

基于改进 DFN 的巨星座遥控任务规划方法

安芷函, 刘 蕾, 房 皓

(中国电子科技集团公司 第十五研究所, 北京 100083)

摘要: 针对巨星座发展给星地任务规划带来的挑战, 开展了卫星遥控任务规划方法研究; 传统方法难以应对卫星数量庞大、任务动态性强与规模巨大的新需求; 为实现高效可靠的星地指令调度, 对跟踪计划所生成的遥控任务, 结合指令类型与卫星状态等多重约束, 采用深度前馈神经网络与贪心算法相结合的改进方法进行任务规划; 通过设立仿真实验, 将结果与其他算法进行对比, 结果表明该方法在任务分配成功率和计算时间效益方面具有优势, 能够满足巨星座大规模任务规划的实际工程需要。

关键词: 巨型星座; 卫星遥控; 任务规划; 神经网络; 深度前馈神经网络

A Deep Feedforward Neural Network and Greedy Algorithm-Based Approach for Task Scheduling of Satellite-Ground Remote Control in Mega-Constellations

AN Zhihan, LIU Lei, FANG Hao

(The 15th Research Institute, China Electronics Technology Group Corporation, Beijing 100083, China)

Abstract: In response to the challenges posed by mega-constellations in satellite-to-ground mission planning, a method for remote control task scheduling is investigated. Traditional approaches struggle to meet the requirements of large satellite numbers, highly dynamic tasks, and enormous task volumes. To achieve efficient and reliable satellite command scheduling, an improved approach integrating deep feedforward network and greedy algorithm is proposed, which processes remote control tasks derived from tracking plans while considering constraints such as command types and satellite status. Simulation experiments compared with other algorithms demonstrate that the proposed method achieves higher task computation success rate and greater computational efficiency, thus fulfilling the practical engineering needs of large-scale mission planning in mega-constellations.

Keywords: giant constellation; satellite remote control; mission planning; neural network; DFN

0 引言

在空天地一体化信息网络的建设中, 大规模低轨卫星通信系统(以下称“巨星座”)凭借其全球覆盖、低传输延迟和高系统容量, 正逐步成为未来通信、导航与遥感服务的关键基础设施。随着卫星数量呈指数级增长, 从传统的数十颗增至当今星座的成千上万颗, 系统的运行模式发生了根本性变化。这一变化使得传统以地面为中心的任务规划方法在应对海量卫星、动态任务与有限星地资源之间的协同调度时, 面临前所未有的计算与时效性挑战。

在实际应用中, 低轨卫星和地面站的遥控任务的大致流程^[1]为根据任务信息、地面站天线情况(是否有卫星遥控任务)、卫星属性信息(卫星轨道预报、卫星有

效载荷状态等)和相关约束条件(能量约束、星载存储器容量约束等)进行星地遥控任务规划, 形成地面站天线对卫星的遥控方案, 实现地面站和卫星资源的最大化利用^[2]。

目前现有关于卫星任务规划问题方面的研究, 文献[3]提出一种关于任务和资源先验信息的调度算法, 通过调度系统能够预先评估任务的紧急度和价值, 并感知天线的状态变化, 从而将复杂的全局优化问题分解为更易处理的局部优化问题。显著提高了遥测任务的满足率, 同时降低了调度计算的复杂度和时间消耗, 增强了大规模卫星网络资源调度的可行性。文献[4]提出一种“基于事件触发的滚动式任务管控”新模式, 使得卫星任务管理系统能够打破传统的固定周期规划模式, 在应急事件发生时立即启动响应, 动态插入应急任务, 并

收稿日期: 2025-09-17; 修回日期: 2025-11-03。

作者简介: 安芷函(1999-), 男, 硕士。

引用格式: 安芷函, 刘 蕾, 房 皓. 基于改进 DFN 的巨星座遥控任务规划方法[J]. 计算机测量与控制, 2026, 34(2): 203-210.

滚动调整原有计划, 构建了“应急—常态”一体化的管控体系, 大幅提升了卫星系统对自然灾害等应急任务的快速响应能力和执行效率。文献 [5] 对单星地一体化自主任务调度规划研究现状进行梳理与分析后延展至多星领域, 综述了多星地面离线协同任务调度规划、多星星上在线协同任务调度规划、多星地一体协同任务调度规划研究现状。文献 [6] 利用星间链组网, 提出了一种适应于断续猝发特征星间网络的路由任务规划算法。该算法以遥感卫星成像任务规划结果为驱动, 结合卫星网络拓扑结构的时变性特点, 改进基于快照序列的拓扑划分方法, 降低路由切换频率, 减少链路切换代价损失。并将卫星与地面站之间的路由看作低轨遥感星座路由的一部分, 在选择星地最短路由路径的同时实现星地建链总时长最大化。文献 [7] 提出了一种算法, 用于设计可提供高时间分辨率观测的卫星星座, 以研究植被和水环境中的日周期变化。关键要求是在目标区域实现每小时重访, 这需要多卫星星座。该算法确定满足观测要求的最佳卫星数量及其轨道参数, 同时考虑性能指标和部署成本之间的权衡。与随机优化方法相比, 该方法提供了更好的物理直观。所提出的方法被应用于 SOLVE 任务, 该任务旨在部署配备高光谱传感器的小型卫星群, 以监测植被和水过程。文献 [8] 提出了一种基于动态规划算法对卫星的区域目标成像多卫星协作任务规划方法。文献 [9] 提出了一种针对有人值守无人操作的遥感卫星移动接收站的运行特点和任务需求利用任务节点矩阵驱动数据接收流程, 自动化控制和调度移动站内各系统完成遥感卫星数据接收工作的软件设计方法; 文献 [10] 出了通过采用云服务架构及天地安全互联技术, 实现用户终端以调用服务的方式远程快速使用资源, 通过重构快响空间地面控制流程, 实现快响力量的统一控制的地面控制系统设计; 文献 [11] 提出了规划架构设计在任务规划研究中的重要性, 阐述和分析了应急卫星任务规划、低轨组网卫星任务规划一体化、基于机器学习的任务规划等技术挑战。

当前研究在应对上述挑战时存在明显局限: 首先, 传统中心化规划方法由于计算复杂度高, 在资源冲突消解时面临困难; 其次, 多数现有规划策略在建模时未充分考虑星上资源与任务间的复杂耦合关系, 也缺乏对多目标 (如覆盖率、数据时效性等) 的协同优化; 最后, 基于简单规则或静态分解的算法虽计算轻量, 却难以适应高动态任务场景, 在规划成功率与效率上存在瓶颈。这些不足严重制约了巨星座系统性能的充分发挥。

因此, 为提升星地任务规划在巨星座场景下的计算成功率和计算效率, 需研究新型高效任务规划算法。本文在分析现有算法不足的基础上, 提出一种深度前馈神经网络 (DFN) 结合贪心算法的改进策略, 以应对大规

模约束下任务规划的实时性与可靠性需求, 并通过仿真实验验证其有效性。

1 深度前馈神经网络 (DFN)

1.1 神经网络

神经网络 (Neural Network) 是一种受生物大脑启发的机器学习模型^[12], 通过多层连接的“神经元”模拟人脑的信息处理机制, 用于识别复杂模式 (如图像、语音、文本等)。其核心思想是: 从数据中自动学习特征表示, 而非依赖人工设计规则。大多数神经网络是由 3 个模块构成, 即输入层、隐藏层及输出层。

作为一种机器学习程序或模型, 它以类似于人脑的方式做出决策, 通过使用模仿生物神经元协同工作方式的过程来识别现象、权衡利弊并得出结论。每个神经网络都由多个节点层或人工神经元组成一个输入层、一个或多个隐藏层和一个输出层。神经网络模型通过信息输入是否反馈, 可以分为两种, 即反馈神经网络和前馈神经网络。常见的前馈神经网络包括卷积神经网络 (CNN, convolutional neural network)^[13]、全连接神经网络 (FCN, fully connected network)、生成对抗神经网络 (GAN, generative adversarial network) 等。常见的反馈神经网络包括循环神经网络 (RNN, recurrent neural network)^[14] 和玻尔兹曼机 (Boltzmann Machine) 等。

1.2 深度前馈神经网络

深度前馈神经网络^[15-16] (DFNN, deep feedforward neural network), 也称为多层感知机 (MLP, multilayer perceptron), 是深度学习领域中最核心的模型架构之一。其通过一系列隐藏层的非线性变换, 能够以极高的精度逼近任何复杂的连续函数, 从而在图像识别、自然语言处理、金融预测等众多领域取得了卓越的成功。尽管深度前馈网络具有强大的表示能力, 其训练过程却面临着显著的挑战。最常见的就是梯度消失/爆炸问题, 即在基于误差反向传播的训练过程中, 梯度需要从输出层逐层反向传播至输入层。对于深层网络, 梯度的连乘计算极易导致其值变得过小或过大, 从而使底层网络的参数难以得到有效更新。

通过传统算法进行约束, 有效缓解梯度消失, 保证了梯度信号的有效传播, 每一层在预训练阶段都学习到了输入数据的某种有意义的结构, 为后续的微调提供了一个极佳的起始点。因此本文采用了贪心算法结合训练的策略, 与深度前馈网络相结合。

2 问题建模

2.1 卫星遥控任务规划

卫星的任务规划从本质上讲是一种非线性规划方法^[17], 其核心是贝尔曼最优性原理。该原理指出, 最优策略的解可通过满足贝尔曼最优方程来获得, 一个状

态的最优价值等于在该状态下所有可能行动中, 能立即获得的奖励与后续状态折扣价值之和的最大值。即整体最优, 则其每一步的局部也必然最优。根据最优性原理的结论, 多级最优决策问题可以化为多个单级最优决策问题, 这为导出递推方程提供理论支持。

根据业务场景, 低轨卫星星座星地任务规划分为跟踪任务规划和遥控任务规划两种, 跟踪任务规划是指对地面天线与低轨卫星之间进行建链规划并生成跟踪计划, 遥控任务规划是指在跟踪计划的基础上, 进行路由规划并生成遥控计划, 以便进行遥控指令上注。遥控计划的生成由上注任务驱动, 即在有任务输入进入任务规划系统时, 基于已有跟踪计划, 规划计算出满足任务需求(注入数据量、截止时间、时效性最高等)的星地链路及时段, 本文在跟踪计划的基础上, 开展巨型低轨星座的星地遥控任务规划问题分析, 并针对具体问题, 开展巨型低轨星座星地遥控任务规划方法研究。

卫星遥控任务规划问题可以建模为一个约束优化问题, 基本定义设计如下:

$S = \{S_1, S_2, \dots, S_N\}$ 为卫星合集, 其中 N 为卫星数量。

$G = \{G_1, G_2, \dots, G_M\}$ 为地面站集合, 其中 M 为地面站数量。

$C = \{C_1, C_2, \dots, C_K\}$ 为指令集合, 其中 K 为指令数量。

$W = \{W_1, W_2, \dots, W_L\}$ 为指令集合, 其中 L 为窗口数量。

卫星遥控任务规划面临关键在于在时间约束和资源约束的情况下达成最大化目标函数。

1) 时间约束:

任务规划中最重要的资源即时间资源, 需要确保任务在时间窗口中的规划符合规划要求, 同时时间窗口彼此间也存在不同情况的时间约束条件。

约束优化问题的约束条件是对于每个指令 $c_k = (s_k, d_k, \tau_k, \delta_k)$, 其中 s_k 是目标卫星, d_k 是指令持续时间, τ_k 是最晚上传时间, δ_k 是指令优先级。时间约束条件为即对确定进行规划的指令的安排的最后时间不能晚于放入规划窗口的最后时间, 也不能晚于本指令要求的最后上传时间, 数据模型表示如下:

$$t_{start} + d_k \leq \min(\tau_k, t_{window.end}) \quad (1)$$

2) 资源约束:

资源约束通常是指在任务规划问题中除时间以外的其他资源限制, 在本文的卫星遥控任务规划问题上, 受限于地面站和卫星简练要求, 同一时刻地面站天线只能跟随一个卫星, 数学模型表示如下:

$$\forall \omega_i, \omega_j \in W, \text{if } g_i = g_j \\ \text{then}[t_{start,i}, t_{end,i}] \cap [t_{start,j}, t_{end,j}] = \emptyset \quad (2)$$

同时要求指令必须分配给卫星可用的跟踪窗口。

在优化目标中设置加权目标, 即不同权重系数下将几个优化目标进行总体规划, 最终优化目标中的最大化目标函数为:

$$max \sum_{k=1}^K (\alpha \cdot x_k + \beta \cdot \frac{d_k}{D_{max}} \cdot x_k + \gamma \cdot \delta_k \cdot x_k) \quad (3)$$

其中: $x_k \in 0, 1$ 表示 c_k 是否被成功安排。

α, β, γ 为权重系数, D_{max} 为最大指令持续时间。

2.2 深度前馈神经网络

深度前馈神经网络核心目的是需按照近似函数 f , 神经网络通过学习和调整参数逼近最理想的函数 f , 通常信息从输入层进入开始, 通过一系列中间计算, 最终达到输出层的单向流动过程, 网络通常包含多个隐藏层, 将每一层计算串联起来^[18], 即:

$$\begin{aligned} a^{[0]} &= x \\ z^{[1]} &= W^{[1]} a^{[0]} + b^{[1]} \\ a^{[1]} &= g^{[1]}(z^{[1]}) \\ z^{[2]} &= W^{[2]} a^{[1]} + b^{[2]} \\ a^{[2]} &= g^{[2]}(z^{[2]}) \\ &\vdots \\ z^{[L]} &= W^{[L]} a^{[L-1]} + b^{[L]} \\ \hat{y} &= a^{[L]} = g^{[L]}(z^{[L]}) \end{aligned} \quad (4)$$

将每一层计算串联得到从输入到输出的完整映射函数。

3 基于 DFN 结合传统贪心算法的遥控任务规划

神经网络模型训练流程如图 1 所示。首先将巨型星座系统的星地跟踪计划和指令进行数据分析和训练, 得到神经网络模型, 后通过调度和冲突检测进行巨型星座星地遥控任务规划。

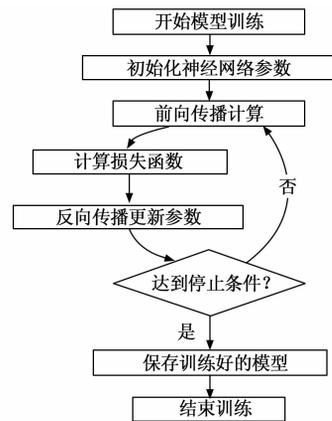


图 1 神经网络模型训练流程

输入层接收原始数据, 隐藏层是核心计算层, 每层包含多个神经元, 对输入数据进行加权求和非线性变换。输出层生成最终结果。

为了将问题建模与神经网络模型联系起来, 将卫星

遥控任务规划问题转化为特征空间中的表示。对每个卫星遥控指令—时间窗口 (c_k, w_j) 定义特征向量为：

$$f_{kl} = [d_k, \delta_k, \Delta t_{\text{deadline}}, \Delta t_{\text{window}}, r_{\text{duration}}, \Delta t_{\text{start}}, I_{\text{sufficient}}]^T \quad (5)$$

其中： d_k 表示卫星遥控指令要求的持续时间， δ_k 表示指令优先级：

$$\Delta t_{\text{deadline}} = \tau_k - t_{\text{window, start}} \quad (6)$$

为卫星执行遥控任务截止时间余量：

$$\Delta t_{\text{window}} = t_{\text{window, end}} - t_{\text{window, start}} \quad (7)$$

为地面站和卫星建链窗口持续时间：

$$r_{\text{duration}} = \frac{d_k}{\Delta t_{\text{window}}} \quad (8)$$

表示持续时间比率：

$$\Delta t_{\text{start}} = t_{\text{window, start}} - t_{\text{current}} \quad (9)$$

表示窗口开始时间偏移：

$$I_{\text{sufficient}} = I(d_k \leq \Delta t_{\text{window}} \wedge t_{\text{window, start}} + d_k \leq \tau_k) \quad (10)$$

作为可行性指示器。神经网络架构采用深度前馈神经网络进行调度决策。输入层特征向量为：

$$x = f_{kl} \in R^7 \quad (11)$$

隐藏层使用 3 个全连接层。第一隐藏层为：

$$h_1 = \sigma(W_1 x + b_1) \quad (12)$$

其中： $W_1 \in R^{128 \times 7}$ 为权重矩阵， $b_1 \in R^{128}$ 作为偏置向量， σ 作为激活函数。

由于巨星座任务规划符合调度决策的稀疏性特点，同时为了保持梯度能力，缓解梯度消失问题，使用 ReLU 激活函数。相比较相比 Sigmoid/Tanh，计算复杂度更低，也可以保证大部分神经元在特定模式下激活。 σ 作为 ReLU 激活函数：

$$\sigma(z) = \max(0, z) \quad (13)$$

第二隐藏层为：

$$h_2 = \sigma(W_2 h_1 + b_2) \quad (14)$$

其中：

$$W_2 \in R^{64 \times 128} \quad (15)$$

$$b_2 \in R^{64} \quad (16)$$

第三个隐藏层为：

$$h_3 = \sigma(W_3 h_2 + b_3) \quad (17)$$

其中：

$$W_3 \in R^{32 \times 64} \quad (18)$$

$$b_3 \in R^{32} \quad (19)$$

输出层为 $\hat{y} = \phi(W_4 h_3 + b_4)$ 。

其中：

$$W_4 \in R^{1 \times 32} \quad (20)$$

$$b_4 \in R \quad (21)$$

ϕ 表示激活函数：

$$\phi(z) = \frac{1}{1 + e^{-z}} \quad (22)$$

在每个隐藏层后加入批量归一化：

$$h_i^{BN} = BN(h_i) = \gamma \cdot \frac{h_i - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta \quad (23)$$

其中： μ ， σ^2 分别表示批次的均值和方差， γ ， β 为可学习的缩放和偏移参数， ϵ 是小常数用于数值稳定性。

为了防止过拟合，添加丢弃层：

$$h_i^{\text{drop}} = \text{Dropout}(h_i^{BN}, p) \quad (24)$$

其中： p 为丢弃率。

建立损失函数为：

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \|\theta\|_2^2 \quad (25)$$

其中： $y_i \in 0, 1$ 作为真实标签， \hat{y}_i 为预测概率， λ 是 L2 正则化系数， θ 是所有模型参数。

建模中使用 Adam 优化：

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (26)$$

其中： \hat{m}_t 是偏差校正的一阶矩估计， \hat{v}_t 是偏差校正的二阶矩估计， η 作为学习率， ϵ 作为数值稳定性常数。

由于需要避免冲突问题，对窗口和时间区间执行冲突检测函数：

$$\text{Conflict}(\omega_i, t_s, t_e) = \exists \omega_j \in O \text{ such that } g_j = g_i \text{ and } [t_s, t_e] \cap [t_{j, \text{start}}, t_{j, \text{end}}] \neq \varnothing \quad (27)$$

当检测到冲突时，执行冲突检测调整策略，即窗口重搜索策略，在当前任务的可用窗口集合中寻找下一个最优窗口；考虑到新鲜度优先级重计算，考虑时间推移对优先级的影响，动态更新任务优先级更新函数如下：

$$P_{\text{dynamic}}(t_i, t) = P_{\text{global}}(t_i) \cdot e^{-\lambda \cdot (\tau_i - t)} \quad (28)$$

在极端情况下，回退部分低优先级任务的分配，为高优先级任务腾出资源。

具体流程伪代码如下。

Algorithm 1: DFN-GA

Input: 任务集合 T , 资源窗口集合 W , 训练好的 DFN 模型。

Output: 调度方案 S

- 1: //阶段一: DFN 优先级计算
- 2: for each $t_i \in T$ do
- 3: $P_{\text{max}} \leftarrow 0$
- 4: for each $w_j \in W$ do
- 5: if Compatible(t_i, w_j) then
- 6: $\varphi \leftarrow \text{ExtractFeatures}(t_i, w_j)$
- 7: $P_{ij} \leftarrow f_{\theta}(\varphi)$ // DFN 前向传播
- 8: $P_{\text{max}} \leftarrow \max(P_{\text{max}}, P_{ij})$
- 9: end if
- 10: end for
- 11: $P_{\text{global}(t_i)} \leftarrow P_{\text{max}}$
- 12: end for

```

13://阶段二:贪心调度
14: $T_{sorted} \leftarrow \text{Sort } T \text{ by } P_{global} \text{ descending}$ 
15: $S \leftarrow \emptyset$  // 已调度任务集合
16: $R \leftarrow W$  // 可用资源集合
17:for each  $t_i \in T_{sorted}$  do
18:    $scheduled \leftarrow \text{false}$ 
19:    $W_{candidate} \leftarrow \text{GetCandidateWindows}(t_i, R)$ 
20:    $W_{sorted} \leftarrow \text{Sort } W_{candidate} \text{ by start time ascending}$ 
21:
22:   for each  $w_j \in W_{sorted}$  do
23:     if not  $\text{Conflict}(t_i, w_j, S)$  then
24:       // 分配任务到窗口
25:        $S \leftarrow S \cup \{(t_i, w_j)\}$ 
26:        $R \leftarrow R \setminus \{w_j\}$  // 标记窗口为占用
27:        $scheduled \leftarrow \text{true}$ 
28:       break
29:     end if
30:   end for
31:
32:   if not  $scheduled$  then
33:     // 动态调整:尝试抢占低优先级任务
34:      $success \leftarrow \text{TryPreemption}(t_i, S, R)$ 
35:     if not  $success$  then
36:        $\text{MarkAsFailed}(t_i)$ 
37:     end if
38:   end if
39: end for
40: return  $S$ 

```

本文使用神经网络为 3 个隐藏层, 第 1 隐藏层 128 个神经元, 实现特征的充分展开和交互, 允许网络学习复杂的特征组合, 第 2 个隐藏层 64 个神经元, 实现特征选择和抽象特征降维, 保留最重要的调度相关特征, 第三个隐藏层 32 个神经元, 完成高级特征融合, 进一步压缩信息, 提取调度决策的核心特征, 输出层为而分配问题, 后对每一层都进行归一化处理, 这一设计基于信息瓶颈理论和特征层次提取原理, 与卫星任务规划问题的特性高度匹配, 选择 128-64-32 是在准确率和泛化能力间取得最佳平衡, 在计算效率上参数量适中, 适合实时调度需求, 同时相比更宽更深的网络, 具有更好的泛化性能。

为防止过拟合, Dropout 为 0.3, 损失函数方面, 由于卫星任务优先级预测本质上是一个二元分类问题, 即任务可成功调度和不可成功调度两种, 在该方向上使用 BCELoss, 可以直接优化预测概率与真实标签的差异, 通过样本加权处理类别不平衡, 提供平滑的梯度信号控制训练稳定性。为确定 BCELoss 的性能表现, 先单独对样本数据使用消融实验检验性能, 检验效果如表

1 所示。

表 1 损失函数性能

| | 调度准确率/% | 收敛速度 | 鲁棒性 |
|------------|---------|------|-----|
| BCELoss | 86.3 | 快 | 高 |
| Focal Loss | 85.8 | 中 | 中 |
| MSE | 83.1 | 慢 | 低 |

可见在当前环境下 BCELoss 在各项指标上表现最优, 更适合卫星调度的概率预测任务。

优化器方面, Adam 稀疏特征获得更大更新, 稠密特征(如时间差)获得更稳定的更新, 相比较 SGD 收敛速度更快, 训练效率更高, 由于其对初始学习率不敏感, 更加适配工程应用, 调度数据集上对比优化器结果如表 2 所示。

表 2 损失函数性能

| | 最终准确率/% | 收敛轮数 | 训练时间/min |
|--------------|---------|------|----------|
| Adam | 86.3 | 35 | 12.3 |
| SGD+Momentum | 84.1 | 72 | 24.8 |

可以验证 Adam 优化器更适配当前工程应用。采用学习率为 0.001 的 Adam 优化器。

算法空间复杂性为:

$$7 \times 128 + 128 + 128 \times 64 + 64 + 64 \times 32 + 32 + 32 \times 1 + 1 + 128 * 2 + 64 * 2 + 32 * 2 = 11841 \quad (29)$$

特征维度适合大规模问题。

时间复杂性为:

$$O(7 \times 128 + 128 \times 64 + 64 \times 32 + 32 \times 1) = O(11937) \quad (30)$$

调度决策方面为 $O(K \cdot L \cdot C)$, 其中 C 是冲突检测成本。

使用本神经网络算法理论期望误差界定为:

$$R(\theta) \leq \hat{R}(\theta) + O\left(\sqrt{\frac{d_{vc}}{N}}\right) \quad (31)$$

其中: $R(\theta)$ 为期望风险, $\hat{R}(\theta)$ 为经验风险, d_{vc} 为 Vapnik-Chervonenkis 维度, N 为训练样本数。

为了保证监控验证损失最小值, 当验证损失停止下降时降低学习率, 学习率减低因子为 0.1, 训练 50 epoch。训练后并不单独使用神经网络对所有任务进行规划, 而是通过神经网络对任务合集中任务进行优先级排序, 后使用贪心算法对总任务合集进行规划。在进行训练后生成 200 个任务合集验证训练成果。

本方法创新将复杂的调度问题重构为机器学习问题, 利用神经网络从历史数据中学习调度策略, 实现了从“基于规则”到“基于学习”的范式转变, 提出了神经网络优先级的贪心调度框架, 结合了深度学习的表示学习能力和传统调度算法的高效性, 使用神经网络负责

学习复杂的优先级评估函数后贪心算法负责高效的资源分配和冲突解决。传统算法使用显式冲突检测，计算复杂度为 $O(N^2)$ 。本方法使用神经网络隐式学习冲突模式，通过优先级排序避免潜在冲突，计算复杂度为 $O(N) + O(N \log N)$ ，即前向传播复杂度加排序复杂度。在算法动态适应方面，传统算法需要根据新地面站上线，卫星状态变化，紧急指令插入和天气条件等问题重新求解，而使用神经网络可以通过前向传播进行快速调整。通过建立损失函数实现多目标优化，与传统算法的单一目标相比在现实使用中更具有优势。同时，在约束满足学习方面提出了隐式约束学习方法，无需显式编码约束进行学习可行解：

$$P[f_{\theta}(x) \in F] \geq 1 - \epsilon \tag{32}$$

其中： F 为可行解空间。

虽然使用黑盒神经网络，但通过特征重要性分析和可视化工具提供决策解释，即 $I_i = \frac{\partial \hat{y}}{\partial x_i}$ ，同时可以持续使用新的数据进行学习和训练，即：

$$\theta_{t+1} = \theta_t - \nabla \mathcal{L}(\theta_t; D_{new}) \tag{33}$$

4 仿真验证

4.1 模拟仿真

为验证算法性能和规划效果，本文通过构建巨星座仿真，通过巨星座卫星轨道数据和地面站坐标得到星地可见弧段，再通过星地可见弧段分析得到巨星座跟踪计划，通过工程仿真数据构建仿真遥控任务集对巨星座跟踪计划进行遥控计划的任务规划，具体流程如图 2 所示。

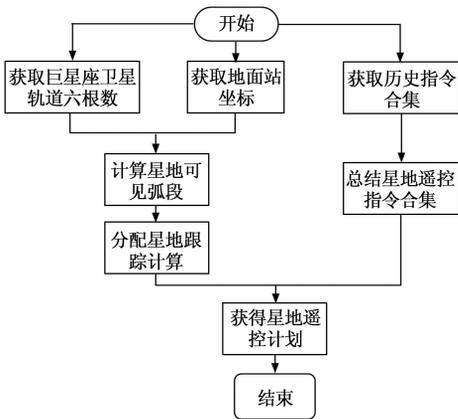


图 2 巨星座遥控计划生成流程

卫星轨道六根数^[19]（也称为开普勒轨道根数或开普勒元素）是用来唯一确定一个人造卫星（或任何天体）绕中心天体（通常是地球）运行轨道的 6 个参数。它们基于开普勒行星运动定律，描述了一个理想的、未受扰动的二体问题中的椭圆轨道。由于在实际工程中，在太空中，卫星轨道会受到各种摄动力（如地球非球形引力、大气阻力、太阳光压、日月引力等）的影响，导

致这些根数会随时间缓慢变化（称为轨道摄动，实际的轨道根数通常都附带一个特定的历元时刻，表示这些参数在该时刻是准确的^[20]）。为了减小计算量和方便仿真，使用 30 分钟作为弧段，在本弧段时间内忽略轨道摄动。根据 Starlink 一期部署目标为约 1 584 颗卫星（550 km 轨道），其他商业星座数量通常在数百颗到数千颗规模，因此仿真使用 STARLINK 数据库中抽取 1 500 颗在轨低轨卫星作为巨星座卫星轨道六根数仿真数据库，规模代表性达到了 94.7%，组成模拟巨星座数据库的样本量符合标准，由于调度问题的计划复杂性随卫星数量呈超线性增长，设置过多的卫星样本数量实验周期过长。

巨星座中地面站数量较多，星地指令发送传输通过地面站进行星地互联，出于对全国地理分布的考虑，取中国城市北京和成都，再取东南西北最有代表性的城市作为各个方位地面站，共 6 个地面站。地面站坐标如表 3 所示。

表 3 地面站坐标

| | 北京 | 成都 | 抚远 | 三亚 | 喀什 | 漠河 |
|----|-------|-------|-------|-------|-------|-------|
| 经度 | 116.4 | 104.0 | 134.2 | 109.3 | 75.6 | 122.2 |
| 纬度 | 39.9 | 30.7 | 48.2 | 18.1 | 39.3 | 53.3 |
| 海拔 | 43.5 | 503 | 40 | 100 | 1 100 | 600 |

通过分析取得的 1 500 颗在轨卫星轨道六根数和地面站坐标，基于 TLE 数据和 Skyfield 库进行可行性分析获得星地可见弧段。

由于同一天线同一时刻只能执行一个上注任务、同一卫星同一时刻只能执行一个上注任务。为避免冲突，通过星地可见弧段时间对地面进行跟踪计划规划，仿真中使用动态规划算法。由于低轨卫星速度较快，选取时间不同星地可见弧段也有较大的随机性，因此规划的跟踪计划数量具有较大的波动性，平均每个地面站约有 400 个跟踪计划。

指令工程建模方面，通过对一段时间内历史遥控指令分析，归纳总结每类指令的约束特点及分布规律，基于此进行指令的仿真及指令集构建。生成遥控指令种类和持续时间与历史遥控指令合集相同，同时统计历史遥控指令合集，规划总结历史遥控指令约束特点和分布规律，为了防止生成与历史合集完全一致的任务合集，生成的指令数量遵循以种类为区别的正态分布。指令种类分为 3 个大类组合形成的各种类型，即紧急程度、功能类型、卫星类型。其中紧急程度的要求如表 4 所示。

表 4 地面站坐标

| 紧急程度 | 具体要求 |
|------|--------------|
| 紧急指令 | 需在卫星下次过境内执行 |
| 常规指令 | 24 小时内完成 |
| 计划指令 | 执行窗口大于 24 小时 |

功能类型分为遥控下载、软件更新、姿态调整、载荷控制、轨道维持及应急处理 6 个大类。卫星类型适配方面包括通信卫星、遥感卫星、科学实验卫星、技术验证卫星 4 个卫星种类。各指令生成的比例遵循过去工程数据库中的指数数量的对数正态分布。

为了方便计算和仿真,对指令上传时间补齐到秒,即指令时间有秒为单位的小数时自动补一秒后取整秒。

4.2 其他算法分析对比

本文使用了贪心算法、遗传算法和模拟退火算法,并与 DFN 结合传统算法的贪心算法进行性能和规划效果对比。

贪心算法的特点是每一步选择中都采取当前状态下最好或最优的选择。该算法的核心在于,在决策过程的每一阶段,都选择当前看来最优的局部解^[21],并期望通过这种局部最优选择的迭代,最终导向全局最优解。其优势在于算法简单、计算高效,但适用性依赖于问题是否具备贪心选择性质与最优子结构。本文使用贪心算法重点在于最终规划的遥控指令数量。

遗传算法 (GA, genetic algorithm) 是一种模拟自然界生物进化流程的优化算法。它通过模拟“自然选择”和“遗传机制”(如基因交叉、变异、选择)来解决复杂的优化问题,尤其适用于传统数学方法难以处理的非线性、多峰值、大规模组合优化问题^[22]。核心思想是类似于生物进化原理:适应环境的个体更易生存并繁衍后代。种群 (Population) 可以映射为组候选解;染色体 (Chromosome) 映射为解的编码 (如二进制串、浮点数);基因 (Gene) 映射为解中的单个参数;适应度 (Fitness) 映射为解的质量评价函数。具体流程如图 3 所示。

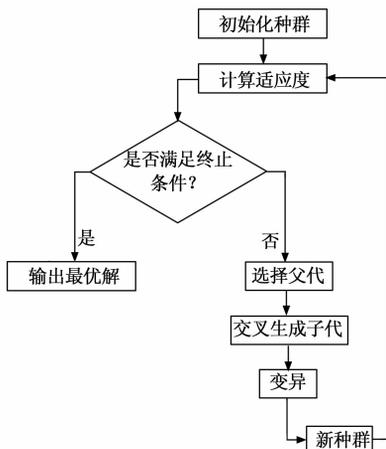


图 3 遗传算法流程图

本文进行遗传算法仿真时,使用每个个体表示指令的分配顺序(即一个排列)。设定适应度函数为:适应度 = 分配指令数 × 10 000 + 总提前时间,确保优先最大化分配指令数,其次优化提前时间,遗传操作锦标赛选

择 (Tournament size = 3) 部分匹配交叉。变异为索引位置交换 (Indpb = 0.05)。算法参数中,种群大小为 30,代数 50,交叉概率为 0.7,变异概率为 0.2,使用 DEAP 框架高效实现遗传算法,记录算法进化过程以便分析同时限制每个文件的处理时间。

模拟退火算法^[23] (SA, simulated annealing) 是一种受金属退火过程启发的全局优化算法,由 Kirkpatrick 等人于 1983 年提出。它通过模拟固体物质在高温下缓慢冷却(退火)时原子逐渐达到能量最低态的物理过程,解决组合优化问题(如旅行商问题、调度问题)和复杂函数极值搜索问题,特别适合处理多局部最优解的场景。具体流程如图 4 所示。

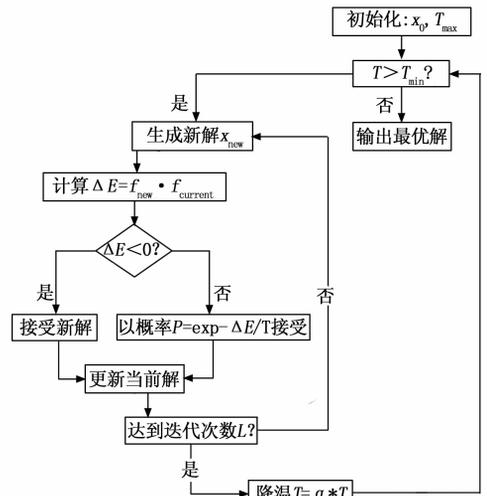


图 4 模拟退火流程图

本文使用模拟退火算法进行仿真,相关参数如下:初始温度 (Initial_temp) 设置为 10 000,这样可以控制算法早期接受劣质解的概率的同时较高的初始温度有助于探索解空间;冷却率 (Cooling_rate) 设置为 0.95,表示每次迭代温度降低的比例,0.95 表示每次迭代温度降低 5%;迭代次数 (Iterations) 设置为 500 次,意味着算法总共执行的迭代次数,迭代次数主要关乎平衡计算时间和求解质量,解 (Solution) 表示为指令的排列顺序;模拟退火能量公式设置为:能量 (Energy) = 适应度 (Fitness) = 已分配指令数 × 10 000 + 总提前时间^[24]。

4.3 实验设置与数据分析

本文使用 STARLINK 数据库中轨道面分布情况抽取 1 500 颗在轨低轨卫星作为巨星座卫星轨道六根数仿真数据库,基于 TLE 数据和 Skyfield 库进行可行性分析获得星地可见弧段,通过分析现有的卫星指令调度数据生成随机的遥控指令合集,分析历史遥控指令合集中不同指令任务出现的概率,随机生成遥控指令,每次生成 7 000~8 000 个指令,指令集数量 1 000 个,通过动态规划算法对星地可见弧段时间生成星地跟踪计划。

由于各个算法建模阶段的指标均以遥控任务规划数量为重要指标,因此主要依据任务规划数量为标准,同时由于巨星座在实际工程中对规划时间有一定要求,因此将不同算法的规划成功时间同样作为指标,计算时间不包括 DFN 的模拟训练时间。

总结不同算法性能表现如表 5 所示。

表 5 算法对比

| 数据 算法 | 任务分配成功百分比中位数 | 任务分配成功百分比平均数 | 规划消耗时间中位数/s | 规划消耗时间平均数/s |
|----------|--------------|--------------|-------------|-------------|
| 贪心算法 | 28.48 | 28.49 | 7.03 | 7.04 |
| 遗传算法 | 29.48 | 28.48 | 7.05 | 7.35 |
| 模拟退火算法 | 26.50 | 26.48 | 1.74 | 1.75 |
| DFN+贪心算法 | 36.30 | 35.94 | 3.14 | 3.25 |

4.4 验证结果

通过分析数据可以得到,贪心算法在遥控任务指令分配成功百分比中相比较其他算法(遗传算法和模拟退火算法)没有明显差别。深度前馈神经网络配合传统贪心算法在任务分配成功率上有较为鲜明的优势,与传统算法中相比在时间上优势较为明显。比较启发式算法中模拟退火算法,深度前馈神经网络配合传统贪心算法可以有更高的任务分配率。

5 结束语

本研究成功地深度前馈神经网络与传统算法中的贪心算法结合起来解决巨星座遥控任务规划问题,实证研究证实了该策略兼具可行性与高效性。尽管实际工程应用面临更为复杂的场景,包含诸多不确定性和新增约束条件,但神经网络具备良好的可扩展性,能够灵活适应这些挑战。在此基础上,研究人员可针对具体应用问题构建不同类型的神经网络,持续探索并优化解决方案。由此,能够开发出更为灵活高效的策略,以应对日益增长的大规模卫星系统数量及任务规划复杂性所带来的挑战。

参考文献:

[1] 杨琳,陆荣,沈庆丰,等.一种低轨卫星测控遥测作业时序优化方法[P].中国:201610667768,2016-10-27.

[2] 金光,武小悦,高卫斌.卫星地面站资源调度优化模型及启发式算法[J].系统工程与电子技术,2004,26(12):1839-1841.

[3] 王官龙,宋建国,陈慧伟,等.一种基于先验信息的中高轨卫星遥测任务资源调度方法[J].电信工程,2023,63(9):1329-1333.

[4] 岳群彬,尚希杰,林晓勇,等.分段分策略调度的遥感卫星任务规划架构[J].天地一体化信息网络,2024,5

(2):63-69.

[5] 向尚,陈盈果,李国梁,等.卫星自主与协同任务调度规划综述[J].自动化学报,2019,45(2):252-264.

[6] 齐星,柳震,梁家辉,等.一种面向低轨遥感星座的路由任务规划算法研究[J].电信科学,2024,38(4):30-38.

[7] 孙慧,李晖,赵曼,等.基于贪心—线性规划算法的敏捷卫星观测时段选取[J].无线电工程,2018,48(9):803-809.

[8] 刘洋,陈英武,谭跃进.卫星地面站系统任务调度的动态规划方法[J].中国空间科学技术,2005,25(1):44-47.

[9] 李爽.基于出口遥感卫星移动接收站站监控软件设计[J].计算机测量与控制,2023,31(11):142-150.

[10] 陈韬亦.基于快响空间的地面控制系统设计[J].计算机测量与控制,2020,28(7):132-137.

[11] 邓宝松,孟志鹏,义余江,等.对地观测卫星任务规划的实践及发展[J].计算机测量与控制,2019,27(11):130-139.

[12] 尹宝才,王文通,王立春.深度学习研究综述[J].北京工业大学学报,2015,41(1):48-59.

[13] 周飞燕,金林鹏,董军.卷积神经网络研究综述[J].计算机学报,2017,40(6):1229-1251.

[14] 朱大奇,史慧.神经网络原理及应用[M].北京:科学出版社,2006.

[15] 毛勇华,桂小林,李前,等.深度学习应用技术研究[J].计算机应用研究,2016,33(11):3201-3205.

[16] 张蕾,章毅.大数据分析的无限深度神经网络方法[J].计算机研究与发展,2016,53(1):68-79.

[17] 邢立宁,陈英武.任务规划系统研究综述[J].火力与指挥控制,2006,31(4):1-4.

[18] 刘建伟,刘媛,罗雄麟.深度学习研究进展[J].计算机应用研究,2014,31(7):1921-1930.

[19] 刘林,王彦荣.卫星轨道预报的一种分析方法[J].天文学报,2005,46(3):307-313.

[20] NADOUSHAN M J, ASSADIAN N. Repeat ground track orbit design with desired revisit time and optimal tilt[J]. Aerospace Science & Technology, 2015, 40: 200-208.

[21] 董军军.动态规划算法和贪心算法的比较与分析[J].软件导刊,2008(2):129-130.

[22] 高新洲,郭延宁,马广富,等.采用混合遗传算法的敏捷卫星自主观测任务规划[J].哈尔滨工业大学学报,2021,53(12):1-9.

[23] 徐欢,祝江汉,王慧林.基于模拟退火算法的电子侦察卫星任务规划问题研究[J].装备指挥技术学院学报,2010,21(3):62-66.

[24] SUTTON R S, BARTO A G. Reinforcement learning: an introduction[J]. Robotica, 1999, 17(2): 229-235.