学号\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

密级\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

武汉大学本科毕业论文

面向动态TSP问题的深度强化学习优化算法研究

院（系）名 称：弘毅学堂

专 业 名 称 ：计算机科学与技术

学 生 姓 名 ：徐哲豪

指 导 教 师 ：王峰 副教授

二○二二年四月

**郑 重 声 明**

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名： 日期：

摘要

旅行商问题（TSP）被广泛用来测试求解组合优化问题的方法好坏，如今旅行商问题已经渗透到运筹学、人工智能等众多领域。传统的旅行商问题假定城市的位置和数量是固定不变的。然而，在现实生活中，访问的需求和交通状况会随着时间的推移而改变，这就带来了动态旅行商问题(DTSP)。随着问题规模的增大和实时性的要求变高，传统的运筹优化方法面临的计算压力过大。而深度强化学习在游戏、无人驾驶、工业自动化等领域都取得了重大成功，展现了强大的求解能力和泛化能力。

因此，本文对深度强化学习应用在动态旅行商问题的求解展开研究，在指针网络的基础上进行了改进，提出用来解决动态旅行商问题的交替编码解码模型，舍弃了编码器的循环神经网络结构，改为简单的特征嵌入，同时在解码器中的循环神经网络之前增加了一层嵌入来提取上一步指针指向的输入的特征。

实验将本文模型应用于10、20、50个城市的动态旅行商问题，选择贪心策略的结果来与本模型的结果进行对比，发现模型有了较大程度的学习，对于10、20个城市的问题，模型能够找到十分接近最优解的路径，然而对于50个城市的动态旅行商问题，模型在短时间的训练效果不是很理想，后面还会出现过拟合现象。总的来说，本模型能够有效的用来解决小规模的动态旅行商问题。

关键词：动态旅行商问题；深度强化学习；指针网络；行动者-评价者算法；序列到序列

**ABSTRACT**

The travelling salesman problem (TSP) is widely used to test the methods for solving combinatorial optimization problems. Today the travelling salesman problem has permeated many fields such as operations research and artificial intelligence. The traditional travelling salesman problem assumes that the location and number of cities are fixed. However, in real life, the demand for visits and traffic conditions change over time, which brings about the dynamic travelling salesman problem (DTSP). As the problem size increases and the requirement of real-time becomes higher, traditional operations optimization methods face excessive computational pressure. In contrast, deep reinforcement learning has achieved significant success in the fields of gaming, unmanned vehicles, and industrial automation, demonstrating powerful solving and generalization capabilities.

Therefore, this paper investigates the application of deep reinforcement learning to the solution of the dynamic travelling salesman problem, and improves the pointer networks by proposing an alternating encoding and decoding model for solving the dynamic travelling salesman problem, discarding the recurrent neural network structure of the encoder and replacing it with a feature embedding, while adding a layer of embedding prior to the recurrent neural network in the decoder to extract the features of input pointed by the pointer in the previous step.

The model is applied to the dynamic travelling salesman problem with 10, 20, and 50 cities, and the greedy strategy are selected to compare with this model. It is found that the model has a great degree of learning. For 10 and 20 cities, the model is able to find a path very close to the optimal solution, however, for 50 cities, the model is not very effective in a short time of training, and overfitting will occur later. Overall, the model can be effectively used to solve the small-scale dynamic travelling salesman problem.

**Key words:** dynamic travelling salesman problem； deep reinforcement learning； pointer networks；actor-critic algorithm； seq2seq

目 录

1 绪论

1.1 概述 1

1.2 相关研究 2

1.3 问题定义 5

1.4 论文结构安排 5

2 背景知识

2.1 背景知识 6

2.2 本章小结 8

3 交替编码解码模型

3.1 模型流程 9

3.2 模块介绍 10

2.3.1 编码器 10

2.3.2 解码器 11

2.3.3 注意力机制 11

2.3.4 指针计算 12

3.3 强化学习训练 13

3.4 本章小结 14

4 实验与分析

4.1 数据集 15

4.2 实验步骤 15

4.3 实验结果以及分析 17

4.4 本章小结 19

5 总结和展望

5.1 总结 20

5.2 未来的工作 20

参考文献 21

致谢 24

1 绪论

1.1 概述

在实际的生活应用中，有一类需要从所有离散空间组合的集合中找到结果最优的方案，这类问题就是组合优化问题。常见的组合优化问题有旅行商问题、车辆路径问题、背包问题、装箱问题、施泰纳树问题、最小控制集问题、最小顶点覆盖问题、最小集合覆盖问题、最大割问题、平行机排序问题。旅行商问题是典型的NP难的组合优化问题，在国防、电力、通信、产品制造、路径规划、交通物流等方面有广泛的应用背景。例如机器人来进行电力巡检、物流盘点等场景的路径规划问题就可以建模为旅行商问题。出租车、物流公司的车辆调度问题也可以理解为在旅行商问题的基础上叠加了商人数量，取送货需求和多种约束条件。因此快速高效地求解组合优化问题有很高的实用价值。更进一步，很多实际生活中的场景需要考虑动态调度问题，访问的需求和每次访问的代价会随着时间而变化[14]，此时静态优化方法具有明显的局限性。问题一旦发生改变，传统方法就需要重新搜索求解，这样会大大地增加计算代价，很不利于问题求解。

随着问题规模的不断增大、问题更新速度的变快，运用传统方法求解组合优化问题在求解的精度、速度、泛化能力等方面都受到很大的冲击。近年来深度学习和强化学习越来越火热，深度强化学习对很多领域的传统方法有了颠覆性地影响，尤其是AlphaGo、Atari的成功应用，表明深度学习和强化学习的结合会有令人惊喜的效果，展现了强大的决策能力和学习能力，这些成功的经验为求解旅行商等组合优化问题提供了全新的思路和方法。深度强化学习可以用来帮助指导决策过程，这与在离散空间获得最优决策的组合优化问题十分相似和契合。虽然这种端到端的方法提供的解的最优性难以保证，但很多研究已经证明可以接近最优解，并且在求解速度，泛化能力等方面都要远远优于传统方法。

上述研究大多数都是基于静态旅行商问题，对动态旅行商问题的研究较少，因此本文对深度强化学习在动态旅行商问题的优化展开研究，对指针网络的模型进行适当的修改，然后将修改后的模型应用于动态旅行商问题，在对结果进行分析后得到了积极的结论。本文的主要贡献有两个方面：

1. 将指针网络的框架应用于动态旅行商问题，采用编码解码交替进行的方法，移除编码器的循环神经网络，将解码器中的循环神经网络选用门控循环单元，在其之前对上一步指针增加特征嵌入层。
2. 在编码器的输入端输入了所有城市的初始坐标信息，由于坐标改动较小，输入的初始坐标信息将在模型全程指导输出，同时在指针计算的时候有选择的选择编码结果参与计算，忽略掉一些不重要的和冗余信息。

1.2 相关研究

旅行商这一类组合优化问题的主要特点是决策空间是离散的数量有限的集合，理论上可以通过穷举法得到问题的最优解。但是，随着问题规模的逐步增大，可行解会呈指数级增长，无法在多项式时间内求解。

研究旅行商问题的传统方法主要可以分为精确求解和近似求解两大类，其中精确求解主要包括动态规划法（Dynamic programming）和分支界定法（Branch and bound）[5]等等，其思想都是将原问题的规模减小，分解为子问题，然后通过不断迭代的方式求解。精确求解可以得到全局最优解，但是在应用于大规模问题时计算量消耗巨大。近似求解主要为近似算法（Approximate algorithms）和启发式算法（Heuristic algorithms）[6]，近似算法顾名思义，只能找到有一定质量的解，包括贪心算法、线性规划和松弛算法等等，近似算法虽然不能找到最优解，但是能够保证在多项式时间内找到与最优解的路径长度比值不超过一定上限的解，这个上限定义为近似度，近似度为大于1的值，且越接近于1就表明该算法的性能越好。例如有一个近似算法为Christofides 算法，近似度为1.5。用来求解旅行商问题的启发式算法主要有禁忌搜索、演化算法、蚁群优化算法、粒子群算法、模拟退火算法等。这一类方法只能在可行时间内找到较理想的解，对最优性没有保证。比较领先的启发式搜索算法为迭代局部搜索（iterated local search， ILS）算法。多个算例集的测试结果证明了其求解旅行商问题的强大能力。迭代局部搜索算法是从构造的初始解开始，通过一系列操作对解的局部进行变换，得到一些邻域解，然后计算邻域解并选择有改进效果的解作为当前解，之后迭代执行前面的步骤，直至没有改进的解，这时就表明得到了一个局部最优的解，然后重新构造初始解，重复上述流程，直到满足停止条件[25]。这类迭代型搜索算法在求解速度方面不如端到端的方法，但是解的优化效果强于端到端的方法。后来Gao等人利用深度强化学习对大规模邻域搜索的destroy和repair两个算子进行学习，选择图注意力网络（graph attention network）对问题的输入进行特征编码，采用基于循环神经网络的解码器输入destroy和repair两个算子[26]，其中destroy算子是从当前解集中移除一些元素，repair算子是将destroy算子移除的元素按照特定的顺序重新插入到当前解中，模型对destroy的选择策略和repair算子的排序策略进行学习。训练方法采用近端策略优化（proximal policy optimization）算法， 结果表明该模型相较于传统启发式搜索算法有更快的收敛性能，结果接近但未到达最优解。

在机器学习领域，Hopfield等人提出一种Hopfield网络，首次将神经网络的方法应用于求解小规模的旅行商问题。Vinyals等人与2015年将组合优化问题类比为机器翻译的序列映射问题，针对序列到序列（Seq2Seq）模型输入输出的维度固定的限制，改进并提出了可以用来求解组合优化问题的指针网络模型[10]（Pointer Networks，Ptr-Net）这一新架构，加入注意力机制使得序列模型不受输入输出维度的限制。其核心思想是通过编码器对问题的输入向量进行编码得到高维度的特征向量，然后利用解码器基于注意力的计算方式以自回归（autoregressive）的方式逐步构造解集。自回归是指每次选择解集的一个元素，然后在已选择的元素基础上添加下一个元素，直至构造完整的解集。此后很多基于神经网络来求解组合优化问题的研究都是基于此架构来展开的，指针网络在车辆路径规划问题、Knapsack、多目标旅行商问题等其他组合优化问题也显示了显著的优化效果。然而指针网络不适合用于求解动态输入的问题，因为编码器为循环神经网络，在每次输入改变时网络都要重新计算输出。另一个问题是深度学习需要有大量已有的高质量解集来监督训练，而强化学习是通过智能体不断地与环境进行交互，同时用奖励信号来指导学习，这就克服了需要大量高质量解集的问题。很多将深度学习与强化学习相结合的研究都得到了接近甚至等同于最优解的结果，并且运算速度和泛化程度相较于传统方法大大提高。

Transformer框架也被用来求解组合优化问题，其延续了基于注意力机制的编码器-解码器结构，Transformer的编码器和解码器都采用自注意力（self-attention mechanism）和全连接层（fully connected layers, FC）组成，模型中的多头注意力机制（multi head attention， MHA）的计算方法，增加了更多的计算层数，这样能够深入挖掘隐藏的特征信息，有效避免信息丢失的问题。Kool等人的改进主要包括两方面，第一个是采用了多头注意力的机制来实现编码器的功能，第二个是解码器在每一步解码过程中考虑的是第一步的决策和前两步的决策。另外在训练时，考虑到同时训练actor网络和critic网络很低效，设计了一种rollout baseline来代替critic网络，即在之前训练过程中得到的模型里，选择在测试集表现最好的一个，利用这个模型，贪心地选择每个动作，将以此方式得到的奖励作为b（s），如果当前的策略比历史最优的策略好，那么就对模型进行正向激励，这样训练的收敛能力远远强于传统方法，性能也超过了之前的端到端模型。

近年来图神经网络（graph neural networks， GNN）的流行为解决旅行商问题提供了新的思路和工具[3]。图神经网络能够根据图的原始信息以及不同节点之间的关系来计算得到各个节点的特征向量，根据所有节点的特征向量进行节点预测、边预测等任务，应用于旅行商问题就是能够根据城市的坐标信息、访问信息和城市之间的距离计算不同节点之间存在边的概率，也就是路径解包含这条边的概率，但这种方法还需要其他辅助来构造符合问题规则的解，例如可以借助波束搜索、树搜索等方式来根据选择概率进行可行解的构造。另外一种方法是Dai等人根据图神经网络计算旅行商问题中剩余可选城市的Q值，随后用贪婪策略选择Q值最大的城市添加到现有的解集中，重复此步骤直到获得一个完整的解集。然后采用深度Q学习对图神经网络的参数进行训练，让图神经网络能够准确估计每个时刻每个城市的Q值。随后Mittal等用图卷积神经网络（graph convolutional networks， GCN）对模型进行了改进，结果获得了41%的优化能力提升。Ma等人根据图神经网络主要用于计算节点的特征向量这一条件，将图神经网络与指针网络相结合[27]，该模型编码器包含节点编码器和图编码器两部分，其中节点编码器对城市坐标进行线性映射，输入到长短期记忆网络中得到每个城市的点特征向量，图编码器则是通过图神经网络对所有城市编码，得到该问题下所有城市的图特征向量，模型再根据点特征向量和图特征向量用注意力机制计算选择每个城市的概率，另外还引入了向量语义来提高模型的泛化能力。采用分层强化学习对该模型进行训练，结果证明了该模型的有效性。

虽然基于强化学习来求解组合优化问题的研究已经取得了重大的进展，但有一部分研究领域仍在探索阶段，有几个方面需要继续开展研究：

1. 扩大组合优化问题的规模，可以采用分治的思想，将大规模问题分解为若干个相关的小规模问题，然后学习层次化的策略，之后可以对小问题的策略进行组合，得到全局最优策略，这个过程中重点在于如何有效地实现问题的分解与组合，防止信息丢失。
2. 求解更复杂的组合优化问题，实际生活中的组合优化问题具有动态性、多约束条件、多目标等特征，对这类组合优化问题的研究相对于经典静态的组合优化问题较少，如何对实际问题进行动态在线求解是一个重要的研究热点。
3. 模型改进，如今大部分优化模型都是端到端的形式，解的质量没有保证，如何改进模型，提高网络的性能，压缩模型的大小，提高问题求解的效率需要解决。

1.3 问题定义

旅行商问题可以描述为存在一个商人和一系列城市，商人要按照顺利遍历每个城市，所有城市只能访问一次，然后回到出发点。问题的优化目标是找到使得总路径最短的访问顺序。对于动态旅行商问题，变化量可以为城市的数量或者位置[15]，本文仅考虑城市位置在离散时间的变化。

假设所有城市之间都存在通路，将n个城市的集合表示为 ，同时第i个城市在时间t的坐标为，在时间t城市i与城市j之间的欧几里得距离，其中，变量定义为[1]当t时刻的路径为从城市i到城市j时为1，否则为0。

旅行商问题的优化目标即公式（1.1）：

（公式1.1）

1.4 论文结构安排

本论文内容按以下顺序展开：第一章先对旅行商问题的研究意义和研究背景进行概括论述，然后提出动态旅行商问题的数学描述。为了让读者更好了解本文模型的设计思路，第二章对模型相关的一些背景知识进行补充介绍。第三章详细介绍本文的模型。第四章给出用模型进行实验的步骤和结果，最后第五章进行总结和展望。

2 背景知识

2.1 背景知识

1）序列到序列模型

序列到序列模型又称为Seq2Seq模型[9]。这个模型主要是为了学习序列到序列映射函数，应用场景包括机器翻译、语音合成、手写体生成等等。模型的实现一般都基于编码器-解码器结构，通过编码器将输入序列进行编码，得到语义向量，解码器通过给定的语义向量和已经做出的历史预测来对下一步的输出进行预测。具体而言就是利用两个循环神经网络，一个循环神经网络作为编码器，另一个循环神经网络作为解码器，获得语义向量最简单的方式就是直接将最后一个输入的隐状态作为语义向量，也可以对其进行加工变换，还可以利用注意力机制来得到语义向量。解码器的方式有两种，一种是只在初始解码时刻输入语义向量，后面的运算都与语义向量无关。第二种方式是语义向量参与序列解码所有时刻的运算。

2）注意力机制

专家学者们通过类比人类的注意力提出了注意力机制，本质上就是每个特定时刻对某些特定信息的关注。就像人类观察某个场景时，会去重点关注一些特征，这就是注意力分配，注意力机制能够帮助我们在有限的计算资源条件下高效地处理信息。注意力机制主要应用于图像处理领域和自然语言处理领域，并取得了重大成功。基于注意力机制的编码器-解码器模型[28]通过实验证实比传统的编码器-解码器模型在性能上有十分显著的提升。

3）强化学习

强化学习定义为智能体（agent）与环境（environment）不断交互，通过每次交互得到奖励（reward）。学习的过程就是智能体在与环境交互的过程中不断尝试，通过所得到的奖励来指导策略，达到让奖励最大的目标[4]。强化学习可以建模为马尔可夫决策过程[13]（Markov Decision Process, MDP）,过程用五元组（S,A,P,R,γ）组成，其中S表示环境的状态集合或者智能体对环境的感知集合，A表示智能体的动作集合，P表示环境各个状态之间的转移概率，R是对应状态对应动作下智能体获得的奖励。γ是折扣因子，表示时间越久远的奖励越没有价值[20]，当γ接近于0时，表示智能体注重短期回报，当γ接近于1时，表示智能体注重长期回报。强化学习根据马尔可夫决策过程，可以分类为基于模型的强化学习算法和无模型的强化学习算法。

近年来深度学习模型越来越多地应用于强化学习中，形成了深度强化学习，在深度强化学习中，深度学习用来表示状态和策略，用神经网络来拟合强化学习中的值函数或者策略函数，然后强化学习来对问题建模，同时构造优化函数。

4）循环神经网络

循环神经网络的发明目的是为了处理序列当前输出与之前的历史信息的关系。从网络的结构而言，循环神经网络能够记忆历史信息，并对这个历史信息加以利用，影响后面的输出序列。也就是说，循环神经网络每个隐藏层之间的结点都是有联系的，隐藏层的输入不但包括输入层的输出，还有上一时刻隐藏层的输出。循环神经网络对每一个时刻的输入处理都是结合当前模型的状态来给出输出，也就是说，循环神经网络可以理解为是同一个神经网络被复制多次的结果，网络中的参数在不同时刻是共享的。在循环神经网络中可以加入随即失活策略（Dropout），如果模型的结构复杂，参数较多，同时提供的训练样本又较少，那这样训练得到的模型很容易出现过拟合现象，随机失活策略可以帮助循环神经网络适当程度地缓解过拟合现象的发生。在每个训练批次中，通过忽略掉一定比例的特征检测器，就是让一些节点的值为0，也就是让它失活，就可以很显著地减少过拟合现象。这样能够让模型泛化性加强，不会过于依赖某些局部的特征。循环神经网络一般只在不同层循环体之间使用随机失活策略，在同一层的循环结构之间一般不使用。常见的循环神经网络有长短期记忆网络、门控循环单元网络等等。循环神经网络主要用于自然语言处理、计算机视觉以及计算生物学等领域。

5）门控循环单元网络

循环神经网络的梯度很容易出现衰减或者爆炸的情况，虽然有裁剪的方式能够缓解梯度爆炸，但是梯度衰减无法解决。由于这个原因的存在，循环神经网络实际中很难捕捉时间相隔较长的时间序列中的依赖关系。这样的需求就诞生了基础循环神经网络的升级版，也就是门控循环单元网络。它的提出是为了能够更好地捕捉序列中时间间隔大的元素之间的依赖关系。它包含可以学习的门，通过这些门来控制信息的流动传播。门控循环单元引入了更新门（update gate）和重置门（reset gate）两个新概念[24]，这样就可以修改循环神经网络中隐藏状态的计算方法。门控循环单元中更新门和重置门的输入都是当前时间的输入和上一时间的隐藏状态，输出由全连接层加以激活函数得到。大量的实验证明，门控循环单元比长短期记忆网络的结构更简单，参数更少，实践的效果也没有变差，甚至某些特定的任务性能更好。

6）策略梯度

用表示一组状态-行为序列,表示轨迹的回报，表示轨迹出现的概率，那么强化学习的目标函数可以示为

表示轨迹的概率随参数变化最陡的方向，参数在该方向进行更新时，若沿着正方向，则该轨迹的概率会变大，而沿着负方向进行更新时，该轨迹的概率会变小。再看第二项，该项控制了参数更新的方向和步长。为正且越大则参数更新后该轨迹的概率越大；为负则降低该轨迹的概率，抑制该轨迹的发生。所以策略梯度能够提高高回报路径的概率，降低低回报路径的概率。轨迹回报就是一个评价器（critic），该评价器评价参数更新后，该轨迹出现的概率应该变大还是变小。如果变大，应该变大多少；如果减小，应该减小多少。也就是说，策略的参数调整幅度由轨迹回报进行评价。所以需要通过减去一个baseline，令其有正有负。而通常这个baseline选取平均值。减去平均值之后，值就变成有正有负了。也可以通过critic网络来估计，这就是actor-critic算法。

2.2 本章小结

本章介绍了与交替编码解码模型相关的背景知识，序列到序列模型是本文模型的基础，注意力机制是本文模型在特征嵌入后处理信息的重要手段，循环神经网络和门控循环单元网络各有特点，通过对其优缺点的思考以及本研究的需求来 进行网络模型的取舍，强化学习和策略梯度是最后训练本文模型的手段，这一方式相比监督学习有巨大优势。

3 交替编码解码模型

本章先介绍为了研究动态旅行商问题构建的交替编码解码模型的整体流程，然后对各个模块进行详细的说明。

3.1 模型流程

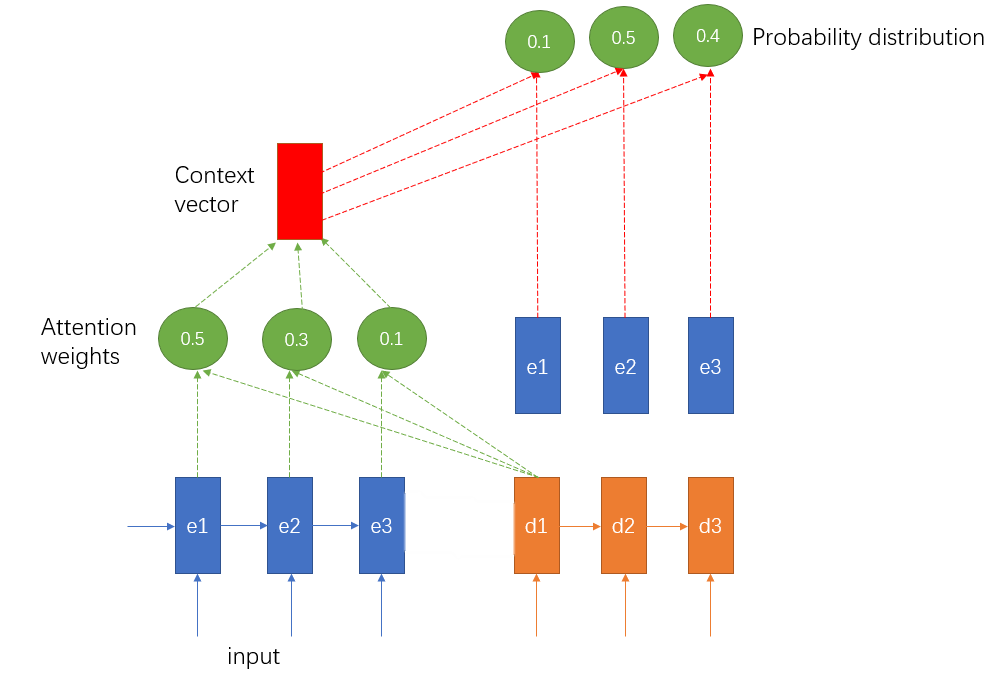


图3.1 模型流程图

本文的交替编码解码模型为带有注意力机制的编码器-解码器结构[17]，其中编码和解码过程交替进行。左下角的编码器读取当前输入的动态城市信息序列，一次一个城市，将输入编码为中间形式的序列，在每次解码完成信息更新后都会重新计算编码。在每个解码时刻i，解码器基于前一时刻解码器的输出即选择访问的城市来输出,通过和e来计算当前解码时刻与所有编码时刻对应的相似度系数（Attention weights），当网络得到当前解码时刻的所有相似度系数后，再以加权求和的形式将所有编码状态累加起来，就得到了语义向量（Context vector），然后将语义向量与每个编码时刻的状态按照指针网络的方法计算，得到的输出看作指向每个城市的指针，输出就是选择各个城市的概率分布。按照这个概率分布在可选城市集合中选择下一个要访问的城市，重复上述过程直到构造一个完整的问题解。

这个深度神经网络的输入是城市的信息序列，输出是城市的访问序列，通过对网络模型的参数训练就能够实现问题序列到解序列的准确映射。

3.2 模块介绍

3.2.1 编码器

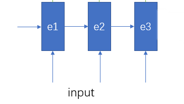


图3.2 编码器

编码器采用一层一维卷积嵌入层的结构，将动态旅行商问题的输入即城市的坐标信息转换为高维度的向量，这样做的目的是充分利用城市的结构信息，由于是动态旅行商问题，该部分的输入分为三部分，一部分是每个城市的欧式初始坐标，第二部分是每个城市动态的实时坐标，第三部分是城市的访问信息，也可以理解为当前解集信息，即关于城市是否已经被访问的掩码。

在指针网络一文中所采用的编码器结构为循环神经网络LSTM（长短期记忆），这样的结构主要是用于处理文本类问题，这一类问题输入的前后具有高度相关性和有序性。然而，对于旅行商问题这一类组合优化问题而言，其输入为所有城市的信息的组合而非排列，其输入次序与问题的解没有任何关系，只需要关心输入的内容。再加上动态旅行商问题的输入会一直变化，在每次变化后都需要重新计算新的编码向量，而循环神经网络不适合这样的重复计算过程。为了简化结构，也为了更好的效果，选择舍弃掉编码器部分的循环神经网络，只选用一层卷积嵌入[12]，这样会更好的符合组合优化问题与输入顺序无关的特性。通过实验结果也发现这样做确实有更好的效果。

假定表示第i个城市的初始二维坐标，表示在解码时刻t第i个城市的掩码以及实时坐标，用表示第i个城市的初始二维坐标编码后的高维度向量，表示在解码时刻t第i个城市的掩码以及实时坐标输入编码器得到的高维度向量。编码器在编码时刻i的输出为。

3.2.2 解码器

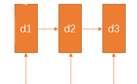


图3.3 解码器

解码器的工作一般是结合编码器的输入以及当前解的信息输出下一阶段对全部城市的评价，与一般的解码器结构有所不同，本模型采用的是一维卷积嵌入层连接门控循环单元[21]。由于编码器没有了循环神经网络，所以没有隐状态输入到解码器。旅行商问题具有马尔可夫性质[7]，即下一阶段的选择只与当前的状态有关，与历史无关，所以解码器在每一解码时刻的输入是上一时刻选择访问的城市的坐标，通过卷积层将输入的坐标信息进行嵌入后输入门控循环单元得到中间向量d，然后与编码器的输出一起参与注意力机制的计算得到语义向量，利用语义向量计算指向每个输入的条件概率。所以实际上本模型中的解码器工作只是概括历史的选择信息，整合当前解集，可以将本模型的解码器视为当前解集的编码器。真正的解码输出由后面的指针网络来负责。

3.2.3 注意力机制

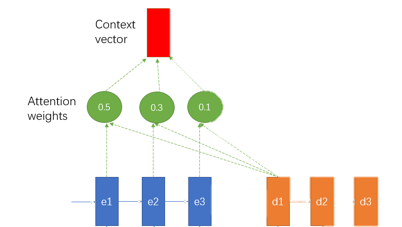


图3.4 注意力计算

注意力机制在深度学习中有着非常重要的作用。神经网络往往都会接收和处理非常庞大的数据量。然而类比人类的注意力，具体到每一个特定的时刻，需要关注的往往是一小部分，这就是注意力机制的用武之地[23]，它能够帮助我们利用有限的注意力资源从大量数据中筛选获得出高价值的重要信息。

在本模型中，注意力机制的运用目的是整合编码输出与解码器的输出，编码输出代表城市的位置信息和可否被访问信息，解码器的输出代表当前的解集信息也就是历史的选择，注意力机制将这两者综合，计算得到序列到序列模型中的语义向量。具体注意力的计算分为两步，第一步是计算所有输入信息的注意力分布a，也就是前面的attention weights，这个反映了当前解码时刻与所有编码时刻输出的相关程度[8]。第二步是根据计算得到的注意力的分布来计算输入编码后的信息的加权平均，得到当前解码时刻的语义向量c，计算方式见公式（3.1）、（3.2）、（3.3），即基于内容的注意力机制：

（公式3.1）

（公式3.2）

（公式3.3）

其中为模型中可训练的参数，这样得到的语义向量实际上代替了原有的编码器中循环神经网络的隐向量，并且语义向量在动态旅行商问题上相较于后者有更好的概括性。

3.2.4 指针计算

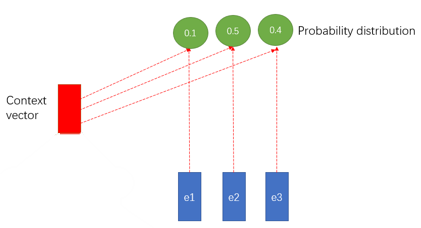


图3.5 指针计算

指针计算是一种简化的注意力计算，其核心思想是可以将注意力计算得到的权值分布即当前时刻与每个编码输出的相关程度直接当作条件概率的输出，也就是指向每个输出的指针，这个方法被证实比传统的方法更简单的同时有更好的效果。计算方式与前面的基于内容的注意力计算相同，只是这时得到的结果不代表注意力权重，直接代表当前解码时刻选择每个城市的条件概率，即：

（公式3.4）

（公式3.5）

会，也就是在指针计算的时候只输入城市的初始坐标，舍弃城市的实时坐标和访问信息，实际上这些是冗余信息，已经在注意力计算的时候吸收处理过了。实验结果表明，当问题规模增大时，有选择的输入一部分关键信息参与计算有更好的效果，这样让网络模型的参数更简单，提取特征信息更有效。为在每一解码时刻t选择j城市的概率，根据的概率分布来指导神经网络进行选择。在强化学习领域有一个很重要的概念为探索&利用（explore & exploit），这需要去权衡（tradeoff），如果总是选择输出概率最大的城市，即exploit，随着网络的收敛，很难获得新的可能更优的解,就好比去餐厅吃饭，如果一直去同一家，那就永远也发现不了更好吃的餐馆，但如果只知道一味探索，新的餐馆的味道很大可能都差于原来的餐馆。这点在推荐系统中十分重要，需要选择是推荐顾客感兴趣的还是推荐新的去发掘顾客的新兴趣。所以具体到本文研究的动态旅行商问题，选择按照输出的概率分布来取样，而不是贪婪地选概率最大的，同时在每个解码时刻之间都更新动态输入，实现坐标变化，然后重新输入编码器。

3.3 强化学习训练

Vinyals采用监督学习来训练指针网络，然而这样的缺陷在于：（1）难以获得大量最优路径的标签数据，（2）网络的性能极大地依赖标签的质量。相比之下，强化学习是用来训练神经网络的更好方法。

首先需要将旅行商问题建模为马尔可夫过程，其中状态就是城市的坐标信息以及每个城市是否已经访问，动作就是每一决策时刻选择访问的城市，所有动作组成的序列就是旅行商问题的解，反馈可以定义为路径长度的负数，这样与目标最小化路径长度一致。策略定义为从状态到动作的映射，本文采取随机策略，因为神经网络得到的是选择每个城市的概率。具体而言，假设动态旅行商问题每次都以第一个城市为起点，在当前解集的情况下，通过参数为θ的神经网络选择下一个要访问的城市，根据概率的链式法则[18]，最终生成完整的访问序列，假设n为动态旅行商问题的规模大小，则解的生成过程可以表示为：

（公式3.6）

其中表示在神经网络的参数θ下，对于输入为S的动态旅行商问题，生成访问序列的概率，为在当前访问序列的情况下以θ为参数的神经网络选择为下一个访问的城市的概率。假设 为访问序列的反馈，那么反馈的期望[19]可以表示为：

（公式3.7）

神经网络模型的最优参数组合应满足。

由于网络输出的是完整的决策序列，计算出的是总的路径长度，于是可以采用基于策略梯度的强化学习actor-critic算法来完成训练。在actor-critic算法中，需要做两个方面的近似，第一个是策略函数的近似： ，这就是行动者网络也就是指针网络负责的工作，其中θ为指针网络模型的参数。第二个是价值函数的近似：，其中w为critic网络的参数，critic网络使用三层卷积层的结构通过最后的求和来将输入的嵌入信息映射到对总反馈的估计[2]。

对于actor网络的参数，更新方法为[11]：

（公式3.8）

对于critic网络的参数，用均方差损失函数[16]作为w的梯度更新。

3.4 本章小结

本章主要介绍了交替编码解码模型，模型主要由编码器、解码器、注意力、和指针网络四部分组成。对于动态问题的输入，模型流程为编码—解码—更新坐标过程的循环，直至构造完整的问题解。编码器中移除循环神经网络能够让编码过程更迅速，在多次编码的情况下更高效。每次更新后重新编码能够让下一步的解码更对应当前情况的动作，输出更准确。

4 实验与分析

4.1 数据集

在本实验中，城市的二维坐标数据采取均匀分布随机生成在正方形单元里，取城市的数量分别为10，20，50来进行实验，训练集与测试集选用不同的随机数种子。需要指出的是，本文作为对动态旅行商问题研究的尝试，仅考虑全连通问题的情况。由于考虑的是动态TSP，所以在数据集中维护static和dynamic两个张量，static用来记录初始生成的坐标且在整个过程中始终保持不变，dynamic除了记录实时动态的坐标外，还增加一位掩码位来表示该城市是否已经被访问，dynamic允许在每个解码时刻之间改变。同时在数据集中构造三个函数：

1. 更新掩码函数update\_mask，方便起见假设商人总是从第一个城市出发。初始状态下将掩码第一位置0，其他位置1。每次访问后将当前访问的城市对应掩码置0。掩码除了输入编码器，还可以指导神经网络按照旅行商问题的规则选择访问的城市。
2. 更新坐标函数update\_dynamic，在每个解码时候之后，对未访问的城市进行坐标更新，对已经访问的城市更新坐标不会影响接下来的决策，还会让计算路径总长度变得麻烦，没有必要。更新方法为每次赋予一个服从正态分布N（0，0.02）的增量，取小增量是因为实际情况中，路径代价的变化量跟总代价比往往很小，在本实验中坐标经过这样的改变可能会超过0-1这个范围。
3. 计算路径长度函数reward，通过给定的访问序列和dynamic，计算该序列经过的总路径长度，以此为reward指导强化学习。

数据集采用Python torch.utils.data来处理。Dataset是代表数据的抽象类，可以继承和重写这个类来实现自己的目的。需要定义\_\_len\_\_和\_\_getitem\_\_这两个函数。如果在类中重写了\_\_getitem\_\_这个方法，那么就可以通过索引来对我们的实例对象进行取值。复写\_\_len\_\_能够获取数据的个数。Dataloader是用来处理模型的输入数据的工具类，由数据集（dataset）和采样器（sampler）组成。

4.2 实验步骤

1）实验参数

训练集的大小为4000，测试集的大小为1000。本文采用随机策略的Actor-critic（AC）算法来对模型进行训练。AC算法中的actor就是本文的指针网络模型，以20个城市为例，参数设置如下：

编码器1：Conv1d（input\_size = 2, filter=128, kernel\_size=1, stride=1）

编码器2：Conv1d（input\_size = 3, filter=128, kernel\_size=1, stride=1）

解码器1：Conv1d（input\_size = 2, filter=128, kernel\_size=1, stride=1）

解码器循环神经网络选择门控循环单元：GRU（input\_size = 128, output\_size=128, num\_layers=1）,同时对输出增加Dropout防止过拟合。

Critic网络总共包含五个编码器

编码器1：Conv1d（input\_size = 2, filter=128, kernel\_size=1, stride=1）

编码器2：Conv1d（input\_size = 3, filter=128, kernel\_size=1, stride=1）

编码器3：relu(Conv1d（input\_size = 256, filter=20, kernel\_size=1, stride=1）)

编码器4：relu(Conv1d（input\_size = 20, filter=20, kernel\_size=1, stride=1）)

编码器5：Conv1d（input\_size = 20, filter=1, kernel\_size=1, stride=1）

所有神经网络中的参数采用Xavier初始化，训练的轮数是20，批训练量（Batch\_size）默认为1，使用Adam优化器对网络参数进行更新，actor和critic的学习率都是固定的0.0005，城市数量为50时学习率为0.0001，取最大梯度为2进行梯度裁剪。

在具体训练时，编码阶段将static和dynamic同时输入到编码器参与编码。解码器初始输入取0，之后输入为上一步选择城市的坐标信息。解码器、编码器以及注意力层共同组成了指针网络。在具体实验时发现，当问题规模n较大时，对于最后指针的计算，只使用静态变量static的编码输出而忽略动态变量dynamic的编码输出参与计算时网络更快地收敛，最后求出的路径长度也更短。在训练阶段，每次动作的选择按照指针网络输出的概率分布取样，然后判断动作是否符合规则，是则停止，否则重复取样，直到取到合规的动作为止，在测试阶段就直接在可选集合中选择概率最大的动作。

2）实验环境

模型的训练和测试在同一设备完成，设备配置：系统为64位Windows 11，GPU为NVIDIA GeForce GTX 1050 Ti，CPU为Intel i7-8750H。编译语言采用Python 3.7，编译器使用的是Pycharm，深度学习使用的框架是Pytorch，同时使用了CUDA来实现GPU硬件加速。

4.3 实验结果以及分析

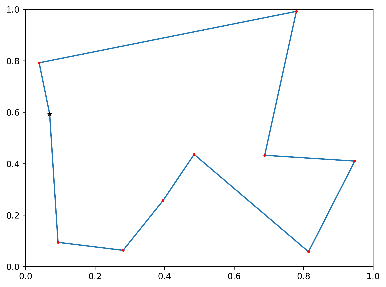
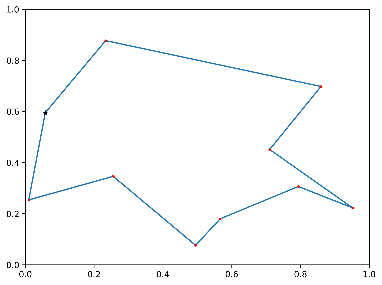
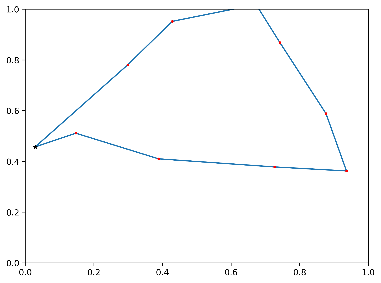
本文选用贪心策略或最近邻策略作为对比算法，该算法每次选择距离当前最近的城市访问，该算法应用于动态旅行商问题有比较良好的结果。另外借用指针网络文中的静态旅行商问题实验结果来进行粗略对比。实际上，动态旅行商问题的路径长度应高于静态TSP，因为动态旅行商无法对变化进行预测，只能根据当前的不完全信息来估计，理论上来说动态旅行商问题没有最优解，因为坐标变化是不可预知和控制的，只能从统计学角度用大量问题实例求解得到的平均结果来估计模型性能好坏。

将各个算法对数量1000的验证集求解，以结果的平均路径长度作为标准来进行比较，具体实验结果见表4.1：

表4.1 实验结果表

|  |  |  |  |
| --- | --- | --- | --- |
| 研究问题 | 指针网络(静态TSP) | 贪心策略 | 本文模型 |
| TSP10 | 2.88 | 3.39 | 3.08 |
| TSP20 | 3.88 | 5.37 | 4.74 |
| TSP50 | 7.66 | 10.14 | 12.37 |

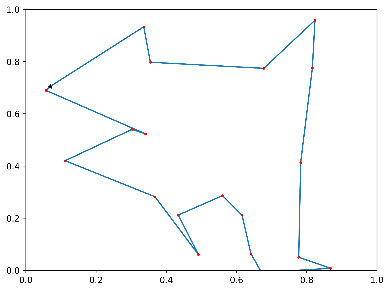
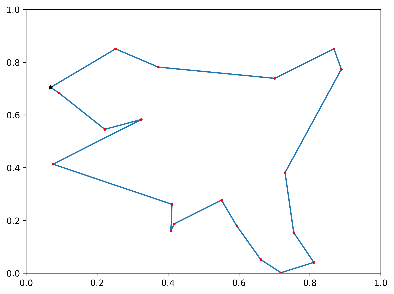
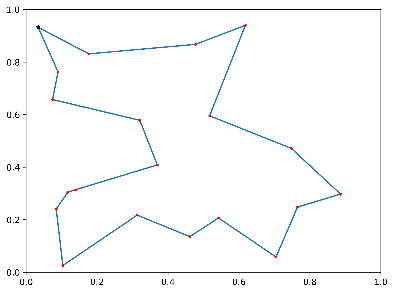
当城市数量为10时，网络在第37个训练轮次（epoch）收敛，每一百个样本平均训练时间为8秒，图4.1为10个城市的动态旅行商问题的三个实例解，图中是横纵坐标均为[0，1]区间的二维平面，用点来表示每个城市，坐标对应城市的位置，图中的闭合回路即为求解的解路径。



（a） （b） （c）

图4.1 10个城市的解示例

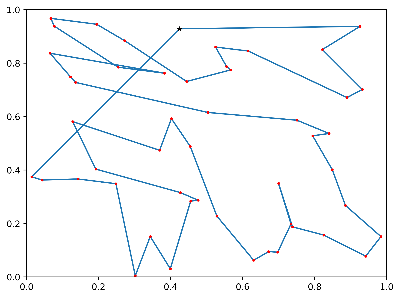
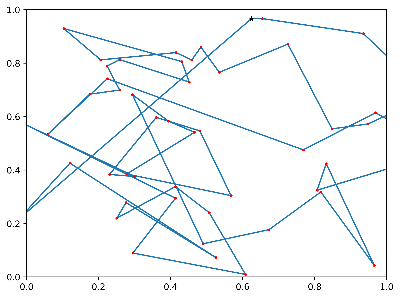
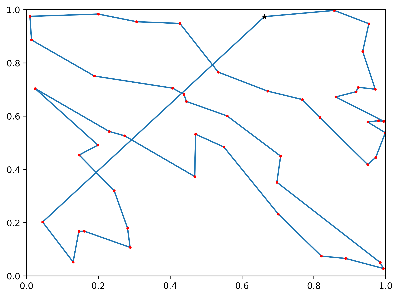
当城市数量为20时，网络在第37个训练轮次收敛，每一百个样本平均训练时间为14秒，图4.2为20个城市的动态旅行商问题的三个实例解：



（a） （b） （c）

图4.2 20个城市的解示例

当城市数量为50时，每一百个样本平均训练时间达到了47秒，训练时间较长，模型从第17个训练轮次开始有明显的学习效果，说明此时critic网络逐渐收敛，在第38个训练轮次达到最好结果，之后出现过拟合现象。最后得到表现最好的解路径长度较训练之前大幅减小，然而与最优解还有一定差距。图4.3为50个城市的动态旅行商问题训练38个轮次后的三个实例解：



（a） （b） （c）

图4.3 50个城市的解示例

从上述解的图示中可以看出，当n为10、20、50的时候模型都展现了显著的学习效果，n为10、20时路径基本符合最优解，从数据上也优于贪心策略的结果，n为50时路径有一定质量，但差于对比算法。当n者过大时，模型训练时间开始变长，效果也开始变差，这可能是网络结构或参数的原因，也有可能是actor-critic算法的局限性所致，因为同时训练两个网络的效率很低，模型的效果极大地依赖critic网络的收敛，而critic网络的收敛效果并不好。

4.4 本章小结

本章介绍了本文的交替编码解码模型解决动态旅行商问题的实验。在正方形单元里，取城市的数量分别为10，20，50来进行动态旅行商实验，说明了实验步骤及环境，然后给出了三种实验情况下求解得到的几个路径图例。选择贪心策略来作为对比算法进行实验结果分析，结果表明，当城市数量为10，20这些较小值时，模型求解得到的结果非常接近最优解，明显优于贪心策略的结果，当城市数量增大到50时，模型在求解速度和解的质量两方面都开始体现局限性，开始差于对比算法。

4 总结和展望

4.1 总结

在这篇文章中主要采用了指针网络的思想，将旅行商这一组合优化问题当作序列到序列的映射问题，通过编码和解码过程实现输入的状态序列到输出的动作序列的映射。不同于指针网络，由于动态旅行商问题输入的次序无关性，在编码器中移除了不必要的循环神经网络，而是简单的卷积嵌入层，另外利用注意力机制将解码向量加工得到了语义向量，再将这一语义向量当成新的解码向量，即用了两层注意力，其中第二层注意力作为指针。问题的动态性要求编码和解码交替进行，得到实时的信息。实验的结果表明，模型对于动态旅行商问题有一定程度的学习，对于小规模的动态旅行商问题的输出解非常接近最优解。本文证明了改进后的编码解码器加以两层注意力计算的模型结合好的强化学习训练方法能够胜任小规模的动态旅行商等这一类具有动态元素的组合优化问题。

4.2 未来的工作

由于动态旅行商问题相较于静态的有了更复杂的输入规模和变化形式，传统的序列到序列的模型可能难以胜任，未来的工作主要有：

（1）将模型改为图指针网络，图能够更好的处理这种平面对称的旅行商问题，在大规模问题时能够更有条理地梳理大量数据，另外可以尝试Transformer模型。

（2）actor-critic算法中两个网络相互依赖，难以收敛，计划改为选择DDPG和A3C算法来完成强化学习的训练。

（3）动态这一因素体现在城市位置和数量两个方面[22]，其实和指针网络一样，当前的模型已经支持将任意长度的输入序列转化为对应规模的输出序列，只是由于城市数量改变的数据集难以处理，也难以通过结果评价模型性能便没有尝试，这点将来有机会再实现。

（4）将模型尝试应用于其他组合优化问题，例如车辆路径规划问题，车间调度问题等。

参考文献

[1]Xianghu Meng,Jun Li,Xianzhong Dai. A Dynamic Colored Traveling Salesman Problem and Its Solution[C]//.第36届中国控制会议论文集（B）.[出版者不详],2017:1372-1377.

[2]陈斌,刘卫国.基于SAC模型的改进遗传算法求解TSP问题[J].计算机科学与探索,2021,15(09):1680-1693.

[3]李凯文,张涛,王锐,覃伟健,贺惠晖,黄鸿.基于深度强化学习的组合优化研究进展[J].自动化学报,2021,47(11):2521-2537.DOI:10.16383/j.aas.c200551.

[4]王扬,陈智斌,吴兆蕊,高远.强化学习求解组合最优化问题的研究综述[J].计算机科学与探索,2022,16(02):261-279.

[5]王若愚,陈勇全.基于强化学习的旅行商问题解构造方法[J].计算机工程,2020,46(11):293-300.DOI:10.19678/j.issn.1000-3428.0055910.

[6]中山大学. 基于元学习的深度强化学习求解多目标旅行商问题的方法:CN202110142988.5[P]. 2021-05-14.

[7]王原,陈名,邢立宁,吴亚辉,马武彬,赵宏.用于求解旅行商问题的深度智慧型蚁群优化算法[J].计算机研究与发展,2021,58(08):1586-1598.

[8]王轲,钟海旺,余南鹏,夏清.基于seq2seq和Attention机制的居民用户非侵入式负荷分解[J].中国电机工程学报,2019,39(01):75-83+322.DOI:10.13334/j.0258-8013.pcsee.181123.

[9] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[10] Vinyals O, Fortunato M, Jaitly N. Pointer networks[J]. Advances in neural information processing systems, 2015, 28.

[11] Bello I, Pham H, Le Q V, et al. Neural combinatorial optimization with reinforcement learning[J]. arXiv preprint arXiv:1611.09940, 2016.

[12] Nazari M, Oroojlooy A, Snyder L, et al. Reinforcement learning for solving the vehicle routing problem[J]. Advances in neural information processing systems, 2018, 31.

[13] Mele U J, Chou X, Gambardella L M, et al. Reinforcement learning and additional rewards for the traveling salesman problem[C]//2021 The 8th International Conference on Industrial Engineering and Applications (Europe). 2021: 198-204.

[14] Dazhi W, Shixin L. An agent-based evolutionary search for dynamic travelling salesman problem[C]//2010 WASE International Conference on Information Engineering. IEEE, 2010, 1: 111-114.

[15] Yan X S, Kang L S, Cai Z H, et al. An approach to dynamic traveling salesman problem[C]//Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826). IEEE, 2004, 4: 2418-2420.

[16] Miki S, Yamamoto D, Ebara H. Applying deep learning and reinforcement learning to traveling salesman problem[C]//2018 international conference on computing, electronics & communications engineering (ICCECE). IEEE, 2018: 65-70.

[17] Li K, Zhang T, Wang R. Deep reinforcement learning for multiobjective optimization[J]. IEEE transactions on cybernetics, 2020, 51(6): 3103-3114.

[18] Zhang R, Prokhorchuk A, Dauwels J. Deep reinforcement learning for traveling salesman problem with time windows and rejections[C]//2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020: 1-8.

[19] Woo S, Yeon J, Ji M, et al. Deep reinforcement learning with fully convolutional neural network to solve an earthwork scheduling problem[C]//2018 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, 2018: 4236-4242.

[20] Ottoni A L C, Nepomuceno E G, Oliveira M S, et al. Reinforcement learning for the traveling salesman problem with refueling[J]. Complex & Intelligent Systems, 2021: 1-15.

[21] Zhang Z, Liu H, Zhou M C, et al. Solving Dynamic Traveling Salesman Problems With Deep Reinforcement Learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021.

[22] Yan X S, Zhou A M, Kang L S, et al. Tsp problem based on dynamic environment[C]//Fifth World Congress on Intelligent Control and Automation (IEEE Cat. No. 04EX788). IEEE, 2004, 3: 2271-2274.

[23] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[24] Dey R, Salem F M. Gate-variants of gated recurrent unit (GRU) neural networks[C]//2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS). IEEE, 2017: 1597-1600.

[25] Brandão J. A deterministic iterated local search algorithm for the vehicle routing problem with backhauls[J]. Top, 2016, 24(2): 445-465.

[26] Shaw P. A new local search algorithm providing high quality solutions to vehicle routing problems[J]. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 1997, 46.

[27] Ma Q, Ge S, He D, et al. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning[J]. arXiv preprint arXiv:1911.04936, 2019.

[28] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems![J]. arXiv preprint arXiv:1803.08475, 2018.

致谢

时光飞逝，转眼间，四年忙碌且快乐的本科学习生活即将结束。临近毕业，论文完成之时，心中有无以言表的感激和眷恋。武汉大学是一个美丽温馨的大家庭，在武汉大学的四年对我人生有着意义非凡的影响，这里有我尊敬的老师们，有我朝夕相处的同学朋友们，在此我要向他们表示我心中最诚挚的感谢。

感谢王峰老师对本论文的指导，从选题到开题到实验到撰写论文都不遗余力地指导我，让我对学习研究充满热情。我的论文作品还有很多不足之处，但这次写论文的经历让我终身受益。这是我难得的一次独立研究，我感受到需要真正用心去对待，没有本科学习的基础就不可能独立开展研究，没有研究也就不会有这篇论文。这次论文经历会在我以后的学习生活中激励我。

感谢我的同学们，四年来我们一起上课、学习、考试、生活，在我们互相学习的过程中给予了我欢乐、信心和榜样。尤其感谢陈嘉诺同学，这四年来我们朝夕相处，从你身上我学到了很多，我们之间的互相帮助和欢乐也很多。

最后感谢我的父母、朋友在我求学过程中对我的无条件支持。你们学业上、生活上对我的帮助我铭记在心。

再次感谢所有人。