****

**研究生（ 文献综述与选题 ）报告**

**题 目：基于学习机制求解组合优化问题研究**

**学 号 D201780818**

**姓 名 周建荣**

**专 业 计算机软件与理论**

**指 导 教 师 何琨**

**院（系、所） 计算机科学与技术**

**华中科技大学研究生院制**

**填表注意事项**

一、本表适用于攻读硕士学位研究生选题报告、学术报告，攻读专业硕士学位研究生实践环节报告，攻读博士学位研究生文献综述、选题报告、论文中期进展报告、学术报告等。

二、以上各报告内容及要求由相关院（系、所）做具体要求。

三、以上各报告均须存入研究生个人学籍档案。

四、本表填写要求文句通顺、内容明确、字迹工整。

一、 课题的来源、目的、意义、国内外研究现状

二、 预计需达到的要求、技术指标，预计的技术关键、技术方案和主要实验研究情况

三、 课题研究进展计划

**一、 课题的来源、目的、意义**

在管理科学、计算机科学、分子物理学和生物学以及超大规模集成电路(VLSI)设计、代码设计、图象处理和电子工程等科技领域中，存在着大量组合优化问题。其中许多问题如货郎担问题、图着色问题、设备布局问题以及布线问题等，至今没有找到有效的多项式时间算法。这些问题已被证明是NP完全问题。用最优算法如线性规划求NP完全问题的最优解，需要问题规模的指数阶时间，在问题规模增大时，往往由于计算时间的限制而丧失可行性。用近似算法如贪心法求解NP完全问题，在多项式界的时间里，只能给出近似最优解。

随着机器学习和深度学习的发展，在众多人工智能领域应用成果取得了重大突破，例如无人驾驶、Alpha Go人工智能下围棋、语音助手、人脸识别、医学图像检测等。这标志着人工智能在一些关键技术上取得重大突破，并且人工智能发展在进一步走向成熟，机器学习和深度学习应用领域越来越广泛，促进了更多专业和学科对人工智能的交叉发展，越来越多的学者进入人工智能领域。

常见的组合优化问题有可满足性问题、旅行商问题、0-1背包问题、最小顶点覆盖问题、最大独立集问题、集合覆盖问题等，这些问题广泛存在于控制领域、工程领域等，存在大量的应用场景。又因为这些组合优化问题被证明为NP完全问题，更多是属于NP难问题，求解这些问题的时间会随着输入数据规模成指数增长，在一定规模之上就难以求解。随着科技水平的发展，信息和数据规模爆炸式增长，能在合理时间内求解得令人满意的结果便是现在组合优化问题一个非常重要的目标，而利用传统的启发式方法求解得到结果差强人意，而且设计启发式算法需要大量的专业领域知识储备，因此越来越多研究人员考虑使用机器学习和深度学习方法优化和求解组合优化问题。

组合优化与强化学习的交叉融合。经典的组合优化问题在求解过程中策略是固定不变、不具备“智能”的，搜索效率相对较低。启发式算法虽然具有一定的灵活性，但是其策略依赖于算法设计者对问题的理解和经验，需要多次进行重复的实验来获得一个较为理想的估价函数和搜索策略，对算法设计者的要求非常高。拟将强化学习融入组合优化，让智能体在搜索过程中自己学会一种高效、合理的搜索策略，通过智能体的自学习的搜索策略提高算法本身的运行效率和求得解的质量。拟围绕高维数据计算、智能调度等方向展开研究，提供更高效的计算模型和智能算法去解决大规模系统调度、任务分配、约束可满足性优化的一系列组合优化问题，产生原创性学术成果，并在现代智能打车、轮船飞机调度等行业产生应用，推动自动化物流、智慧城市的发展。

**二、国内外研究现状**

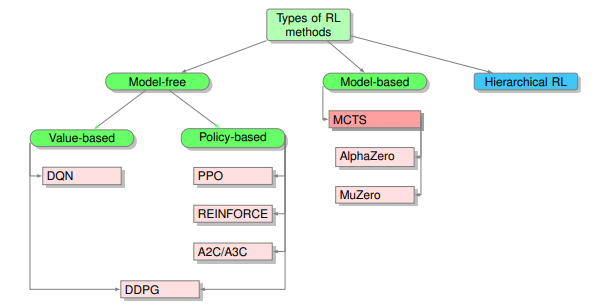
对于传统组合优化问题有许多传统经典的求解算法框架和启发式算法框架，例如禁忌搜索[1]（Tabu Search）、遗传算法[2]（Genetic Algorithm，GA）、蚁群算法[3]（Ant Colony Algorithm，AC）、分支限界算法[4]（Branch and Bound，BnB，B&B）等。

对于传统组合优化问题有经典的求解算法。例如旅行商问题有LKH算法[44]通过一组初始解进行不断搜索增广路改进解质量并且使用1-tree得到一个搜索下界对搜索过程进行优化剪枝，在[45]中0-1背包问题转化为Linear Programming（LP）问题使用拉格朗日松弛和对偶得到初始解并且通过3-flip的local search操作优化解质量，可满足性问题使用单子句推理[46,47]方法加速搜索并使用冲突子句学习[48-50]提高搜索效率等。

一部分学者使用学习机制和框架优化求解器和传统算法或者直接求解组合优化问题。Zhou等人[5]利用RL的policy-base策略，通过顶点-染色的一种选择概率策略构造一个近似state-action的概率矩阵，利用这样的一个RL启发式方法去求解图染色问题。Zhou等人[6]利用概率学习模型作为一种奖惩机制去引导禁忌搜索求解图染色问题。Shimomura等人[7]使用蒙特卡洛树（Monte-Carlo Tree Search, MCTS）和随机采样的方式，以此通过不断采样提升MCTS搜索树的采样结果来求解旅行商问题。Liu等人[8]利用学习基础的贝尔曼等式（Bellman Equation）作为一个启发式函数优化最大相同子图的求解算法。

近年来越来越多学者使用机器学习（Machine Learning，ML）、深度学习（Deep Learning，DL）和强化学习（Reinforcement Learning，RL）等理论去求解组合优化问题，尤其是通过深度强学习（Deep Reinforcement Learning，DRL）。

对于组合优化和学习机制结合的方法已经有一些学者[16-20]提出整体的算法框架和方法论，Y. Bengio等人[16]提出一种方法近似组合优化问题并使用机器学习方法求解，J. Zhou等人[19]提出了图神经网络模型（Graph Neural Network，GNN）对组合优化的一些应用的框架，N. Vesselinova等人[18]和T. Guo等人[20]直接使用机器学习方法对组合优化问题进行求解。

其中一个比较经典深度学习求解组合优化问题的模型是Sutskever等人[21]提出的指针网络（Pointer Network），指针网络起初是为了解决自然语言处理（Nature Language Process，NLP）的序列到序列（Sequence to Sequence，Seq2Seq）的输出模型，因为部分组合优化问题模型属于Seq2Seq，所以指针网络也常被用于求解组合优化问题。

另一个方向就是使用强化学习和深度学习求解组合优化问题，强化学习作为一个研究方向已经有超过五十年的时间，其研究基础——马尔科夫决策过程（Markov Decision Process，MDP）在上世纪八十年代被提出。作为机器学习领域另一个研究热点，强化学习已经被广泛应用于工业制造[9]、仿真模拟[10]、机器人控制[11]、优化与调度[12-13]、游戏博弈[14-15]等领域，RL的基本思想是通过最大化智能体（智能体）从环境中获得的累计奖赏值，以学习到完成目标的最优策略。传统的强化学习被分为基于模型的和非基于模型的两大类。在基于模型的强化学习中，智能体使用收集到的数据训练模型，将模型用于策略优化。非基于模型的强化学习中，智能体直接从搜集到的数据中学习最优的状态-动作序列。近些年，强化学习的研究都取得了丰硕的成果，对强化学习的研究主要是对强化学习的理论、强化学习的算法、以及强化学习的应用三方面。在理论研究方面，主要研究强化学习算法的收敛性和强化学习泛化方法。Sutton等人[22]首次给出了时序差分(Temporal Difference，TD)学习的形式化描述和TD(λ)学习算法，并且已经成为其他强化学习算法如Q值学习算法的基础。后有文献从各方面证明了其收敛性。目前提出的强化学习泛化方法主要有：值函数逼近方法、策略空间逼近方法、同时进行值函数和策略空间逼近的泛化方法[23]。强化学习算法可以分为折扣型回报指标强化学习和平均回报强化学习。折扣型回报指标强化学习算法主要有：TD(λ)学习算法、Q-学习（Q-learning）算法[24]、Sarsa学习算法[25]和Actor-Critic学习算法[26]。平均型回报指标强化学习算法主要包括：基于平均回报指标的时域差值学习算法[27]、R-学习算法[28]。目前，大量优化的强化学习算法被提出，例如：贝叶斯强化学习算法[29]，该算法利用概率提供了先验知识以及对于不确定的“状态-序列”选择提供了更简洁的方法。将强化学习算法与其他机器学习算法相结合也成为目前热门的研究方向，例如强化学习与在线学习结合、强化学习与迁移学习结合、强化学习与深度学习结合等。RL方法把模型划分为了model-based和Model-free两种分类，model-based方式依赖于交互的环境，其实环境的模型是已知的但是环境的参数是未知的，Model-based方式通过已知模型拟合模型参数进行决策，而Model-free则是不考虑环境模型，通过一些方式进行对状态、行为和策略的估计进行决策，在这方典型的两个例子就是AlphaZero[30]和MuZero[31]。

在强化学习应用中，常常把value-based方式和policy-based方式分为两个大类，

这两类在组合优化方面的应用也往往不同。

对于value-based方式，RL通过估计每个状态s的值V(s)或者对于state-action估计Q-table的Q(s,a)进行对行为和动作的决策，对于V(s)和Q(s,a)的估计采用贝尔曼等式进行迭代更新，通常采用TD(0)方式[32]进行迭代更新，这样就得到了Q-learning的算法，其中每个状态用神经网络拟合可得到Deep Q-networks（DQN）算法[33]。DQN广泛应用于求解各种问题，其中包括组合优化问题，例如最大割问题[34]，最小顶点覆盖问题[35,39]，旅行商问题[35]，集合覆盖问题[38]，最大独立集问题[37]，最大公共子图问题[36]等。其中大多数工作都是使用学习方式或者策略在组合优化问题里找到可行解，但是还有另一种方式[37]采用学习方式寻找最优搜索顺序用来收紧最大独立集和最大割的求解过程的松弛范围，以此对算法进行优化。

对于policy-based方式，RL通过迭代直接优化决策动作π，以此为代表的算法就是REINFORCE[40]，Actor-Critic[26]一类算法（A2C，A3C），以及其他更进一步的扩展算法，例如Proximal Policy Optimization（PPO）[41]，Deep Deterministic Policy Gradient（DDPG）[42]，其中PPO可以在带约束的情况下进行更新，DDPG是一种off-policy的A2C算法。Policy-based首次使用梯度算法求解旅行商问题[43]和0-1背包问题[43]。

**三、主要研究内容**

圆装箱（Equal Circle Packing Problem，ECPP）问题，采用拟物理方式求解该问题，圆与圆之间通过嵌入深度和方向计算圆的梯度和势能，并采用梯度下降等方法（例如拟牛顿法，BFGS法）对势能函数进行优化，最后获得一个局部最优解。通过采用bandit模型对局部最优解进行收缩-跳坑处理，使得该解跳出局部最优并重新进行梯度势能下降，以此不断重复搜索可行解。

最大公共子图问题（maximal common subgraph，MCS），采用蒙特卡洛树搜索对分支限界算法进行优化和剪枝。

**四、研究的关键技术**

蒙特卡洛树搜索又称随机抽样或统计试验方法，属于计算数学的一个分支，它是在上世纪四十年代中期为了适应当时原子能事业的发展而发展起来的。传统的经验方法由于不能逼近真实的物理过程，很难得到满意的结果，而蒙特卡洛树搜索方法由于能够真实地模拟实际物理过程，故解决问题与实际非常符合，可以得到很圆满的结果。这也是以概率和统计理论方法为基础的一种计算方法，是使用随机数（或更常见的伪随机数）来解决很多计算问题的方法。将所求解的问题同一定的概率模型相联系，用电子计算机实现统计模拟或抽样，以获得问题的近似解。

蒙特卡罗树搜索大概可以被分成四步。选择(Selection)，拓展(Expansion)，模拟(Simulation)，反向传播(Backpropagation)。

在开始阶段，搜索树只有一个节点，也就是我们需要决策的局面。

搜索树中的每一个节点包含了三个基本信息：代表的局面，被访问的次数，累计评分。

[1]选择(Selection)

在选择阶段，需要从根节点，也就是要做决策的局面R出发向下选择出一个最急迫需要被拓展的节点N，局面R是是每一次迭代中第一个被检查的节点；

对于被检查的局面而言，他可能有三种可能：

1)该节点所有可行动作都已经被拓展过

2)该节点有可行动作还未被拓展过

3)这个节点游戏已经结束了

对于这三种可能：

1)如果所有可行动作都已经被拓展过了，那么我们将使用UCB公式计算该节点所有子节点的UCB值，并找到值最大的一个子节点继续检查。反复向下迭代。

2)如果被检查的局面依然存在没有被拓展的子节点(例如说某节点有20个可行动作，但是在搜索树中才创建了19个子节点)，那么我们认为这个节点就是本次迭代的的目标节点N，并找出N还未被拓展的动作A。执行步骤[2]

3)如果被检查到的节点是一个游戏已经结束的节点。那么从该节点直接执行步骤{4]。

每一个被检查的节点的被访问次数在这个阶段都会自增。

在反复的迭代之后，我们将在搜索树的底端找到一个节点，来继续后面的步骤。

[2]拓展(Expansion)

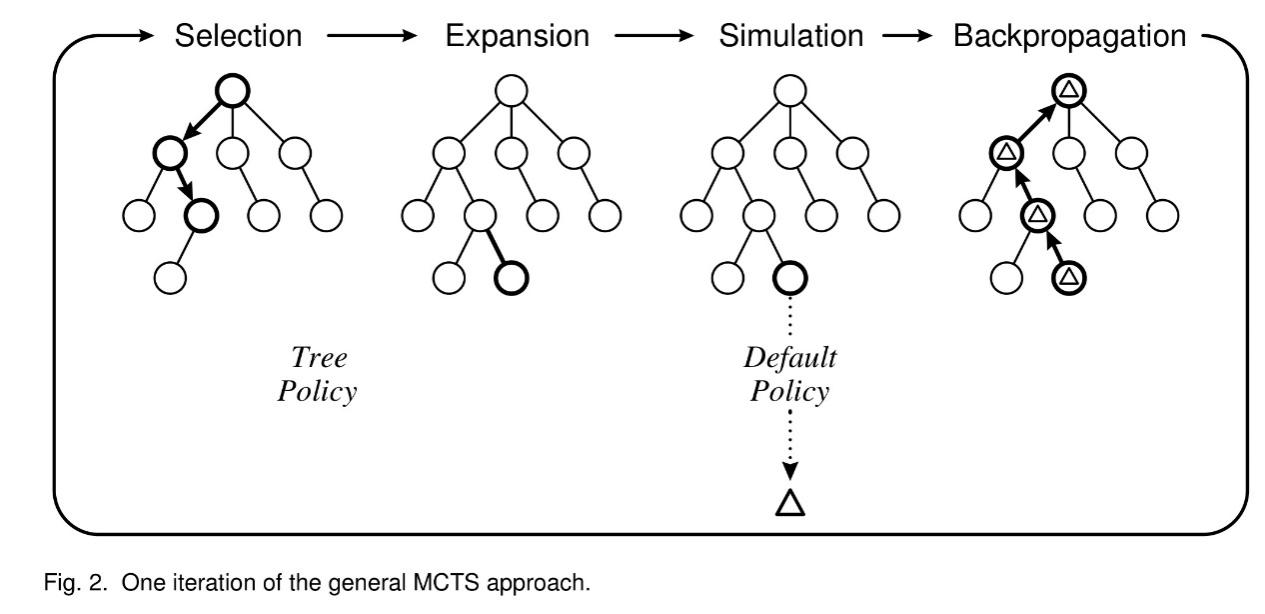
在选择阶段结束时候，我们查找到了一个最迫切被拓展的节点N，以及他一个尚未拓展的动作A。在搜索树中创建一个新的节点Nn作为N的一个新子节点。Nn的局面就是节点N在执行了动作A之后的局面。

[3]模拟(Simulation)

为了让Nn得到一个初始的评分。我们从Nn开始，让游戏随机进行，直到得到一个游戏结局，这个结局将作为Nn的初始评分。一般使用胜利/失败来作为评分，只有1或者0。

[4]反向传播(Backpropagation)

在Nn的模拟结束之后，它的父节点N以及从根节点到N的路径上的所有节点都会根据本次模拟的结果来添加自己的累计评分。如果在[1]的选择中直接发现了一个游戏结局的话，根据该结局来更新评分。

每一次迭代都会拓展搜索树，随着迭代次数的增加，搜索树的规模也不断增加。当到了一定的迭代次数或者时间之后结束，选择根节点下最好的子节点作为本次决策的结果。

**五、研究的技术方案**

拟结合机器学习领域的降维算法对组合优化问题的搜索空间进行降维，并且在基本保持解空间的一些特性的前提下，降低求解复杂度，提高算法运行效率。对于启发式或元启发式搜索算法，若其在可行解上进行一系列变换和更新，通过对每个变元进行评分（赋予概率）的机制，引入强化学习的值评估和策略评估对评分进行更新， 同时结合强化学习的ε贪心等方式进行策略选择，在强化学习的基础上，提出新的组合强化模型的框架。对于在搜索过程或者求解推导过程得到最优解的算法，拟将蒙特卡洛树搜索方法和强化学习中值函数和TD(λ)的思想结合对算法搜索树的分支经行评估和决策，使得智能体在搜索过程中更倾向优先搜索最有可能搜索到最优解的分支而且能把一些无用分支或者劣分支进行裁剪，从而提高每次扩展搜索分支的质量和效率。

**六、主要实验研究情况**

目前已对国内外相关研究现状进行了较为充分的调研，同时对关键技术有较为深入的了解，正在对相关技术方案进行设计与改进。

截止到目前，已以发表一篇，在投一篇：

[1]何琨, 邹晟昊, **周建荣**. 大规模稀疏图的极大团枚举算法. 2017年全国理论计算机科学学术年会, 华中科技大学学报(自然科学版), 2017, 45(12):1-6.

[2]Jiongzhi Zheng, Kun He, **Jianrong Zhou**, Yan Jin, Chumin Li.  Combining Reinforcement Learning with Lin-Kernighan-Helsgaun Algorithm for the Traveling Salesman Problem, AAAI 2021, under phase II review.

**七、课题研究进展计划**

（1） 2020.10.16 - 2020.10.31 对国内外研究现状中的算法进一步深入研究，复现其中的关键细节；

（2） 2020.11.1 - 2020.11.30 设计和完善技术方案，并论证其可行性，筛选出可实施的有效技术方案；

（3） 2020.12.1 - 2020.12.31 针对有效技术方案，设计合理的模型并实现，同时进行模型调优，结构优化；

（4） 2021.1.1 - 2021.1.31 与国内外最新研究的算法进行实验对比，分析实验结果，进一步改进设计的模型与机构；

（5） 2021.2.1 – 2021.3.31 撰写并发表相关论文；

（6） 2021.4.1– 2022.1.31 以此论文为基础发表一系列相关论文；

（7） 2022.2.1 – 2022.6.30撰写博士学位论文， 完成博士学位论文的后续相关工作。

**参考文献**

1. Glover, Fred, and Manuel Laguna. "Tabu search." Handbook of combinatorial optimization. Springer, Boston, MA, 1998. 2093-2229.
2. Whitley, Darrell. "A genetic algorithm tutorial." Statistics and computing 4.2 (1994): 65-85.
3. Dorigo, Marco, Mauro Birattari, and Thomas Stutzle. "Ant colony optimization." IEEE computational intelligence magazine 1.4 (2006): 28-39.
4. Lawler, Eugene L., and David E. Wood. "Branch-and-bound methods: A survey." Operations research 14.4 (1966): 699-719.
5. Zhou, Yangming, Jin-Kao Hao, and Béatrice Duval. "Reinforcement learning based local search for grouping problems: A case study on graph coloring." Expert Systems with Applications 64 (2016): 412-422.
6. Zhou, Yangming, Béatrice Duval, and Jin-Kao Hao. "Improving probability learning based local search for graph coloring." Applied Soft Computing 65 (2018): 542-553.
7. Shimomura, Masato, and Yasuhiro Takashima. "Application of monte-carlo tree search to traveling salesman problem." The 20th Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI). 2016.
8. Liu, Yan-li, et al. "A Learning based Branch and Bound for Maximum Common Subgraph Problems." arXiv preprint arXiv:1905.05840 (2019).
9. University N, Nanjing. Study on an Average Reward Reinforcement Learning Algorithm[J]. Chinese Journal of Computers, 2007, 30(8):1372-1378.
10. Fu Q M, Liu Q, Wang H, et al. A novel off policy Q(λ) algorithm based on linear function approximation[J]. Chinese Journal of Computers, 2014.
11. Kober J, Peters J. Reinforcement Learning in Robotics: A Survey[J]. International Journal of Robotics Research, 2013, 32(11):1238-1274.
12. Wei YZ, Zhao MY. A Reinforcement Learning-based Approach to Dynamic Job-shop Scheduling[J]. 自动化学报, 2005, 31(5):765-771.
13. Ipek E, Mutlu O, Martínez J F, et al. Self-Optimizing Memory Controllers: A Reinforcement Learning Approach[J]. ACM SIGARCH Computer Architecture News, 2008, 36(3):39-50.
14. Tesauro G. TD-Gammon, a Self-Teaching Backgammon Program, Achieves Master-Level Play[J]. Neural Computation, 1989, 6(2):215-219.
15. Scheffer T, Spiliopoulou M. Proceedings of the 17th European conference on Machine Learning[C]// European Conference on Machine Learning. Springer-Verlag, 2006.
16. Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: a methodological tour d’horizon, arXiv preprint arXiv:1811.06128.
17. M. Lombardi, M. Milano, Boosting combinatorial problem modeling with machine learning, arXiv preprint arXiv:1807.05517.
18. N. Vesselinova, R. Steinert, D. F. Perez-Ramirez, M. Boman, Learning combinatorial optimization on graphs: A survey with applications to networking, arXiv preprint arXiv:2005.11081.
19. J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, arXiv preprint arXiv:1812.08434.
20. T. Guo, C. Han, S. Tang, M. Ding, Solving combinatorial problems with machine learning methods, in: Nonlinear Combinatorial Optimization, Springer, 2019, pp. 207–229.
21. Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems. 2014.
22. Sutton R S, Barto A G. Reinforcement learning : an introduction[J]. IEEE Transactions on Neural Networks, 2005, 16(1):285-286.
23. 陈学松, 杨宜民。强化学习研究综述[J]。计算机应用研究, 2010, 27(8):2834-2838
24. Watkins C J C H. Learning from Delayed Rewards[J]. Robotics & Autonomous Systems, 1989, 15(4):233-235.
25. Singh S, Jaakkola T, Littman M L, et al. Convergence Results for Single-Step On-Policy Reinforcement-Learning Algorithms[J]. Machine Learning, 2000, 38(3):287-308.
26. Barto A, Sutton R, Anderson C. Neuronlike Elements that Can Solve Difficult Learning Control Problems[J]. IEEE Transactions on Systems Man & Cybernetics, 1983, SMC-13(5):834-846.
27. Preux P, Girgin S, Loth M. Feature discovery in approximate dynamic programming[C]// IEEE Symposium on Adaptive Dynamic Programming & Reinforcement Learning. IEEE, 2009:109-116.
28. Aissani N, Beldjilali B, Trentesaux D. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach[J]. Engineering Applications of Artificial Intelligence, 2009, 22(7):1089–1103.
29. Ghavamzadeh M, Mannor S, Pineau J, et al. Bayesian Reinforcement Learning: A Survey[J]. Foundations & Trends® in Machine Learning, 2016, 8(5-6):359-483.
30. D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, D. Hassabis, Mastering chess and shogi by self-play with a general reinforcement learning algorithm (2017). arXiv:1712.01815.
31. J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, T. Lillicrap, D. Silver, Mastering atari, go, chess and shogi by planning with a learned model (2019). arXiv:1911.08265.
32. R. S. Sutton, Learning to predict by the methods of temporal differences, Machine learning.
33. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning, Nature.
34. T. D. Barrett, W. R. Clements, J. N. Foerster, A. Lvovsky, Exploratory combinatorial optimization with reinforcement learning, arXiv preprint arXiv:1909.04063.
35. E. Khalil, H. Dai, Y. Zhang, B. Dilkina, L. Song, Learning combinatorial optimization algorithms over graphs, in: NeurIPS, 2017
36. Q. Cappart, T. Moisan, L.-M. Rousseau, I. Prémont-Schwarz, A. Cire, Combining reinforcement learning and constraint programming for combinatorial optimization, arXiv preprint arXiv:2006.01610.
37. Q. Cappart, E. Goutierre, D. Bergman, L.-M. Rousseau, Improving optimization bounds using machine learning: Decision diagrams meet deep reinforcement learning, in: AAAI, 2019.
38. Y. Bai, D. Xu, A. Wang, K. Gu, X. Wu, A. Marinovic, C. Ro, Y. Sun, W. Wang, Fast detection of maximum common subgraph via deep q-learning, arXiv preprint arXiv:2002.03129.
39. J. Song, R. Lanka, Y. Yue, M. Ono, Co-training for policy learning, arXiv preprint arXiv:1907.04484.
40. R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Machine Learning.
41. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms (2017). arXiv:1707.06347.
42. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning (2015). arXiv:1509.02971.
43. I. Bello, H. Pham, Q. V. Le, M. Norouzi, S. Bengio, Neural combinatorial optimization with reinforcement learning (2016). arXiv: 1611.09940.
44. Applegate, D. L.; Cook, W. J.; and Rohe, A. 2003. Chained Lin-Kernighan for Large Traveling Salesman Problems. INFORMS Journal on Computing 15(1): 82–92.
45. Yamada, Takeo, Seija Kataoka, and Kohtaro Watanabe. "Heuristic and exact algorithms for the disjunctively constrained knapsack problem." Information Processing Society of Japan Journal 43.9 (2002).
46. Dowling, William F., and Jean H. Gallier. "Linear-time algorithms for testing the satisfiability of propositional Horn formulae." The Journal of Logic Programming 1.3 (1984): 267-284.
47. H. Zhang and M. Stickel (1996). An efficient algorithm for unit-propagation. In Proceedings of the Fourth International Symposium on Artificial Intelligence and Mathematics.
48. Davis, Martin, and Hilary Putnam. "A computing procedure for quantification theory." Journal of the ACM (JACM) 7.3 (1960): 201-215.
49. Davis, Martin, George Logemann, and Donald Loveland. "A machine program for theorem-proving." Communications of the ACM 5.7 (1962): 394-397.
50. Moskewicz, Matthew W., et al. "Chaff: Engineering an efficient SAT solver." Proceedings of the 38th annual Design Automation Conference. 2001.

|  |  |  |
| --- | --- | --- |
| **研究生签字**  **指导教师签字**  **院(系、所)领导签字** |  |  |

**年 月 日**