning. Baltimore: PMLR, 2022, 17584–17600
 - 68 Prates M, Avelar P H C, Lemos H, et al. Learning to solve NP-complete problems: A graph neural network for decision TSP. AAAI, 2019, 33: 4731–4738
 - 69 Qin J, Tang S, Han C, et al. Partial fingerprint matching via phase-only correlation and deep convolutional neural network. In: Processing of the 24th International Conference on Neural Information. Berlin: Springer, 2017, 602–611
 - 70 Qiu X, Han C, Zhang Z, et al. DiffBFR: Bootstrapping diffusion model for blind face restoration. In: Proceedings of the 31st ACM International Conference on Multimedia. New York: ACM, 2023, 7785–7795
 - 71 Qiu X, Han C, Zhang Z, et al. BlazeBVD: Make scale-time equalization great again for blind video deflickering. In: Computer Vision—ECCV 2024. Lecture Notes in Computer Science, vol. 15075. Cham: Springer, 2025, 37–53
 - 72 Selsam D, Bjørner N. Guiding high-performance SAT solvers with unsat-core predictions. In: Proceedings of the Theory and Applications of Satisfiability Testing-SAT 2019. Lisbon: Springer, 2019, 336–353
 - 73 Selsam D, Lamm M, Bünz B, et al. Learning a SAT solver from single-bit supervision. arXiv:1802.03685, 2018
 - 74 Shi Y, Han C, Guo T. CARSS: Cooperative attention-guided reinforcement subpath synthesis for solving traveling salesman problem. arXiv:2312.15412, 2023
 - 75 Shi Y, Han C, Guo T. NeuroPrim: An attention-based model for solving NP-hard spanning tree problems. Sci China Math, 2024, 67: 1359–1376
 - 76 Song J, Lanka R, Zhao A, et al. Learning to search via self-imitation with application to risk-aware planning. https://lld-workshop.github.io/2017/papers/LLD_2017-paper-18.pdf, 2017
 - 77 Song W, Cao Z, Zhang J, et al. Learning variable ordering heuristics for solving constraint satisfaction problems. Eng Appl Artif Intelligence, 2022, 109: 104603
 - 78 Sun L, Gerault D, Benamira A, et al. NeuroGIFT: Using a machine learning based sat solver for cryptanalysis. In: Proceedings of the Cyber Security Cryptography and Machine Learning. Berlin: Springer, 2020, 62–84
 - 79 Suriyanarayana V, Tavaslioglu O, Patel A B, et al. DeepSimplex: Reinforcement learning of pivot rules improves the efficiency of simplex algorithm in solving linear programming problems. <https://openreview.net/forum?id=SkgvvCVtDS>, 2019
 - 80 Tang S Q, Han C Y, Guo T D. Point matching algorithm based on machine learning method (in Chinese). J Univ Chine Acad Sci, 2020, 37: 450–457 [唐思琦, 韩丛英, 郭田德. 基于机器学习的点集匹配算法. 中国科学院大学学报, 2020, 37: 450–457]
 - 81 Tang S Q, Han C Y, Guo T D, et al. Reinforcement learning of graph matching. In: Proceedings of the 8th International Conference on Computing and Artificial Intelligence. New York: ACM, 2022, 689–697
 - 82 Tang S Q, Qin J, Liu Y, et al. An efficient slap fingerprint segmentation algorithm based on convnets and knuckle line. In: Proceedings of the 12th Chinese Conference on Biometric Recognition. Cham: Springer, 2017, 231–239
 - 83 Vinyals O, Fortunato M, Jaitly N. Pointer networks. In: Advances in Neural Information Processing Systems. La Jolla: NIPS, 2015, 2692–2700
 - 84 Wang C, Han C, Guo T, et al. Solving uncapacitated P-median problem with reinforcement learning assisted by graph attention networks. Appl Intell, 2023, 53: 2010–2025
 - 85 Wang C, Yang Y, Slumbers O, et al. A game-theoretic approach for improving generalization ability of TSP solvers. In: ICLR 2022 Workshop on Gamification and Multiagent Solutions. <https://openreview.net/forum?id=rfeLi91pec>, 2023

- 86 Wang Q, Hao Y, Cao J. Learning to traverse over graphs with a Monte Carlo tree search-based self-play framework. *Eng Appl Artif Intelligence*, 2021, 105: 104422
- 87 Wu Y, Song W, Cao Z, et al. Learning improvement heuristics for solving routing problems. *IEEE Trans Neural Netw Learn Syst*, 2021, 33: 5057–5069
- 88 Xing Z, Tu S. A graph neural network assisted Monte Carlo tree search approach to traveling Salesman problem. *IEEE Access*, 2020, 8: 108418
- 89 Xu H, Hui K H, Fu C W, et al. TilingGNN: Learning to tile with self-supervised graph neural network. *ACM Trans Graph*, 2020, 39: 129
- 90 Yang C, Szeto K Y. Solving the traveling salesman problem with a multi-agent system. In: *Proceedings of the 2019 IEEE Congress on Evolutionary Computation (CEC)*. New York: IEEE, 2019, 158–165
- 91 Yang C, Wang X, Lu Y, et al. Large language models as optimizers. *arXiv:2309.03409*, 2023
- 92 Yolcu E, Póczos B. Learning local search heuristics for boolean satisfiability. In: *Advances in Neural Information Processing Systems*. La Jolla: NIPS, 2019, https://proceedings.neurips.cc/paper_files/paper/2019/hash/12e59a33dea1bf0630f46edfe13d6ea2-Abstract.html
- 93 Zhang K, He F, Zhang Z, et al. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transp Res Part C-Emerging Technologies*, 2020, 121: 102861
- 94 Zheng J, He K, Zhou J, et al. Combining reinforcement learning with Lin-Kernighan-Helsgaun algorithm for the traveling Salesman problem. *AAAI*, 2021, 35: 12445–12452
- 95 Zheng J, Zhong J, Chen M, et al. A reinforced hybrid genetic algorithm for the traveling salesman problem. *Comput Oper Res*, 2023, 157: 106249
- 96 Zhu Y, Swami A, Segarra S. Free energy node embedding via generalized skip-gram with negative sampling. *IEEE Trans Knowl Data Eng*, 2023, 35: 8024–8036
- 97 Zong Z, Zheng M, Li Y, et al. MAPDP: Cooperative multi-agent reinforcement learning to solve pickup and delivery problems. *AAAI*, 2022, 36: 9980–9988

Machine learning method for combinatorial optimization problems

Tiande Guo, Anqi Li & Congying Han

Abstract Optimization is a very classical research field, and many problems in engineering, management, and other fields are essentially optimization problems. At present, the rapid development of machine learning methods represented by deep learning and reinforcement learning has brought key promotion and development in many subject areas. Optimization is one of the most important supporting disciplines of machine learning, and machine learning is ultimately transformed into solving an optimization problem, which belongs to the research content of science for AI (artificial intelligence). In turn, AI technologies, including machine learning, are already having a significant impact on the development of science and will form a new paradigm of scientific research—AI for science. The artificial intelligence method of optimization problems is to design a fast and efficient algorithm of optimization problems using the theory of artificial intelligence. In this paper, we summarize the research progress of machine learning methods for combinatorial optimization in recent years, and deeply analyze the difficulties of solving combinatorial optimization problems and the bottlenecks of artificial intelligence solving methods. Some important research directions are given based on the current research status of image processing, game theory, reinforcement learning, and other popular fields.

Keywords combinatorial optimization, machine learning, optimization

MSC(2020) 90C27

doi: 10.1360/SSM-2024-0180