

分类号_____ 密级 _____

UDC _____

学 位 论 文

基于生物可信性的脉冲超启发算法的研究

作 者 姓 名：

指 导 教 师：

申请学位级别： 硕士

学科专业名称：

论文提交日期：

论文答辩日期：

学位授予日期：

答辩委员会主席：

评 阅 人：

东 北 大 学

2023 年 5 月

A Thesis in Computer Software and Theory

**Research on Spiking Hyper-heuristic Algorithm
Based on Biological Credibility**

**Northeastern University
May 2023**

独创性声明

本人声明，所呈交的学位论文是在导师的指导下完成的。论文中取得的研究成果除加以标注和致谢的地方外，不包含其他人已经发表或撰写过的研究成果，也不包括本人为获得其他学位而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：

日 期：

学位论文授权使用授权书

本学位论文作者和指导教师完全了解东北大学有关保留、使用学位论文的规定：即学校有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人同意东北大学可以将学位论文的全部或部分内容编入有关数据库进行检索、交流。

作者和导师同意网上交流的时间为作者获得学位后：

半年 ☐ 一年 ☐ 一年半 ☐ 两年 ☐

学位论文作者签名：

导师签名：

签字日期：

签字日期：

摘 要

组合优化是一类广泛存在于计算机科学、运筹学、数学以及工程等领域的优化问题。借助对其的研究,人们能够掌握一种有力的决策工具。然而,这些问题通常包含离散或组合特性,其解空间通常很大,求解效率往往难以令人接受。因此,如何高效地对组合优化问题进行求解成为一项长久被研究者们所关注的课题。

超启发算法相较于其他求解算法,具有与问题域解耦的特点。这一优良特性使得它具有成为组合优化问题的通用求解算法的可能。然而,其设计使得高层选择策略需要负责进行探索和利用的平衡、算子的决策和对低层算子的求解结果的评估等任务,因此对高层选择策略的决策能力要求较高,现有的高层策略往往难以满足这些需求。近年来,随着计算机领域的发展,脉冲神经网络作为第三代神经网络受到学界的广泛关注。其丰富的神经元动力学特性和低功耗的特点,使得其在求解组合优化问题时拥有得天独厚的优势,能够在节省计算资源的同时对问题进行高效求解。因此,本文提出了基于脉冲神经网络的脉冲超启发算法框架,并基于生物可信性对其进行改进。本文的主要贡献点如下:

(1) 考虑到脉冲神经网络的高能效低功耗特性和其携带的丰富时空信息,本文提出了基于脉冲神经网络的超启发算法框架。首先基于待求解的任务特性设计有针对性的低层求解算子,其直接对解进行更新。脉冲神经网络则是低层求解算子的高层决策结构,根据低层算子的求解情况选择适于当前求解进程的算子对当前解进行下一步更新。这一结构利用了脉冲神经网络的优良特性,能够较好解决当前超启发算法存在的高层选择策略决策能力不足的问题。

(2) 脉冲神经网络具有低功耗的特性,这一特性主要是由于网络中脉冲的发放是稀疏的。然而,这同样会导致网络中各层间脉冲发放的极度不平衡。本文从仿生学角度入手,由生物体内的神经系统中广泛存在的短时程可塑性机制推导出改进后的 LIF 脉冲神经元动力学机制。通过在脉冲神经元上引入短时程突触可塑性机制,能够很好地改善网络中存在的脉冲发放不平衡问题。

(3) 脉冲神经网络中丰富的时空信息能够更好地指导对低层算子的选择。然而,这也会增大脉冲神经网络的训练负担,其反向传播算法需要在时间和空间两个维度上来对网络进行权重更新。为了提升脉冲神经网络的学习效率,本文通过时域上的阈下记录信息和空域上的层内激活率信息,来对脉冲神经网络进行时空负反馈调节,从而加快了权重的更新速度,提升了训练效率。

本文在 TSP 求解任务和 EVRPC 求解任务上对基于生物可信性的脉冲超启发算法的求解能力进行验证，证明了上述模块的有效性。并与现有最新的超启发算法进行对比。通过实验证明，本文提出的方法能够求得质量更好的解，并在求解时间上具备优势，验证了本文方法的有效性和先进性。

关键词：脉冲神经网络；超启发算法；短时程突触可塑性；时空负反馈机制

Abstract

Combinatorial optimization is a class of optimization problems that widely exist in the fields of computer science, operations research, mathematics, and engineering. With the help of its study, people can master a powerful decision-making tool. However, these problems usually contain discrete or combinatorial characteristics, their solution space is usually large, and their solution efficiency is often unacceptable. Therefore, how to efficiently solve combinatorial optimization problems has become a topic that researchers have been paying attention to for a long time.

Compared with other solving algorithms, the hyper-heuristic algorithm has the characteristics of decoupling from the problem domain. This excellent property makes it possible to become a general solution algorithm for combinatorial optimization problems. However, its design makes the high-level selection strategy responsible for tasks such as balancing exploration and exploitation, operator decision-making, and evaluation of the solution results of low-level operators. Therefore, the decision-making ability of the high-level selection strategy is high. Policies often struggle to meet these needs. In recent years, with the development of the computer field, the spiking neural network, as the third-generation neural network, has attracted extensive attention from the academic circle. Its rich neuron dynamic characteristics and low power consumption make it have unique advantages in solving combinatorial optimization problems, and can efficiently solve problems while saving computing resources. Therefore, this thesis proposes a spiking hyper-heuristic algorithm framework based on spiking neural networks and improves it based on biological credibility. The main contributions of this thesis are as follows:

(1) Considering the high energy efficiency and low power consumption characteristics of the spiking neural network and the rich spatiotemporal information it carries, this thesis proposes a hyper-heuristic algorithm framework based on the spiking neural network. First, a targeted low-level solution operator is designed based on the characteristics of the task to be solved, which directly updates the solution. The pulse neural network is the high-level decision-making structure of the low-level solution operator. According to the solution of the low-level operator, an operator suitable for the current solution process is selected to update the current solution in the next step. This structure takes advantage of the excellent

characteristics of the spiking neural network, and can better solve the problem of insufficient decision-making ability of high-level selection strategies existing in current hyper-heuristic algorithms.

(2) The spiking neural network has the characteristic of low power consumption, which is mainly due to the fact that the spiking in the network is sparse. However, this also leads to an extreme imbalance in the bursting of the various layers in the network. In this thesis, starting from the perspective of bionics, the improved dynamic mechanism of LIF spiking neuron is deduced from the short-term plasticity mechanism widely existing in the nervous system in organisms. By introducing a short-term synaptic plasticity mechanism on spiking neurons, the problem of unbalanced spiking in the network can be well improved.

(3) The rich spatiotemporal information in the spiking neural network can better guide the selection of low-level operators. However, this will also increase the training burden of the spiking neural network, and its backpropagation algorithm needs to update the weight of the network in two dimensions of time and space. In order to improve the learning efficiency of the spiking neural network, this thesis adjusts the spatio-temporal negative feedback of the spiking neural network through the Subthreshold Recording information in the time domain and the Fire-Rate in Layer information in the space domain, thereby speeding up the update speed of the weight and improving the learning efficiency of the spiking neural network training efficiency.

In this thesis, the solving ability of the pulse hyperheuristic algorithm based on bio-credibility is verified on the TSP solving task and the EVRPC solving task, which proves the effectiveness of the above modules. And compared with the existing state-of-the-art hyperheuristic algorithms. The experiment proves that the method proposed in this thesis can obtain a better solution, and has an advantage in the solution time, which verifies the effectiveness and advancement of the method in this thesis.

Key words: Spiking neural network; Hyper-heuristic algorithm; Short-term synaptic plasticity; Spatio-temporal negative feedback mechanism

目 录

独创性声明	I
摘 要	II
Abstract	IV
第 1 章 绪 论	1
1.1 研究背景和意义	1
1.2 国内外研究现状	3
1.2.1 组合优化问题求解的研究现状	3
1.2.2 脉冲神经网络研究现状	6
1.3 问题提出	7
1.4 本文研究内容	8
1.5 本文组织结构	9
1.6 本章小结	10
第 2 章 预备知识	11
2.1 脉冲神经网络	11
2.1.1 脉冲神经元	11
2.1.2 脉冲编码	14
2.2 超启发算法	15
2.2.1 低层求解算子	15
2.2.2 高层选择策略	16
2.3 强化学习	17
2.3.1 基于价值的学习	17
2.3.2 基于策略的学习	18
2.4 本章小结	19
第 3 章 整体框架和优化模型	21
3.1 整体框架	21
3.2 优化模型	22
3.2.1 旅行商问题	22
3.2.2 带充电的电动车辆路径问题	23

3.3 本章小结	28
第 4 章 基于脉冲神经网络的超启发算法框架	29
4.1 研究动机	29
4.2 低层算子设计	30
4.2.1 破坏-修复算子的设计	30
4.2.2 节点变换算子的设计	32
4.2.3 子序列变换算子的设计	34
4.3 基于脉冲神经网络的高层选择策略	34
4.3.1 基于强化学习的决策思想	35
4.3.2 脉冲强化学习网络的设计	36
4.3.3 SRLN 的训练方法	40
4.4 本章小结	42
第 5 章 基于短时程突触可塑性机制的改进脉冲神经元	43
5.1 研究动机	43
5.2 短时程突触可塑性机制	44
5.2.1 短时程增强机制	45
5.2.2 短时程抑制机制	45
5.2.3 脉冲神经元的增强-抑制转换	46
5.3 STSP-LIF 神经元的设计	47
5.4 本章小结	49
第 6 章 基于时空信息的负反馈调节机制	51
6.1 研究动机	51
6.2 基于时域负反馈调节的阈下记录	52
6.3 基于空域负反馈调节的层内激活率	53
6.4 脉冲神经网络上的时空负反馈调节机制	54
6.5 本章小结	56
第 7 章 实验测试与结果分析	57
7.1 实验设置	57
7.1.1 实验环境	57
7.1.2 评价指标	57
7.2 TSP 任务上的实验	58

7.2.1 数据集	58
7.2.2 消融实验结果与分析	58
7.2.3 对比实验结果与分析	63
7.3 EVRPC 任务上的实验	65
7.3.1 数据集	65
7.3.2 消融实验结果与分析	65
7.3.3 对比实验结果与分析	69
7.4 本章小结	72
第 8 章 总结与展望	73
8.1 总结	73
8.2 未来工作	74
参考文献	75
致谢	85
攻读硕士学位期间的论文项目情况	86

第1章 绪 论

组合优化是一类广泛存在于计算机科学、运筹学、数学以及工程等领域的优化问题。这类问题的目标通常是在大量的可选项中选择一些最优化的组合，以满足特定的目标或性能要求。例如，旅行商问题、装箱问题、调度问题等都属于组合优化问题，这些问题是实际的生产生活中最为常见的一类问题。然而，这些问题通常包含离散或组合特性，其解空间通常很大，其求解效率往往难以令人接受。因此，如何高效地对组合优化问题进行求解成为一项长久被研究者们所关注的课题。超启发算法相较于其他求解算法，具有与问题域解耦的特点。这一优良特性使得它具有成为组合优化问题的通用求解算法的可能。通过发掘超启发算法的求解潜力，研究者们有望能够更加高效地对多类组合优化问题进行求解，这将在生产调度、流水线安排、物流运输、智能仓储等多项领域带来显著的积极影响，对社会的经济增长和社会运行效率的提升有着重大意义。

本章内容组织如下：首先 1.1 阐述了脉冲超启发算法的研究背景和意义；1.2 节分别介绍了组合优化求解领域和脉冲神经网络领域中的国内外研究现状；1.3 节以目前的研究进展为基础，提出尚未解决的问题；1.4 节将介绍本文的主要研究内容；1.5 节对本文各章的组织结构进行说明；最后 1.6 节对本章进行总结。

1.1 研究背景和意义

组合优化问题（Combination Optimization Problem, COP）是现实世界中最常见的问题之一，其涉及在有限集合中寻求最优解的问题^[1]，在众多工业生产领域，如制造业的生产线优化、资源分配和货物运输等领域均有广泛应用。通过对组合优化问题的研究，我们可以得到一种有力的工具，帮助我们在实践中优化决策，例如如何将有限的资源分配到最需要的地方，如何设计最有效的生产线，如何在时间限制下进行作业调度或路线规划等。通过组合优化理论，我们能够发现这些问题中隐藏的结构、规律和特征。使用这些知识，我们可以针对特定情况开发出有效的算法，从而在计算机和数学方面得到解决。除了常见的工业生产与管理决策领域，组合优化也在各种科学领域中得到广泛应用，如生物学、物理学、化学、社会学等等。它可以在如蛋白质结构预测、遗传编码优化、能源分布优化、社交网络分析等问题上为研究提供帮助。组合优化为我们提供了解决实际问题的权威理论，为生产、科研、社会管理等领域提供了有力的支持，使我们更好地实现了资源优化、效率提升、成本降低、决策科学化等目标。

虽然组合优化能够在众多领域发挥重要作用，但组合优化问题的求解一直是项困难的任务。组合优化问题被认为是 NP-hard (Non-deterministic Polynomial Hard) 问题的一类，它通常是不可能有多项式时间内得到精确解的。这种情况下，我们通常需要使用近似算法来得到次优解或局部最优解。然而，组合优化问题通常需要在非常大的解空间中搜索最优解，这个解空间的大小可能非常庞大，且大小可能随着问题规模的增加呈指数增长，导致计算复杂度非常高，对计算资源的要求也非常高。同时，由于组合优化问题中可能存在多个最优解，不确定最优解的个数会给问题的求解带来很大的难度。而且有些情况下，合理的最优解可能是非常难以判断的。此外，组合优化问题的解必须满足各种约束条件的限制，这些限制有时会使问题更难以解决。对于某些组合优化问题，即便采用近似算法，其计算成本也是非常高的，可以需要大量时间和空间的计算资源。同时，这些方法还需要考虑如何平衡计算的速度和精度之间的关系，以及在哪些情况下终止计算等问题。因此，求解组合优化问题是一项非常具有挑战性的任务，在实践中需要使用灵活的方法和技术进行求解。

近年来，随着计算机领域的发展，人工智能相关的研究不断突破，在多种任务上取得了理想的表现。随着人工智能相关研究的深入，作为其驱动的神经网络 (Neural Network, NN) 也不断发展。其第一个发展阶段是感知机。1958 年，神经学家 Rosenblatt^[2] 首先提出了一种可以模仿人类感知能力的机器，并将其称之为“感知机”。1960 年，其基于感知机的研究，实现了一台能够识别部分英文字母的神经网络计算机^[3]。神经网络发展第二个阶段是有多个网络层组成人工神经网络 (Artificial Neural Network, ANN)，其基于联结主义的思想进行构建，于 20 世纪 80 年代中期开始兴起^[4]，并在 2006 年以后以深度卷积网络 (Deep Convolutional Neural Networks, DCNN)^[5] 为代表，成为近十几年人工智能发展的主要推动者。然而，ANN 虽然在整体上模拟了生物体内的神经网络架构，但在神经元上与真实的神经元存在较大差异，缺少神经元内部丰富的动力学机制，因此在整体活动过程中不能够精准模拟生物大脑内神经元的真实运作机制。脉冲神经网络 (Spiking Neural Network, SNN) 随着类脑计算的兴起，被视为第三阶段的神经网络^[6]。由于其具有丰富的时空领域的动力学特性、多样化的编码机制、低功耗的优势，众多研究者们对它投以关注^[7]。其丰富的神经元动力学特性和低功耗的特点，使得其在求解组合优化问题时拥有得天独厚的优势，能够在节省计算资源的同时对问题进行高效求解。因此，将脉冲神经网络应用于组合优化问题的求解，构造高效的求解框架，对生产规划和决策设计有长远的理论意义与实际应用价值。

综上所述，组合优化问题的求解在理论研究和实际应用上都有一定的研究价值。本

文以脉冲神经网络为研究手段，构建脉冲超启发算法框架，对求解组合优化问题这一挑战性任务提供求解方案。

1.2 国内外研究现状

脉冲神经网络虽然具有许多令人赞叹的特性，但由于这一领域较为新颖，其所能解决的问题范围也十分宽广，因此尚未有研究者将其应用于组合优化问题的求解中。因此，本节将对组合优化问题求解的研究现状和脉冲神经网络的研究现状分别展开介绍。

1.2.1 组合优化问题求解的研究现状

优化问题分为两类问题：连续问题和组合问题。连续优化问题由连续的决策变量组成，可以产生无限多的有效解。组合优化问题，也称为离散优化问题，由离散决策变量及其元素定义，组合优化问题的解决方案由这些元素的排列表示。由于解决方案是元素的组合（或排列），因此组合优化问题具有有限数量的解决方案。组合或排列之间的编码规则的定义将取决于问题的需要及其解决方案表示。

连续优化可以使用导数和梯度来描述优化函数的变化趋势^[8]。但是，组合优化不能借助梯度来对目标函数进行求解。因此，组合问题通常是需要解决的更具挑战性的问题。它们通常是 NP-hard 问题^[9]，并且没有能够在多项式时间内解决它们的确定性方法^[10]。在实际问题中，解空间往往是广阔的，而最优解的搜索方向通常是未知的。此外，组合优化方法的性能需要良好的策略来平衡探索和开发，以加速搜索并找到最佳解决方案。这些特性将组合优化问题求解变成了一项具有挑战性的任务^{[11][12]}。

求解组合优化问题的优化算法包括精确算法和近似算法。根据其所用技术的不同，可以将应用于组合优化问题的算法分为三个发展阶段。第一阶段是 1959 年到 1970 年，主要使用简单的启发式算法，如局部启发式算法和贪心算法；第二阶段为 1970 到 1980 年，该阶段主要应用精确算法；第三阶段是 1980 年以后，以元启发式算法为代表的各类新型求解算法开始应用于组合优化问题的求解。

根据文献[13]，求解组合优化问题的精确算法可分为三类：直接树搜索（如分支定界算法^[14]）、动态规划^[15]和整数线性规划（如列生成算法^[16]）。在 20 世纪 70 年代，使用精确算法实现了许多突破。Christofides 等人^[17]通过设计分支定界法、节省法和 3-最佳游览法研究了组合优化问题中的有容车辆路径规划问题（Capacitated Vehicle Routing Problem, CVRP）和动态车辆路径规划问题（Dynamic Vehicle Routing Problem, DVRP）。Christofides 等人^[18]使用树搜索算法来求解组合优化问题。尽管这些精确的算

法很容易理解,且能够通过求解得到组合优化问题的精确解,但它们的计算开销在大规模问题中令人难以接受^{[19][20]}。

传统的启发式算法可以分为构建式算法和两阶段式算法。构建式算法以最小化成本为目标,不断生成更加逼近目标的可行解,如储蓄算法^[21]。两阶段式算法主要针对以旅行商为代表的欧几里得空间上的组合优化问题。其首先对节点进行聚类,然后构造多条可行路线以满足约束。该算法通过改变路线之间或路线内的节点位置来提高解决方案的质量,如文献^[22]。构建解决方案的邻域是启发式算法的重要组成部分。然而,传统的启发式算法会在没有焦点的情况下执行整个搜索。因此,其只以有限的方式探索解空间。这种低效的探索在大规模问题的求解中表现有所不足。

元启发式通常被描述为独立于问题的算法^[23],可以合并多种特定组合优化问题的定义和与它们相关的先验知识^{[24][25]}。因此,其可以以灵活的方式有效地解决几类优化问题,因为它们与任何特定问题没有“强”联系^[26]。元启发式可以被视为更高级别的启发式^[27],由主启发式和从属启发式组成^[28]。主启发式驱动和协调一组从属启发式,通过不断地迭代循环,来高效探索解空间^[29],每个从属启发式在元启发式的搜索策略中扮演不同的角色^{[30][31]}。因此,主启发式在更高的抽象层次上安排各个从属启发式协同工作,以协调的方式为给定问题搜索最优解。

一种基于群体的元启发式算法是萤火虫算法 (Firefly Algorithm, FA)。FA 算法的灵感来自于模拟萤火虫的社交闪光行为。萤火虫可以通过发光传递社交信息,它们的光强变化区分了每只萤火虫的身份。FA 算法由于其固有的细分能力,有能力将萤火虫分为几个小组,这就是 FA 算法可以很好地处理多峰和非线性问题的主要原因^[32]。Trachanatzi 等人^[33]在车辆路径规划问题 (Vehicle Routing Problem, VRP) 上设计了一种改进的 FA 算法,以找到具有最小碳排放因子的最大奖励收集。修改后的 FA 的性能优于差分进化算法、蝙蝠算法和 Gurobi 优化器。为了求解 CVRP 问题,Goel 和 Maini^[34]使用混合算法 HAFA,将 FA 算法与蚁群算法进行结合。其中 FA 算法作为探索工具,蚁群算法中的信息素摇动方法用于避免局部最优。对于时间相关的 VRP,Alinaghian 等人^[35]提出了一种基于高斯 FA 算法的改进 FA 算法 (Improved Firefly Algorithm, IFA),考虑了燃料消耗的约束。作者使用了两种启发式方法进行随机初始化,从而加速了各种解决方案。Altabeeb 等人^[36]设计了一种增强型 FA,用于解决与两阶段局部搜索策略相结合的 CVRP 问题。此外,作者还实施了交叉和变异算子,以更好地平衡探索与利用。Khalifehzadeh 等人^[37]讨论了具有需求不确定因素的多阶段产品分销网络 (Multi-stage Product Distribution Network, MPDN) 模型。作者实现了一种基于启发式的 FA 算法,

其中客户需求用模糊逻辑处理。基于启发式的选择性 FA 算法 (Selective Firefly Algorithm, SFA) 在创建最佳解决方案的改进百分比指数方面优于经典 FA 算法。Tokhmehchi 等人^[38]使用成本最小化方法解决了与逆向物流有关的闭环供应链网络问题。作者将 FA 算法与遗传算法混合, 为优化提供了更好的解决方案。

和声搜索 (Harmony Search, HS) 也是一种常见的元启发算法, 由 Geem 等人提出^[39]。其概念是由音乐家在和声即兴创作中寻找最佳和声结果而衍生出来的。和声的即兴创作方法通常包括以下步骤: 首先, 从乐师的记忆中选择合适的音调; 其次, 音乐的音高与合理的音高相关, 可以稍微调整并形成新的或随机的音高。Z. Guo 等人^[40]讨论了 MTO 供应链中的集成输出和运输调度问题。为了解决这个问题, 开发了一种基于 HS 算法的模因优化模型, 借助启发式程序将集成输出问题转化为订单分配问题, 同时还提出了一种新的即兴创作方法来提高 HS 算法的效率。与遗传算法和其他自然启发算法相比, 数值测试证明了所提出模型的有效性。S. Chen 等人^[41]为带有时间窗的动态车辆路径问题 (Dynamic Vehicle Routing Problem with Time Windows, DVRPTW) 构建了一个改进的 HS 算法, 其可以实时监控客户请求。该算法结合变邻域下降 (Variable Neighborhood Descent, VND) 算法, 修改了 HS 算法的局部搜索策略。变邻域 HS (Harmony Search Variable Neighborhood Descent, HSVND) 算法在求解 DVRPTW 实例时, 在探索和利用的平衡上表现出良好的效果。

元启发式算法虽然与具体问题解耦, 能够对多个种类的组合优化问题进行求解。但由于其主启发式结构简单、从属启发式结构复杂, 使其在进行迭代求解时, 解的质量更多依赖于从属启发式的求解效果。当组合优化问题结构复杂且规模极大时, 元启发式算法求解效果将跌落到与传统启发式类似的境地。为了改善元启发式存在的问题, 超启发算法应运而生。

与元启发式算法不同, 超启发算法寻求整合简单且计算成本低的方法来解决问题^[42]。元启发式算法与超启发算法之间的一个基本区别是前者探索问题的解空间, 而后者关注求解器空间^{[43][44]}。因此, 超启发算法并不能直接解决问题^[45]。相反, 在求解过程的每个步骤中, 它都会选择一种启发式方法来处理此类步骤^[46]。这种思想使得超启发算法的结构可分解为低层求解算子和高层选择策略, 从而令超启发算法框架形成一种近似于模块化的结构^[47]。根据不同的待求解问题, 只需要更换对应的低层求解算子, 便可有效规避“没有免费午餐 (No Free Lunch, NFL)”定律, 从而使超启发算法能够在多类组合优化问题上表现良好^{[48][49]}。

装箱问题 (Bin Packing Problem, BPP) 是一类典型的优化问题, 其目的是最小化

使用的箱子数量来完成对一组物品的打包任务^{[50][51]}。Burke 等人^[52]通过使用遗传编程实现探索了 BPP 问题上的超启发算法。他们的结果证明，确实有可能产生比人工启发式算法具有更高通用性的打包启发式算法。Sim 等人^[53]探讨了如何使用免疫系统为一维 BPP 产生超启发算法。他们的结果与通过最先进的方法获得的结果相当，也证明了这种超启发是处理动态数据集的可靠策略。Lopez-Camacho 等人^[54]提出了一个统一的超启发框架，用于在一维和二维 BPP 问题上创建和测试超启发算法。

作业车间调度问题（Job Shop Scheduling Problem, JSSP）是制造业中一类至关重要的组合优化问题，其能够根据客户需求来定制高效生产计划。这一问题被证明是 NP 完全问题^[55]，因此其传统求解技术难以对其进行处理。Chaurasia 等人^[56]实现了一种进化算法，将引导突变作为高层选择策略来解决 JSSP 问题。这种方法在他们 80% 的实验中超越了其他常见的方法。GarzaSantisteban 等人^[57]提出了一个由模拟退火支持的超启发模型，该模型可以处理不同大小的 JSSP 实例。

1.2.2 脉冲神经网络研究现状

与 ANN 类似，SNN 也受到生物学中观察到的神经网络的启发^[58]。然而，与 ANN 不同的是，SNN 采用的单元使用更接近其生物对应物的神经元单元来处理信息^[59]。生物神经元使用动作电位（也称为脉冲）处理和传输信息，这是大脑表现出高能效、低功耗的基础。脉冲神经元和生物神经元之间的这些相似性也意味着有可能开发出与人脑具有相似功率要求的 SNN^[60]。在脉冲神经网络中处理信息的基本单元称为脉冲神经元。受生物神经元的启发，脉冲神经元利用由称为脉冲的二元事件组成的时间信号作为它们的输入和输出。脉冲神经元利用脉冲的精确时间来编码信息。脉冲神经元接收到的每个脉冲都会改变其状态，这被称为膜电位。当一个神经元的膜电位达到某个阈值时，脉冲神经元会产生一个脉冲，该脉冲通过突触传递给其他脉冲神经元。

神经网络中的学习涉及神经元连接性的修改。与可以通过随机梯度下降和反向传播成功训练的 ANN 不同，SNN 仍然没有可靠的训练方法。SNN 的原生训练方法可以分为基于替代函数的监督学习，以脉冲时间依赖可塑性（Spike-timing Dependent Plasticity, STDP）为代表的在突触处使用局部学习规则的无监督学习，以及基于间接训练思想的 ANN-to-SNN 转换^[61]。突触可塑性是突触活动的特定模式导致突触强度变化的生物学过程。突触可塑性是在理论分析的基础上首次提出作为学习和记忆机制的^[62]。虽然突触处的局部学习规则在生物学上更合理，但学习性能通常低于监督学习和 ANN-to-SNN 转换。

类似于 ANN 的反向传播算法, **SpikeProp**^[63]通过线性处理神经元接收的输入和由此产生的脉冲发放在时间上的关系, 将目标脉冲与实际脉冲在时间上的差距定义为误差函数, 从而避开了在脉冲激活时产生的不连续性问题。然而, 所有基于事件的方法^[64], 包括 **SpikeProp**, 具有仅在脉冲激发时间附近定义的导数项, 却忽略了脉冲信号的时间效应^{[65][66]}。文献[67]提出了一种称为 **SuperSpike** 的改进的 **SpikeProp** 方法, 该方法利用膜电位的导数而不是脉冲生成附近的导数, 从而允许在没有脉冲发生的情况下训练模型。在训练时, **SuperSpike** 使用输出和所需脉冲序列之间的 **van Rossum** 距离^[68]作为损失函数, 而 **SpikeProp** 使用平方和误差。

STDP 是一种无监督的 **Hebbian** 学习机制, 它根据突触前和突触后脉冲产生的时间顺序调整突触权重^{[69][70]}。当突触前脉冲先于突触后脉冲到达时, 突触权重增加。如果突触上脉冲的到达时间颠倒, 突触权重就会减少。尽管在各种神经元系统中发现了 **STDP** 机制, 但并非所有系统都遵循 **STDP** 规则。例如, 小脑中平行纤维和浦肯野细胞之间的突触遵循反 **Hebbian** 时间顺序^[71]。与 **STDP** 相比, 反 **STDP** 机制对突触前输入和突触后脉冲的相对时间表现出相反的依赖性。对于反 **STDP**, 在突触前脉冲之后发生的突触后脉冲会导致突触权重的减少, 而突触前脉冲之前发生的突触后脉冲会导致突触权重的增加。

大多数 ANN 到 SNN 的转换方法都侧重于将 **ReLU** 转换为脉冲神经元。文献[61]提出了一种忽略偏差和最大池化的 ANN 到 SNN 的转换方法。在随后的工作中, 文献[72]提出了基于数据的归一化来提高深度 SNN 的性能。文献[73]提出了批量归一化和脉冲最大池化的转换方法。文献[74]将转换方法扩展到 **VGG** 和残差架构。几种 ANN 到 SNN 转换设计的一个核心假设是, 现有 ANN 的大量计算成本是由于网络中连接节点之间实值活动的持续传输, 以及随后的矩阵乘法或卷积^[75]。因此, 实现 ANN 到 SNN 的转换可以实现相同的信息传输和功能, 但会降低信号传输和计算的 cost。二进制值脉冲既可以通过将实值信号转换为二进制信号来减少每次传输的位数, 又可以通过脉冲信号在时间步上稀疏的特性来使 SNN 中的脉冲神经元在整个仿真过程中的大部分时间步上是不活跃的。这些 ANN 到 SNN 的转换方法将训练好的参数从 ANN 导入 SNN, 从而能够使 ANN 到 SNN 的转换方法在深度 SNN 中取得与原始 ANN (如 **VGG** 和 **ResNet**) 相近的结果。

1.3 问题提出

通过对 1.2 节中现有研究方法的研究和分析, 可以发现, 目前组合优化问题的求解

方法虽然已有很多,但由于其 NP-hard 的特性,仍然存在着巨大的改进空间。超启发算法作为一种泛用性强的求解框架,有望在组合优化问题的求解中表现优良。然而,其设计使得高层选择策略需要负责进行探索和利用的平衡、算子的决策和对低层算子的求解结果的评估等任务,因此对高层选择策略的决策能力要求较高,现有高层策略往往难以满足这些需求。本文将脉冲神经网络作为超启发算法的高层决策结构,下面详细地对目前方法中存在的问题进行探讨:

(1) 现有高层选择策略决策能力不足。目前常用的超启发算法的高层选择策略仍然多基于启发式的思想,如模拟退火等。这种选择策略虽然能够实现对低层算子的选择,但面对结构复杂的组合优化问题时,由于解空间过于庞大,且需要进行多轮迭代,启发式的高层选择策略在每轮迭代后很难选取到真正适合当前解结构的下一低层算子。因此,求解效率会随问题规模和复杂程度的提升而急剧衰弱。

(2) 若将脉冲神经网络作为超启发算法的高层选择策略,虽然其决策能力更高,但现有的脉冲神经网络中存在脉冲发放率不平衡问题。在脉冲神经网络的前几层,脉冲序列较为密集,因此该位置的神经元激发难度过低,容易造成过激发现象。而在脉冲神经网络的后几层,脉冲序列过于稀疏,该位置的神经元的脉冲发放率衰减严重。如何对脉冲神经网络前后的脉冲激发率进行较好的平衡是一个值得研究的问题。

(3) 现有的脉冲神经网络在训练和推理时会在多个时间步上进行。这一举措虽然符合生物中真实神经元的活动机制,也能够为脉冲神经网络带来丰富的时序信息。但这一机制也使得脉冲神经元在进行反向传播的训练时需要将突触权重在多个时间步上更新,造成训练较慢的问题。

1.4 本文研究内容

针对 1.3 节中提到的当前组合优化求解的任务中存在的问题与挑战,本文对超启发算法框架进行了深入研究,提出了脉冲超启发算法 (Spiking Hyper-heuristic, SHH),其高层选择策略为脉冲神经网络,且该网络通过引入生物可信性机制来改善传统脉冲神经网络中存在的不足。本文提出的方法包括基于强化学习思想的脉冲神经网络,能够作为高层选择策略进行高效的低层算子选择;提出基于短时程突触可塑性机制的改进脉冲神经元,能够对脉冲神经网络中前后层的脉冲发放率进行平衡;提出基于时空负反馈调节机制的权重更新方法,能够加快脉冲网络的收敛速度。下面就本文的主要内容进一步阐述:

(1) 针对现有高层选择策略决策能力不足的问题,本文提出基于强化思想的脉

冲神经网络作为超启发算法的高层决策结构。这一高层决策结构能够利用脉冲神经网络的高效低耗和时间信息丰富的特性,实现对低层算子的高效选择,从而能够快速且高质量地求解组合优化问题。

(2) 针对脉冲神经网络中存在的激活不平衡问题,本文借鉴了生物神经系统内存在的短时程突触可塑性机制,借助其对神经元的短时程增强-抑制作用,对脉冲神经网络中原有的神经元进行动力学机制上的改进。改进后的脉冲神经元能够根据输入脉冲的疏密程度动态调整激活阈值,从而实现不同层间的脉冲神经元在激活率上的平衡。

(3) 针对脉冲神经网络中存在的训练速度较慢的问题,本文借鉴了生物体内广泛存在的多种混合负反馈调节机制,提出了时空负反馈调节,能够从时间和空间两个维度上对脉冲神经网络内的神经元进行调控。通过基于时域的阈下记录和基于空域的层内激活率,该机制能够恰当地评价脉冲神经元的激活难易程度和实际活跃程度,从而适时调控其权重更新情况,来实现网络的快速训练。

1.5 本文组织结构

本文内容总共为8章,各章的组织结构如下:

第1章是绪论部分,首先介绍了脉冲超启发算法的研究背景和意义,然后对国内外研究现状进行分析,对现有研究进行深度分析的基础上针对目前该领域存在的问题,提出本文的研究内容,并对文本的主要研究内容进行介绍。

第2章是对预备知识的介绍,包括脉冲神经网络的相关内容(脉冲神经元模型和脉冲编码方法)。同时也介绍了超启发算法和强化学习的相关内容。

第3章是整体架构和优化模型,对本文提出的各个部分的内容进行了梳理,并且介绍本文实验所针对的组合优化求解任务,并对任务进行抽象,建立与之对应的优化模型。

第4章提出了基于脉冲神经网络的超启发算法框架,从研究动机、低层算子设计、高层选择策略三个方面介绍本文设计的脉冲超启发算法框架。

第5章提出了基于短时程突触可塑性机制的改良脉冲神经元模块,介绍短时突触可塑性的动力学机制,并将其融合到脉冲神经元中。

第6章提出了基于时空信息的负反馈调节机制,通过时域上的域下记录和空域上的层内激活率,提出时空负反馈调节因子来加速训练。

第7章为实验测试与结果分析,通过本文方法在TSP求解任务和EVRPC求解任务上的表现,验证了基于生物可信性的脉冲超启发算法的有效性。

第8章对本文主要研究内容进行总结,并提出未来能够进一步改进的方向。

1.6 本章小结

本章首先对脉冲超启发算法的研究背景和意义进行了简单介绍,其次从组合优化求解和脉冲神经网络两个方面对国内外研究现状进行叙述。然后针对目前该领域的研究现状提出问题,并介绍基于这些问题本文的研究内容。最后简要说明本文的整体组织结构。

第2章 预备知识

在本文对基于生物可信性的脉冲超启发算法的研究过程中，涉及到很多的理论知识和相关概念。本文的脉冲神经网络采用 LIF 神经元作为基本运算单元，采用由高层选择结构和低层求解算子组成的超启发框架结构，采用强化学习的思想作为超启发算法高层决策结构的选择和决策机制。本章将从脉冲神经网络、超启发算法和强化学习这三个方面对这些预备知识进行介绍，以便能更好的理解本文内容。

2.1 脉冲神经网络

作为最新一代神经网络的脉冲神经网络，其具有高生物可信性、丰富的时空领域动力学特性、高能效比、低功耗等特性。这些特性主要依赖于其与 ANN 相比，更贴近真实生物体神经元的脉冲神经元的动力学特性，以及与这些神经元相适配的脉冲编码方式。

2.1.1 脉冲神经元

目前，脉冲神经网络已包含多种脉冲神经元模型，它们拥有不同的生物学准确性和计算可行性之间的权衡。根据研究任务不同，研究者们可以选择自己所需要的脉冲神经元模型^[76]。

(1) Hodgkin-Huxley 模型

Hodgkin 和 Huxley 对乌贼的巨型轴突进行了实验，得出结论： K^+ 通道和 Na^+ 通道这两种离子通道参与了动作电位的产生^[77]。虽然离子通道通透性的变化实际上是由于蛋白质的结构变化，但可以通过类比门的开闭来描述。其动力学机制如公式(2.1)和公式(2.2)所述。

$$C_m \frac{dv_m(t)}{dt} = I_{ion}(t) + I_{syn}(t) \quad (2.1)$$

$$I_{ion}(t) = G_K n^4 (v_m - E_K) + G_{Na} m^3 h (v_m - E_{Na}) + G_L (v_m - E_L) \quad (2.2)$$

其中 C_m 为膜电容， v_m 为膜电位， I_{syn} 为突触输入电流， G_K 代表 K^+ 离子电导， E_K 代表 K^+ 离子反转电位， G_{Na} 表 Na^+ 离子电导， E_{Na} 表示 Na^+ 离子的反转电位， G_L 表示漏电导， E_L 表示漏电反转电位，现在认为是 Cl^- 离子的反转电位。 n 、 m 和 h 是介于 0 和 1 之间的无量纲量，分别与 K^+ 离子通道激活、 Na^+ 离子通道激活和 Na^+ 离子通道失活相关。

它们的微分方程可表述为如公式 (2.3) 所示的形式。

$$\frac{dg}{dt} = \alpha_g(v_m)(1-g) - \beta_g(v_m)g \quad (2.3)$$

$$\alpha_m(v_m) = \frac{0.1(25-v_m)}{\exp\left(\frac{(25-v_m)}{10}\right) - 1} \quad (2.4)$$

$$\beta_m(v_m) = 4 \exp\left(\frac{-v_m}{18}\right) \quad (2.5)$$

$$\alpha_h(v_m) = 0.07 \exp\left(\frac{-v_m}{20}\right) \quad (2.6)$$

$$\beta_h(v_m) = \frac{1}{\exp\left(\frac{(30-v_m)}{10}\right) + 1} \quad (2.7)$$

$$\alpha_n(v_m) = \frac{0.01(10-v_m)}{\exp\left(\frac{(10-v_m)}{10}\right) - 1} \quad (2.8)$$

$$\beta_n(v_m) = 0.125 \exp\left(\frac{-v_m}{80}\right) \quad (2.9)$$

其中 g 表示门控变量 n 、 m 和 h ， $\alpha_g(v_m)$ 表示门从非激活状态转换到激活状态的转换率，而 $\beta_g(v_m)$ 表示门从激活状态转换到非激活状态的转换率。对于每个门的转换率 $\alpha_g(v_m)$ 和 $\beta_g(v_m)$ 的详细动力学模型，在公式 (2.4) 到公式 (2.9) 中给出。

通过求解这些方程，Hodgkin-Huxley 模型可以模拟脉冲生成过程中的脉冲神经元上的膜电位行为。尽管 Hodgkin-Huxley 模型在生物学上是准确的^{[78][79]}，但它需要大量计算资源，在大规模模拟中是不可行的。

(2) 累积-发放模型

计算效率最高的模型，也是生物学上最不合理的模型，是累积-发放 (Integrate and Fire, IF) 模型。该模型的动力学机制最为简单，如公式 (2.10) 所示。

$$C_m \frac{dV}{dt} = I(t) \quad (2.10)$$

dV/dt 是神经元电位随时间的导数， I 是来自突触的总电流。当脉冲神经元膜上的点位达到某个阈值时，就会触发一个脉冲^[80]。

(3) 泄露-累积-发放模型

泄露-累积-发放 (Leaky Integrate-and-Fire, LIF) 模型通过在 IF 模型中引入“泄漏”

项来反映当细胞中未达到某种平衡时发生的离子扩散。由于其简单性和低计算成本, LIF 模型及其变体是脉冲神经元模型广泛使用的实例之一。该模型的动力学机制可表示为如公式 (2.11) 和公式 (2.12) 所示的形式。

$$C_m \frac{dv_m}{dt} = -G_L (v_m - E_L) + I_{syn}(t) \quad (2.11)$$

$$\begin{cases} v_m \leftarrow v_{peak}, & v_m \geq v_\theta \\ v_m \leftarrow v_{reset}, & v_m < v_\theta \end{cases} \quad (2.12)$$

其中 v_θ 是阈值电压, v_{peak} 是动作电位, v_{reset} 是复位膜电位。当电压达到阈值 v_θ (为简单起见, 通常取值为 1) 时, 神经元会发射脉冲信号, 然后在将电压重置为复位电位 v_{reset} 。

当突触输入电流恒定 ($I_{syn}(t) = I$) 且 $v_{reset} = 0$ 时, 可以得到膜电位的动力学机制如公式 (2.13) 所示:

$$v_m(t) = R_m I \left(1 - \exp\left(-\frac{t}{\tau_m}\right) \right) \quad (2.13)$$

其中 R_m 是膜电阻, $\tau_m = R_m / C_m$ 是膜时间常数。由于当膜电位达到阈值时神经元会发放脉冲, 因此可以通过设置 $v_m(t) = v_\theta$ 找到第一个脉冲的发放时间 $t^{(1)}$, 如公式 (2.14) 所示:

$$t^{(1)} = \tau_m \ln \frac{R_m I}{R_m I - v_\theta} \quad (2.14)$$

因此, 可以得到稳态发放率如公式 (2.15) 所示:

$$f = \left(\tau_{ref} + \tau_m \ln \frac{R_m I}{R_m I - v_\theta} \right)^{-1} \quad (2.15)$$

从理论上讲, 可以利用公式 (2.15) 作为静态非线性来训练深度脉冲神经网络, 并在脉冲神经元中对网络进行合理的近似^[81]。

(4) Izhikevich 模型

Izhikevich 提出了一个模型, 该模型结合了 Hodgkin-Huxley 模型动力学的生物学合理性和 LIF 神经元的计算效率^[82]。Izhikevich 模型^[83]呈现为常微分方程表示的二维自治系统, 常可以表示为公式 (2.16) 和公式 (2.17) 所示的形式:

$$C_m \frac{dv_m}{dt} = k(v_m - E_L)(v_m - v_t) - u + I_{syn}(t) \quad (2.16)$$

$$\frac{du(t)}{dt} = a(b(v_m - E_m) - u) \quad (2.17)$$

其中 u 代表 K^+ 离子电流的激活和 Na^+ 离子电流的失活, v_t 是瞬时阈值电位。通过对参数 a 和 b 的调整, Izhikevich 模型可以展现出所有已知类型的脉冲神经元的放电模式。

2.1.2 脉冲编码

由于 SNN 利用脉冲和脉冲序列来传递信息, 因此将真实数据编码为脉冲是将信息输入到 SNN 前的重要步骤。目前的编码方案主要有两种, 分别是速率编码和时间编码。此外, 一些传感器, 例如动态视觉传感器 (Dynamic Vision Sensor, DVS), 可以生成原始脉冲序列。

(1) 速率编码

速率编码方案基于随时间推移的脉冲的平均数量。如图 2.1 所示, 该方案设置一个事件窗口, 通过统计在时间窗口上生成的脉冲数量作为发放速率, 从而对信息进行编码。在这一机制中, 发放速率与信号的输入强度正相关。

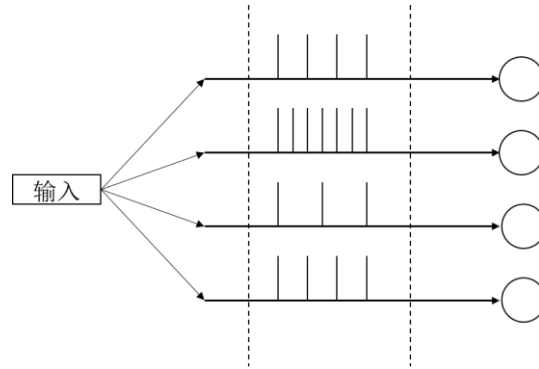


图 2.1 速率编码

Fig. 2.1 Rate Encoding

(2) 时间编码

使用速率编码传输信息需要较长的时间窗口, 并且编码后的脉冲序列较为稠密。另一种常见的编码方法是时间编码^[84]。如图 2.2 所示, 在这种编码中, 借助脉冲发放的时间传输信息。在这一机制中, 强度较高的输入被转换为较早的发放的脉冲。时间编码更加迅速, 且编码后的脉冲序列更为稀疏的, 这种稀疏性可以加快 SNN 推理的速度。

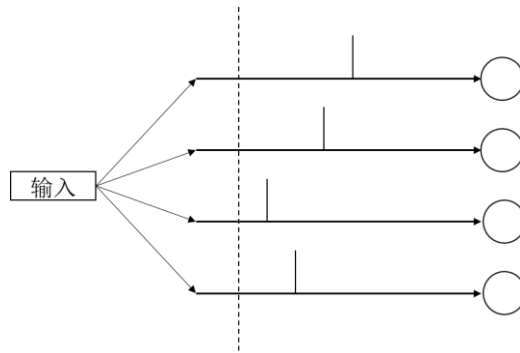


图 2.2 时间编码

Fig. 2.2 Time Encoding

2.2 超启发算法

超启发算法的出现是为了尝试通过在一组问题上表现出色来为组合优化问题提供更广义的解决方案，而不是推导出只对该领域的几个问题实例产生良好结果的技术。超启发通过在求解器空间而不是解空间中工作来实现这一点。因此，超启发选择或生成用于解决手头问题的低级启发式。超启发使用不同的技术，如基于案例的推理、局部搜索和遗传编程来选择或生成低级启发式。其整体结构如图 2.3 所示。

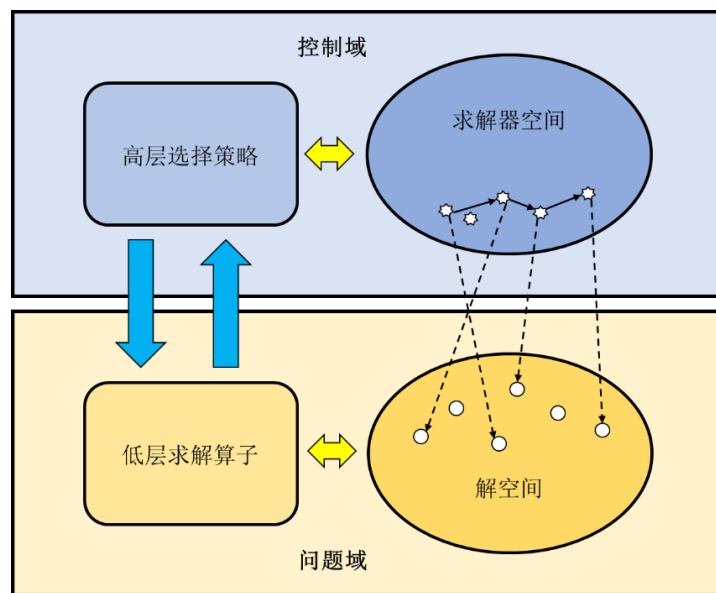


图 2.3 超启发算法框架

Fig. 2.3 Hyper-heuristic Algorithm Framework

2.2.1 低层求解算子

位于超启发算法框架下层的是低层算子，这些优化算子用来直接生成解。低层算子是超启发算法构建高层策略的基础，负责在生成解空间内进行搜索和扩展，并对每个生成的解进行评价和记录。根据不同的问题和算法特点，低层算子的设计和实现方式也会有所不同。以下是超启发算法中常见的低层算子：

(1) 变异算子：变异算子是一种最基本的低层算子，通常用来从当前解中随机选择一个部分，并对其进行变异以生成新解。变异操作可以针对解中的某些元素进行替换、删除、插入、交换等操作，用以产生具有多样性的新解。

(2) 重组算子：重组算子用于生成新解，其过程是将两个或更多个当前的解合并，产生新的解。重组算子的设计和实现可以通过交叉、混合、合并、分割等方式进行。重

组算子可以加强算子在解空间中的探索性和多样性，并帮助算法跳出局部最优点。

(3) 邻域搜索算子：邻域搜索算子是通过当前解附近的邻域进行搜索来产生新解。邻域搜索算子常用于欧几里得空间上的组合优化问题的求解。

(4) 子群启发式算子：子群启发函数是一种基于群体智能的低层算子，其中问题解被表示为个体的一种组合，通过考虑解的子群中特定的启发式，以产生新的解。子群启发式算子常用于多目标优化问题的求解中。

低层算子的设计和实现直接影响超启发算法的效率和成功率，需要针对具体的问题和算法进行选择和优化。

2.2.2 高层选择策略

超启发框架的高层选择策略会依据一定的评价指标来从算子池中选择当下情况最适合的低层求解算子。之后，被选择的低层算子负责直接对优化问题进行求解，并进行解的更新。更新后的解会被低层算子呈送到高层选择策略，高层策略会依据一定的接受机制来判别是否接受这一更新。整个决策过程主要由选择机制和接受机制共同驱动。

选择机制由当前状态对低层求解算子进行选择。虽然算子池往往不大，但选择机制仍需要面对探索-利用的平衡问题。从利用的角度来评价，选择机制应当尽可能多地选择对解改善最大的低层算子；而从探索的角度来看，选择机制需要对当前非最优的算子进行尝试，探索其能否为之后的求解带来更大的好处。此外，利用何种指标来判断当前算子的好坏也是其亟待解决的一个问题。目前较为常用的评价指标为低层算子对解的改进效果，然而这一评价指标过于单一，且在不同领域的问题上，解的改进率往往有着尺度上的巨大差异。仅使用这一指标作为选择机制的评判标准，将会对超启发算法的泛用性带来较大损害。

接受机制决定了超启发算法在解空间上的探索方式，并对超启发算法的性能有重大影响^[85]。尽管只接受更好解决方案的策略在直觉上能指导算法沿着最快的方向收敛，但它通常会导致局部最优。因此，接受恶化的解决方案是必要的，但要以可控的方式进行。目前常用的接受机制为 ε -greedy，其思想是以一个较小的 ε 概率对恶化的解进行接受，而在多数情况下对其拒绝。

目前多数的超启发算法需要人工设定选择与接受机制，这使得求解效率往往依赖于设计的精妙与否。且人工设计的选择与接受机制往往会对某些组合优化领域有所偏重，弱化了超启发算法本身所具有的泛用性。因此，基于机器学习相关理论的自组织选择接受机制是未来具有广阔发展前景的方向。

2.3 强化学习

强化学习是一种机器学习技术，其用于训练智能体（agent）从环境中学习，以制定决策策略并最大化其期望的奖励。在强化学习中，智能体通过与环境进行交互，观察环境状态和奖励信号，并相应地采取行动来学习。强化学习强调的是学习的连续过程，而不是单个的训练任务。

强化学习的基本组件包括智能体、环境、状态、行动、奖励信号等，组件间的关系如图 2.4 所示。其智能体可以根据当前的环境状态来做出决策和行动，并接收环境发出的反馈信号和奖励信号。状态表示当前环境的观察结果，行动表示智能体的响应，奖励信号则表示行动的实际效果和目标。强化学习的目标是让智能体制定出最优的策略，从而最大化期望的奖励值。

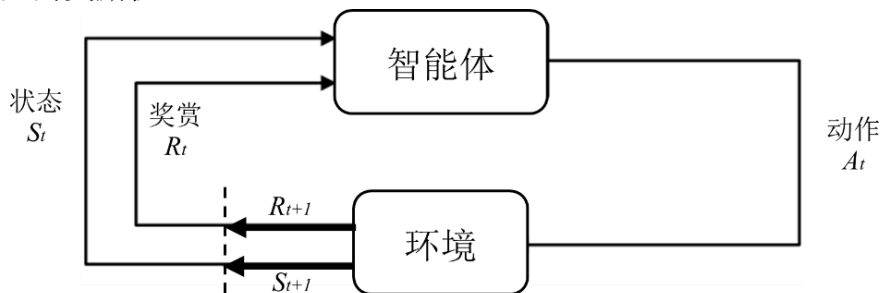


图 2.4 强化学习组件间关系

Fig. 2.4 Component Relationships for Reinforcement Learning

根据学习方案的不同，强化学习算法可分为基于价值的学习和基于策略的学习。

2.3.1 基于价值的学习

基于价值的强化学习通过最优动作价值函数（Optimal Action-Value Function）来学习如何进行最优决策。其先对每个动作的价值（常被称为 Q 值）进行评价，通过对能够在每个状态下造成最大 Q 值的动作的选择，来完成对最优策略的选取。如图 2.5 所示，传统的 Q 学习通过当前状态和已选择的动作得到对应的 Q 值，而深度 Q 网络（Deep Q-network, DQN）则基于当前状态得到每个动作的离线 Q 值，其大大降低了前向传播时的计算成本。

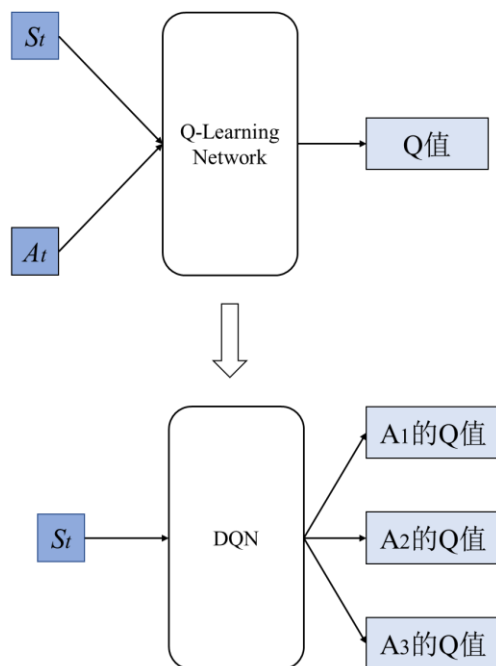


图 2.5 Q-Learning 与 DQN
Fig. 2.5 Q-Learning vs. DQN

2.3.2 基于策略的学习

DQN 虽然在各类离散空间的任务中表现良好，但由于其学习基准是当前状态下每个动作所对应的 Q 值，因此在连续空间中就难以适用了。若将连续空间进行离散化以适应 DQN 的学习过程，则会造成动作空间爆炸式地增加，依然会造成学习效率的降低。同时，这种离散化也会带来信息的损失，对最终的任务表现造成损害。

面对连续空间上的学习任务，基于策略的学习更加适用。相较于价值学习中对确定动作的学习，在策略学习中，学习的对象是由一组状态-动作序列组成的轨迹 π 。而学习目标可以表示为如公式 (2.18) 所示的形式：

$$U(\theta) = \sum_{\pi} P(\pi; \theta) R(\pi) \quad (2.18)$$

其中， $P(\pi; \theta)$ 为轨迹 π 出现的概率， $R(\pi)$ 为轨迹的回报。对公式 (2.18) 求导，可以得到公式 (2.19)：

$$\begin{aligned}
 \nabla_{\theta} U(\pi) &= \nabla_{\theta} \sum_{\pi} P(\pi; \theta) R(\pi) \\
 &= \sum_{\pi} \nabla_{\theta} P(\pi; \theta) R(\pi) \\
 &= \sum_{\pi} \frac{P(\pi; \theta)}{P(\pi; \theta)} \nabla_{\theta} P(\pi; \theta) R(\pi) \\
 &= \sum_{\pi} P(\pi; \theta) \frac{\nabla_{\theta} P(\pi; \theta) R(\pi)}{P(\pi; \theta)} \\
 &= \sum_{\pi} P(\pi; \theta) \nabla_{\theta} \log P(\pi; \theta) R(\pi)
 \end{aligned} \tag{2.19}$$

通过公式（2.19）得到的策略梯度，可以直接对策略进行优化，沿着梯度最陡的方向进行优化，从而逐步迭代得到最优策略。

相较于价值学习，策略学习由于基于策略梯度进行，因此具有更好的收敛性。然而其同样具有收敛至局部最优的不足。同时，价值学习能够学习到确定策略，而策略学习学习到的是随机策略。前者更符合离散空间中任务的要求，而后者更能模拟连续空间中的随机扰动。

2.4 本章小结

本章主要介绍了本文方法中涉及到的基础知识，2.1 节主要介绍了脉冲神经网络中的基本概念，包括集中常见的脉冲神经元模型和脉冲编码方案，LIF 神经元后续作为本文脉冲神经网络的基础单元。2.2 节介绍了超启发算法框架，主要介绍了低层求解算子和高层选择策略，本文的脉冲超启发算法框架结构也遵从这一组织结构。2.3 节介绍了强化学习的相关概念，本文的脉冲超启发算法框架的高层决策结构在对低层求解算子进行选择时，借鉴了相关的思想。

第3章 整体框架和优化模型

第2章介绍了本文涉及的相关理论知识,本章将对本文关于组合优化求解中的相关概念进行定义,介绍本文所提出的脉冲超启发算法的框架,并对本文研究所面向的组合优化任务以及对它们的优化模型的建立进行介绍。

3.1 整体框架

本文提出的脉冲超启发算法的整体框架如图3.1所示,该算法的框架主要包括优化模型、低层求解算子、脉冲决策结构、短时程突触可塑性神经元、负反馈调节模块等五个部分。

第一部分是优化模型。优化模型是从待求解的实际问题中抽象出优化目标和约束条件,是整个超启发算法框架中与问题域结合最紧密的部分,也为整个超启发算法确定解空间的大小和优化方向。

第二部分是低层求解算子。低层算子通过对当前解执行邻域变换操作来对解的结构进行扰动和变换,从而对解进行更新。低层求解算子是整个超启发算法框架中直接面对问题域进行求解的部分,与问题结构适合的低层算子能够加速解的更新,便于更快寻找到最优解。本文根据所面对的欧几里得空间上的组合优化求解任务,设计出与其结构相适应的丰富的低层算子,为后续求解带来便利。

第三部分是脉冲决策结构。脉冲决策结构是超启发算法框架的高层策略选择结构,能够对当前解的好坏进行评估,并根据当前解的结构选择恰当低层算子,来进行下一步的解的更新。本文构建的脉冲决策结构是基于强化学习思想构造的脉冲神经网络,利用脉冲神经元丰富的动力学特性提升对低层算子的决策能力,提高求解效率。

第四部分是短时程突触可塑性神经元。脉冲神经网络虽然由于脉冲信息传递时的稀疏性而具有低能耗的特性,但也造成网络的浅层和深层间存在脉冲发放的不均衡。本文利用生物体中存在的短时程突触可塑性机制,将脉冲神经网络中的传统脉冲神经元替换为短时程突触可塑性神经元,解决了脉冲神经网络中存在的脉冲发放不平衡问题,提升了求解效果。

第五部分是时空负反馈调节模块,利用生物体内常见的负反馈调节机制,从时域和空域两个维度对脉冲神经网络的权重更新方式进行调整,引入时空负反馈调节因子将脉冲神经元激活的难易程度和实际激活表现的活跃程度信息融入到脉冲神经网络的权重

更新中，提升了网络的学习速度。

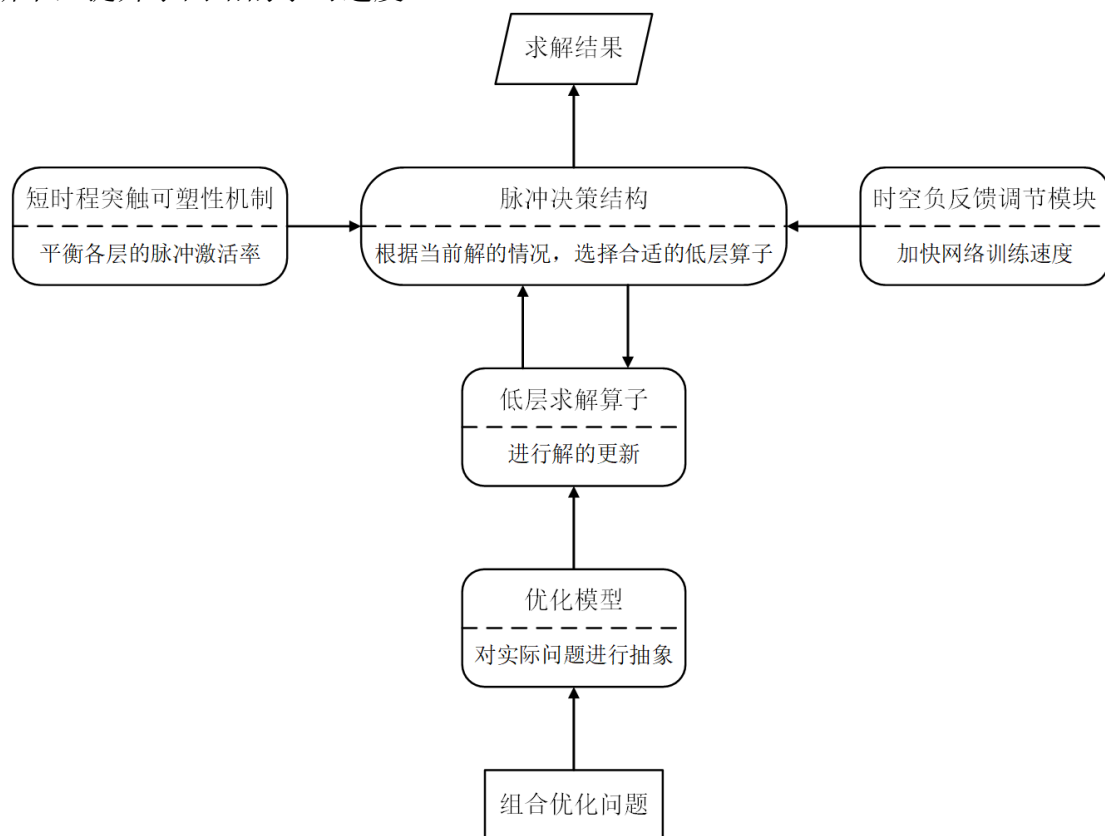


图 3.1 系统整体框架图

Fig. 3.1 Structure of system

3.2 优化模型

本节首先介绍本文所要面对的组合优化求解任务：旅行商问题的求解和带充电的电动车辆路径问题的求解。这两个问题都是在欧几里得空间上的组合优化问题，因此具有类似的问题结构，便于后续的低层算子的设计。接下来，本文对这两个实际问题进行抽象，建立了指导求解的优化模型。

3.2.1 旅行商问题

3.2.1.1 问题定义

旅行商问题（Travelling Salesman Problem, TSP）是一个典型的组合优化问题，常被用作离散优化中许多算法性能分析的基准问题^[85-87]。此外，TSP 对于车辆路径^[88]、生产计划^[89]、客户服务时间^[90]、设计硬件设备和计算机网络^[86]等实际应用至关重要。

TSP 表现为设计访问路线中的最短行程：它寻找访问每个城市的顺序，从第一个城市开始，访问每个城市一次，然后返回起始城市。其可以形式化定义为在一个带权的无

向完全图中寻找一个最小花费的哈密顿回路的问题。

问题定义（旅行商问题）给定一个带权无向图 $G=(V,E)$ ，其中 V 表示其所要遍历的节点集合 $V=\{v_0, v_1, v_2, \dots, v_{n-1}\}$ ， E 表示无向图上所有边的集合。对于 $0 \leq i < n, 0 \leq j < n$ ，若有 $i \neq j$ ，则有节点 v_i 与 v_j 的连线所构成的无向边 $(v_i, v_j) \in E$ ，其权值记为边长 d_{ij} 。从给定的任意节点 v_i 出发，要求对点集 V 中所有其他节点访问且仅访问一次后返回出发节点 v_i ，寻找一条最短访问路线，使该路线上所有边的边长之和最小。

3.2.1.2 优化模型的建立

通过以上讨论和定义，可以建立 TSP 任务上的优化模型如 (3.1) 所示：

$$\min D(L) = \sum_{i \neq j} d_{ij} s_{ij} \quad (3.1)$$

目标函数确定优化目标为最小化总路径 L 的路径长度。其中， $D(L)$ 是总路径长度函数， s_{ij} 是对子路径的存在性判断。对于节点 i 到节点 j 的子路径 $(i \rightarrow j)$ ，定义其存在性如公式 (3.2) 所示：

$$s_{ij} = \begin{cases} 1, & (i \rightarrow j) \in L \\ 0, & (i \rightarrow j) \notin L \end{cases} \quad (3.2)$$

公式 (3.3) 对各个节点的访问情况进行了约束，限制所有节点被访问且只能被访问一次。

$$s.t. \begin{cases} \sum_{i \in V, i \neq j} x_{ij} = 1 \\ \sum_{j \in V, j \neq i} x_{ji} = 1 \end{cases} \quad (3.3)$$

其中， x_{ij} 是总路径 L 中节点 i 到节点 j 的有向边。通过公式 (3.3)，可以约束每个节点必为一条且仅为一条有向边的起点，并必为一条且仅为一条有向边的终点。这意味着在路径中，每个节点均被访问且仅被访问一次。

公式 (3.4) 对访问序列进行限制，其中 $|L|$ 表示总路径 L 中的元素个数。这一约束限制了访问回路中不包含任何子回路。

$$s.t. \sum_{i, j \in V} x_{ij} = |L|, L \in E \quad (3.4)$$

通过公式 (3.1) 到公式 (3.4) 的目标函数和约束条件，可以建立 TSP 求解任务上的优化模型。

3.2.2 带充电的电动车辆路径问题

3.2.2.1 问题定义

车辆路径规划问题（Vehicle Routing Problem, VRP）也是一个典型的组合优化问题，其在供应链网络、物流运输等领域发挥着极大作用。根据所面对的现实任务的不同，VRP 已发展出不同的变体。

由于电子商务市场的扩大，劳动力成本的增加是物流中心的一个严重问题。传统仓库采用的是人工拣选系统，拣选人员走入仓库的通道拣选物品。拣选成本大约占仓库总运营费用的 55%，其中 60%是由于在仓库中行走造成的^[40]。为了减少订单拣选的成本和时间，机器人移动履行系统（Robotic Mobile Fulfillment System, RMFS）已经被引入，它采用了自动引导车（Automated Guided Vehicle, AGV）。RMFS 是一种“货到人”的拣选系统，通过它，工人不需要在仓库中行走。相反，AGV 在货架下方行驶，提升并将它们运送到工作站，在那里进行人工拣选。如何规划 AGV 的拣选路径，使其更加高效地完成拣选和运送任务，成为提高 RMFS 效率的关键。

为了对 AGV 的拣选路径进行规划，本文提出一个 VRP 任务的变体：带充电的电动车辆路径问题（Electric Vehicle Routing Problem with Charge, EVRPC）。由于电池供电的 AGV 在“货到人”系统中的广泛使用，本问题是一个考虑到充电需求的 AGV 调度问题。在这个问题中，每个节点代表仓库中的一个位置，包括货架、AGV 停靠点和充电站。AGV 从停靠点出发，依次搬运一组货架，完成搬运任务后，在最大的时间限制内返回停靠点。所有目标货架必须由 AGV 提供服务。当 AGV 行驶时，电池的电量会随着行驶距离的增加而成比例地减少，如果电量水平不足以维持 AGV 到下一个任务货架位置，AGV 必须对其电池进行充电。其目标是最小化所有 AGV 的总行走距离。

3.2.2.2 优化模型的建立

通过以上的讨论，可以建立 EVRPC 任务的优化模型如（3.5）所示：

$$\min \left\{ \max_{k \in O} v_k \right\} \quad (3.5)$$

其中， O 表示 AGV 的停靠点集合， v_k 表示 AGV 从停靠点 k 出发，完成搬运任务后返回该停靠点所用的时间。考虑到 AGV 在货架节点执行搬运操作、在充电站进行充电操作、在节点间行驶等均需要耗费时间，因此目标函数确定优化目标为最小化 AGV 的最大完成时间。由于所有 AGV 运行速度相同且设定它们均为匀速运行，因此这一目标可等效为最小化 AGV 的最长运行路径。

$$s.t. \begin{cases} \sum_{j \in V, j \neq i} x_{ij} = 1 & \forall i \in S \\ \sum_{j \in V, j \neq i} x_{ji} = 1 & \forall i \in S \end{cases} \quad (3.6)$$

公式（3.6）对货架的被服务状态进行了约束。其中， V 表示所有节点集合， S 表示

货架节点集合。 x_{ij} 为路径判断标记, 若服务路径中存在节点 i 到节点 j 的边, 则 $x_{ij} = 1$, 否则 $x_{ij} = 0$ 。这条约束限制了每个货架节点必出现在服务路径中, 且仅被服务一次。

$$s.t. \begin{cases} \sum_{k \in O} y_{ik} = 1 & \forall i \in S \\ \sum_{i \in S} y_{ik} \geq 1 & \forall k \in O \end{cases} \quad (3.7)$$

公式 (3.7) 对 AGV 访问货架的情况进行了约束。其中, y_{ij} 为访问判断标记, 若一条服务回路中, 存在节点 i 到节点 j 的路径, 则 $y_{ij} = 1$, 否则 $y_{ij} = 0$ 。这条约束限制了每个货架都只被一个 AGV 访问, 而每个 AGV 都执行访问货架的任务, 至少访问一个货架。

$$s.t. \begin{cases} \sum_{i \in S} x_{ki} = 1 & \forall k \in O \\ \sum_{i \in F} x_{ik} = 1 & \forall k \in O \end{cases} \quad (3.8)$$

公式 (3.8) 对 AGV 的出入站进行了约束。这条约束限制了每个停靠点每次只出发 1 台 AGV, 且每次只有 1 台 AGV 返回。这一约束避免了 AGV 在进出停靠点时的出入站冲突。

$$s.t. \sum_{i \in V, i \neq j} x_{ij} - \sum_{i \in V, i \neq j} x_{ji} = 0 \quad \forall j \in F \quad (3.9)$$

公式 (3.9) 为流量平衡约束。其中, F 表示货架节点和充电站节点的集合。这条约束限制了每个货架或充电站的流入和流出是相等的, 避免了 AGV 在非停靠点的不正常停靠。

$$s.t. \begin{cases} x_{ki} \leq y_{ki} & \forall i \in F, k \in O \\ x_{ik} \leq y_{ik} & \forall i \in F, k \in O \end{cases} \quad (3.10)$$

公式 (3.10) 对 AGV 非正常的出入站进行约束。这条约束限制了 AGV 从停靠点出发后访问的第一个货架或充电站节点必须在其服务路径中, AGV 返回停靠点之前最后访问的货架或充电站节点也必须在其服务路径中, 从而避免了在出入站时不正常地同时访问多个货架或充电站节点。

$$s.t. \begin{cases} x_{ij} + y_{ik} - y_{jk} \leq 1 & \forall i \in F, j \in F, i \neq j, k \in O \\ x_{ij} + y_{jk} - y_{ik} \leq 1 & \forall i \in F, j \in F, i \neq j, k \in O \end{cases} \quad (3.11)$$

公式 (3.11) 对连续节点的访问情况做出约束, 这个约束限制了若服务路径上的两个货架或充电站节点需要被连续访问, 则这两个节点应当被同一个 AGV 服务。这一约束避免了需要连续访问的节点被错误安排到不同的 AGV 上, 从而造成错误的异步访问和无谓的冗余路径。

$$s.t. \ x_{ij} = 0 \quad \forall i \in R, j \in R \quad (3.12)$$

公式 (3.12) 对充电站的连续访问做出限制。其中, R 表示充电站节点的集合。这一约束限制 AGV 不应在访问一个充电站节点之后继续访问另外一个充电站节点, 其应当在第一个访问到的充电站节点完成所有的充电操作, 电量充满。这一限制避免了连续访问充电站带来的冗长路径。

$$s.t. \ \sum_{j \in V, j \neq i} x_{ij} \leq 1 \quad \forall i \in R \quad (3.13)$$

公式 (3.13) 对充电站的使用做出限制, 其约束在同一次访问中, 不应当有多条流向同一充电站的边。这一限制避免了不同 AGV 在使用同一个充电站时有可能产生的冲突访问。

$$s.t. \ \begin{cases} \sum_{j \in V, j \neq i} x_{ij} = z_i & \forall i \in R \\ \sum_{k \in O} y_{ik} \leq z_i & \forall i \in R \\ a_i \leq Mz_i & \forall i \in R \end{cases} \quad (3.14)$$

公式 (3.14) 能够对充电站的使用情况做出调整。其中, z_i 是充电站 i 的使用标记, 其标记为 1 时代表可使用, 标记为 0 时代表不可使用。 a_i 是 AGV 到达节点 i 的时间, M 是一个大整数系数。通过这一限制, 可以通过 z_i 对充电站的使用情况做出调整, 将其设定为 0 时可以将该充电站从 AGV 的访问路径中移除, 并约定 AGV 在其上的到达时间为 0。

$$s.t. \ Ma_k \leq x_{kj} \quad \forall k \in O, j \in S \quad (3.15)$$

公式 (3.15) 对 AGV 的出发时间进行限制。通过该限制, 所有 AGV 从停靠点出发时, 其在该停靠点上的到达时间 (即为出发时间) 必定为 0。

$$s.t. \ \begin{cases} a_j \geq a_i + h_i + t_{ij} - M(1 - x_{ij}) & \forall i \in V, j \in F, j \neq i \\ a_j \leq a_i + h_i + t_{ij} + M(1 - x_{ij}) & \forall i \in V, j \in F, j \neq i \end{cases} \quad (3.16)$$

公式 (3.16) 对 AGV 到达其访问节点的时间进行约束。其中, h_i 是 AGV 在节点 i 上的服务时间。当节点 i 为货架节点时, 服务时间 h_i 是搬运用时; 当节点 i 是充电站节点时, 服务时间 h_i 是充电用时; 当节点 i 是停靠点时, 服务时间 h_i 是停靠用时。 t_{ij} 是 AGV 从节点 i 运行到节点 j 时所用时间。设定所有 AGV 运行速度相同, 且它们在运行时都保持匀速, 因此 t_{ij} 仅与节点 i 与节点 j 之间的距离有关, 该值与边长成正比。通过这一限制, 对 AGV 在每个节点上到达时间进行计算。

$$s.t. \begin{cases} v_k \geq a_i + h_i + t_{ik} - M(1 - x_{ik}) & \forall i \in F, k \in O \\ v_k \leq a_i + h_i + t_{ik} + M(1 - x_{ik}) & \forall i \in F, k \in O \end{cases} \quad (3.17)$$

公式(3.17)对AGV的任务完成时间进行约束。由于AGV对货架访问后返回原停靠点视为一次任务的完成,故将其在停靠点上的到达时间作为任务完成的时间。其计算逻辑与其他节点上的到达时间的计算相同。

$$v_k \leq T_{max} \quad \forall k \in O \quad (3.18)$$

公式(3.18)对AGV的任务最大任务完成时间进行约束,这一约束能够限制AGV完成任务的时间上限,对超过这一上限的AGV视为无法完成该任务,从而避免无限等待等极端情况。

$$s.t. \begin{cases} q_j \leq q_i - [\mu(t_{ij} + h_i)x_{ij}] + M(1 - x_{ij}) & \forall i \in S, j \in S, j \neq i \\ q_j \geq q_i - [\mu(t_{ij} + h_i)x_{ij}] - M(1 - x_{ij}) & \forall i \in S, j \in S, j \neq i \end{cases} \quad (3.19)$$

公式(3.19)对AGV在货架节点间移动时的电量变化情况进行约束。其中, q_i 表示AGV在节点*i*时的剩余电量, μ 为AGV的耗电速率,约定AGV在进行搬运操作和运行时的耗电速率相同。通过这一约束,能够计算AGV在货架节点间移动并执行搬运操作时的电量变化情况,从而得到其到达某一货架节点时的剩余电量。

$$s.t. \begin{cases} q_j \leq Q - [\mu(t_{ij}x_{ij})] + M(1 - x_{ij}) & \forall i \in P, j \in S \\ q_j \geq Q - [\mu(t_{ij}x_{ij})] - M(1 - x_{ij}) & \forall i \in P, j \in S \end{cases} \quad (3.20)$$

公式(3.20)对AGV在充满电之后的电量变化情况进行约束。其中, P 表示停靠点和充电站节点的集合, Q 表示AGV的电池最大电量。当AGV从停靠点或充电站出发前往下一节点时,其到达下一节点时的剩余电量只与其在路上的运行时间有关。通过这一约束,能够计算AGV在充满电之后运行到下一节点时的电量情况。

$$s.t. q_i \geq \sum_{j \in V, j \neq i} \mu(t_{ij} + h_i)x_{ij} \quad \forall i \in S \quad (3.21)$$

公式(3.21)对节点剩余电量的计划进行约束。当其到达货架节点*i*时,会预计到达下个节点*j*后所要消耗的电量,该电量不会超过其在节点*i*时的剩余电量 q_i 。通过这一约束,可以使AGV不至于在某一货架节点或运行过程中将电量耗尽,从而使得其电量总能支持其完成一个完整的访问任务。

$$s.t. \begin{cases} x_{ij} \in \{0,1\} & \forall i \in V, j \in V \\ y_{ik} \in \{0,1\} & \forall i \in F, k \in O \\ z_i \in \{0,1\} & \forall i \in R \\ q_i \geq 0 & \forall i \in S \\ a_i \geq 0 & \forall i \in V \\ v_k \geq 0 & \forall k \in O \end{cases} \quad (3.22)$$

公式 (3.22) 对各变量的取值情况进行约束, 使它们的取值合理且符合问题的实际情况。

通过公式 (3.5) 到公式 (3.22) 的目标函数和约束条件, 可以建立 EVRPC 求解任务上的优化模型。

3.3 本章小结

本章 3.1 节介绍了本文的整体框架结构, 3.2 节对本文所面对的组合优化问题求解任务和其上的优化模型的建立进行了介绍。

第4章 基于脉冲神经网络的超启发算法框架

第 3 章中介绍了本文所面对的组合优化求解任务的优化模型，本章将基于优化模型和问题结构提出基于脉冲神经网络的超启发算法框架，将从研究动机开始，依次介绍整个超启发算法的低层求解算子的设计和高层选择策略的确定，并给出高层的脉冲决策层的训练方法。

4.1 研究动机

组合优化的求解是一项价值高且有挑战性的任务，如何高效求解是众多研究者们一直追寻的目标。对组合优化问题求取精确解的方法，因为其本身 NP-hard 的特性而难以在大规模问题上进行求解，因此，以各种启发式算法为代表的求取近似解的算法在对组合优化问题的求解上更具备研究前景。

超启发算法由于将关注点放在低层算子所处的求解器空间，而非通常算法所关注的解空间，因此具有与具体问题弱相关的特性，能够做到与问题域解耦。这使得超启发算法泛化性更强，更有望成为求解组合优化问题的通用求解框架。

然而，由于要根据解在不同求解阶段多变的结构特性来选择合适的低层算子进行解的更新，超启发算法的高层选择策略需要具备较强的决策能力。目前常见的超启发算法仍将启发式算法（如模拟退火算法、大洪水算法等）作为高层选择结构，其决策能力较弱，在面对复杂问题或大规模问题时，并不能够高效地对低层算子进行选择，制约了超启发算法的求解效果。

脉冲神经网络作为第三代神经网络，由于其脉冲神经元的类脑特性，具有类似生物脑的低功耗特性，能够在计算资源受制约的情况下依旧具备较高的计算能力，这一特性使得它适合面对 NP-hard 问题。同时，脉冲神经元由于脉冲发放的时序性，具有丰富的时域信息，这一信息能够辅助超启发算法在求解的不同阶段适时选择不同的低层求解算子，因此能够很好地与超启发算法框架相适配。

综上所述，通过合适的设计，将脉冲神经网络作为超启发算法的高层选择策略，能够有效地提升高层选择策略的决策能力，从而提高超启发算法的求解效率。基于脉冲神经网络的超启发算法框架有望成为组合优化问题的通用求解框架。

4.2 低层算子设计

低层算子位于超启发算法框架的底层，其直接面向问题域。低层算子不仅是控制域与问题域的连接结构，更是解的生成与更新的直接运算单元。因此，恰当地设计低层算子能够显著提高求解效率。

设计低层算子时应遵循三个原则：首先，低层算子不应过于简单，也不宜过于复杂。过于简单的低层算子对问题进行初步求解时，不能有效对当前解做出更新，从而使得高层选择部分需要较多的迭代，耗时较为严重；而过于复杂的低层算子则会使最终形成的整个超启发框架过于复杂笨重，在每次解的更新时消耗过多时间，不利于发挥超启发框架中高层选择部分的优势。其次，低层算子设计时应当尽量丰富。只有多样化的丰富的低层算子，才能在对当前解进行更新时充分对潜在的解空间进行探索，从而有利于更快探索到最优解。最后，设计底层算子时应当充分考虑待求解问题本身的特性。与高层选择策略的解耦思想不同，低层算子在求解时极度依赖于问题域，只有与待求解问题相适应的算子结构才能有效地对当前解进行更新，有助于更加高效地寻找到最优解。

通过第 3 章的定义与分析可以看出，本文所面向的两个问题具有相似性。不论是 TSP 问题还是 EVRPC 问题，对服务节点（客户，货架）和路径子序列进行扰动时均能有效地进行解的更新。当对服务节点（客户，货架）进行扰动时，对解的邻域变换影响范围较小；当对路径子序列进行扰动时，解的邻域变换影响范围较大。

基于上述设计三原则和对待求解问题的分析，本文设计了破坏-修复、节点交换和子序列交换这三类低层算子，并依照邻域变换的影响范围，对所有算子进行升序排列。

4.2.1 破坏-修复算子的设计

破坏-修复算子是超启发算法中一类常见的扰动算子，其思想是通过对当前解的可行性的“破坏”和对被破坏的解的“修复”，来完成对当前解的小范围扰动，从而实现解的更新。在破坏阶段，破坏算子能够按照一定规则移除当前解中的部分服务节点，被删除后的解从可行解变为不可行解；而在修复阶段，修复算子从被移除的节点集中按照一定规则将节点插入路径中，从而将被破坏的解重新变为可行解。为了保证更新后的解能够维持可行性，破坏算子与修复算子必须成对出现。

本文设计如下 5 个破坏算子：

LLH-a：随机移除算子

该算子随机选择 m 个节点，从当前解中移除。

LLH-b: 最差距离移除算子

该算子计算每个节点与其前后节点的距离之和作为该节点的距离成本，选择所有节点中距离成本最高的 m 个节点，将它们从当前解中移除。

LLH-c: 最差节点移除算子

该算子计算每个节点在移除前与移除后该节点所在路径的距离变化，将距离之差作为该节点的破坏力。选择所有节点中破坏力最高的 m 个节点，将它们从当前解中移除。

LLH-d: 最长弧移除算子

该算子计算每对相邻货架间的距离作为弧距，从弧距最高的两个节点开始，按弧距降序依次移除节点，直至移除 m 个节点。

LLH-e: 最相关移除算子

定义节点间差异性如公式 (4.1) 所示：

$$R_{ij} = \varphi_1 d_{ij} + \varphi_2 l_{ij} \quad (4.1)$$

其中 d_{ij} 表示节点间距离， φ_1 表示距离权重。 l_{ij} 反映两个节点是否位于同一路径，若它们在同一路径上，则 $l_{ij} = 0$ ，否则 $l_{ij} = 1$ 。 φ_2 表示路径权重。 R_{ij} 越低，则两节点间相关性越强。该算子首先随机选择一个节点 i 进行移除，并计算其他节点与节点 i 的差异性，从中挑选最相关的 $m-1$ 个节点进行移除。

同时，本文设计了如下 2 个修复算子：

LLH-x: 贪婪插入算子

该算子计算移除集中节点插入到每个可能位置后总路径的变化长度，将其中最小变化长度作为该节点的插入成本，其对应的位置为该节点的最优插入位置。从移除集中挑选插入成本最低的节点，并将其插入最优插入位置中。

LLH-y: 后悔插入算子

该算子计算移除集中节点插入到每个可能位置后总路径的变化长度，将其中最小变化长度作为该节点的最优插入成本，其对应的位置为该节点的最优插入位置；第二小的变化长度为次优插入成本。每个节点的最优插入成本与次优插入成本的插值为该节点的后悔成本。从移除集中挑选后悔成本最大的节点，并将其插入最优插入位置中。

将破坏算子与修复算子进行组合，并筛选出其中表现更好的算子，就是本文所设计的 8 个破坏-修复算子。根据邻域变换的影响范围从小到大原则，对算子排列如下：

LLH-1: 执行 LLH-c 算子移除路径上的 m 个最差节点，将它们放入移除集。之后执行 m 次 LLH-y 算子，将移除集内的节点基于后悔规则插入路径。

LLH-2: 执行 LLH-c 算子移除路径上的 m 个最差节点，将它们放入移除集。之后执

行 m 次 LLH-x 算子，将移除集内的节点基于贪婪规则插入路径。

LLH-3: 执行 LLH-b 算子移除路径上的 m 个最差距离节点，将它们放入移除集。之后执行 m 次 LLH-y 算子，将移除集内的节点基于后悔规则插入路径。

LLH-4: 执行 LLH-b 算子移除路径上的 m 个最差距离节点，将它们放入移除集。之后执行 m 次 LLH-x 算子，将移除集内的节点基于贪婪规则插入路径。

LLH-5: 执行 LLH-e 算子移除路径上的 m 个最相关节点，将它们放入移除集。之后执行 m 次 LLH-y 算子，将移除集内的节点基于后悔规则插入路径。

LLH-6: 执行 LLH-e 算子移除路径上的 m 个最相关节点，将它们放入移除集。之后执行 m 次 LLH-x 算子，将移除集内的节点基于贪婪规则插入路径。

LLH-7: 执行 LLH-a 算子随机移除路径上的 m 个节点，将它们放入移除集。之后执行 m 次 LLH-x 算子，将移除集内的节点基于贪婪规则插入路径。

LLH-8: 执行 LLH-d 算子移除路径上 m 个位于最长弧上的节点，将它们放入移除集。之后执行 m 次 LLH-x 算子，将移除集内的节点基于贪婪规则插入路径。

4.2.2 节点变换算子的设计

节点变换类算子能够对路径内或路径间的算子进行交换操作，这类交换操作只影响该节点与其前后节点间的关系，而不改变解的可行性。通过节点变换算子对当前路径上节点间关系的变动，能够实现对当前解的更新。本文设计如下 4 个节点变换算子，根据邻域变换的影响范围从小到大原则，对算子排列如下：

LLH-9: 交换算子

该算子会随机选择一个节点，并将其与该节点之后的相邻节点交换位置。

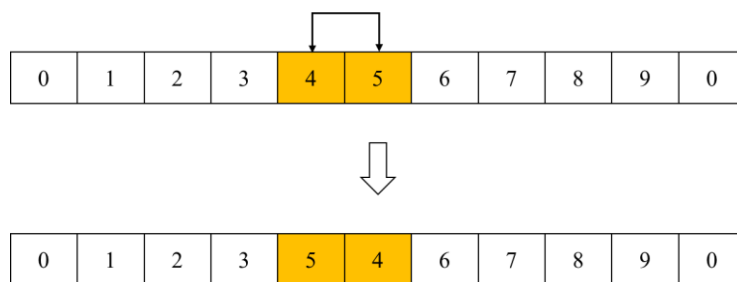


图 4.1 交换算子

Fig. 4.1 Swap operator

LLH-10: 移位算子

该算子会随机选择一个节点，并将其插入到它原本所在路径的其他位置上。

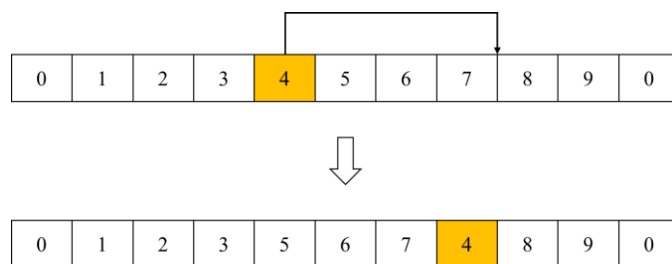


图 4.2 移位算子

Fig. 4.2 Shift operator

LLH-11: 重定位算子

该算子会随机选择一个节点，并将其插入到另一条路径的其他位置上。

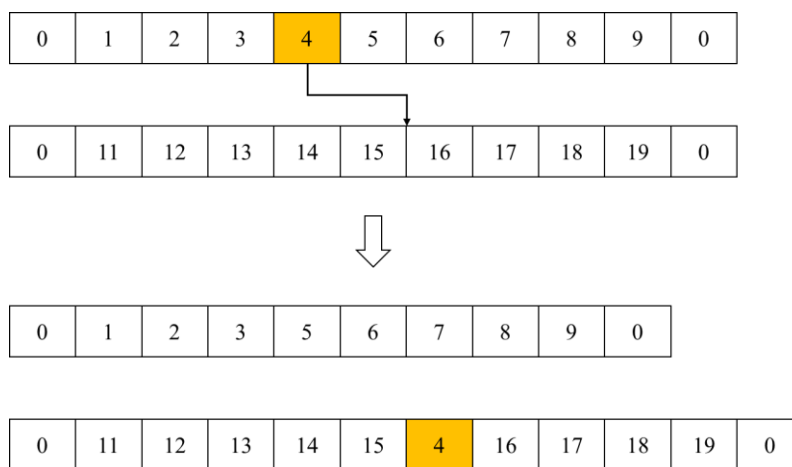


图 4.3 重定位算子

Fig. 4.3 Relocation operator

LLH-12: 交叉算子

该算子会随机选择两个位于不同路径的节点，并交换它们的位置。

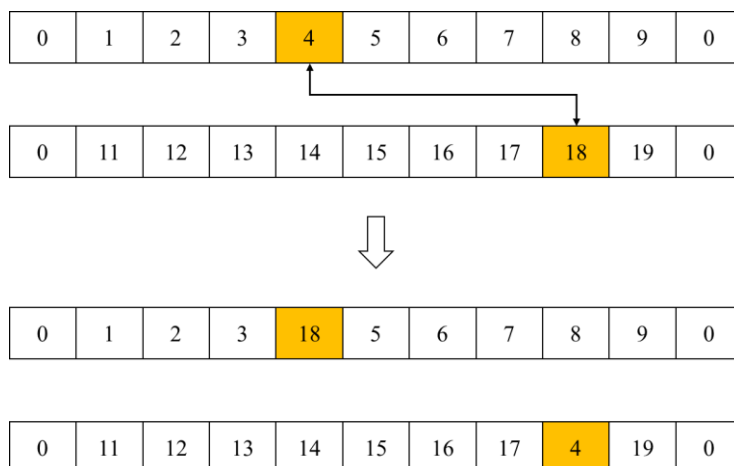


图 4.4 交叉算子

Fig. 4.4 Crossover operator

4.2.3 子序列变换算子的设计

子序列变换类算子能够对路径上的一段子序列进行变换，这种变换可以在路径内进行，也可以在路径间进行。该变换会对当前解造成大范围扰动，能够对当前解进行较大幅度的更新。本文设计如下 2 个子序列变换算子，根据邻域变换的影响范围从小到大原则，对算子排列如下：

LLH-13: 反转算子

该算子将从一条路径内随机选择两个节点，对两个节点之间的子序列顺序进行反转。

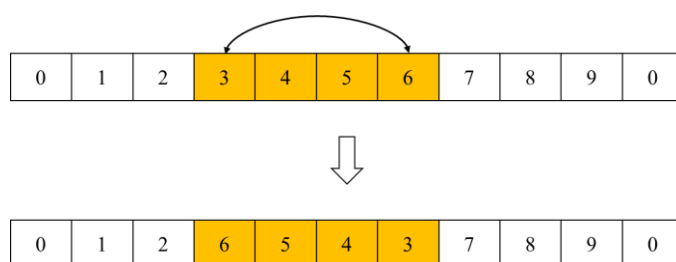


图 4.5 反转算子

Fig. 4.5 Reserve operator

LLH-14: 重连接算子

该算子将从不同的路径上随机选取两条子序列，将子序列与其原本所在路径上的前后节点断开，并重新连接到另一条路径上悬空的节点之中。

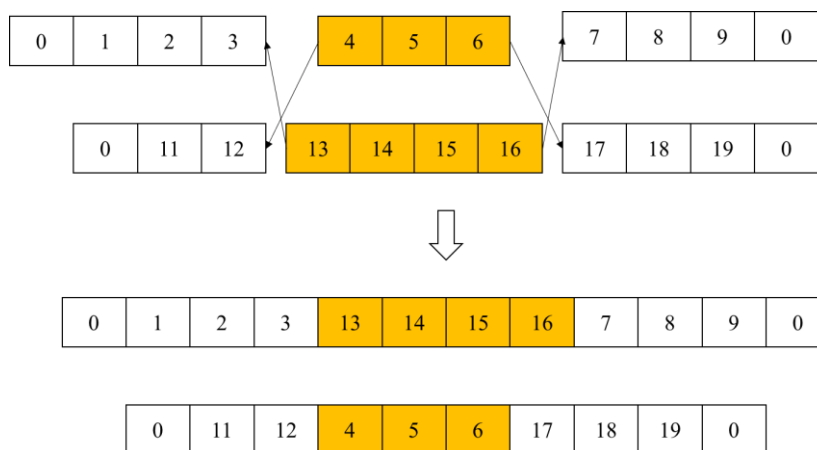


图 4.6 重连接算子

Fig. 4.6 Reconnection operator

4.3 基于脉冲神经网络的高层选择策略

本章前文介绍了超启发算法中待求解问题结构相适应的 14 个低层算子的设计，这一节将介绍基于脉冲神经网络的高层选择策略，这一决策结构能够在求解的不同阶段

观察到不同的求解状况，从而实现对上述 14 个低层求解算子的高效选择。本节将从设计思想、网络设计和网络的训练方法三个方面来对脉冲高层决策结构进行介绍。

4.3.1 基于强化学习的决策思想

在强化学习的价值学习过程中，智能体通过观察到的当前状态 s 、动作 a 和奖赏 r 与环境进行交互，交互的结果为下一状态 s' 。Q 值函数 $Q(s, a)$ 能够根据当前状态和选择的动作给出在这个情况下的整体收益的预期值，即为 Q 值。因此，设计合理的状态、动作和奖赏能够使智能体有效工作。

在超启发框架中，作为面向控制域的部分，高层选择部分所作的工作是选择合适的低层算子对当前解进行更新。因此不难看出，动作空间 \mathcal{A} 即为 14 个低层算子的集合。约定算子编号为动作 a 的值，则有如公式 (4.2) 所示的动作集合：

$$a \in \mathcal{A} = \{1, 2, \dots, 14\} \quad (4.2)$$

而状态为当前解的描述，该描述应能充分反映当前解的优劣及复杂程度，并体现上一动作对当前解的影响，为指导下一动作的选择提供参考。为此，本文设计了一个三元组表述的状态 $s = \langle s_{ir}, s_{cut}, s_{la} \rangle$ ，其能全面地描述当前解的各项特征。

第一个特征 s_{ir} 反映了当前解的改进效果。如公式 (4.3) 所示，该项的值为当前解的改进率 (Improvement Rate, IR)：

$$IR = \frac{cost_{new} - cost_{old}}{cost_{old}} \quad (4.3)$$

其中， $cost_{new}$ 表述当前解的成本（总路程长度）， $cost_{old}$ 表示当前解更新前的成本。这项特征不仅能够反映当前解的优劣，还能反映更新的快慢程度。这一指标对指导如何更高效地更新到更优质的解有所帮助。

第二个特征 s_{cut} 反映了当前解所用的更新时间。如公式 (4.4) 所示，该项的值为当前求解用时 (Current Update Time, CUT)：

$$CUT = currentTimeMillis(UpdateEnd) - currentTimeMillis(UpdateStart) \quad (4.4)$$

其中函数 $currentTimeMillis(\bullet)$ 能够获取精确到毫秒的本机实时时间。这项特征能够反映低层算子将解更新为当前解的过程中所用的时间，这不仅能够在一定程度上反映最近阶段解的结构复杂程度，还能够一定程度上反映在这一阶段使用之前低层算子的求解效率。这一指标能够指导如何选择与当前解的结构更加适宜的低层算子。

第三个特征 s_{la} 反映了更新到当前解所使用的低层算子。该项的值为最近动作 (Last

Action, LA)。在超启发算法中,算子的性能往往与过去所应用的算子有关。例如,在使用子序列变换算子后,使用破坏-修复算子往往会比再次使用子序列变换算子表现更好。因此,使用 s_{la} 对更新到当前解的动作进行记录,有助于指导如何寻找后续的更适宜动作。

奖赏必须是一种度量,其可以令智能体得知自己决策的好坏。解的改进率能够反映决策的好坏,其满足奖赏的定义。然而由于其数值同样会反映更新的快慢,若直接使用解的改进率作为奖赏,会导致解在原本更新就较慢的情况下会产生较小的奖赏 r ,从而使后续的更新更加缓慢。因此,设计奖赏 r 为如公式 (4.5) 所示的形式:

$$r = \begin{cases} +1, & cost_{new} - cost_{old} > 0 \\ 0, & cost_{new} - cost_{old} = 0 \\ -1, & cost_{new} - cost_{old} < 0 \end{cases} \quad (4.5)$$

其中, $cost_{new}$ 表述当前解的成本(总路程长度), $cost_{old}$ 表示当前解更新前的成本。通过这一设计,可以使得智能体在求解的所有阶段内都获得较为稳定的奖赏,奖赏反映了其要学习的方向。

通过设定好的状态、动作和奖赏,智能体进行学习过程。它可以通过在训练过程中最小化 Bellman 方程的均方误差来学习。因此,损失函数定义如公式 (4.6) 所示:

$$\begin{aligned} L(\theta) &= \mathbb{E}_{s,a,r} \left[\left(\mathbb{E}_{s'} [y|s,a] - Q(s,a;\theta) \right)^2 \right] \\ &= \mathbb{E}_{s,a,r,s'} \left[\left(y - Q(s,a;\theta) \right)^2 \right] + \mathbb{E}_{s,a,r} \left[\mathbb{V}_{s'} [y] \right] \end{aligned} \quad (4.6)$$

其中:

$$y = r + \gamma \max_{a'} (s', a'; \hat{\theta}) \quad (4.7)$$

其中, s' 和 a' 表示下一状态步的状态和动作, $\mathbb{E}[\cdot]$ 表示期望, $\mathbb{V}[\cdot]$ 表示方差, θ 是当前网络的权重, $\hat{\theta}$ 是目标网络的权重, γ 是折现因子。当损失函数对权重求导,可以得到如公式 (4.8) 所示的梯度:

$$\nabla_{\theta} L(\theta) = \mathbb{E}_{s,a,r,s'} \left[\left(y - Q(s,a;\theta) \right) \nabla_{\theta} Q(s,a;\theta) \right] \quad (4.8)$$

4.3.2 脉冲强化学习网络的设计

脉冲神经网络可以与传统神经网络拥有相同的结构,它们之间最大的不同是脉冲神经网络中的神经元是脉冲神经元。基于转换思想的 ANN-to-SNN 网络的训练原理便是据此进行:在相同结构的 ANN 上进行训练,将训练好的 ANN 中的神经元激活值等效转

换为 SNN 的脉冲神经元激活率，从而将训练后的 ANN 转换成 SNN。

为了不丧失脉冲神经网络的生物可信性，并保持脉冲神经网络的低功耗特性，本文将直接使用脉冲神经网络进行学习，从而实现端到端的基于脉冲的强化学习。这一学习机制将通过脉冲强化学习网络（Spiking Reinforcement Learning Network, SRLN）来实现。

脉冲神经网络的基本计算单元是脉冲神经元。一个典型的脉冲神经元能够从与其相连的上一个神经元接受刺激，进行神经元膜上电位的积累。当膜上积累的电位超过激活阈值时，该神经元被激活，能够发放一个脉冲，同时该神经元膜上电压复位为静息状态。脉冲神经元模型可形式化描述如公式（4.9）到公式（4.12）所示：

$$H_t = f(V_{t-1}, X_t) \quad (4.9)$$

$$S_t = \Theta(H_t - V_{th}) \quad (4.10)$$

$$V_t = H_t(1 - S_t) + V_{reset}S_t \quad (4.11)$$

$$\Theta(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (4.12)$$

其中， V_{t-1} 表示在时间步 t 之前该神经元膜上的电位， X_t 表示时间步 t 时该神经元收到的输入， H_t 表示神经元在时间步 t 时累积的膜上电位， V_{th} 表示神经元的激活阈值， $\Theta(\cdot)$ 为神经元的激活函数。 S_t 表示神经元在时间步 t 的输出脉冲，若神经元在时间步 t 能够输出脉冲则为 1，否则为 0。 V_{reset} 为静息电压， V_t 为时间步 t 之后神经元膜上的实际电压。

其中，公式（4.9）描述了脉冲神经元的亚阈值状态，其表示神经元在没有达到激活阈值时神经元的动力学状态。不同类型的脉冲神经元的主要差异便是亚阈值状态的不同。为了平衡仿生的真实性和训练的复杂程度，本文采用的是 LIF 神经元。当未达到激活阈值时，神经元膜上电位随时间泄露。LIF 神经元模型的 $f(\cdot)$ 可定义如公式（4.13）所示：

$$f(V_{t-1}, X_t) = V_{t-1} + \frac{1}{\tau}(-(V_{t-1} - V_{reset}) + X_t) \quad (4.13)$$

其中， τ 为膜电位时间常数。

虽然 LIF 神经元在生物可信性和训练复杂度上做到了很好的平衡，但从其形式化模型不难看出，该神经元的输出为脉冲序列。然而，在强化学习中，最后输出的 Q 值应为一个实数。为了弥合这两种数据形式间的差异，需要一个脉冲解码器来完成数据转换工

作。为了尽量减少网络的训练难度，更好的发挥脉冲神经网络的低功耗特性，本文设计了膜电位（Membrane Potential, MP）神经元作为脉冲神经网络最后一层的神经元。该神经元在完成数据转换工作的同时，还能通过膜电位的累积与泄露充分发掘神经元中潜藏的时间信息。其模型可形式化为如公式（4.14）和公式（4.15）的形式：

$$V_t = \left(1 - \frac{1}{\tau}\right) V_{t-1} + \frac{1}{\tau} X_t \quad (4.14)$$

$$O_t = V_t \quad (4.15)$$

其中， O_t 表示神经元在时间步 t 时的输出。可以看出，相较于 LIF 神经元，MP 神经元的输出不再为一个脉冲序列，而是神经元在每个时间步之后的膜上电压，而这是一个实数。对于不同的 MP 神经元，膜电压越大则意味着采取相应动作的概率越大，这与 Q 值的作用是相同的。

同时，对比公式（4.10）、公式（4.12）和公式（4.15），可以看出 LIF 神经元在其膜上电位达到阈值时会激活并发放脉冲，同时将膜上电位重置为静息电位。而 MP 神经元则不存在激活与静息的活动机制。因此，在进行网络的反向传播时，MP 神经元可以看作是一个激活阈值无穷大、静息电位为零的 LIF 神经元。这意味着在后续的训练过程中，MP 神经元可以与 LIF 神经元一同参与训练。与单独设置解码器相比，这个操作可以大大降低脉冲神经网络的训练复杂度。

然而，在整个仿真时间 T 中，可以得到每个时间步 t 的膜电压 V_t 。为了表示最终的 Q 值，需要在这 T 个膜电压中选择一个最优统计量作为输出。为了满足这个需要，本文最终设计了三个有意义的统计量作为候选统计量：

（1）最终膜电压（Last Membrane Voltage, LastMV）：如公式（4.16）所示，使用最后时刻 MP 神经元上膜电压作为 Q 值。这一统计量计算量最小，同时可以反映整个仿真时间之后 MP 神经元膜上的累积和泄露效应。

$$Q_{LastMV} = V_T \quad (4.16)$$

（2）最大膜电压（Maximum Membrane Voltage, MaxMV）：如公式（4.17）所示，通过收集整个仿真时间内的 MP 神经元的膜电压，取其中的峰值作为 Q 值。这一统计量能够避免一些较早激活的次末层 LIF 神经元对 MP 神经元的影响被时间弱化。

$$Q_{MaxMV} = \max_{1 \leq t \leq T} V_t \quad (4.17)$$

（3）平均膜电压（Mean Membrane Voltage, MeanMV）：如公式（4.18）所示，与最大膜电压类似，同样收集整个仿真过程内的 MP 神经元的膜电压，取它们的均值作为

Q 值。

$$Q_{MeanMV} = \frac{1}{T} \sum_{t=1}^T V_t \quad (4.18)$$

为了从这三个统计量中选择更适宜的统计量作为 Q 值, 本文使用了一个简单的案例来评估它们的好坏。以一个简单的网络为例, 其仅包含一个 LIF 神经元和一个 MP 神经元。其中, LIF 神经元前突触的权重为 w_1 , MP 神经元前突触的权重为 w_2 。当对网络随时间 t 输入电流 $I(t)$ 时, LIF 神经元接受一个输入 $X(t) = w_1 I(t)$, 而 MP 神经元则接受 LIF 神经元的加权脉冲输出 $w_2 S(t)$ 。在这个例子中, 设置整个仿真时长为 8 个时间步, 膜时间常数为 2.0, 静息电压为 0, LIF 神经元的激活阈值为 1.0, 权重 w_1 和 w_2 均为 1.0, $I(t)$ 为实数且 $I(t) \in [0, 2]$ 。此时, 可以绘制出输入电流 $I(t)$ 与 MP 神经元上膜电压的三个统计量的关系如图 4.7 所示。

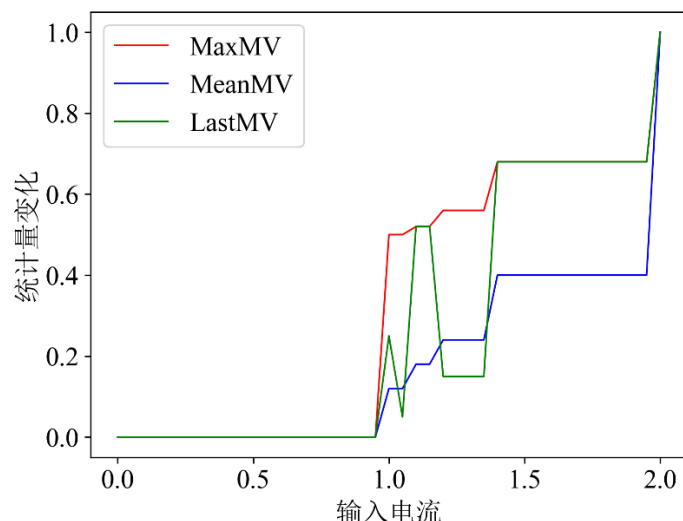


图 4.7 输入电流引发的变化

Fig. 4.7 Input Current Induced Changes

可以看出, 在某些时刻, 随着输入电流的轻微扰动, LIF 神经元的激活时间会提前, 但又不足以产生新的脉冲。这时, MP 神经元上的最终膜电压会急剧衰减。这种不稳定性对于训练是有害的, 因此本文不采用。而最大膜电压与平均膜电压都能很好工作, 且它们的走势基本一致。考虑到最大膜电压更能够反映 MP 神经元的时间信息, 而平均膜电压由于对每一个时间步上的 MP 神经元膜电压都进行计算, 对于神经元随时间的电位累积和电位泄露并不能很好地反映, 因此最终选择最大膜电压这一统计量作为最终 Q 值。

因此, 可以将 LIF 神经元作为隐含层的组织单元, 而将 MP 神经元作为输出层的组织单元, 构建出超启发算法的高层决策结构 SRLN, 如图 4.8 所示。

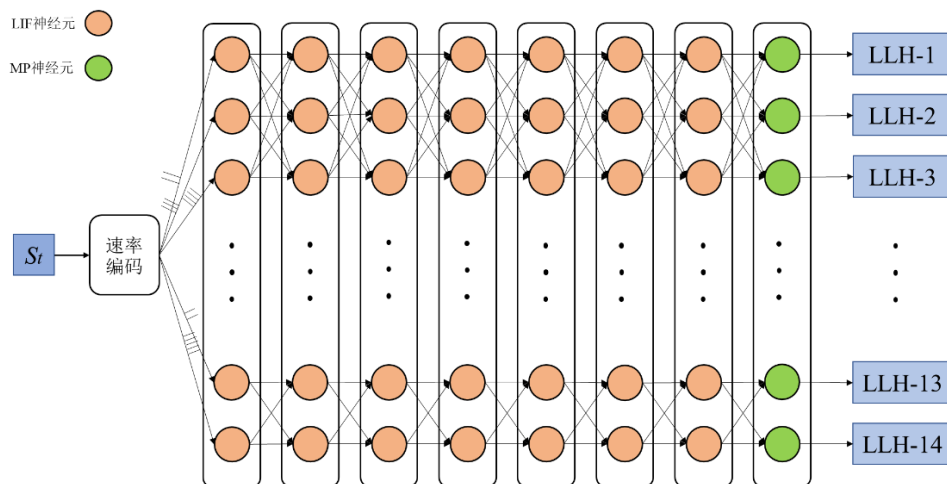


图 4.8 SRLN 结构图

Fig. 4.8 SRLN structure diagram

4.3.3 SRLN 的训练方法

通过上述设计，可以推导出 SRLN 的反向传播算法，从而对网络进行高效训练。由公式（4.6）可知，只需要推导出 Q 值对于权重的梯度，便可以借此对网络进行训练。

传统脉冲神经元的激活函数为阶跃函数，因此不能进行求导操作。为了使 SRLN 的反向传播成为可能，本文使用替代函数 $\sigma(x)$ 替代了原本的激活函数 $\Theta(x)$ 进行神经元的激活和权值的反向传播。定义替代函数 $\sigma(x)$ 如公式（4.19）所示：

$$\sigma(x) = \frac{1}{\pi} \arctan \pi x + \frac{1}{2} \quad (4.19)$$

该函数图像如图 4.9 所示。可以看出，函数的值在 0 到 1 范围之内发生变化，并在点(0,0.5)附近变化剧烈，因此可以较好地对本原的阶跃函数进行模拟。

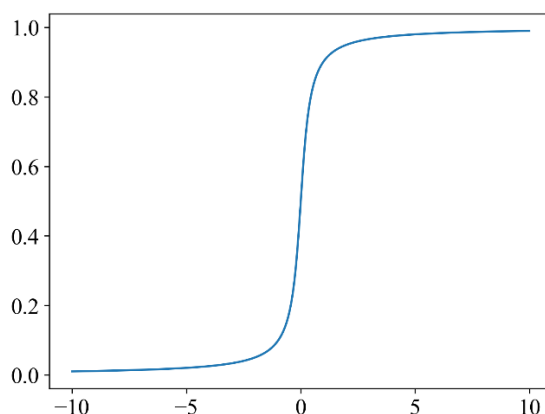


图 4.9 替代函数

Fig. 4.9 Substitution function

对公式 (4.19) 求导, 可得公式 (4.20) :

$$\sigma'(x) = \frac{1}{1 + (\pi x)^2} \quad (4.20)$$

由于选择最大膜电压作为最终的 Q 值, 可以得到公式 (4.21) 和公式 (4.22) :

$$t' = \arg \max_{1 \leq t \leq T} V_t^l \quad (4.21)$$

$$\frac{\delta V_t^L}{\delta \theta^L} = \begin{cases} \frac{\delta V_t^L}{\delta V_{t-1}^L} \frac{\delta V_{t-1}^L}{\delta \theta^L} + \frac{\delta V_t^L}{\delta X_t^L} \frac{\delta X_t^L}{\delta \theta^L}, & t > 1 \\ \frac{\delta V_t^L}{\delta X_t^L} \frac{\delta X_t^L}{\delta \theta^L}, & t = 1 \end{cases} \quad (4.22)$$

其中, L 为网络总层数, V_t^l 为第 l 层神经元在时间步 t 时的膜电压, X_t^l 为第 l 层神经元在时间步 t 时的输入。当 $l > 1$ 时, 约定 $I_t^l = S_t^{l-1}$, 从而保证 $X_t^l = \theta^l I_t^l$ 。 θ^l 为第 l 层神经元的权重。为了降低网络的训练难度, 本文设定脉冲神经元在所有时间步上共享权重。

由公式 (4.7) 到公式 (4.11)、公式 (4.15) 和公式 (4.22), 可以得到 Q 值对脉冲神经元膜上累积电位的偏导数如公式 (4.23) 所示:

$$\frac{\delta Q}{\delta H_t^l} = \begin{cases} \frac{\delta Q}{\delta H_{t+1}^l} \frac{\delta H_{t+1}^l}{\delta H_t^l} + \frac{\delta V_{t'}^l}{\delta S_t^{l-1}} \frac{\delta S_t^{l-1}}{\delta H_t^l}, & t > t' \\ 0, & t < t' \\ \frac{\delta V_{t'}^l}{\delta S_t^{l-1}} \frac{\delta S_t^{l-1}}{\delta H_t^l}, & t = t' \end{cases} \quad (4.23)$$

其中:

$$\frac{\delta H_{t+1}^l}{\delta H_t^l} = \frac{\delta H_{t+1}^l}{\delta V_t^l} \frac{\delta V_t^l}{\delta H_t^l} = \left(1 - \frac{1}{\tau}\right) \frac{\delta V_t^l}{\delta H_t^l} \quad (4.24)$$

$$\frac{\delta V_{t'}^l}{\delta S_t^{l-1}} = \frac{\delta V_{t'}^l}{\delta V_t^l} \frac{\delta V_t^l}{\delta X_t^l} \frac{\delta X_t^l}{\delta S_t^{l-1}} = \left(1 - \frac{1}{\tau}\right)^{t'-t} \frac{\theta^l}{\tau} \quad (4.25)$$

$$\frac{\delta S_t^j}{\delta H_t^i} = \begin{cases} \frac{\delta S_t^j}{\delta H_t^j} \frac{\delta H_t^j}{\delta X_t^j} \frac{\delta X_t^j}{\delta S_t^{j-1}} \frac{\delta S_t^{j-1}}{\delta H_t^i}, & j > i \\ \sigma'(H_t^j - V_{th}), & j = i \end{cases} \quad (4.26)$$

而由公式 (4.9) 和公式 (4.11), 可以得到公式 (4.24) 中脉冲神经元膜上实际电位对脉冲神经元膜上累积电位的偏导数如公式 (4.27) 所示:

$$\frac{\delta V_t^i}{\delta H_t^i} = 1 - S_t^i + (V_{\text{reset}} - H_t^i) \frac{\delta S_t^i}{\delta H_t^i} \quad (4.27)$$

此外，由公式（4.22）到公式（4.24），可以得到脉冲神经元膜上累积电位对突触权重的偏导数如公式（4.28）所示：

$$\frac{\delta H_t^i}{\delta \theta^i} = \begin{cases} \frac{\delta H_t^i}{\delta V_{t-1}^i} \frac{\delta V_{t-1}^i}{\delta H_{t-1}^i} \frac{\delta H_{t-1}^i}{\delta \theta^i} + \frac{\delta H_t^i}{\delta X_t^i} \frac{\delta X_t^i}{\delta \theta^i}, & t > 1 \\ \frac{\delta H_t^i}{\delta X_t^i} \frac{\delta X_t^i}{\delta \theta^i}, & t = 1 \end{cases} \quad (4.28)$$

最终，由公式（4.23）和公式（4.28），可以推导出 Q 值对于权重的梯度如公式（4.29）所示：

$$\frac{\delta Q}{\delta \theta^l} = \begin{cases} \sum_{t=1}^T \frac{\delta Q}{\delta H_t^l} \frac{\delta H_t^l}{\delta \theta^l}, & l < L \\ \frac{\delta V_{t'}^l}{\delta \theta^l}, & l = L \end{cases} \quad (4.29)$$

通过在脉冲神经网络中对上述权重梯度进行反向传播，SRLN 可以实现权重的快速更新，从而使网络能够高效训练。

4.4 本章小结

本章首先在 4.1 节介绍了本章方案提出的动机，然后在 4.2 节介绍了脉冲超启发算法框架的低层求解算子的设计，4.3 节中介绍了超启发算法框架中基于脉冲神经网络的高层决策结构的设计，并对其用于推理的前向传播机制和用于训练的反向传播算法进行了推导，证明了该算法的可学习性。

第5章 基于短时程突触可塑性机制的改进脉冲神经元

5.1 研究动机

脉冲神经网络因为其丰富的动力学机制，在消耗计算资源相同的情况下具有更强的决策能力。且由于其在进行脉冲发放时，不同神经元在整个仿真过程中的不同时间步上活跃，因此生成的脉冲序列天然具有时序信息，这一特性能够适应组合优化求解任务在不同求解阶段上解的结构的不同，对低层求解算子的选择带来帮助。

然而，脉冲神经网络的动力学机制也造成了其脉冲发放的稀疏性。这一特性虽然会降低网络在整个仿真时间上的平均活跃水平，使脉冲神经网络拥有极佳的能效比，但对于深层脉冲神经网络而言，其也带来一些弊端。由于脉冲发放是稀疏的，且 LIF 神经动力学机制的泄露项会导致膜电位随时间衰减，因此随着网络层数的加深，脉冲神经网络各层的激活率亦会不断衰减。如图 5.1 所示，脉冲神经网络各层间的激活数存在着极大的不平衡。第一层由于直接接受外界输入，因此激活数量远远高于网络的后几层。这种神经元激活的不平衡会对网络的决策效率带来不良影响。

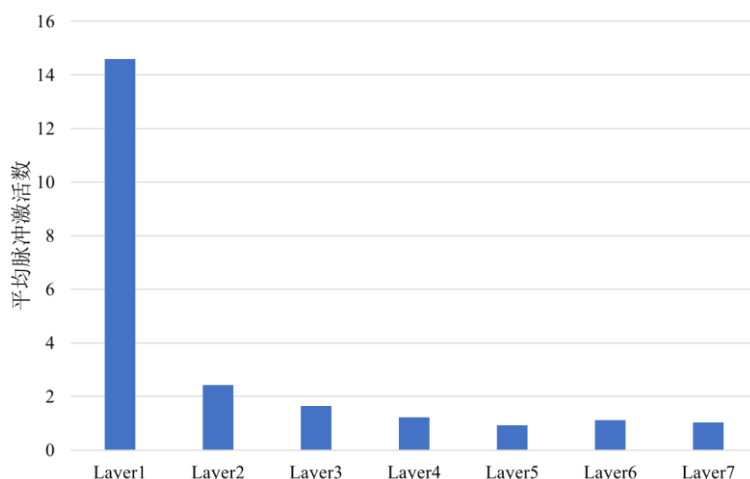


图 5.1 脉冲神经网络中不同层的平均激活数

Fig. 5.1 Average number of fires at different layers in spiking neural network

在神经科学领域的研究中，研究人员发现脑皮层内的神经元的群体活动，往往处在一种平衡的临界点附近，常表现出一种自组织临界（Self-Organized Criticality, SOC）的状态。比如神经系统在优化处理信息的神经雪崩时，其尺寸和持续的时间往往呈现幂律分布^[91]，其幂指数在不同个体和不同实验情况下依旧服从一定的动力学标度关系^[92]。

这种平衡的临界态是生物为了在抗外界干扰的鲁棒性和响应外界刺激的灵活性之间达成平衡的进化结果^[93]，因此，当生物脑内的神经群体处于这种平衡状态时，神经系统能够展现最佳的计算能力^[94]。作为对生物体神经元的模拟，脉冲神经元组成的脉冲神经网络也具有与神经系统类似的性质。处于不平衡状态的脉冲神经网络无疑不能发挥其最优的决策能力。

综上所述，寻求一种合理的机制来解决脉冲神经网络中的不平衡问题，使网络中不同层的神经元的激活率回归到平衡状态，对提升整个脉冲超启发算法的求解能力有着极大裨益。

5.2 短时程突触可塑性机制

在生物体内的神经元动力学机制中，长时程突触可塑性（Long-term Synaptic Plasticity, LTSP）机制^[95]被首先发现，其被认为与突触间强度的变化乃至突触的形成有着重要关联。而记忆，尤其是长期记忆的形成，其生理基础常被认为与突触的更新紧密相关。基于此，长时程突触可塑性被广泛研究，并被应用于类脑计算领域。长时程突触可塑性可视为由兴奋和抑制两个机制共同组成：长时程增强（Long-term Potentiation, LTP）机制和长时程抑制（Long-term Depression, LTD）机制。前者表现为在长时间使用高频电信号刺激突触前神经元后，突触后神经元产生的电位相较于刺激前会有大幅度增强；而 LTD 机制则相反，当长时间使用低频电信号刺激突触前神经元，则突触后神经元产生的电位相较于刺激前会有大幅度削弱。之所以会有这样的差异，是因为高频刺激和低频刺激对神经细胞的诱导，会使得进入细胞的 Ca^{2+} 的粒子浓度产生差异。这种差异会激活细胞内不同种类的酶，从而使细胞电位变化不同。

而 LTP 机制与 LTD 机制的共同作用，则会导致脉冲时间依赖可塑性（Spike-timing Dependent Plasticity, STDP）的形成。这是一种由于突触前神经元和突触后神经元之间紧密的时间相关性所造成的非对称 Hebbian 学习规则^[62]。当突触前神经元的激活先于突触后神经元激活的发生时，会诱导 LTP 机制的产生，从而实质上使得这两个神经元间的突触连接强度增加；而当突触前神经元的激活比突触后神经元的激活更迟时，则会诱导 LTD 机制的产生，从而使得这两个神经元间的突触连接强度受到削弱。这一机制被认为是大脑记忆信息和学习知识的基础，因此被广泛应用于类脑计算领域。实际上，很大一部分脉冲神经网络的学习机制都是基于 STDP 的学习机制。

而相较于 LTSP 机制，短时程突触可塑性（Short-term Synaptic Plasticity, STSP）机制^[96]所受到的关注则要少得多。但在生物体内，如神经肌肉接头处^[93]，STSP 机制并不

罕见。与 LTSP 机制类似，STSP 机制也可看作由两个拮抗的机制共同组成，它们分别是短时程增强（Short-term Enhancement, STE）机制和短时程抑制（Short-term Depression, STD）机制^[97]。

5.2.1 短时程增强机制

STE 机制常在神经递质释放概率低的突触中表现明显。这类突触的突触前神经元的激活很难导致突触后神经元的激活。在这类突触中，若对突触前神经元重复刺激，使其在短时间内连续激活，则其在后续的激活中引发的突触后电位变化会显著增强。STE 机制的产生很可能与突触间隙内离子浓度的改变有关^[96]。以两个短间隔的突触前动作电位为例，若第一个突触前动作电位未引起突触后神经元的激活（突触后动作电位未达到阈值），则当配对脉冲对平行纤维进行刺激时，第二突触前刺激所引发的突触后电位改变可达到第一刺激所引发的突触后电位的改变的两倍。此时， Ca^{2+} 离子通过 Ca^{2+} 离子通道进入到突触间隙中，导致突触间隙中 Ca^{2+} 离子浓度的增高。虽然随着 Ca^{2+} 离子通道的关闭，突触前末端的 Ca^{2+} 离子浓度会逐渐达到平衡状态，但残留的 Ca^{2+} 离子会使得新平衡态时的突触间隙内 Ca^{2+} 离子浓度整体提升。这一高 Ca^{2+} 离子浓度的平衡态突触间隙会放大刺激所造成的电位改变，从而使得第二刺激引发的突触后电位的改变增强。

不难看出，位于整个脉冲神经网络后几层的神经元间连接与神经递质释放概率低的突触处在类似的情形。在这里，脉冲神经元的激活率显著衰减。根据神经递质的浓度变化，可以将其看作突触前神经元的激活能够为突触后神经元提高资源利用率。这一利用率的提升能够提高突触后神经元对刺激的敏感性，导致膜上电位变化的提升。同时，突触后神经元超常的电位变化将消耗浓度过高的神经递质，促进新稳态的形成，使资源利用率最终回归到基线值。因此，可以建立如公式（5.1）所示的神经元动力学模型：

$$\frac{du(t)}{dt} = \frac{U - u(t^-)}{\tau_E} + a_E(1 - u(t))\delta(t - t_s) \quad (5.1)$$

其中， $\delta(\bullet)$ 是狄拉克函数， $u(t)$ 表示神经元在当前时刻 t 的资源利用率， $u(t^-)$ 是本时刻之前的资源利用率， a_E 表示增强因子。当神经元在时间步 t_s 激活时，资源利用率瞬时增加 $a_E(1 - u(t^-))$ ，并随着增强时间常数 τ_E 逐渐落回到资源利用率基线值 U 。

5.2.2 短时程抑制机制

STD 机制则常在神经递质释放概率高的突触中表现明显。这类突触的突触前神经元的激活往往导致突触后神经元的激活。在这类突触中，若对突触前神经元重复刺激，使

其在短时间内重复激活，则在其后续的激活中引发的突触后电位变化的幅值会降低。STD 机制的产生或许与突触前神经元囊泡的损耗及突触后神经元受体的失敏有关^[96]。以两个短间隔的突触前动作电位为例，第一刺激的产生会令突触前神经元释放大囊泡，将神经递质送入突触间隙。此时，当第二刺激到达时，囊泡的损耗使得突触前神经元不具备充足的囊泡用以释放神经递质，从而导致释放到突触间隙的神经递质不足，继而不足以引起突触后电位的大幅变化。同时，当突触后神经元接受到足够的神经递质得以激活时，神经元细胞膜上的受体会在这一过程中进入短暂的失敏状态。此时第二刺激释放的神经递质通过突触间隙传导到突触后神经元，由于膜上大部分受体仍处在失敏状态，不能对神经递质进行及时的响应，因此造成神经元的电位变化幅度较小。

不难看出，位于整个脉冲神经网络前几层的神经元间连接与神经递质释放概率高的突触处在类似的情形。在这里，脉冲神经元的激活率异常高。根据囊泡的减少和受体的失敏，可以将其看作突触前神经元的激活将消耗突触后神经元的激活资源。这种消耗将影响突触后神经元的受激反应，使得其电位变化减小。同时，被消耗的资源将因为生物体内的动态平衡而逐渐增加，最终回归到基线值（ $R=1$ ）。因此，可以建立如公式（5.2）的神经元动力学模型：

$$\frac{dR(t)}{dt} = \frac{1-R(t)}{\tau_D} - u(t^-)R(t^-)\delta(t-t_s) \quad (5.2)$$

其中， $R(t^-)$ 是本时刻之前的激活资源。当神经元激活时，激活资源因为被消耗而瞬时降低 $u(t^-)R(t^-)$ ，并随着抑制时间常数 τ_D 逐渐回归到基线值。

5.2.3 脉冲神经元的增强-抑制转换

在生物体内部，神经元实际上会受到 STE 机制和 STD 机制的共同影响^[93,95-97]。表现为增强的神经元，其 STSP 机制中的 STD 机制弱于 STE 机制；表现为抑制的神经元，其 STSP 机制中的 STE 机制弱于 STD 机制。在特定情况下，某些神经元会发生增强-抑制态的转换。

为了更好地贴近生物神经元的实际情况，设计脉冲神经网络中突触前神经元的激活在突触后神经元上诱发的电流振幅由 STE 机制和 STD 机制共同影响。令电流振幅 I_{syn} 的动力学机制如公式（5.3）所示：

$$I_{syn}(t_s) = AR(t_s)u(t_s) \quad (5.3)$$

其中， t_s 表示神经元激活时所位于的时间步， A 表示在所有激活资源都不被消耗且神经元的资源利用率达到峰值时瞬时激发的情况下，突触前神经元可以唤起突触后神经

元的最大电流。

由公式 (5.1) 到公式 (5.3) 可看出, 通过调整四元组 $\langle \tau_E, \tau_D, a_E, U \rangle$, 可以调整突触前神经元的激活在突触后神经元上诱发的电流振幅 I_{syn} 的大小, 从而使得神经元呈现增强或抑制效果。这即能够更加贴切地模拟生物神经元中真实存在的活动机制, 也将为后续的脉冲神经元的改良提供便利。

5.3 STSP-LIF 神经元的设计

在上一节中, 本文对脉冲神经元的 STSP 机制进行了动力学推导。这一普遍存在于各类型脉冲神经元的动力学机制, 也可适用于 LIF 神经元。然而, 正如 LIF 神经元本身的动力学设计考虑到了生物可信性与训练复杂度的平衡, 为了使高效训练成为可能, LIF 神经元上的 STSP 机制也应进行针对性设计。在不失其平衡作用下, 本节对其进行了便于网络训练的简化。

从上一节的讨论可以看出, 虽然 STSP 机制中的 STE 机制和 STD 机制在神经元上共同存在, 但神经元会因二者影响力的不同而表现出增强或抑制效果。同时, 从图 5.1 可以看出, 脉冲神经网络中神经元激活的不平衡主要体现在其浅层的异常激活, 尤其是第一层。因此, 较为理想的使整个网络脉冲激活平衡的手段是对过于密集的脉冲输入进行抑制。故在 LIF 神经元组成的脉冲神经网络中, 令神经元的 STD 机制占据主导作用, 使神经元表现为抑制性神经元。

为了便于训练, 在对 LIF 神经元设计 STD 机制时, 将激活资源的消耗等效转化为神经元激活的难易程度。神经元每接收一个脉冲的输入, 会增加该神经元的激活阈值, 使得神经元的激活更加困难; 增加的阈值会逐渐回归到基线阈值, 且当前神经元达到激活状态时, 也会将激活阈值重置为基线阈值。因此, 将 LIF 神经元上的 STD 机制设计为如公式 (5.4) 和公式 (5.5) 所示的动态阈值机制:

$$U_{th}(t) = V_{th}(t-1) + \frac{\beta}{\tau_d} \cdot \left(-\left(V_{th}(t-1) - V_{th}^0 \right) + X_t \right) \quad (5.4)$$

$$V_{th}(t) = U_{th}(t) \cdot (1 - S_t) + V_{th}^0 \cdot S_t \quad (5.5)$$

其中, $U_{th}(t)$ 表示时间步 t 时 LIF 神经元的动态累积阈值, V_{th}^0 表示 LIF 神经元的基线阈值。 β 是神经元的阈值累积因子, 当神经元接受一个输入 X_t 时, 该神经元的动态累积阈值 $U_{th}(t)$ 会随之提高, 并随着神经元的抑制时间常数 τ_d 逐渐落回基线阈值 V_{th}^0 。 $V_{th}(t)$ 表示时间步 t 时 LIF 神经元的实际动态阈值。当神经元处于亚阈值状态时, 实际动态阈值 $V_{th}(t)$ 为当前时间步 t 时的动态累积阈值 $U_{th}(t)$; 当神经元激活时, 实际动态阈

值 $V_{th}(t)$ 重置为基线阈值 V_{th}^0 。

由公式 (4.10) 可以看出, 脉冲神经元激活与否的关键是膜上累积的电压与激活阈值的差距。且由公式 (4.11) 和公式 (5.5) 可以看出, 当神经元激活时, 其膜电压 V_t 会重置为静息电压 V_{reset} , 其动态阈值 $V_{th}(t)$ 也会重置为基线阈值 V_{th}^0 。因此, 可以将 LIF 神经元的动态阈值变化等效转化为膜上电压的动态变化。由公式 (4.13) 和公式 (5.4), 可以得到 LIF 神经元新的亚阈值状态下的膜电压动力学模型如公式 (5.6) 和公式 (5.7) 所示:

$$f_{new}(V_{t-1}, X_t) = V_{t-1} + \frac{1}{\tau}(-(V_{t-1} - V_{reset}) + X_t) - \frac{\beta}{\tau_d}(-\alpha(V_{t-1} - V_{th}^0) + X_t) \quad (5.6)$$

$$\alpha = \frac{V_{th}(t-1) - V_{th}^0}{V_{th}^0 - V_{t-1}} \quad (5.7)$$

其中, α 是膜电位-阈值比例因子, 该因子的值由三元组 $\langle \beta, \tau, \tau_d \rangle$ 共同确定。通过这一比例因子, 可以将动态阈值直接转化为膜上电位的变化。这种转化能够使 STD 机制成为可学习的。由公式 (5.6) 和公式 (4.24), 可以得到存在 STD 机制的神经元在进行梯度的反向传播时, 神经元的下一时刻膜上累计电位对上一时刻膜上累计电位的偏导数如公式 (5.8) 所示:

$$\frac{\delta H_{t+1}^i}{\delta H_t^i} = \frac{\delta H_{t+1}^i}{\delta V_t^i} \frac{\delta V_t^i}{\delta H_t^i} = \left(1 - \frac{1}{\tau} + \frac{\alpha\beta}{\tau_d}\right) \frac{\delta V_t^i}{\delta H_t^i} \quad (5.8)$$

而由公式 (5.6) 和公式 (4.25), 可以得到存在 STD 机制的神经元在进行梯度的反向传播时, 神经元的下一层膜上实际电位对上一层脉冲输出的偏导数如公式 (5.9) 所示:

$$\frac{\delta V_{t'}^l}{\delta S_t^{l-1}} = \frac{\delta V_{t'}^l}{\delta V_t^l} \frac{\delta V_t^l}{\delta X_t^l} \frac{\delta X_t^l}{\delta S_t^{l-1}} = \left(1 - \frac{1}{\tau} + \frac{\alpha\beta}{\tau_d}\right)^{t'-t} \left(\frac{1}{\tau} - \frac{\beta}{\tau_d}\right) \theta^l \quad (5.9)$$

借助上述转化, 可以将 STD 机制转变为 LIF 神经元膜上电压随抑制时间常数 τ_d 变化的动态抑制, 从而将 STSP 机制纳入到整个训练模型中。将第 4 章中神经元反向传播时在时域上的反向传播公式 (4.24) 替换为带 STSP 机制的时域上的反向传播公式 (5.8), 在空域上的反向传播公式 (4.25) 替换为带 STSP 机制的空域上的反向传播公式 (5.9), 即可将 STSP 机制融入到第 4 章中的梯度反向传播过程。

借助带 STSP 机制的梯度反向传播, 使得 LIF 神经元在利用生物可信性平衡网络各层之间脉冲激活率的同时, 能够不过多增加训练复杂度, 从而令脉冲神经网络依旧能够高效地学习。

5.4 本章小结

本章首先在 5.1 节介绍了本章方法提出的研究动机,然后在 5.2 节对 STSP 机制进行了详细的介绍,并在 5.3 节中将 STSP 机制引入到了常见的 LIF 神经元中,得到了改良后的 STSP-LIF 神经元。这种改良在考虑到生物可信性的同时,还兼顾了计算上的便利。STSP-LIF 神经元依旧能够像 LIF 神经元那样参与网络的训练。本章最后由 STSP-LIF 神经元的动力学机制推导出了新的反向传播算法,该反向传播算法在融入了短时程突触可塑性的同时,依旧能够很好地与第 4 章中 SRLN 原有的反向传播过程相结合,证明了该算法在训练上的便捷性。

第6章 基于时空信息的负反馈调节机制

6.1 研究动机

脉冲神经网络在引入丰富时序信息的同时，也为训练带来了困难。相较于传统神经网络仅需要在层间进行梯度的反向传播，脉冲神经网络的结构可视为在时域和空域上都进行了组织，因此在梯度的反向传播时，需要如图 6.1 所示，从时间和空间两个维度进行梯度的反向传播。这无疑加重了训练的负担。

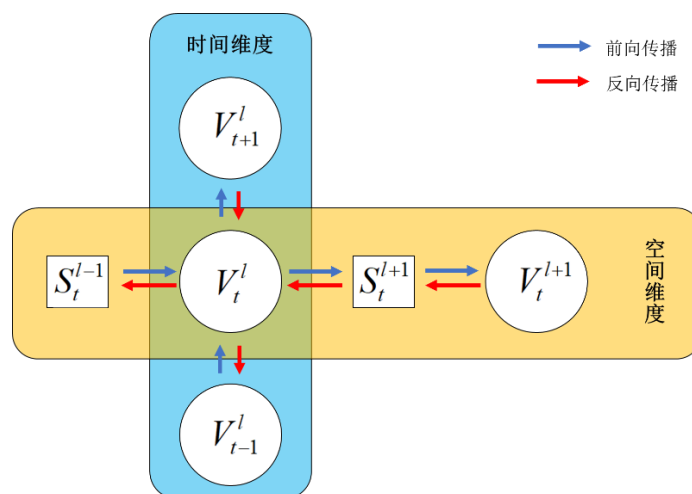


图 6.1 SNN 在时空上的信息传播

Fig. 6.1 Information Propagation of SNN in Time and Space

负反馈调节机制是生物体内广泛存在的机制，其表现为生物体的自我调节的过程。该机制通过各类感受器感知内部或外部的环境变化，并通过电信号或激素的控制来调节生物体内的各类生化反应，对环境变化进行削弱或减轻，从而维持生物体内环境的相对恒定。生物体内的负反馈调节机制是非常重要的生理机制，该机制能够使生物体整体上表现出相对稳定的状态，是生物体内稳态的基础。反向传播算法就是对生物体内负反馈调节机制的一种模拟，其遍布整个网络的权重更新类似于负反馈调节机制中的激素调节。其作用范围大、训练时间长，更新机制可类比生物体内的长反馈调节。然而，生物体内存在多种负反馈调节机制，如单个神经元上的自突触（autapse）机制^{[98][99]}，能够将神经元本身的动作电位转变为抑制性刺激作用于自己，从而实现单个神经元上的负反馈调节^{[100][101]}。这些多样化的负反馈机制联合作用于神经系统，能够发掘大量时空信息，从而使生物体能够对外界刺激高效学习^[102-105]。

本章模拟生物体内多样化的负反馈调节机制，将基于时域负反馈调节的阈下记录和基于空域负反馈调节的层内激活率融入到反向传播算法中，通过这种仿生的混合负反馈

调节机制，充分挖掘时空信息，通过时空信息中所隐含的神经元活动状态来加快突触的权重更新速度，从而加速脉冲神经网络的学习过程。

6.2 基于时域负反馈调节的阈下记录

脉冲神经元相较于传统神经元更加复杂的动力学机制，使得其天然携带大量的时序信息。然而，为了降低网络训练的难度，在之前的训练中，本文约定网络中的 LIF 神经元和 MP 神经元在各时间步上共享权重。这种设计能够避免时间步的增加导致的梯度爆炸，但这样使得神经元的时序信息只能够在前向传播过程中体现，而在反向传播过程中，网络只能发掘到脉冲神经元在整个仿真时间中神经元活动的统计量，难以学习到更多时序信息。通过对时序信息进一步的挖掘，能够发现更多与神经元活动相关的隐含信息，这些信息有望对神经元的权重更新带来帮助。

同时，从前两章的讨论可以看出，神经元的亚阈值状态对脉冲神经元尤为重要。各脉冲神经元间的差异主要体现在神经元处于亚阈值状态时其膜上电位的动力学模型的不同。且通过巧妙地设计脉冲神经元处于亚阈值状态的动力学机制，可以将生物神经元的多种生理机制融入到脉冲神经元中。然而，由于脉冲神经元只有在膜上电位达到激活阈值时才会产生脉冲，因此脉冲神经元各层之间的信息传递只有在达到阈值的状态才能进行。这使得在反向传播过程中，网络训练的依据是神经元处于阈值或超阈值状态时传递的脉冲活动信息，其处于亚阈值状态的膜上电位活动信息对网络的反向传播过程是不可见的。这同样也是一种信息的浪费。

考虑到对时序信息和亚阈值状态的利用，本节设计了阈下记录机制。如图 6.2 所示，该记录从突触前脉冲和突触后脉冲的时序关系出发，记录在突触后神经元处于亚阈值状态时，突触前神经元发放的脉冲对突触后神经元的阈下记录的改变。

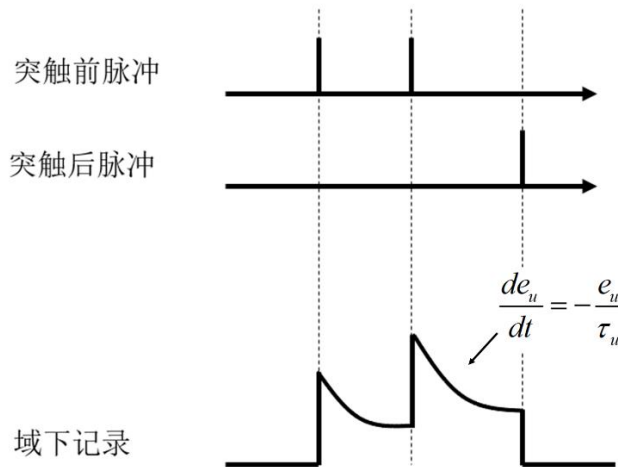


图 6.2 阈下记录机制

Fig. 6.2 Subthreshold Recording Mechanism

阈下记录是对脉冲神经元处在亚阈值状态时的输入记录。每一个不引起突触后神经元激活的脉冲输入都会增加突触后神经元的阈下记录，且这一记录会随时间逐渐衰减。而当突触后神经元激活时，阈下记录将被清零，并等待之后的突触前脉冲。这一动力学机制可形式化如公式（6.1）和公式（6.2）：

$$e_u(t) = \left(1 - \frac{1}{\tau_u}\right) \cdot E_u(t-1) + \frac{1}{\tau_u} \cdot X_t \quad (6.1)$$

$$E_u(t) = e_u(t) \cdot (1 - S_t) \quad (6.2)$$

其中， $e_u(t)$ 是阈下累积记录，其会因时间步 t 上的脉冲输入 X_t 累加，并随阈下记录时间常数 τ_u 逐渐衰减。 $E_u(t)$ 是实际阈下记录，其记录神经元在亚阈值状态下每个时刻的阈下累积记录，并会因神经元在时间步 t 上的脉冲发放 S_t 而清零。

阈下记录的大小能够有效地反映脉冲神经元的激活难易程度。越多、越密集的脉冲输入，能够造成越大的阈下权重累积记录。而神经元本身的激活会造成阈下记录的清零。因此，只有那些接受了密集的脉冲输入，且这些脉冲输入不能引发其激活的脉冲神经元，才能积攒到足够大的阈下记录。通过分析整个仿真时间 T 内各时间步上不同脉冲神经元的阈下记录，可以得到它们在时域上的激活难易度。

如公式 6.3 所示，整个仿真时间 T 内各神经元的最大阈下记录 UTS 信息，是神经元在时域上的重要特征。若将该特征通过恰当的方式融入到脉冲神经网络的训练过程中，则能够有效地实现神经元在时域上的负反馈调节机制。

$$UTS = \max_{1 \leq t \leq T} E_u(t) \quad (6.3)$$

6.3 基于空域负反馈调节的层内激活率

在前两章的设计中，针对脉冲神经元的讨论往往只细化到某层，对同层之内各脉冲神经元的情况并未过多讨论。这种细化程度针对传统的非脉冲神经元是足够的，因为其构成的网络仅在空间这一个维度上进行组织和延伸。然而脉冲神经元在组织成脉冲神经网络后，其学习过程不仅在空域上的层间关系进行前向和反向传播，同时还在时域上的时间步关系间进行前向和反向的传播。因此，对其在空域上进行进一步的细化能够对其训练提供帮助。

由于神经元在整个仿真过程中各时间步上的活动不同，因此，即便处于同一层的脉冲神经元，其在经过整个仿真时间的累计后也可能表现出截然不同的特性。同时，由于各神经元在层间的信息传递是通过脉冲信息来体现，因此对于同层的神经元，不妨以其

在整个仿真过程中总激活次数的多寡作为衡量其特性的标志。如公式 (6.4) 所示，以层内激活率 (Fire-Rate in Layer, FRL) 表示。

$$FRL_i = \frac{\sum_{1 \leq t \leq T} CF_i(t)}{\sum_{i \in LIF_l} \sum_{1 \leq t \leq T} CF_i(t)} \quad (6.4)$$

其中， FRL_i 表示神经元 i 的层内激活率， $CF_i(t)$ 表示神经元 i 在时间步 t 上的激活计数。 T 为整个仿真过程的总时间步长度， LIF_l 为网络第 l 层的所有 LIF 脉冲神经元。

由层内激活率的信息，可以看出同层之内各个脉冲神经元的实际活动情况。层内激活率越大的神经元，其活跃程度越高，同时也贡献了更多的层间信息。若将该特征通过恰当的方式融入到脉冲神经网络的训练过程中，则能够有效地实现神经元在空域上的负反馈调节机制。

6.4 脉冲神经网络上的时空负反馈调节机制

通过时域和空域两个维度的负反馈调节，可以在脉冲神经网络上高效地实现负反馈调节机制，从而加速训练过程。

首先，通过时域上的阈下记录信息，可以提炼出时域上的负反馈调节因子：阈下记录因子。为了更好地将阈下记录信息应用到负反馈调节中，以基线阈值的一半作为基准对脉冲神经元的激活难易程度进行评价。高于这一基准值的脉冲神经元可视为难激活神经元，低于这一基准值的脉冲神经元可视为易激活神经元。通过该设计可确定阈下记录因子如公式 (6.5) 所示：

$$F_{uts}^i = \frac{UTS_i}{V_{th}^0/2} \quad (6.5)$$

其中， F_{uts}^i 为神经元 i 的阈下记录因子， UTS_i 为该神经元的阈下记录。当 $F_{uts}^i > 1$ 时，该脉冲神经元为难激活神经元；当 $F_{uts}^i \leq 1$ 时，该神经元为易激活神经元。同时，由于脉冲在网络中的稀疏性，可能存在有神经元在整个仿真过程中都没有接受到脉冲，其阈下记录为 0。为了避免此时的异常更新，约定 $UTS_i = 0$ 时， $F_{uts}^i = 1$ 。由后文的公式 (6.7) 可知，此时，当前神经元的阈下记录因子不会对时空负反馈调节因子的大小产生影响。

其次，通过空域上的层内激活率信息，可以提炼出空域上的负反馈调节因子：层内激活率因子。为了更好地将层内激活率信息应用到负反馈调节中，以同层脉冲神经元的平均层内激活率作为基准对脉冲神经元的活跃程度进行评价。高于这一基准值的脉冲神经元可视为活跃神经元，低于这一基准值的神经元可视为不活跃神经元。通过该设计可

确定层内激活因子如公式 (6.6) 所示:

$$F_{frl}^i = \frac{FRL_i}{\sum_{0 \leq i < N_l} FRL_i / N_l} \quad (6.6)$$

其中, F_{frl}^i 为神经元 i 的层内激活因子, FRL_i 为该神经元的层内激活率, N_l 为网络第 l 层的所有脉冲神经元的个数。当 $F_{frl}^i > 1$ 时, 该脉冲神经元为活跃神经元; 当 $F_{frl}^i \leq 1$ 时, 该神经元为不活跃神经元。同时, 由于脉冲在网络中的稀疏性, 可能存在有神经元在整个仿真过程中都没有发送脉冲, 其层内激活率为 0。为了避免此时的异常更新, 约定 $FRL_i = 0$ 时, $F_{frl}^i = 1$ 。同样可由后文的公式 (6.7) 得知, 此时, 当前神经元的层内激活因子不会对时空负反馈调节因子的大小产生影响。

由阈下记录因子和层内激活因子, 可以定义从时域和空域两个维度对脉冲神经网络进行调节的时空负反馈调节因子 (Factor of Spatiotemporal Negative Feedback Regulation) 如公式 (6.7) 所示:

$$F_{STNFR}^i = F_{uts}^i \cdot F_{frl}^i \quad (6.7)$$

由公式 (4.29) 和公式 (6.7), 加入时空负反馈调节因子后, 反向传播过程中权重的更新如公式 (6.8) 所示:

$$\Delta w_i = \eta \cdot F_{STNFR}^i \cdot \frac{\delta Q}{\delta \theta^i} \quad (6.8)$$

相对于传统反向传播的权重更新算法, 带负反馈调节机制的更新算法在学习率 η 之外, 施加了时空负反馈调节因子 F_{STNFR}^i 的影响。根据脉冲神经元的阈下记录因子 F_{uts}^i 和层内激活因子 F_{frl}^i 的不同, 时空负反馈调节因子 F_{STNFR}^i 可呈现为如下几种情况:

(1) $F_{uts}^i > 1, F_{frl}^i > 1$ 时, 该神经元是难激发但活跃的神经元。作为一个难激发的神经元, 其依然能够在同层神经元里处于较平均值更加活跃的状态, 说明其是一个高价值的神经元, 且所在突触权重过低, 应尽快增加该神经元所在突触的权重。此时, $F_{STNFR}^i > 1$, 能放大反向传播时对权重的改变。

(2) $F_{uts}^i < 1, F_{frl}^i < 1$ 时, 该神经元是易激发但不活跃的神经元。作为一个易激发的神经元, 其却在同层神经元里处于较平均值更加不活跃的状态, 说明其是一个低价值的神经元, 且处于类似 Dropout 的状态。对于该神经元, 应避免进行较大的权重改变, 从而在节省计算资源的同时使网络更快收敛。此时, $F_{STNFR}^i < 1$, 能缩小反向传播时对权重的改变。

(3) $F_{uts}^i < 1, F_{frl}^i > 1$ 或 $F_{uts}^i > 1, F_{frl}^i < 1$, 该神经元是易激发的活跃神经元或难激发的不活跃神经元。此时, 该神经元的激活难易程度与其实际活跃情况一致, 反向传播时的

权重改变能够较好地实现这类神经元上的权重更新。此时, F_{STNFR}^i 接近 1, 能够较为真实地反映反向传播时权重的实际改变。

通过上述分情况讨论, 可以看出时空负反馈调节因子可以根据神经元本身在整个仿真过程中活动所带有的时空信息来判断其训练情况。加入时空负反馈因子后, 连接该神经元的突触可以根据神经元的特性来自适应地改变其上的权重, 从而实现基于时空信息的自适应负反馈权重更新。这一机制能够帮助网络更快训练。

6.5 本章小结

本章 6.1 节介绍了生物体内多样化的负反馈调节机制以及本章方案思想的来源。6.2 节介绍了基于时域负反馈调节机制的阈下记录, 能够刻画脉冲神经元激活的难易程度。6.3 节介绍了基于空域负反馈调节机制的层内激活率, 能够刻画脉冲神经元实际的活跃程度。6.4 节将时域与空域上的负反馈调节信息融合为时空负反馈调节因子, 能够从时空两个维度对脉冲神经网络进行负反馈调节, 加速网络的权重更新。

第7章 实验测试与结果分析

本章将依据第3章所抽象出的两个优化模型，通过在这两个组合优化问题上的求解效果，验证本文所提出的基于生物可信性的脉冲超启发算法框架的求解能力。本章将在两个任务上分别对本文的脉冲超启发算法的各模块进行消融实验和对比实验。通过消融实验，来验证本文根据仿生思想所提出的短时程突触可塑性机制和时空负反馈机制的有效性，证实它们能够改善作为高层决策结构的脉冲神经网络的决策能力。通过对比实验，将本文所提出的脉冲超启发算法框架与其他超启发算法进行对比，验证本文所设计的算法框架结构的有效性和其求解能力的优越性。

7.1 实验设置

7.1.1 实验环境

(1) 本文的所有实验均在如下所示的硬件环境中进行：

实验所用计算机为惠普 CPU 为单块 x86 架构 CPU，型号为 Intel Core i9-9900K，基准速度为 3.60GHz，核心数量为 8 核 16 线程。GPU 为单块 NVIDIA Turing 架构通用 GPU，型号为 NVIDIA GeForce RTX 2080Ti，显存容量为 11.0GB。RAM 容量为 16.0GB，速度为 2666MHz。硬盘为两块 NVMe 固态硬盘和一块 SATA 机械硬盘。两块 NVMe 固态硬盘型号均为 TOSHIBA KBG30ZMV，容量均为 256GB。SATA 机械硬盘型号为 TOSHIBA DT01ACA200，容量为 2TB。

(2) 本文的所有实验均在如下所示的软件环境中进行：

操作系统为 Windows 10 专业版 64 位系统，版本号为 20H2.19042.1237。实验所用的编程语言为 Python 3.8.11，集成开发环境为 Pycharm，深度学习框架为 PyTorch 1.8.2 LTS，脉冲学习框架为 SpikingJelly，依赖环境为 CUDA 11.2 和 cuDnn 8.1.0。

7.1.2 评价指标

为了评估本文所提出的脉冲超启发算法框架的求解效果，本文针对两个求解任务的特性，采用了欧几里得空间上的组合优化问题常用的两个评价指标：运行时间和距离成本。

运行时间反映了算法的求解速度。由于本文的实验都是在同样的环境下进行，因此

本文直接将本机的运行时间作为各算法的求解时间，借此对不同算法的求解用时进行评估。

作为欧几里得空间上的组合优化模型，无论是 TSP 求解任务还是 EVRPC 求解任务，其求得的解的总路径长度总能反映解的好坏。作为 NP-hard 问题，当问题规模较小时，能够通过精确算法求解出 TSP 任务和 EVRPC 任务的最优解。此时，与最优解差距越小的解，其质量越高。然而，面对大规模的复杂问题，精确算法难以在可接受时间内对问题进行求解。此时，在满足各项约束的条件下，可以认为求得的解的总路径长度越短，则其距离成本越小，也越贴近其理论上的最优解。因此，基于近似解总是右逼近最优解的特性，本文将距离成本作为评价解的好坏的指标，距离成本越小的解质量越高。

第 3 章的 EVRPC 优化模型虽然将 AGV 的最大运行时间作为优化目标，但 AGV 的运行时间可以等价转化为它运行的路径长度。在 EVRPC 任务中，AGV 的运行时间由在货架节点的搬运时间、在充电节点的充电时间和在节点间移动的移动时间组成。由于 AGV 是非异构的，因此不同 AGV 在同一货架节点进行搬运时所消耗的搬运时间相同。同时，由于它们的非异构性，所有 AGV 的电池容量和充电速率是相同的，因此它们在充电节点进行充电所消耗的时间仅与其耗电量有关。而耗电量与移动距离成正比，因此充电时间可等价转化为路径长度。同时，由于各 AGV 移动速度相同，且令它们在行驶时均为匀速，因此 AGV 在节点间的移动时间与移动距离成正比，也可转化为路径长度。综上，为了评价指标的统一，后续进行 EVRPC 任务的实验时，将最大运行时间转换为最大路径长度进行求解。

7.2 TSP 任务上的实验

7.2.1 数据集

为了充分验证本文提出的算法面对复杂问题和大规模数据时的求解效果，本文所用的 TSP 求解任务的数据集由生成数据组成。依据 Kool W 等人^[106]提出的生成算法形成生成规则，为不同规模的 TSP 任务生成大量实例。其中，当节点数为 20、50、100 时，可将其视为较小规模问题，利用生成规则为每个问题生成 10240 个实例；当节点数为 150、200 时，将其视为大规模问题，利用生成规则为每个问题生成 5120 个实例。

7.2.2 消融实验结果与分析

本小节对第 5 章提出的短时程突触可塑性机制和第 6 章提出的时空负反馈机制进行

消融实验，验证它们在脉冲超启发算法求解 TSP 任务上的有效性。

首先，本文由脉冲神经网络中不同层的脉冲发放情况对网络内神经元活动的平衡性进行评估。对比引入短时程突触可塑性机制前后脉冲神经网络各层的平均脉冲激活数如图 7.1 所示。可以直观地看出，在对 TSP 进行求解时，引入短时程突触可塑性前，脉冲神经网络的第一层脉冲激活数远高于其后第二层到第七层的激活数；而引入短时程突触可塑性之后，由于 STD 机制能够对短时间内接受到密集输入的脉冲神经元进行抑制，故网络中第一层脉冲神经元的异常激活现象受到遏制，整个网络内的脉冲激活的不平衡性得到了极大改善。

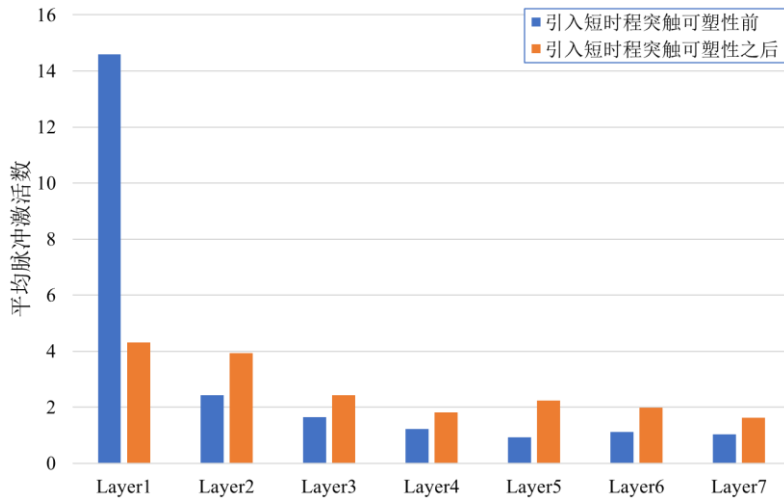


图 7.1 TSP 上的短时程突触可塑性机制对平均脉冲激活数的影响

Fig. 7.1 Effects of short-term synaptic plasticity mechanisms on average number of spike fires on TSP

对第 4 章中作为超启发算法高层决策机构的脉冲神经网络 SNN、引入第 5 章短时程突触可塑性机制的 STSP-SNN、引入第 6 章时空负反馈机制的 STNF-SNN 和同时引入这两个机制的生物可信性 (Biological Credibility, BC) 网络 BC-SNN 的脉冲发放情况进行统计，可以得到不同网络中各层的平均脉冲激活数如图 7.2 所示。

由图 7.2 可以看出，在 TSP 的求解任务上，直接将 SNN 作为高层决策结构会出现明显的脉冲发放不平衡问题，各层间脉冲激活数的平均值为 3.27，标准差为 4.64。STSP-SNN 由于短时程突触可塑性机制的存在，能够显著地改善脉冲激活的平衡性，其各层间脉冲激活数的平均值为 2.61，标准差为 0.98，与 SNN 相比不平衡性大大降低。而 STNF-SNN 中存在的时空负反馈机制能够使部分脉冲神经元陷入到类似 Dropout 的状态，因此也能对脉冲发放的平衡做出些许贡献，其各层间脉冲激活数的平均值为 2.96，标准差为 3.42，对 SNN 的脉冲激活的不平衡有略微的改善。融合了两个机制的 BC-SNN 表现最好，其各层间脉冲激活数的平均值为 2.88，标准差为 0.98，在显著改善脉冲激活

不平衡问题的同时，还维持了较高的平均脉冲激活数，这对提高网络的决策能力大有裨益。

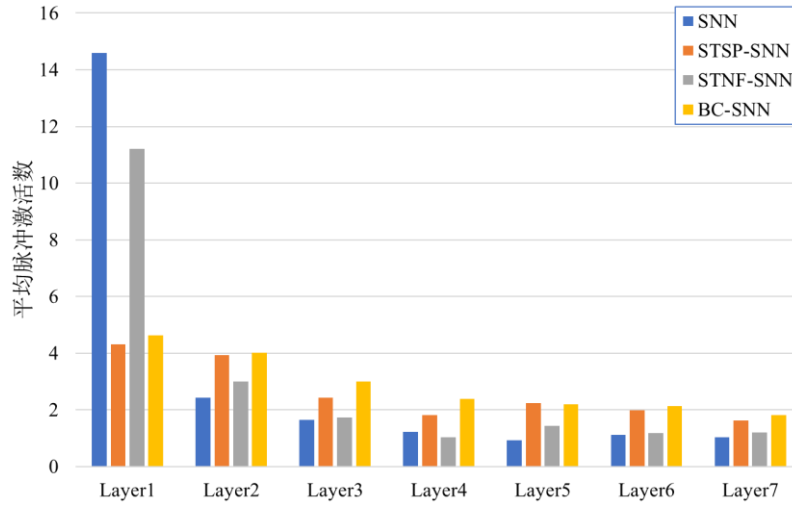


图 7.2 TSP 上的不同网络中各层的平均脉冲激活数

Fig. 7.2 Average number of spike fires at different layers in different network on TSP

之后，本文通过训练时的平均奖赏对高层决策结构的学习效率进行评估。如公式 (7.1) 所示，平均奖赏 \bar{R} 对一个回合中的每个动作的奖赏 r 进行记录，并统计它们的平均值。

$$\bar{R} = \frac{1}{N} \sum_{i=1}^N r_i \quad (7.1)$$

其中， N 为一个回合中所包含的动作或奖赏数量， r_i 表示第 i 个动作所对应的奖赏，奖赏 r_i 的计算方式由公式 (4.5) 给出。对于本文所设计的动作和奖赏，若 $\bar{R} > 0$ ，则说明在寻找最优解的过程中，所选择的大部分低层算子对解的更新会让解的质量变好；若 $\bar{R} < 0$ ，则说明在寻找最优解的过程中，所选择的大部分低层算子对解的更新会让解的质量变差。平均奖赏 \bar{R} 越高，则说明计算过程中的解向质量更优的更新次数越多。因此，训练后的高层决策的平均奖赏 \bar{R} 应保持在一个较高值，以便令解能够被快速优化。

为了验证时空负反馈机制对训练效率的提升，本文对比引入时空负反馈机制前后的网络在训练过程中的平均奖赏变化情况，如图 7.3 所示。可以看出，在 TSP 的求解任务上，引入时空负反馈机制的网络能够更快地将平均奖赏收敛到较高值，加快了网络的训练速度。

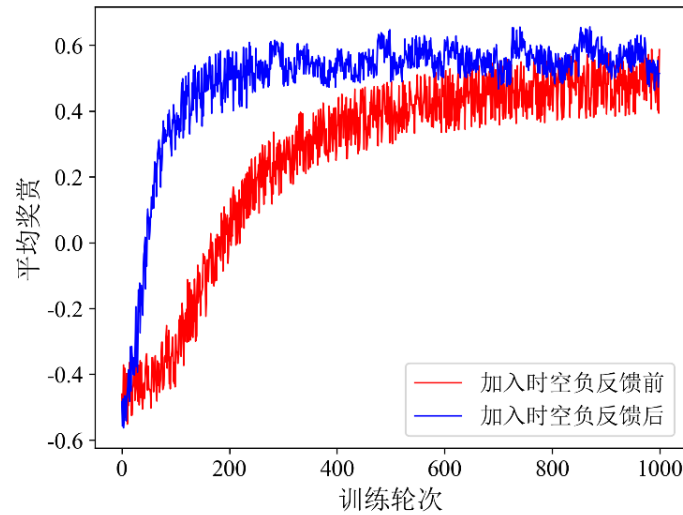


图 7.3 TSP 上的时空负反馈机制对训练速度的影响

Fig. 7.3 Effects of short-term synaptic plasticity mechanisms on average number of spike fires on TSP

对 SNN、STSP-SNN、STNF-SNN 和 BC-SNN 训练时的平均奖赏情况进行统计，可以得到不同网络在训练时的平均奖赏变化，如图 7.4 所示。

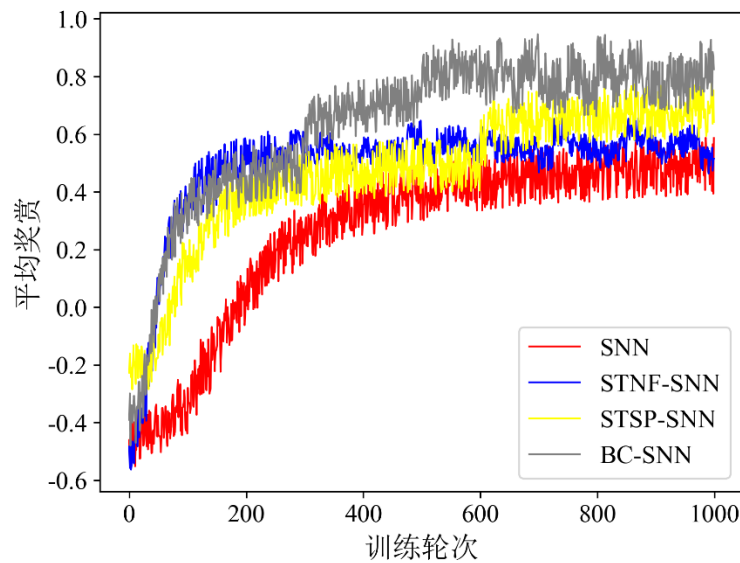


图 7.4 在 TSP 上训练时不同网络的平均奖赏变化

Fig. 7.4 Average reward variation across different networks during training on TSP

由图 7.4 可以看出，在 TSP 的求解任务上，直接对 SNN 进行训练时，平均奖赏收敛较慢，且稳定在一个较低值。STNF-SNN 由于时空负反馈机制的存在，能够快速将平均奖赏收敛至一个较高值。而 STSP-SNN 由于平衡了脉冲神经网络中各层的激活率，提升了网络整体的计算能力，因此虽然收敛速度较 STNF-SNN 稍慢，但最后可以使平均奖赏收敛到一个较高的值。融合了两个机制的 BC-SNN 在四个网络中表现最好，其在提升训练速度的同时，还能使得平均奖赏的数值更高，兼具了效率与效果上的优势。

为了评判各模块的实际求解效果，对比分别由上述四个网络作为高层决策结构的脉冲超启发算法（Spiking Hyper-heuristic, SHH）、基于短时程突触可塑性的脉冲超启发算法（Spiking Hyper-heuristic based on Short-term Synaptic Plasticity, STSP-SHH）、基于时空负反馈的脉冲超启发算法（Spiking Hyper-heuristic based on Spatio-temporal Negative Feedback, STNF-SHH）和基于生物可信性的脉冲超启发算法（Spiking Hyper-heuristic based on Biological Credibility, BC-SHH）的求解情况，如表 7.1 所示。由于求解过程只能体现前向计算的速度，并不能反映训练速度。为了更好地体现 STNF-SHH 在训练速度上的优势，将训练轮次设定为 600 轮，通过模型在不充分训练时的求解表现，来验证四个网络的各方面性能。

表 7.1 TSP 上的消融实验
Tab. 7.1 Ablation experiments on TSP

Instances	SHH		STSP-SHH		STNF-SHH		BC-SHH	
	cost	time(s)	cost	time(s)	cost	time(s)	cost	time(s)
TSP-20	3.88	691	3.87	424	3.85	612	3.85	367
TSP-50	5.81	938	5.79	554	5.78	802	5.78	467
TSP-100	8.17	1416	8.11	981	8.06	1262	8.02	637
TSP-150	21.47	2243	18.99	1377	18.09	1970	17.26	985
TSP-200	43.63	7084	42.28	3845	41.14	6361	38.74	3140

从表 7.1 中可以看出，加入短时程突触可塑性机制之后，STSP-SHH 在求解速度上较 SHH 显著提升，且在 TSP-150 和 TSP-200 这样的大规模问题上优势愈加明显。证明平衡网络间的脉冲激活之后，对网络的决策能力带来了提升，能够依据当前解的结构选择更适宜的算子，加快求解速度。然而其解的质量提升较小，与 STNF-SHH 和 BC-SHH 相比处于劣势，证明在训练到 600 轮时，STSP-SHH 的脉冲神经网络训练程度较差。而加入了时空负反馈调节机制之后，STNF-SHH 求得的解相较于 SHH 和 STSP-SHH 在距离成本上更低，说明此时 STNF-SHH 的脉冲神经网络在 600 轮时已训练到较好程度。这验证了通过带有时空信息的负反馈调节，对训练中不同类型的神经元实施不同速度的权重更新，能够有效加快网络的训练速度，使得 STNF-SNN 在相同的训练轮次下学习到更优的权重参数。但其求解速度相较于 STSP-SHH 和 BC-SHH 有所不足，证明其因为脉冲发放不平衡程度较高，导致网络整体的决策能力有所不足。而 BC-SHH 求得的解在距离成本和运行时间上都取得了比其他三个网络更好的表现，证明通过融合两种生物可信性机制，作为高层决策结构的网络的性能可以得到显著提升。

7.2.3 对比实验结果与分析

为了验证本文提出的 BC-SHH 框架的性能水平，本节将与最新的求解 TSP 问题的超启发算法进行对比，通过解的质量来验证基于生物可信性的脉冲神经网络作为超启发算法的高层结构的优越性。

Concord^[107]是一款针对 TSP 问题的专用求解器，其能够对给出的 TSP 问题求出其精确解。然而，其在大规模问题上的求解能力不足，所用时间往往是不可接受的。因此，本节仅在小规模问题上使用其进行求解。

SAD-GED 是 Saputra W 等人^[108]在 2020 年提出的一个自适应大洪水超启发算法。其通过一个自适应的选择机制对低层算子进行选择，并采用大洪水算法作为解的接受机制。该算法能够完成 TSP 和 VRP 上的求解任务，并获得良好的表现。当求解 TSP 问题时，SAD-GED 采用了 Hyflex 框架^[109]中提出的 15 个对应的低层求解算子。

CMS-HH 是 Zhao F 等人^[110]在 2021 年提出的一种合作多阶段超启发算法。其将多臂老虎机和中继杂交技术进行混合，作为选择机制对底层算子进行选择，以获得更大的未来收益。CMS-HH 同样在 TSP 和 VRP 的优化任务上都取得了良好表现。当求解 TSP 任务时，CMS-HH 采用了专为 TSP 求解任务设计的 13 个低层求解算子。

AQHH 是 Dantas A 等人^[111]在 2022 年提出的一个基于 Approximate Q-Learning 思想的超启发算法。其高层选择策略采用 DQN 来增强决策能力，能够在 TSP 和 VRP 上取得最优的结果。当求解 TSP 问题时，AQHH 同样采用了 Hyflex 框架中提出的 15 个低层求解算子。

为验证本文提出的 BC-SHH 的求解能力，将 BC-SHH 与上述四种求解算法的求解效果进行对比。其中，Concorde 仅在小规模 TSP 任务上进行求解。在小规模 TSP 和大规模 TSP 任务上的求解情况如表 7.2 和表 7.3 所示。

表 7.2 小规模 TSP 上的对比实验

Tab. 7.2 Comparative experiment on small-scale TSP

Algo	TSP-20		TSP-50		TSP-100	
	cost	time(s)	cost	time(s)	cost	time(s)
Concorde	3.84	313	5.70	846	7.76	4985
SAD-GED	3.85	243	5.85	2163	8.17	13508
CMS-HH	3.84	2362	5.72	2757	7.91	3961
AQHH	3.84	111	5.73	343	8.12	613
BC-SHH	3.84	341	5.71	434	7.86	646

表 7.3 大规模 TSP 上的对比实验
Tab. 7.3 Comparative experiment on big-scale TSP

Algo	TSP-150		TSP-200	
	cost	time(s)	cost	time(s)
SAD-GED	20.24	13384	42.63	36967
CMS-HH	17.93	4687	39.68	15605
AQHH	19.16	957	40.38	3351
BC-SHH	16.75	963	37.23	3132

如表 7.2 和表 7.3 所示, 本文所提出的 BC-HH 在距离成本和运行时间两个指标上均有良好的表现。尤其面对大规模问题时, 精确求解算法 Concorde 丧失求解能力, 采用启发式算法作为高层决策结构的 SAD-GED 和 CMS-HH 也存在明显的性能下降的问题。采用 DQN 的 AQHH 虽然能够较好地求解大规模 TSP 问题, 但由于其采用的为 ANN 网络, 其对计算资源的消耗高于本文提出的基于生物可信性的 SNN。因此在相同计算资源的约束下, 其解的质量不如 BC-SHH。

为了更加充分地对各实例的解的质量做出评价, 本文除了评价平均距离成本和平均计算时间之外, 还在所有实例上对各求解算法求得的解进行了统计, 对不同算法求得的解的优越关系进行了比较。若两个算法在同一实例上求得的解的距离成本不同, 则约定距离成本更低的那个解更优越; 若两个解的距离成本相同, 则约定求解所用时间更短的那个解更优越。如图 7.5 所示, 矩阵中每个元素的值代表在该规模下所有实例上, 纵坐标上的算法比横坐标上的算法求得的更优越的解在所有解中的占比。

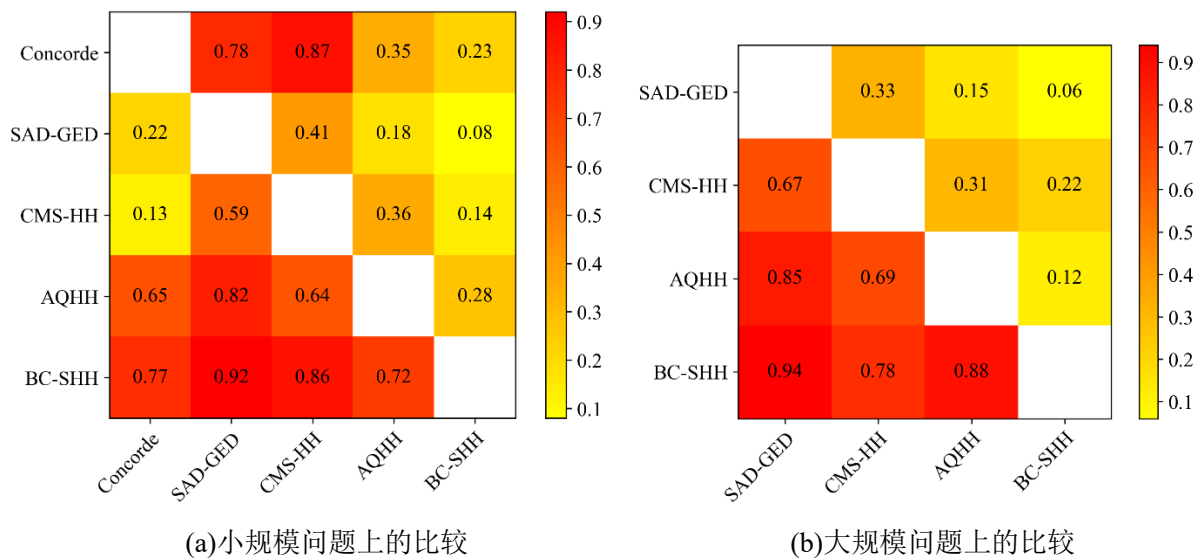


图 7.5 不同算法在 TSP 上的优越性比较

Fig. 7.5 Comparison of the Superiority of Different Algorithms on TSP

由图 7.5 可以看出, 本文所提出的算法在不论在小规模问题的求解上还是大规模问

题的求解上均能维持较好的性能，能够在大多数实例上求得比其他算法更加优越的解。

7.3 EVRPC 任务上的实验

7.3.1 数据集

本文所提出的 EVRPC 问题是电动车辆路径问题（Electric Vehicle Routing Problem, EVRP）的一个变体。由于 EVRP 相关领域的研究起步较晚，因此该领域尚未有较好的公开数据集。本文对 CVRPLIB^[112]中的 Set-A 数据集进行处理，用以生成与 EVRPC 任务相适应的数据集。

CVRPLIB 是一个开源的运筹学数据库，其中包含了多种 CVRP 任务上的标准数据集。Set-A 数据集中的节点分布较为均匀，能够更好地验证本文所建立的求解模型在各种情况下的表现，因此将其作为基准。本文所采用的处理方式随机挑选 Set-A 中原有实例上的部分节点作为充电节点，而将未被选择到的节点上的节点需求作为 AGV 在货架节点上的搬运用时。这种生成方式能够较好地将 CVRP 任务下的数据集转化为适用于 EVRPC 任务的数据集。借助这一生成方案，得到形如“E-nx-cy-kz”的各类型实例。其中，E 代表 EVRPC 任务，nx 表示有 x 个货架节点，cy 表示有 y 个充电站，kz 表示有 z 台 AGV 可供使用。由于 EVRPC 约束较多，在解空间中探索难度较大，因此将货架节点与充电站节点之和不大于 40（ $x + y \leq 40$ ）的实例视为小规模问题，其余视为大规模问题。同时，为了验证本文所提算法对大量实例的求解效果，每个类型下均生成 100 个实例。

7.3.2 消融实验结果与分析

本小节同样对短时程突触可塑性机制和时空负反馈机制进行消融实验，验证它们在脉冲超启发算法求解 EVRPC 任务上的有效性。

首先，本文对网络内神经元活动的平衡性进行评估。对比引入短时程突触可塑性机制前后脉冲神经网络各层的平均脉冲激活数，如图 7.6 所示。

由图 7.6 可以直观地看出，在对 EVRPC 进行求解时，引入短时程突触可塑性同样能够显著改善脉冲激活的不平衡性。

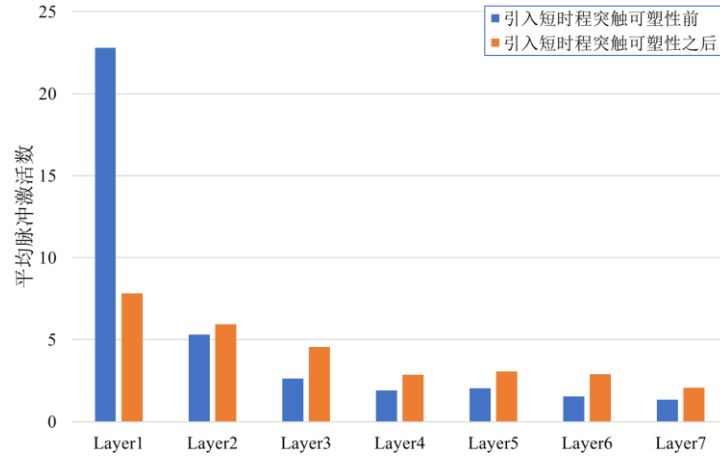


图 7.6 EVRPC 上的短时程突触可塑性机制对平均脉冲激活数的影响

Fig. 7.6 Effects of short-term synaptic plasticity mechanisms on average number of spike fires on EVRPC

对 SNN、STSP-SNN、STNF-SNN 和 BC-SNN 这四个网络的脉冲发放情况进行统计，可以得到不同网络中各层的平均脉冲激活数，如图 7.7 所示。

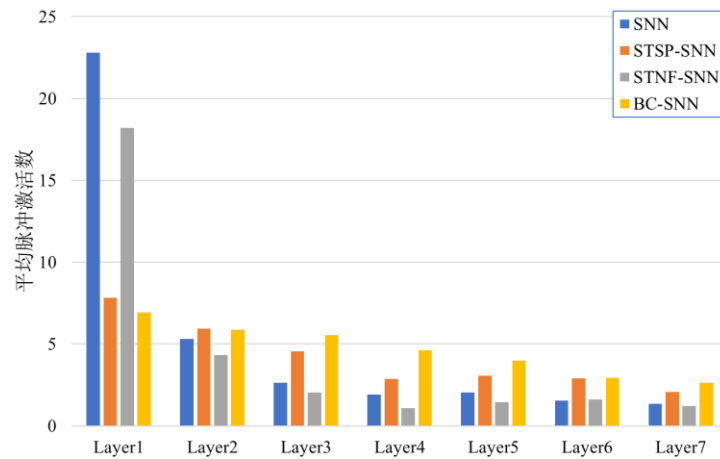


图 7.7 EVRPC 上的不同网络中各层的平均脉冲激活数

Fig. 7.7 Average number of spike fires at different layers in different network on EVRPC

由图 7.7 可以看出，在 EVRPC 的求解任务上，直接将 SNN 作为高层决策结构会出现明显的脉冲发放不平衡问题，各层间脉冲激活数的平均值为 5.36，标准差为 7.23。STSP-SNN 由于短时程突触可塑性机制的存在，能够显著地改善脉冲激活的平衡性，其各层间脉冲激活数的平均值为 4.17，标准差为 1.91，与 SNN 相比不平衡性大大降低。而 STNF-SNN 中存在的时空负反馈机制也能够对脉冲发放的平衡做出一定贡献，其各层间脉冲激活数的平均值为 4.27，标准差为 5.78，对 SNN 的脉冲激活的不平衡稍有改善。融合了两个机制的 BC-SNN 表现最好，其各层间脉冲激活数的平均值为 4.65，标准差为 1.46，在四个网络中脉冲激活不平衡程度最低，同时还维持了较高的平均脉冲激活数，

这能显著改善网络的决策能力。

为了验证时空负反馈机制对训练效率的提升，本文对比引入时空负反馈机制前后的网络在训练过程中的平均奖赏变化情况，如图 7.8 所示。可以看出，在 EVRPC 的求解任务上，引入时空负反馈机制的网络的平均奖赏的收敛速度同样优于未引入时空负反馈机制的网络，证明该机制在求解 EVRPC 任务时同样能够加快网络的训练速度。

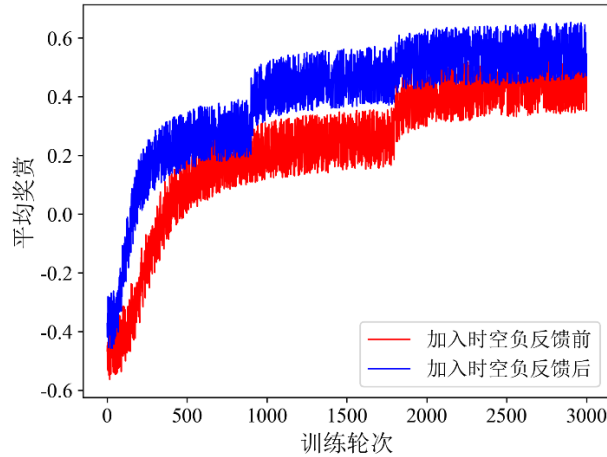


图 7.8 EVRPC 上的时空负反馈机制对训练速度的影响

Fig. 7.8 Effects of short-term synaptic plasticity mechanisms on average number of spike fires on EVRPC

对 SNN、STSP-SNN、STNF-SNN 和 BC-SNN 训练时的平均奖赏情况进行统计，可以得到不同网络在训练时的平均奖赏变化，如图 7.9 所示。

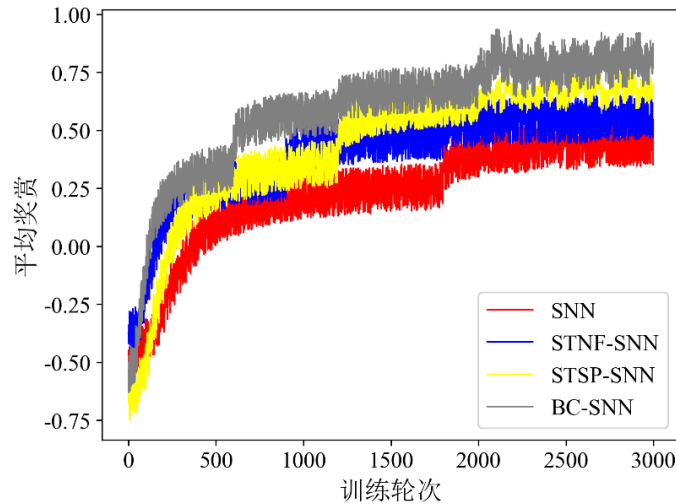


图 7.9 在 EVRPC 上训练时不同网络的平均奖赏变化

Fig. 7.9 Average reward variation across different networks during training on EVRPC

由图 7.9 可以看出，在 EVRPC 的求解任务上，直接对 SNN 进行训练时，平均奖赏收敛较慢，且稳定在一个较低值。STNF-SNN 由于时空负反馈机制的存在，能够快速将平均奖赏收敛至一个较高值。而 STSP-SNN 由于平衡了脉冲神经网络中各层的激活率，提升了网络整体的计算能力，因此虽然收敛速度较 STNF-SNN 稍慢，但最后可以使平均

奖赏收敛到一个更高的值。融合了两个机制的 BC-SNN 在四个网络中表现最好，其在提升训练速度的同时，还能使得平均奖赏的数值更高，兼具了效率与效果上的优势。

为了评判各模块的实际求解效果，本文对比了 SHH、STSP-SHH、STNF-SHH 和 BC-SHH 的求解情况，如表 7.4 所示。与 TSP 求解任务上的设置类似，同样对训练轮次进行限制，将其设定为 1000 轮，通过网络在不充分训练时的表现来展现四个网络在各方面的性能表现。

表 7.4 EVRPC 上的消融实验
Tab. 7.4 Ablation experiments on EVRPC

Instances	SHH		STSP-SHH		STNF-SHH		BC-SHH	
	cost	time(s)	cost	time(s)	cost	time(s)	cost	time(s)
E-n30-c2-k5	697	86	692	76	684	82	684	68
E-n31-c2-k5	673	101	665	86	662	96	663	77
E-n31-c2-k6	762	113	751	94	743	107	742	84
E-n32-c2-k5	791	134	778	106	775	127	771	95
E-n34-c2-k5	809	156	802	124	799	148	794	111
E-n35-c2-k5	732	212	683	154	672	201	669	137
E-n35-c2-k6	964	228	959	174	954	216	951	155
E-n36-c2-k5	753	240	751	196	740	227	740	174
E-n37-c2-k5	865	313	848	241	845	296	842	215
E-n37-c2-k6	873	335	854	265	840	317	831	236
E-n41-c3-k6	1002	429	995	340	982	406	977	303
E-n42-c3-k6	1069	479	1056	385	1048	453	1044	344
E-n42-c3-k7	1407	561	1372	466	1239	530	1157	415
E-n43-c3-k7	996	602	981	529	975	569	932	472
E-n45-c3-k7	1154	685	1138	614	1130	647	1124	548
E-n50-c3-k7	1022	853	1014	769	1013	806	1010	686
E-n51-c3-k7	1174	966	1172	866	1167	913	1167	772
E-n52-c3-k9	1196	1108	1080	976	1077	1047	1073	871
E-n56-c4-k9	1431	2040	1359	1761	1355	1927	1354	1570
E-n57-c4-k9	1250	2259	1158	1980	1042	2134	1036	1766

表 7.4（续表） EVRPC 上的消融实验
Tab. 7.4 (continued) Ablation experiments on EVRPC

Instances	SHH		STSP-SHH		STNF-SHH		BC-HH	
	cost	time(s)	cost	time(s)	cost	time(s)	cost	time(s)
E-n58-c4-k8	1437	2386	1406	2224	1399	2254	1388	1983
E-n59-c4-k9	1705	2681	1643	2498	1631	2532	1616	2227
E-n60-c4-k9	1491	3573	1447	3324	1425	3375	1414	2964
E-n61-c4-k9	1205	4110	1192	3808	1180	3882	1177	3396
E-n64-c5-k9	1293	5706	1264	4768	1262	5389	1259	4253
E-n75-c5-k10	1471	9448	1443	7992	1437	8923	1420	7127

从表 7.4 中可以看出，加入短时程突触可塑性机制之后，STSP-SHH 在求解速度上较 SHH 显著提升，这验证了短时程突触可塑性机制对脉冲网络进行激活平衡后，网络的决策能力得到提升，能够更快求得当前权重参数下的最优解。而加入了时空负反馈调节机制之后，STNF-SHH 求得的解相较于 SNN 在距离成本上更低，这验证了通过带有时空信息的负反馈调节在 EVRPC 求解任务上同样起效，使得 STNF-SNN 在相同的训练轮次下学习到更优的权重参数。而 BC-SHH 求得的解在距离成本和运行时间上都取得了比 SHH 更好的表现，证明通过融合两种生物可信性机制，作为高层决策结构的网络的性能可以得到显著提升。

7.3.3 对比实验结果与分析

EVRPC 问题可看作 EVRP 领域的一个特例，而 EVRP 相关的研究较为新颖，因此尚未有针对其开展的超启发算法相关的研究。然而，由于超启发算法本身与问题较为解耦，且许多 VRP 的变种问题在问题结构和解的更新上存在较大的共同之处，应用于其他类型 VRP 求解任务的超启发算法同样能够在 EVRP 领域表现良好。因此，本节将与 VRP 领域中最新的超启发算法进行对比，通过这些算法在 EVRPC 求解任务上的表现，来验证基于生物可信性的脉冲神经网络作为超启发算法的高层结构的优越性。

Cplex^[113]是 IBM 公司开发的一款专业的高性能求解器，可用于各种优化问题的建模、求解和分析，包括线性规划、混合整数规划、二次规划、约束优化和混合互补问题等。其能够为各种优化问题求得精确解，本文将其应用于 EVRPC 小规模问题上的求解。

7.2.3 小节中用到的 SAD-GED、CMS-HH 和 AQHH 同样能够求解 VRP 问题，并在其上表现良好。当求解 VRP 问题时，SAD-GED 和 AQHH 采用了 Hyflex 框架中提出的

12 个对应的低层求解算子，而 CMS-HH 采用了专为 VRP 求解任务设计的 10 个低层求解算子。

HH-ILS 是 Olgun B 等人^[114]在 2021 年提出的一种求解绿色车辆路径问题（Green Vehicle Routing Problem, GVRP）的超启发算法。GVRP 关注车辆在行驶过程中的油耗，这与 EVRPC 中的电量消耗有着相似之处。HH-ILS 的高层选择策略为迭代局部搜索（Iterative Local Search, ILS）和变邻域下降（Variable Neighborhood Descent, VND）的混合策略，低层算子为 17 个用于 GVRP 任务的求解算子。

为验证本文提出的 BC-SHH 的求解能力，将 BC-SHH 与上述五种求解算法的求解效果进行对比。其中，Cplex 仅在小规模 EVRPC 任务上进行求解。在小规模 EVRPC 和大规模 EVRPC 任务上的求解情况如表 7.5 和表 7.6 所示。

表 7.5 小规模 EVRPC 上的对比实验

Tab. 7.5 Comparative experiment on small-scale EVRPC

Instances		Cplex	SAD-GED	CMS-HH	HH-ILS	AQHH	BC-SHH
E-n30-c2-k5	cost	684	684	684	684	684	684
	time(s)	364	71	967	605	52	61
E-n31-c2-k5	cost	642	642	642	642	642	642
	time(s)	529	88	1061	644	61	69
E-n31-c2-k6	cost	692	701	692	692	692	692
	time(s)	619	92	1167	709	67	75
E-n32-c2-k5	cost	724	792	724	724	782	724
	time(s)	901	109	1276	754	77	85
E-n34-c2-k5	cost	752	811	752	763	805	752
	time(s)	1753	163	1515	847	94	99
E-n35-c2-k5	cost	630	735	641	664	732	630
	time(s)	2519	197	1645	895	120	123
E-n35-c2-k6	cost	925	969	935	948	962	930
	time(s)	3002	210	1809	985	136	139
E-n36-c2-k5	cost	728	751	749	750	743	734
	time(s)	4316	239	1960	1040	157	156
E-n37-c2-k5	cost	806	862	839	841	855	817
	time(s)	6191	287	2119	1096	196	192
E-n37-c2-k6	cost	826	876	833	855	872	831
	time(s)	7449	301	2331	1206	217	211

表 7.6 大规模 EVRPC 上的对比实验

Tab. 7.6 Comparative experiment on big-scale EVRPC

Instances		SAD-GED	CMS-HH	HH-ILS	AQHH	BC-SHH
E-n41-c3-k6	cost	1003	986	990	1003	980
	time(s)	676	3347	1536	306	271
E-n42-c3-k6	cost	1096	1048	1063	1089	1041
	time(s)	791	3581	1609	353	307
E-n42-c3-k7	cost	1388	1163	1172	1307	1159
	time(s)	804	3939	1770	430	371
E-n43-c3-k7	cost	997	948	947	976	930
	time(s)	931	4207	1851	496	421
E-n45-c3-k7	cost	1142	1095	1094	1138	1089
	time(s)	1255	4780	2017	597	489
E-n50-c3-k7	cost	1067	1011	1017	1032	1008
	time(s)	2511	6435	2462	807	612
E-n51-c3-k7	cost	1290	1179	1168	1274	1167
	time(s)	2862	6807	2559	923	689
E-n52-c3-k9	cost	1214	1073	1089	1194	1071
	time(s)	3255	7192	2657	1057	777
E-n56-c4-k9	cost	1468	1355	1358	1431	1354
	time(s)	5985	9337	4165	1907	1401
E-n57-c4-k9	cost	1275	1068	1067	1250	1035
	time(s)	6719	9811	4275	2146	1575
E-n58-c4-k8	cost	1388	1298	1308	1367	1291
	time(s)	7529	10302	4387	2213	1769
E-n59-c4-k9	cost	1803	1624	1649	1795	1612
	time(s)	8421	10809	4500	2513	1987
E-n59-c4-k10	cost	1436	1315	1329	1422	1303
	time(s)	8703	11890	4851	2886	2185
E-n60-c4-k9	cost	1513	1409	1415	1491	1402
	time(s)	9498	12465	5977	3317	2644
E-n61-c4-k9	cost	1287	1178	1185	1265	1172
	time(s)	10586	13058	6107	3548	3029

表 7.6 (续表) 大规模 EVRPC 上的对比实验

Tab. 7.6 (continued) Comparative experiment on big-scale EVRPC

Instances		SAD-GED	CMS-HH	HH-ILS	AQHH	BC-SHH
E-n64-c5-k9	cost	1224	1159	1170	1206	1143
	time(s)	16080	15620	8646	4199	3793
E-n75-c5-k10	cost	1439	1329	1360	1410	1317
	time(s)	45287	24345	13315	6923	6357

如表 7.5 和表 7.6 所示, 本文所提出的 BC-HH 在距离成本和运行时间两个指标上均有良好的表现。尤其面对大规模问题时, 精确求解算法 Cplex 丧失求解能力, 采用启发式算法作为高层决策结构的 SAD-GED、CMS-HH 和 HH-ILS 也存在明显的性能下降的问题。采用 DQN 的 AQHH 虽然能够在求解大规模 EVRPC 问题时保持一定的求解速度, 但由于其采用的为 ANN 网络, 其对计算资源的消耗高于本文提出的基于生物可信性的 SNN。因此在相同计算资源的约束下, 其解的质量不如 BC-SHH。

如图 7.10 所示, 同样在 EVRPC 任务上对不同算法进行解的优越性的评估。可以看出, 在 EVRPC 求解任务上, 本文所提出的算法同样在不同规模的任务上取得优越表现, 在大多数实例上求得的解比其他算法更加优越。

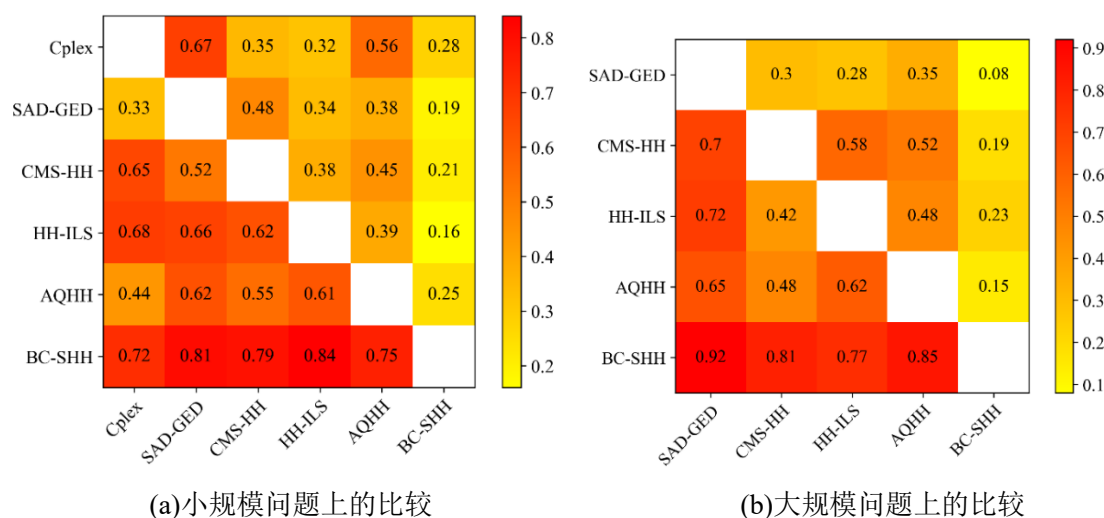


图 7.10 不同算法在 EVRPC 上的优越性比较

Fig. 7.10 Comparison of the Superiority of Different Algorithms on EVRPC

7.4 本章小结

本章 6.1 节介绍了进行实验的软硬件环境和评价指标。6.2 节和 6.3 节分别在 TSP 求解任务和 EVRPC 求解任务上介绍了实验的结果, 通过消融实验验证了各模块的有效性, 并通过对比实验说明本文所提出的算法的优越性。

第8章 总结与展望

组合优化是一类广泛存在于计算机科学、运筹学、数学以及工程等领域的优化问题。这些问题通常包含离散或组合特性，其解空间通常很大，求解效率往往难以令人接受。超启发算法相较于其他求解算法，具有与问题域解耦的特点。然而，现有的超启发算法的高层决策能力往往较弱。本文提出了基于脉冲神经网络的脉冲超启发算法框架，并基于生物可信性对其进行改进，实现了一个高效的脉冲超启发算法。

8.1 总结

本文对基于脉冲神经网络的超启发算法框架进行了研究。根据目前存在的问题，提出了基于生物可信性的脉冲超启发算法，主要工作如下：

(1) 整理组合优化问题求解领域和脉冲神经网络领域的国内外研究现状并对问题建模。

通过对大量国内外组合优化问题求解领域和脉冲神经网络领域文献的阅读与分析，总结该领域研究现状，从脉冲神经网络中蕴含的仿生潜力入手，针对目前方法中存在的问题，对脉冲超启发算法中存在的生物可信性进行研究。组合优化求解的必要过程是将实际问题抽象为由优化目标和约束条件组成的优化模型，因此本文从具有较高应用价值的欧几里得空间上的组合优化问题入手，基于 TSP 求解任务和 EVRPC 求解任务建立了优化模型，为后续的求解指导了优化方向。

(2) 提出基于脉冲神经网络的超启发算法框架

考虑到脉冲神经网络的高能效低功耗特性和其携带的丰富时空信息，本文提出了基于脉冲神经网络的超启发算法框架。首先基于之前的优化模型设计有针对性的低层求解算子，其直接对解进行更新。脉冲神经网络则是低层求解算子的高层决策结构，根据低层算子的求解情况选择适于当前求解进程的算子对当前解进行下一步更新。这一结构利用了脉冲神经网络的优良特性，能够较好解决当前超启发算法存在的高层选择策略决策能力不足的问题。

(3) 提出基于短时突触可塑性机制的改良脉冲神经元

脉冲神经网络具有低功耗的特性，这一特性主要是由于网络中脉冲的发放是稀疏的。然而，这同样会导致网络中各层间脉冲发放的极度不平衡。本文从仿生学角度入手，由生物体内的神经系统中广泛存在的短时程突触可塑性机制推导出改进后的 LIF 脉冲神

神经元动力学机制。通过在脉冲神经元上引入短时程突触可塑性机制，能够很好地改善网络中存在的脉冲发放不平衡问题。

（4）提出基于时空信息的负反馈调节机制

脉冲神经网络中丰富的时空信息能够更好地指导对低层算子的选择。然而，这也会增大脉冲神经网络的训练负担，其反向传播算法需要在时间和空间两个维度上来对网络进行权重更新。为了提升脉冲神经网络的学习效率，本文通过时域上的阈下记录信息和空域上的层内激活率信息，来对脉冲神经网络进行时空负反馈调节，从而加快了权重的更新速度，提升了训练效率。

（5）相关实验与结果分析

本文在 TSP 求解任务和 EVRPC 求解任务上对基于生物可信性的脉冲超启发算法的求解能力进行验证，证明了上述模块的有效性。并与现有的最新的超启发算法进行对比。通过实验证明，本文提出的方法能够求得质量更好的解，并在求解时间上具备优势，具有良好的实践价值。

8.2 未来工作

本文提出的方法相较于一些现有方法取得了一定的提升效果，但是仍然存在着一些问题，有待进一步解决。下面将介绍未来工作中需要改进的内容。

（1）算法的通用性问题

本文重点关注欧几里得空间上的组合优化问题，因此建立的优化模型和设计的低层算子均针对这一类问题进行适配。实际上，脉冲超启发算法具有强大的通用性潜力，在之后的研究中可针对其他类别的组合优化问题设计对应的低层算子，来探索脉冲超启发算法在通用性上的潜力。

（2）生物可信性问题

本文设计的脉冲超启发算法借助生物可信性对网络进行改良，并取得了性能上的实际提高。然而，随着计算神经科学的不断研究，生物脑中存在的众多活动机制被人们所发现，如何将这些机制应用于脉冲神经网络是一个值得深入发掘的问题。在之后的研究中，可深入探索赋予脉冲神经网络更多的生物可信性，以发掘其求解潜力。

参考文献

- [1] Schrijver A. Combinatorial optimization: polyhedra and efficiency[M]. Berlin: Springer, 2003:38-39.
- [2] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain[J]. Psychological review, 1958, 65(6): 386.
- [3] Rosenblatt F. Perceptron simulation experiments[J]. Proceedings of the IRE, 1960, 48(3): 301-309.
- [4] Hopfield J J. Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the national academy of sciences, 1982, 79(8): 2554-2558.
- [5] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [6] Maass W. Networks of spiking neurons: the third generation of neural network models[J]. Neural networks, 1997, 10(9): 1659-1671.
- [7] Roy K, Jaiswal A, Panda P. Towards spike-based machine intelligence with neuromorphic computing[J]. Nature, 2019, 575(7784): 607-617.
- [8] Papadimitriou C H, Steiglitz K. Combinatorial optimization: algorithms and complexity[M]. Courier Corporation, 1998: 3-7.
- [9] Korte B H, Vygen J, Korte B, et al. Combinatorial optimization[M]. Berlin: Springer, 2011: 385-414.
- [10] Woon S F, Rehbock V. A critical review of discrete filled function methods in solving nonlinear discrete optimization problems[J]. Applied mathematics and computation, 2010, 217(1): 25-41.
- [11] Liu Q, Li X, Liu H, et al. Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art[J]. Applied Soft Computing, 2020, 93(1): 106382.
- [12] Sergienko I V, Shylo V P. Problems of discrete optimization: Challenges and main approaches to solve them[J]. Cybernetics and Systems Analysis, 2006, 42(4): 465-482.

- [13] Fomin F V, Grandoni F, Kratsch D. A measure & conquer approach for the analysis of exact algorithms[J]. Journal of the ACM (JACM), 2009, 56(5): 1-32.
- [14] Laporte G, Mercure H, Nobert Y. An exact algorithm for the asymmetrical capacitated vehicle routing problem[J]. Networks, 1986, 16(1): 33-46.
- [15] Burrows C P. Distribution management: Mathematical modelling and practical analysis[J]. 1972, 21(3): 337-338.
- [16] Rao M R, Zionts S. Allocation of transportation units to alternative trips—A column generation scheme with out-of-kilter subproblems[J]. Operations Research, 1968, 16(1): 52-63.
- [17] Christofides N, Eilon S. An algorithm for the vehicle-dispatching problem[J]. Journal of the Operational Research Society, 1969, 20(3): 309-318.
- [18] Christofides N, Mingozzi A, Toth P. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations[J]. Mathematical programming, 1981, 20(1): 255-282.
- [19] Golden B L, Wasil E A, Kelly J P, et al. Fleet management and logistics[M]. New York: Springer, 1998: 33-56.
- [20] Kumar S N, Panneerselvam R. A survey on the vehicle routing problem and its variants[J]. 2012, 4(3): 66-74.
- [21] Clarke G, Wright J W. Scheduling of vehicles from a central depot to a number of delivery points[J]. Operations research, 1964, 12(4): 568-581.
- [22] Kindervater G A P, Savelsbergh M W P. Local search in combinatorial optimization[M]. Princeton University Press, 2018: 337-360.
- [23] Parejo J A, Ruiz-Cortés A, Lozano S, et al. Metaheuristic optimization frameworks: a survey and benchmarking[J]. Soft Computing, 2012, 16(3): 527-561.
- [24] Bianchi L, Dorigo M, Gambardella L M, et al. A survey on metaheuristics for stochastic combinatorial optimization[J]. Natural Computing, 2009, 8(2): 239-287.
- [25] Abd Elaziz M, Elsheikh A H, Oliva D, et al. Advanced metaheuristic techniques for mechanical design problems[J]. Archives of Computational Methods in Engineering, 2022, 29(1): 1-22.
- [26] Boussaïd I, Lepagnot J, Siarry P. A survey on optimization metaheuristics[J]. Information sciences, 2013, 237(1): 82-117.

- [27] Hussain K, Mohd Salleh M N, Cheng S, et al. Metaheuristic research: a comprehensive survey[J]. Artificial intelligence review, 2019, 52(8): 2191-2233.
- [28] Salhi S. Handbook of metaheuristics[J]. Journal of the Operational Research Society, 2014, 65(2): 320-320.
- [29] Adetunji K E, Hofsajer I W, Abu-Mahfouz A M, et al. A review of metaheuristic techniques for optimal integration of electrical units in distribution networks[J]. IEEE Access, 2020, 9(1): 5046-5068.
- [30] Meng Z, Yıldız A R, Mirjalili S. Efficient decoupling-assisted evolutionary/metaheuristic framework for expensive reliability-based design optimization problems[J]. Expert Systems with Applications, 2022, 205(1): 117640.
- [31] Sörensen K. Metaheuristics—the metaphor exposed[J]. International Transactions in Operational Research, 2015, 22(1): 3-18.
- [32] Osaba E, Yang X S, Diaz F, et al. A discrete firefly algorithm to solve a rich vehicle routing problem modelling a newspaper distribution system with recycling policy[J]. Soft Computing, 2017, 21(18): 5295-5308.
- [33] Trachanatzi D, Rigakis M, Marinaki M, et al. A firefly algorithm for the environmental prize-collecting vehicle routing problem[J]. Swarm and Evolutionary Computation, 2020, 57(1): 100712.
- [34] Goel R, Maini R. A hybrid of ant colony and firefly algorithms (HAFA) for solving vehicle routing problems[J]. Journal of Computational Science, 2018, 25(1): 28-37.
- [35] Alinaghian M, Naderipour M. A novel comprehensive macroscopic model for time-dependent vehicle routing problem with multi-alternative graph to reduce fuel consumption: A case study[J]. Computers & Industrial Engineering, 2016, 99(1): 210-222.
- [36] Altabeeb A M, Mohsen A M, Ghallab A. An improved hybrid firefly algorithm for capacitated vehicle routing problem[J]. Applied Soft Computing, 2019, 84(1): 105728.
- [37] Khalifehzadeh S, Fakhrzad M B. A modified firefly algorithm for optimizing a multi stage supply chain network with stochastic demand and fuzzy production capacity[J]. Computers & Industrial Engineering, 2019, 133(1): 42-56.

- [38] Tokhmehchi N, Makui A, Sadi-Nezhad S. A hybrid approach to solve a model of closed-loop supply chain[J]. *Mathematical Problems in Engineering*, 2015, 2015(1): 179102.
- [39] Geem Z W, Kim J H, Loganathan G V. A new heuristic optimization algorithm: harmony search[J]. *simulation*, 2001, 76(2): 60-68.
- [40] Guo Z, Shi L, Chen L, et al. A harmony search-based memetic optimization model for integrated production and transportation scheduling in MTO manufacturing[J]. *Omega*, 2017, 66(2): 327-343.
- [41] Chen S, Chen R, Gao J. A modified harmony search algorithm for solving the dynamic vehicle routing problem with time windows[J]. *Scientific Programming*, 2017, 2017(1): 1021432.
- [42] Ortiz-Bayliss J C, Terashima-Marín H, Conant-Pablos S E. Combine and conquer: an evolutionary hyper-heuristic approach for solving constraint satisfaction problems[J]. *Artificial Intelligence Review*, 2016, 46(3): 327-349.
- [43] Zhang C, Zhao Y, Leng L. A hyper-heuristic algorithm for time-dependent green location routing problem with time windows[J]. *IEEE Access*, 2020, 8(1): 83092-83104.
- [44] Amaya I, Ortiz-Bayliss J C, Rosales-Perez A, et al. Enhancing selection hyper-heuristics via feature transformations[J]. *IEEE Computational Intelligence Magazine*, 2018, 13(2): 30-41.
- [45] Zhou Y, Yang J J, Zheng L Y. Hyper-heuristic coevolution of machine assignment and job sequencing rules for multi-objective dynamic flexible job shop scheduling[J]. *IEEE Access*, 2018, 7(1): 68-88.
- [46] Lin J. Backtracking search based hyper-heuristic for the flexible job-shop scheduling problem with fuzzy processing time[J]. *Engineering Applications of Artificial Intelligence*, 2019, 77(1): 186-196.
- [47] Amaya I, Ortiz-Bayliss J C, Conant-Pablos S, et al. Hyper-heuristics reversed: Learning to combine solvers by evolving instances[C]. *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019: 1790-1797.
- [48] Garza-Santisteban F, Sánchez-Pámanes R, Puente-Rodríguez L A, et al. A simulated annealing hyper-heuristic for job shop scheduling problems[C]. *2019 IEEE congress on evolutionary computation (CEC)*. IEEE, 2019: 57-64.

- [49] Kieffer E, Danoy G, Brust M R, et al. Tackling large-scale and combinatorial bi-level problems with a genetic programming hyper-heuristic[J]. IEEE Transactions on Evolutionary Computation, 2019, 24(1): 44-56.
- [50] Drake J H, Swan J, Neumann G, et al. Sparse, continuous policy representations for uniform online bin packing via regression of interpolants[C]. Evolutionary Computation in Combinatorial Optimization: 17th European Conference, EvoCOP 2017, Amsterdam, The Netherlands, April 19-21, 2017, Proceedings 17. Springer International Publishing, 2017: 189-200.
- [51] Hu H, Zhang X, Yan X, et al. Solving a new 3D bin packing problem with deep reinforcement learning method[EB/OL]. <https://arxiv.org/abs/1708.05930>, 2017-8-20.
- [52] Burke E K, Hyde M R, Kendall G, et al. Automating the packing heuristic design process with genetic programming[J]. Evolutionary computation, 2012, 20(1): 63-89.
- [53] Sim K, Hart E, Paechter B. Learning to solve bin packing problems with an immune inspired hyper-heuristic[C]. ECAL 2013: The Twelfth European Conference on Artificial Life. MIT Press, 2013: 856-863.
- [54] López-Camacho E, Terashima-Marin H, Ross P, et al. A unified hyper-heuristic framework for solving bin packing problems[J]. Expert Systems with Applications, 2014, 41(15): 6876-6889.
- [55] Mohan J, Lanka K, Rao A N. A review of dynamic job shop scheduling techniques[J]. Procedia Manufacturing, 2019, 30(1): 34-39.
- [56] Chaurasia S N, Sundar S, Jung D, et al. An evolutionary algorithm based hyper-heuristic for the job-shop scheduling problem with no-wait constraint[C]. Harmony Search and Nature Inspired Optimization Algorithms: Theory and Applications, ICHSA 2018. Singapore: Springer Singapore, 2018: 249-257.
- [57] Garza-Santisteban F, Sánchez-Pámanes R, Puente-Rodríguez L A, et al. A simulated annealing hyper-heuristic for job shop scheduling problems[C]. 2019 IEEE congress on evolutionary computation (CEC). IEEE, 2019: 57-64.
- [58] Du Z, Ben-Dayana Rubin D D, Chen Y, et al. Neuromorphic accelerators: A comparison between neuroscience and machine-learning approaches[C]. Proceedings of the 48th International Symposium on Microarchitecture. IEEE, 2015: 494-507.

- [59] Deng L, Wu Y, Hu X, et al. Rethinking the performance comparison between SNNs and ANNs[J]. *Neural networks*, 2020, 121(1): 294-307.
- [60] Lee H, Kim C, Lee S, et al. An accurate and fair evaluation methodology for SNN-based inferencing with full-stack hardware design space explorations[J]. *Neurocomputing*, 2021, 455(1): 125-138.
- [61] Cao Y, Chen Y, Khosla D. Spiking deep convolutional neural networks for energy-efficient object recognition[J]. *International Journal of Computer Vision*, 2015, 113(1): 54-66.
- [62] Hebb D O. The organization of behavior: A neuropsychological theory[M]. Psychology press, 2005: 107-139.
- [63] Bohte S M, Kok J N, La Poutre J A. SpikeProp: backpropagation for networks of spiking neurons[C]. 2000 European Symposium on Artificial Neural Networks (ESANN). Springer, 2000: 419-424.
- [64] Sporea I, Grüning A. Supervised learning in multilayer spiking neural networks[J]. *Neural computation*, 2013, 25(2): 473-509.
- [65] Panda P, Roy K. Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition[C]. 2016 International Joint Conference on Neural Networks (IJCNN). IEEE, 2016: 299-306.
- [66] Lee J H, Delbruck T, Pfeiffer M. Training deep spiking neural networks using backpropagation[J]. *Frontiers in neuroscience*, 2016, 10(1): 508.
- [67] Zenke F, Ganguli S. Superspike: Supervised learning in multilayer spiking neural networks[J]. *Neural computation*, 2018, 30(6): 1514-1541.
- [68] van Rossum M C W. A novel spike distance[J]. *Neural computation*, 2001, 13(4): 751-763.
- [69] Bi G, Poo M. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type[J]. *Journal of neuroscience*, 1998, 18(24): 10464-10472.
- [70] Song S, Miller K D, Abbott L F. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity[J]. *Nature neuroscience*, 2000, 3(9): 919-926.
- [71] Bell C C, Han V Z, Sugawara Y, et al. Synaptic plasticity in a cerebellum-like structure depends on temporal order[J]. *Nature*, 1997, 387(6630): 278-281.

- [72] Diehl P U, Neil D, Binas J, et al. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing[C]. 2015 International joint conference on neural networks (IJCNN). IEEE, 2015: 1-8.
- [73] Rueckauer B, Lungu I A, Hu Y, et al. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification[J]. *Frontiers in neuroscience*, 2017, 11(1): 682.
- [74] Sengupta A, Ye Y, Wang R, et al. Going deeper in spiking neural networks: VGG and residual architectures[J]. *Frontiers in neuroscience*, 2019, 13(1): 95.
- [75] Patel K, Hunsberger E, Batir S, et al. A spiking neural network for image segmentation[EB/OL]. <https://arxiv.org/abs/2106.08921>, 2021-6-16.
- [76] Izhikevich E M. Which model to use for cortical spiking neurons?[J]. *IEEE transactions on neural networks*, 2004, 15(5): 1063-1070.
- [77] Hodgkin A L, Huxley A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]. *The Journal of physiology*, 1952, 117(4): 500.
- [78] Meunier C, Segev I. Playing the devil's advocate: is the Hodgkin–Huxley model useful?[J]. *Trends in neurosciences*, 2002, 25(11): 558-563.
- [79] Strassberg A F, DeFelice L J. Limitations of the Hodgkin-Huxley formalism: Effects of single channel kinetics on transmembrane voltage dynamics[J]. *Neural computation*, 1993, 5(6): 843-855.
- [80] Van Pottelbergh T, Drion G, Sepulchre R. Robust modulation of integrate-and-fire models[J]. *Neural Computation*, 2018, 30(4): 987-1011.
- [81] Hunsberger E, Eliasmith C. Training spiking deep networks for neuromorphic hardware[EB/OL]. <https://arxiv.org/abs/1611.05141>, 2016-9-16.
- [82] Izhikevich E M. Simple model of spiking neurons[J]. *IEEE Transactions on neural networks*, 2003, 14(6): 1569-1572.
- [83] Izhikevich E M. Neural excitability, spiking and bursting[J]. *International journal of bifurcation and chaos*, 2000, 10(06): 1171-1266.
- [84] Buzsaki G. *Rhythms of the Brain*[M]. Oxford university press, 2006: 254-257.
- [85] AL-Madi N A, Khader A T. The Traveling Salesman Problem as a Benchmark Test for a Social-Based Genetic Algorithm[J]. *Journal of Computer Science*, 2008, 4(10): 871.

- [86] Bharati T P, Kalshetty Y R. A hybrid method to solve travelling salesman problem[J]. IJIRCCE, 2016, 4(8): 15148-15152.
- [87] Wang Z, Guo J, Zheng M, et al. Uncertain multiobjective traveling salesman problem[J]. European Journal of Operational Research, 2015, 241(2): 478-489.
- [88] Bertsimas D J, Simchi-Levi D. A new generation of vehicle routing research: robust algorithms, addressing uncertainty[J]. Operations research, 1996, 44(2): 286-304.
- [89] Crama Y, Van De Klundert J, Spieksma F C R. Production planning problems in printed circuit board assembly[J]. Discrete Applied Mathematics, 2002, 123(1-3): 339-361.
- [90] Chang T S, Wan Y, Ooi W T. A stochastic dynamic traveling salesman problem with hard time windows[J]. European Journal of Operational Research, 2009, 198(3): 748-759.
- [91] Beggs J M, Plenz D. Neuronal avalanches in neocortical circuits[J]. Journal of neuroscience, 2003, 23(35): 11167-11177.
- [92] Shew W L, Clawson W P, Pobst J, et al. Adaptation to sensory input tunes visual cortex to criticality[J]. Nature Physics, 2015, 11(8): 659-663.
- [93] Munoz M A. Colloquium: Criticality and dynamical scaling in living systems[J]. Reviews of Modern Physics, 2018, 90(3): 031001.
- [94] Legenstein R, Maass W. Edge of chaos and prediction of computational performance for neural circuit models[J]. Neural networks, 2007, 20(3): 323-334.
- [95] Daoudal G, Debanne D. Long-term plasticity of intrinsic excitability: learning rules and mechanisms[J]. Learning & memory, 2003, 10(6): 456-465.
- [96] Zucker R S, Regehr W G. Short-term synaptic plasticity[J]. Annual review of physiology, 2002, 64(1): 355-405.
- [97] Fioravante D, Regehr W G. Short-term forms of presynaptic plasticity[J]. Current opinion in neurobiology, 2011, 21(2): 269-274.
- [98] Tamas G, Buhl E H, Somogyi P. Massive autaptic self-innervation of GABAergic neurons in cat visual cortex[J]. Journal of Neuroscience, 1997, 17(16): 6352-6364.
- [99] Bacci A, Huguenard J R, Prince D A. Functional autaptic neurotransmission in fast-spiking interneurons: a novel form of feedback inhibition in the neocortex[J]. Journal of Neuroscience, 2003, 23(3): 859-866.

- [100] Bacci A, Huguenard J R, Prince D A. Modulation of neocortical interneurons: extrinsic influences and exercises in self-control[J]. Trends in neurosciences, 2005, 28(11): 602-610.
- [101] Saada R, Miller N, Hurwitz I, et al. Autaptic excitation elicits persistent activity and a plateau potential in a neuron of known behavioral function[J]. Current Biology, 2009, 19(6): 479-484.
- [102] Bacci A, Huguenard J R. Enhancement of spike-timing precision by autaptic transmission in neocortical inhibitory interneurons[J]. Neuron, 2006, 49(1): 119-130.
- [103] Deleuze C, Pazienti A, Bacci A. Autaptic self-inhibition of cortical GABAergic neurons: synaptic narcissism or useful introspection?[J]. Current opinion in neurobiology, 2014, 26(1): 64-71.
- [104] Jiang M, Zhu J, Liu Y, et al. Enhancement of asynchronous release from fast-spiking interneuron in human and rat epileptic neocortex[J]. PLoS biology, 2012, 10(5): e1001324.
- [105] Straiker A, Dvorakova M, Zimmowitch A, et al. Cannabidiol inhibits endocannabinoid signaling in autaptic hippocampal neurons[J]. Molecular Pharmacology, 2018, 94(1): 743-748.
- [106] Kool W, van Hoof H, Welling M. Attention, Learn to Solve Routing Problems![C]. 2019 7th International Conference on Learning Representations (ICLR). OpenReview, 2019: 1-25.
- [107] Applegate D, Cook W, Rohe A. Chained Lin-Kernighan for large traveling salesman problems[J]. INFORMS Journal on Computing, 2003, 15(1): 82-92.
- [108] Saputra W, Muklason A, Rozaliya B Z H. Self Adaptive Learning–Great Deluge Based Hyper-heuristics for Solving Cross Optimization Problem Domains[C]. 2020 17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE, 2020: 570-575.
- [109] Ochoa G, Hyde M, Curtois T, et al. Hyflex: A benchmark framework for cross-domain heuristic search[C]. Evolutionary Computation in Combinatorial Optimization: 12th European Conference, EvoCOP 2012, Málaga, Spain, April 11-13, 2012. Proceedings 12. Springer Berlin Heidelberg, 2012: 136-147.

- [110] Zhao F, Di S, Cao J, et al. A novel cooperative multi-stage hyper-heuristic for combination optimization problems[J]. Complex System Modeling and Simulation, 2021, 1(2): 91-108.
- [111] Dantas A, Pozo A. The Impact of State Representation on Approximate Q-Learning for a Selection Hyper-heuristic[C]. Intelligent Systems: 11th Brazilian Conference, BRACIS 2022, Campinas, Brazil, November 28–December 1, 2022, Proceedings, Part I. Cham: Springer International Publishing, 2022: 45-60.
- [112] Ivan Lima, Daniel Oliveira, Eduardo Queiroga, et al. CVRPLIB - Capacitated Vehicle Routing Problem Library[EB/OL]. <http://vrp.atd-lab.inf.puc-rio.br/index.php/en>, 2023-3-23
- [113] Garcia-López J M, Ilchenko K, Nazarenko O. Optimization lab sessions: major features and applications of IBM CPLEX[C]. Optimization and Decision Support Systems for Supply Chains. Springer International Publishing, 2017: 139-150.
- [114] Olgun B, Koç Ç, Altıparmak F. A hyper heuristic for the green vehicle routing problem with simultaneous pickup and delivery[J]. Computers & Industrial Engineering, 2021, 153(1): 107010.

致谢

攻读硕士学位期间的论文项目情况

参加项目：

1. 辽宁省教育厅基础科研重点项目：类脑的神经形态感知计算研究(No. LJKZ0008)

获奖情况：

1. 获得 2020-2021 年度东北大学硕士研究生一等学业奖学金
2. 获得 2022-2023 年度东北大学硕士研究生二等学业奖学金