清 华 大 学

综 合 论 文 训 练

题目：基于强化学习的机组组合问题求解方法研究

系　　别：电机工程与应用电子技术系

专　　业：电气工程及其自动化

姓　　名：公仲泽

指导教师：张　宁　副教授

2021 年 6 月 22 日

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：学校有权保留学位论文的复印件，允许该论文被查阅和借阅；学校可以公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存该论文。

(涉密的学位论文在解密后应遵守此规定)

签　名：公仲泽　导师签名：[illegible]　日　期 2021年6月11日

中文摘要

随着我国电力市场制度的逐步发展以及清洁能源的引入，机组组合问题又面临着新的挑战。使用传统方法进行机组组合求解，能够求得经济上的最优解，但是在求解时间上随着系统规模的增大而迅速增大，难以满足当前电力市场快速出清的要求。针对上述问题，本文提出了基于强化学习的机组组合求解方法，在保证电力系统安全约束的情况下实现机组组合问题的快速求解。

本文首先对机组组合问题进行混合整数线性规划模型与马尔可夫决策过程的建模。在机组组合的问题背景下，引入了保证电力系统安全约束的混合整数线性规划问题模型，使用该分析方法能够利用 Gurobi 求解器实现求解，给出机组组合问题的最优解。引入了强化学习中马尔可夫决策过程的概念，基于机组组合问题的特点给出状态空间、动作空间、转移概率以及奖励函数，为强化学习打下基础。

针对机组组合过程中每一个时段内的决策过程，引入了模仿学习的行为克隆方法。本文给出了一个基于 ResNet 网络的智能体结构，并令其模仿混合整数线性规划问题方法给出的在某些场景下的状态决策对，使之能够求解在单时段内的机组组合问题。通过模仿学习，该智能体为强化学习提供了一个基础的策略网络，便于强化学习的求解。

最后，本文基于模仿学习给出的智能体作为基础的决策网络，引入了强化学习中的策略梯度算法，使用 Actor-Critic 算法对该问题进行求解。提出了使用最优潮流优化限制、屏蔽函数与惩罚函数的三种方法实现了电力系统安全约束在各机组之间上与时序上的安全约束。使用强化学习方法给出的策略模型，能够以远小于优化方法的求解时间给出与优化方法相比成本差不多的解。

综上所述，本文的工作实现了对机组组合问题进行了优化问题及马尔可夫决策过程的建模，使用模仿学习得到一个求解单时段机组组合的模型，并使用强化学习使其能求解多时段的机组组合问题。本文引入数据驱动的方法扩展了电力系统优化调度的分析方法。

关键词：机组组合；MILP；MDP；模仿学习；强化学习

ABSTRACT

Using the traditional method for unit commitment solution, we can find the economically optimal solution, but the solution time increases rapidly with the increase of system size, which is difficult to meet the current requirements of rapid market clearing. To solve the above problems, this paper proposes a reinforcement learning based unit combination method to achieve fast solution of the unit commitment problem while ensuring the security constraints of the power system.

In this paper, we first model the mixed integer linear programming model and Markov decision process for the unit commitment problem. In the context of the unit commitment problem, a MILP model is proposed to ensure the safety constraints of the power system. The concept of MDP in reinforcement learning is introduced.

The behavioral cloning method of imitation learning is introduced for the decision process within each time period of the unit commitment process. In this paper, an agent based on ResNet network is given and made to imitate the statedecision pairs given by the MILP method in certain scenarios to solve the unit commitment problem in a single time period.

Finally, this paper introduces the policy gradient algorithm in reinforcement learning based on the agent given by imitation learning as the underlying decision network, and solves the problem using the Actor Critic algorithm. Three methods using OPF optimization restrictions, shielding functions and penalty functions are proposed to achieve the safety constraints of the power system in cross section and in time sequence.

In summary, this paper implements the modeling of the MDP for the unit commitment problem, using imitation learning to obtain a model for solving the single time unit commitment, and using reinforcement learning to enable it to solve the multi time unit commitment problem. This paper introduces a data driven approach to extend the analytical approach to optimal scheduling of power systems.

Keywords: unit commitment; MILP; MDP; imitate learning; reinforcement learning

目　　录

第 1 章　引言

1.1　研究背景

在最近十多年来，全球都在共同面对着越来越严峻的气候变暖以及能源安全问题。为了实现全球的能源结构优化改革，应对能源安全问题，引入风能、太阳能、水的重力势能等可再生能源成为了重要手段[1]。可再生能源是指与传统的化石能源相对应的，从自然中可以可再生地重复得到的能源，通常包含风能、太阳能、潮汐能等。但是，可再生能源的引入同时也给电力系统从源端带来了更大的随机性。作为如今发电的主要力量，对于传统机组来说，需要同时面对来自清洁能源的随机性与负荷的随机性，使得传统机组的调度难度大大提升。

随着国内电网电力市场改革的进一步加深，国家发改委发布《关于推进电力市场建设的实施意见》，强调“逐步建立起以中长期交易规避风险，以现货市场发现价格，交易品种齐全、功能完善的电力市场”。国内的电力市场提供现货市场和期货市场，立足于规范计划调度与简单电力交易，最终达到现货与中长期的效果[2]。在此背景下，为了实现电力市场的有效性，需要在日内进行短时间内的市场出清。因此，机组组合的快速求解则成为重要问题。

随着大数据时代的到来，数据科学正在成为越来越受到人们重视的科学。在全球数据爆炸的背景下，对已有数据进行收集、分析、研究成为了重要的研究方法。目前，已经有文章[3]提出使用数据驱动的方法对电力系统内问题求解。使用数据驱动的方法进行电力系统问题研究，能够提供一个与以往完全不同的、全新的数据视角来看待电力系统内的问题。

1.2　研究现状

机组组合问题是电力系统优化调度中的重要问题，好的机组组合问题的求解结果可以直接降低发电厂的发电成本，转化为经济效益。因此，机组组合问题一直都是人们研究的热点问题。但是机组组合问题作为优化问题来说，由于其决策变量同时包含连续变量和离散变量，这就使得求解该问题的难度相比连续变量的线性规划大大提升。在过去几十年里，人们提出了许多优化方法对机组组合问题进行求解。

在研究早期，人们主要提出优化方法对该问题进行求解。主要的方法包含优

先顺序法[4-5]，分支界定法[6]，动态规划法[7]，拉格朗日松弛法[8]以及混合整数线性规划法[9-10]。优化方法属于分析方法，使用分析方法求解机组组合问题的时候能够有着充足的理论依据作为基础，这属于分析方法的优点。但是使用分析方法同时存在以下缺陷。

(1) 计算复杂度。分析方法的复杂度会随着问题规模的增大而快速增加，在早期计算机计算资源仍不发达的时候较难处理大系统问题的。以分支界定法为例，其求解时间会随着问题规模的增加而指数级别的增加[11]。

(2) 难以找到全局最优解。由于分析方法通常都对问题进行了一定的假设或近似，则在对大系统进行分析时可能会进一步放大近似时产生的误差，从而直接影响最终优化结果的真实性与有效性。

在近十多年中，人工智能与数据科学的飞速发展。对于数据进行分析、处理、利用的竞争已经渗透到了全球的各行各业中，这也同时为人们带了更多实用人工智能方面的算法。这些方法中主要包含专家系统法[12]，神经网络法[13]，模糊逻辑法[14]，遗传算法[15]以及模拟退火法[16]。但是，这些方法与人工智能算法有着共通的缺点：最终结果对所选用的算法结构、超参调节以及损失函数有着很大的关系[17]。

1.3 研究意义

目前为止，对机组组合问题使用基于数据驱动的人工智能方法的研究还比较少。而随着数据科学的快速方法，使用大数据与数据分析的方法对问题进行研究已经成为十分重要的研究方法。本文基于对机组组合问题的优化问题建模与 MDP 建模，设计并训练得到机组组合问题的决策模型。借由此模型，经过前期训练后便可快速求解新的机组组合问题，从而适应电力市场背景下的快速出清、快速计算的要求。

本文主要的研究意义如下：

(1) 完成对机组组合问题的马尔可夫决策过程（MDP）建模。MDP 是强化学习中的基础，所有的强化学习方法都是基于 MDP 的。本文中将从机组组合问题中提取出包含状态空间、动作空间、转移函数以及奖励函数的四元元组，使用 MDP 描述了机组组合的全过程，并且保证了电力系统安全约束能够得到满足。基于该 MDP，可以使用多种不同的强化学习方法对该问题进行求解。

(2) 通过强化学习的方法，使用 Actor-Critic 网络给出了机组组合的策略模型和

价值函数，从而提升了机组组合问题的求解速度。基于对机组组合问题的优化问题建模与 MDP 建模，本文通过模仿学习和强化学习的方法给出了一个能够求解序贯决策的机组组合问题的算法模型。该算法模型能够在短时间内给出决策结果，从而提升机组组合问题求解的效率。

1.4 研究目标与主要工作

1.4.1 研究目标

本文主要对机组组合问题在全天二十四小时内的调度计划进行求解。本文的研究目标分为以下两点。

1. 对机组组合问题进行建模，分别建模为 MILP 问题和 MDP 过程，使之可以分别使用优化算法和强化学习算法进行求解。
2. 使用强化学习方法求解机组组合问题的 MDP 问题，在保证电力系统安全约束满足的前提下给出高效的机组组合求解模型

1.4.2 主要工作

本文的主要工作分为以下几个部分：

第二章中将机组组合问题分别建模为混合整数线性规划问题以及 MDP 过程。在建模为混合整数线性规划问题的时候，以机组出力成本作为最小化的目标函数，将电力系统的各种安全约束作为该优化问题中的约束条件。在建模为 MDP 问题的时候，给出 $< S, A, P, R >$ 四元元组。S, A, P, R 分别指由状态空间、动作空间、转移概率函数以及奖励函数。以动静态变量作为状态变量，以机组启动概率作为动作空间，以安全约束作为转移核函数，以负运行成本作为奖励函数。

第三章中使用模仿学习的行为克隆方法训练了解决单时段机组组合问题的模型。在行为克隆方法中，将使用 MILP 方法计算得到的特定情境下的决策序列拆分为 $< S, A >$ 状态动作对，以此作为监督学习的输入和标签进行训练，从而得到了可以解决单时段机组组合问题的模型。

第四章中分析了仅仅使用模仿学习的不足，不能生成连贯的决策结果。因此引入强化学习方法。考虑到状态空间含有连续变量，因此使用策略梯度法的 Actor-Critic 算法对该强化学习问题进行求解。并且还使用 OPF 优化限制、屏蔽函数以及惩罚函数来保证电力系统安全约束的实现。最后，给出了一个能求解全天机组组合问题的模型。

第五章中，对全文进行总结。其中，对于第二章到第四章，各章之间的相互关

系如图1.1所示。

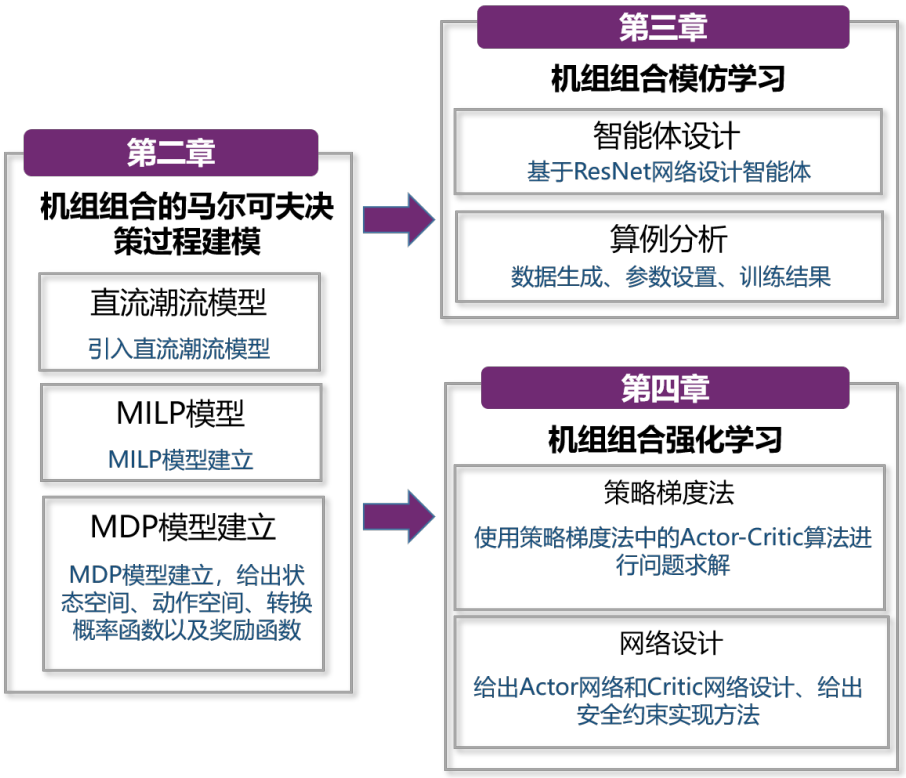


图 1.1　章节结构图

第 2 章　机组组合的马尔可夫决策过程建模

2.1　概述

随着全球气候变暖问题与能源安全问题日益严峻，可再生能源在全球和国内都正在经历着快速的发展，具体表现为国内可再生能源装机容量和装机比例都在迅速提升。伴随着可再生能源的引入，其与传统火力发电所不同的特点也需要引起人们的注意。可再生能源有着随机性与间歇性等特点，从电网的源端引入了随机性。在竞争性电力市场蓬勃发展的大背景下，电力市场要求供给端能够在保证电力系统安全的前提下灵活快速地对需求端的负荷作出响应，对机组组合问题的求解速度提出了新的挑战。

通常对机组组合问题进行建模的方法是将其建模为混合整数线性优化问题。通过将最小化成本设为目标函数，将机组组合的物理约束及安全约束设为约束条件，以及将各个机组的启停状态以及出力大小作为决策变量来构成该优化问题。优化问题建模方法的建模过程比较简单，模型直观理解比较容易，求解方法较为成熟，通常使用商业化的求解器如 Gurobi[19] 对混合整数线性优化问题进行求解。但是，由于混合了整型变量与连续变量，该问题的求解复杂度随着问题规模的增大而急剧增大。在引入可再生能源后，电力市场高频率的出清计算要求下，采用该方法难以在短时间内给出问题的解。

强化学习是机器学习中的一个分支，其通过智能体与环境的交互过程使得智能体能够在该交互过程中逐步学习环境的特性，最终实现理解环境动态的效果。通过引入强化学习，可以充分利用曾经解过的机组组合问题的经验，使得在面对新问题的时候能够实现快速求解。而强化学习的第一步是将机组组合问题建模为马尔可夫决策过程（Markov Decision Process，简称为 MDP）。

综上所述，对机组组合问题的建模有混合整数先行优化建模以及 MDP 建模两种方法，本章分别采用两种模型进行建模。

2.2　主要符号对照表

$\mathbf{P}$	节点功率注入
$\mathbf{B}$	导纳矩阵虚部
θ	节点电压相角

$\mathbf{S}$	线路有功潮流
$\mathbf{G}$	支路有功潮流
C_{net}	网络总成本
$\Omega_N, \Omega_L, \Omega_G, \Omega_T$	节点、线路、机组以及机组组合时间段集合
a_i, b_i, c_i	机组 i 对应的二次成本函数系数
u_i^t	机组 i 在时间 t 的启停状态，0 表示停止，1 表示启动
g_{ij}	矩阵 GB^{-1} 中第 i 行第 j 列元素，说明第 j 节点单位有功注入引起第 i 条线路有功潮流变化量
a_{ij}	矩阵 A 中的第 i 行第 k 列元素。若机组 j 接入节点 i，则 $a_{ij}=1$。对于同列其他元素有 $a_{ik}=0, k\neq j, i\in\Omega_N, j\in\Omega_G$
p_i^t	机组 i 在时间 t 的出力,$i\in\Omega_G$
d_i^t	节点 i 在时间 t 的负荷,$i\in\Omega_N$
$\bar{f}_l, \underline{f}_l$	线路 l 的有功传输上限和下限，$l\in\Omega_L$
$T_{i,on}^t, T_{i,off}^t$	机组 i 在时刻 t 的开机时间和停机时间,$i\in\Omega_G$
$\bar{T}_{i,on}, \underline{T}_{i,off}$	机组 i 的最小开机时间和停机时间,$i\in\Omega_G$
$u_{i,on}^t, u_{i,off}^t$	机组 i 在时刻 t 的开机次数与停机次数
$\bar{u}_{i,on}, \underline{u}_{i,off}$	机组 i 的最大开机次数和停机次数,$i\in\Omega_G$
$\Delta p_{i,up}, \Delta p_{i,down}$	机组 i 的上下爬坡速率,i$\in\Omega_G$
r_{up}^t, r_{down}^t	电力系统在时间 t 的负荷正负备用率
T_i^t	机组 i 在时刻 t 距离上一次状态变换已经过去的时间
$l_{i,j}^t$	机组 i 在时间 t 的 j 小时后的负荷
N_{gen}	系统中总的发电机机组个数
N_{load}	在 MDP 设计的状态中向后看负荷的时间长度

2.3 直流潮流模型

通常，计算电力系统潮流有直流潮流和交流潮流两种计算方法。交流潮流模型在计算时可以更加精确地描述电力系统，提供包括电压、电网潮流等更多的信息。直流潮流模型是对交流潮流模型的化简，其通过线性化电力系统的非线性潮流问题，使得线性代数理论得以使用，从而轻易地对该问题进行求解。

虽然交流潮流模型更加精确，但是求解非线性方程会变得非常复杂，需要投入大量的计算资源以及计算时间。同时，直流潮流模型就能够有较好的近似效果。

因此，本文采用直流潮流模型。

根据习惯，直流潮流模型的完整方程为如下。

$$\begin{aligned} \theta &= \boldsymbol{B}^{-1}\boldsymbol{P} \\ \boldsymbol{S} &= \boldsymbol{G}\boldsymbol{B}^{-1}\boldsymbol{P} \end{aligned} \tag{2.1}$$

式(2.1)中各变量的定义如下。直流潮流矩阵形式为：

$$\boldsymbol{P} = \boldsymbol{B}\theta \tag{2.2}$$

其中，$\boldsymbol{P}$ 是节点功率注入向量，大小为 $n \times 1$ 阶。θ 是电压相角，大小为 $n \times 1$ 阶。$\boldsymbol{B}$ 是节点导纳矩阵的虚部，大小为 $n \times n$。

式(2.3)说明了支路潮流与节点相角的关系。其中，$\boldsymbol{S}$ 为线路有功潮流向量，大小为 $m \times 1$ 阶。$\boldsymbol{G}$ 为表示支路有功潮流和节点相角关系的矩阵，大小为 $m \times n$。

$$\boldsymbol{S} = \boldsymbol{G}\theta \tag{2.3}$$

值得注意的是，矩阵 $\boldsymbol{G}$ 中每一行最多仅有两个非零元。

$$\begin{cases} \dfrac{\partial P_l}{\partial \theta_k} = \dfrac{1}{x_{ij}} \\ \dfrac{\partial P_l}{\partial \theta_j} = -\dfrac{1}{x_{ij}} \end{cases} \tag{2.4}$$

2.4 考虑安全约束的电力系统机组组合优化模型

本节将传统机组的机组组合问题建模为混合整数线性优化问题，使之可以利用商业优化器进行求解。

2.4.1 决策变量

机组组合问题是电力系统中进行优化调度的重要内容。机组组合即是通过适当地规划各机组的启停状态与各机组出力大小以达到在满足负荷要求下实现总成本最低的目标。因此，选择机组 i 在时刻 t 的启停状态 u_i^t 以及出力大小 p_i^t 作为决策变量。因为在该优化问题下，决策变量有整型变量 u_i^t 以及连续变量 p_i^t，因此该问题为混合整数线性规划问题。

2.4.2 目标函数

机组组合问题的目标是最小化整个电力系统的发电成本。为了降低模型的复杂度，本节目标函数不考虑机组的启停成本，仅考虑传统火电机组的燃料成本，并设为机组出力的二次函数。则目标函数如(2.5)所示。其中，a_i, b_i, c_i 为成本函数系数。

$$minC_{net} = \sum_{t\in\Omega_T}\sum_{i\in\Omega_G}(a_i {P_i^t}^2 + b_i P_i^t + c_i)u_i^t \tag{2.5}$$

2.4.3 约束条件

2.4.3.1 线路有功潮流约束

根据直流潮流模型，可以得到以下支流潮流约束。

$$\begin{aligned}\sum_{k\in\Omega_N} g_{lk}\sum_{j\in\Omega_G} a_{kj}p_i^t - \sum_{k\in\Omega_N} g_{lk}d_k^t \leqslant \bar{f}_l, \forall l\in\Omega_L, t\in\Omega_T \\ \sum_{k\in\Omega_N} g_{lk}\sum_{j\in\Omega_G} a_{kj}p_i^t - \sum_{k\in\Omega_N} g_{lk}d_k^t \geqslant \underline{f}_l, \forall l\in\Omega_L, t\in\Omega_T\end{aligned} \tag{2.6}$$

2.4.3.2 网络功率平衡约束

在电力系统中，需要保持网络整体的功率和负荷在每一时刻都相等。

$$\sum_{i\in\Omega_G} p_i^t = \sum_{i\in\Omega_N} d_i^t, \forall t\in\Omega_T \tag{2.7}$$

2.4.3.3 最小开停机时间约束

对于传统机组来说，其发电过程依靠燃烧燃料加热水生成蒸汽，并使用蒸汽推动汽轮机旋转，最后汽轮机带动发电机旋转。该过程实现了从燃料化学能到蒸汽热能到汽轮机机械能，最后转化为电能的能量流动过程。因此，传统机组存在较大惯性，机组的状态转换之间需要有一定时间间隔。

$$\begin{aligned}T_{i,on}^t \geqslant \bar{T}_{i,on} \\ T_{i,off}^t \geqslant \underline{T}_{i,off}\end{aligned} \tag{2.8}$$

2.4.3.4 日内最大启停次数约束

对于传统机组来说，为了保护机组，延长机组使用寿命，一天之内不宜过多次开停机。

$$\sum_{t\in\Omega_T} u_{i,on}^t \leqslant \bar{u}_{i,on}, \sum_{t\in\Omega_T} u_{i,off}^t \leqslant \bar{u}_{i,off}, \forall i \in \Omega_G \tag{2.9}$$

2.4.3.5 爬坡约束

由于传统机组主要依靠燃烧燃料获取能量，因此其输出功率不能在短时间内发生突变，在相邻时间段中出力变化存在上限。因此，对于时刻 t 和时刻 $t-1$ 都开机的机组，满足式(2.10)。

$$-\Delta p_{i,down} \leqslant p_i^t - p_i^{t-1} \leqslant \Delta p_{i,up}, \forall t \in \Omega_G \tag{2.10}$$

2.4.3.6 机组出力上下限约束

由于火力机组能量来源于燃烧燃料，因此其出力存在着上下限。

$$u_i^t \underline{p}_i \leqslant p_i^t \leqslant u_i^t \bar{p}_i, \forall t \in \Omega_T, i \in \Omega_G \tag{2.11}$$

2.4.3.7 正负备用约束

为了保证电力系统运行安全，除了要求线路上功率与负荷匹配之外，还要求开机机组能保留正负功率备用。

$$\begin{aligned} \sum_{i\in\Omega_G} u_i^t \bar{p}_i &\geqslant \sum_{i\in\Omega_N} (1 + r_{up}^t) d_i^t, \forall t \in \Omega_T \\ \sum_{i\in\Omega_G} u_i^t \underline{p}_i &\leqslant \sum_{i\in\Omega_N} (1 - r_{down}^t) d_i^t, \forall t \in \Omega_T \end{aligned} \tag{2.12}$$

2.5 考虑安全约束的电力系统 MDP 建模

强化学习是机器学习的一个分支，通过智能体和动态环境的试错交互过程来学习在不同状态下应该执行的动作，并最终根据最优策略解决问题[20]。在进行强化学习前，首先要将机组组合的具体问题建模为马尔可夫决策过程（Markov Decision Process，MDP），因为大部分强化学习算法是对 MDP 问题才能使用动态规划的方法[21]。将具体问题建模为 MDP，需要给出一个四元元组 $\{S, A, P, R\}$。其中 S 代表环境中所有可能的状态的集合，A 代表智能体可用动作的集合，P 代表在状态之间转换概率的集合（状态转移概率矩阵），R 代表状态转换后得到的奖励。

在进行机组组合优化模型建模时，将机组的启停状态与当时的出力共同作为

决策变量进行优化。实际上，可以将该目标拆分为两步：第一步决策每个时段内机组的启停状态，第二步根据该启停状态来计算对应开机机组的出力情况。为了提升算法性能、降低训练难度，本节中 MDP 建模主要关注第一步，而第二步的过程则通过 python 的电力系统求解软件包 pandapower[22] 实现。

本节将机组组合问题建模为 MDP，给出上述四元元组。

2.5.1 状态空间

在该 MDP 中，希望让智能体在每一个时刻中都能根据给定的输入数据给出该时刻的机组启停状态。因此，每一时刻输入的数据构成状态空间。在机器学习中，有着垃圾进，垃圾出（Garbage In Garbage Out）的问题。因此，在选取状态空间的时候应该充分考虑可能对机组组合问题有影响的变量。基于在求解过程中，变量的值是否会发生变化，本节将选择的变量划分为静态变量和动态变量。

2.5.1.1 静态变量

静态变量应该选择对机组组合问题求解有重要影响，同时在求解过程中不会改变的变量。静态变量刻画了机组组合问题中的静态结构，是对机组组合问题求解全过程中都存在影响的变量。

(1) 机组成本函数

各火力机组的成本函数在求解机组组合问题中占据最重要的地位。机组组合问题的本质就是通过合理地规划各机组的启停状态以实现在安全约束下满足负荷需求的目标。通常，认为机组的出力成本函数为二次函数，如式(2.13)所示，其中一共产生 a_i, b_i, c_i 三个参数。

$$C_i(p_i^t) = a_i {p_i^t}^2 + b_i p_i^t + c_i \tag{2.13}$$

(2) 备用率

备用率是指在机组组合求解过程中，要求某一时刻开机的所有机组提供的出力上限和出力下限要为该时刻的负荷留有一定余量，以应对可能存在的安全问题。对于所有机组来说，同一时刻的向上（下）备用率是相等的。

$$\begin{aligned} r_{up,i} &= r_{up}, \forall i \in \Omega_G \\ r_{down,i} &= r_{down}, \forall i \in \Omega_G \end{aligned} \tag{2.14}$$

(3) 最短开停机时间

由于火电机组惯性较大，启停状态变化不能过于频繁，因此机组存在着最短开停机时间限制。不同机组之间的最短开停机时间不同，大型机组最短

开停机时间较长，而小型机组最短开停机时间较短。而在不同时间段内，相同机组的最短开停机时间是不变的。

$$\begin{aligned} T_{i,on}^{t} &= T_{i,on}, \forall i \in \Omega_G \\ T_{i,off}^{t} &= T_{i,off}, \forall i \in \Omega_G \end{aligned} \tag{2.15}$$

2.5.1.2 动态变量

动态变量应该选择对机组组合问题求解有重要影响，同时在求解过程中会改变的变量。动态变量刻画了机组组合问题中的动力学结构，展现并解释了机组组合问题的动态变化过程。

(1) 机组状态

由于火电机组的惯性，在相邻时刻中某一时刻机组的启停状态通常不会与上一时刻有太大的差别[17]。因此，上一时刻的机组状态将会很大程度上影响当前时刻机组状态。

$$u_i^t, \forall i \in \Omega_G, t \in \Omega_T \tag{2.16}$$

(2) 机组出力

一方面，由于爬坡约束在相邻时间同一机组的出力不能产生太大变化。另一方面，上一时刻机组出力大小也能反映该机组的重要程度。上一时刻出力越大的机组通常在当前时刻中会有更大的出力，上一时刻出力较小的机组通常在当前时刻中会有更小的出力。

$$p_i^t, \forall i \in \Omega_G, t \in \Omega_T \tag{2.17}$$

(3) 状态转变冷却时间

由于最短开停机时间限制，同一机组不能在短时间内多次反转状态。因此状态转变冷却时间记录该机组距离上一次状态转变已经过去的时间长度，正数代表上一次状态转变为启动机组，负数代表上一次状态转变为停止机组。

$$T_i^t, \forall i \in \Omega_G, t \in \Omega_T \tag{2.18}$$

(4) 未来负荷

机组组合问题的目标是在给定负荷的情况下，求解总成本最低的机组调度计划。因此，负荷是对该问题有着非常重要影响的变量。此外，考虑到当未来几个时刻内负荷变化情况不同，也会对机组调度计划产生影响，因此未来 N_{load} 个时刻内的负荷也是会影响当前调度计划的变量。所以，将未来 N_{load} 个时刻的负荷同时作为动态变量。

$$l_{i,j}^{t} = l_{j}^{t}, \forall i \in \Omega_G, t \in \Omega_T, j \in 1, \cdots, N_{load} \tag{2.19}$$

2.5.2 动作空间

在强化学习中，要求动作空间具有完备性、高效性以及合法性。

(1) 完备性

完备性的要求指动作空间应该包含所有可能的动作，使得所有动作都有被探索到的概率。在本问题中，决策目标是机组的启停状态。因此，动作空间应该包含所有的机组启停状态的组合。

(2) 高效性

在优化问题中，决策变量为启停状态（离散变量）与机组出力（连续变量）。出于对高效性的考虑，将启停状态与机组出力分步求解，在强化学习中仅求解启停状态的值。当得到启停状态之后，再对最优潮流进行求解，此时仅为连续变量的优化问题，问题难度相比混合整数线性优化问题大大下降。

(3) 合法性

合法性要求动作空间中的动作能够保证安全约束，在智能体与环境交互之前，需要先保证所给出动作的合法性。在机组组合问题中，既是要保证线路有功潮流约束、网络功率平衡约束、最小开停机时间约束、日内最大启停次数约束、爬坡约束、机组出力上下限约束以及正负备用约束。

每一个机组可能的动作为打开或关闭，因此动作空间为所有机组打开或关闭动作的组合。

$$A = \{0, 1\}^{N_{gen}} \tag{2.20}$$

2.5.3 转换概率

转换过程由转移核决定，如式(2.21)所示。

$$f(s, a) = s' \tag{2.21}$$

该转移核在保证安全约束的前提下根据动作 a 从状态 s 转至状态 s'。在机组组合问题的转换过程中，动作 a 为智能体根据策略 π 给出的机组调度计划解决方案，经过合法性检验[①]之后，则根据该结果计算各机组的出力与成本等信息。最终根据结果调整状态空间中的动态变量部分，从而实现状态的转移。

2.5.4 奖励

强化学习的目标是让智能体在解决问题时，在路径上获得的奖励最大化。在机组组合问题中，目标是最小化一天时间内的总成本。因此，在每一个时间段内给出的奖励应该为负的机组出力成本。

$$R(s,a,s') = -\sum_{t\in\Omega_T}\sum_{i\in\Omega_G}(a_i {p_i^t}^2 + b_i p_i^t + c_i)u_i^t \tag{2.22}$$

2.6 本章小结

对优化问题的建模和 MDP 的建模是使用优化方法和强化学习方法的基础。本章首先根据机组组合问题的特点，将机组运行成本设置为最小化的目标，并且同时根据电力系统的其他安全约束建立了该优化问题的约束条件，完成了对优化问题的建模。在该优化问题中，决策变量同时包括整型变量与连续变量，因此该优化问题属于混合整数线性规划问题。然后，本章对机组组合问题进行 MDP 的建模，提出了表征机组组合问题的动静态变量作为状态空间，以机组启停状态作为动作空间，以电力系统安全约束作为转换的概率函数，并且以电力系统运行成本作为该 MDP 的奖励函数，完成了对 MDP 的建模。基于该 MDP 建模，后文将给出模仿学习方法和强化学习方法，解决机组组合问题。

① 合法性检验将会在4.6.1节中说明

第 3 章　机组组合模仿学习

3.1　概述

为了实现优化策略的目标，经典的强化学习算法通常采用最大化路径累计折现奖励的优化方法。这样的更新方法能够在大部分数据情形下较快地完成对模型的训练，并且给出最优策略[20]。

但是，这样的方法在面对类似机组组合这样的序列决策问题（sequential decision）的时候，可能会智能体受限于不能及时得到反馈，因此而难以较快的更新策略而使得模型更新较慢。当面对规模较大，问题较复杂的情形下，这种基于随机初始策略对广阔的状态空间与动作空间进行探索的做法会使得模型难以在有限时间内收敛，最终无法给出一个较好的结果。

因此，人们提出模仿学习为强化学习进行辅助。模仿学习通过让智能体在进行强化学习之前先通过学习人类专家在不同状态下给出的动作，基于已有经验进行探索。智能体再对基于专家的先验知识得到的初始策略进行探索，从而实现快速求解，快速收敛到最优解的目标。

3.2　主要符号对照表

τ_i	在第 i 个情境下做出的从头到尾的完整决策过程，$\tau_i = < s_1^i, a_1^i, s_2^i, a_2^i, \cdots, s_{n_i}^i, a_{n_i}^i >$
$\mathcal{D}$	将决策序列 τ 拆分成动作状态对后构成的决策经验
p_{ij}	第 i 个机组在第 j 个时段内，模仿学习算法给出的开机概率 p，则 p∈[0,1]
U_{ij}	第 i 个机组在第 j 个时段内，MILP 算法给出的启停状态，1 表示开机，0 表示关机

3.3　模仿学习

在强化学习中，大部分算法通过最大化路径累积奖励的方法来对策略进行优化，以求最终得到最优策略。但是当问题的奖励反馈较慢的时候，这样的训练方法是比较低效的，有可能花了很长时间进行训练却只达到了局部最优或者甚至无

法收敛。此外，在某些问题中人们只能判断一个动作是好是坏，而不能显式地量化地给出一个状态动作的价值，如：学习了两本不同领域的经典教材，对阅读者来说都是有益的，但是却无法用数字给出每本书带来的收获。在这种情况下，人们通常只能人工地给出一个估计值，不能准确反映问题中的实际价值。模仿学习是解决上述问题的一个可行的方法。在模仿学习中，智能体通过观察并模仿专家给出的状态动作对来实现掌握知识、模仿行为的目标，并最终给出状态与最优动作之间的映射。因此，在模仿学习中智能体并不受反馈较慢的奖励或者认为给定的奖励函数的影响，能够解决上述问题。

在模仿学习中，问题给出一系列的情境，专家在给定的情境中进行决策，并且给出决策结果，得到决策序列 $\{\tau_1, \tau_2, \cdots, \tau_n\}$。决策序列中的每一个元素代表在某种情境下专家所面对的一系列状态以及其根据专家策略 π^* 所作出的决策 $\tau_i = < s_1^i, a_1^i, s_2^i, a_2^i, \cdots, s_{n_i}^i, a_{n_i}^i >$。根据以上的决策序列，其中所有的状态动作对可以组成专家决策经验 $D = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), \cdots\}$。

对于专家决策经验，不同的模仿学习算法有着不同的使用方法。行为克隆（Behavioral Cloning）是模仿学习的最简单的形式，直接将专家决策经验的状态作为输入，将动作作为标签进行监督学习。直接策略学习法（Direct Policy Learning）需要一个能够在模型训练期间也能互动的专家，并迭代地使用根据专家的经验学习，然后再让专家对智能体给出的决策进行评价的方法不断地改进智能体的策略。逆强化学习（Inverse Reinforcement Learning）则特别地通过根据专家决策数据来学习环境的奖励函数，并以此来优化策略[23]。考虑到算法的复杂性，本节使用行为克隆的方法来完成模仿学习。

进行行为克隆需要先让专家对一系列情境进行决策，并且给出决策结果，然后再让智能体对其进行监督学习。在机组组合问题中，可以利用第二章中介绍的 MILP 优化问题作为专家，给出一系列情境并使用优化方法求解得到在每一个状态下的机组启停的 0-1 变量作为专家决策序列。将机组启停的 0-1 变量作为输入，令智能体输出预测的每台机组的开机可能性，并以均方误差作为误差来对智能体进行监督训练。因此，在机组组合问题中的行为克隆模仿学习算法如算法3.1所示。

算法 3.1 机组组合中的行为克隆模仿学习

1: 将一系列情境输入给 MILP 模型，并且得到对应的专家决策 τ^*
2: 将专家决策进行处理，得到独立同分布的状态动作对 $(s_0^*, a_0^*), (s_1^*, a_1^*), \cdots$
3: 通过最小化损失误差函数 $SSE(a^*, \pi_\theta(s))$ 的方法来对策略 π_θ 进行监督训练

然而，行为克隆并不总能很好地完成学习目标。由于以下几个原因，仅仅通过

监督学习的方法来得到状态和动作之间的映射不足以得到目标效果。

- 获取专家经验时产生的误差
- 专家经验例子数量太少
- 专家面对的问题与智能体面对的问题不完全相同

因此，在实际训练中，在模仿学习之后通常需要再对该模型进行进一步的训练，也就是说模仿学习是为后续的训练提供了一个较好的初始策略，以降低后续强化学习过程中探索的复杂性。因此，整个训练过程实际的流程图如图3.1所示。

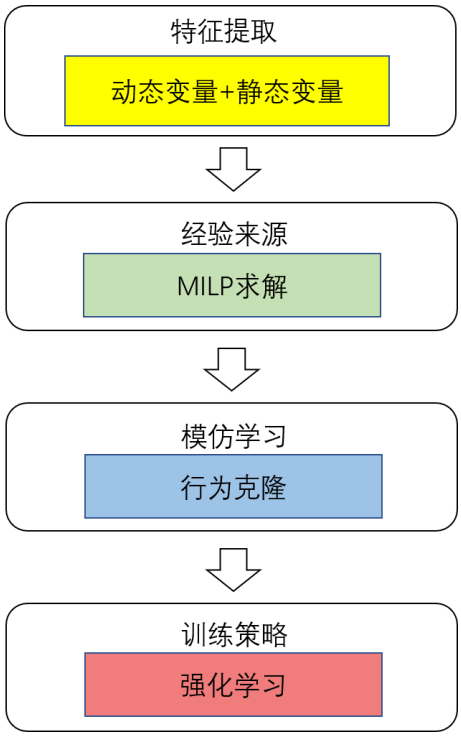


图 3.1　完整训练过程流程图

3.4　网络设计

3.4.1　问题结构

在本文第二章中已经完成了对机组组合问题的 MDP 建模，分别定义了该 MDP 模型中的的状态空间、动作空间、转换概率以及奖励。

在状态空间中，分别用静态变量和动态变量对机组组合问题中重要的影响因素完成了特征提取。在静态变量中，使用机组成本函数、备用率以及最短开停机时间来描述该问题中的静态结构。在动态变量中，使用机组状态、机组出力、状态转变冷却时间以及未来负荷刻画了该问题中的动力学结构，展示并解释了机组组合

问题的动态变化过程。在动作空间中，为了满足完备性、高效性、合法性的要求，选用了当前时刻所有机组开关的启停状态的组合作为动作。

综上所述，MDP 问题的建模已经完成了对该问题的定义。机组组合问题可以使用七个变量（三个静态变量以及五个动态变量）完成对一个机组状态的描述。对于整个电力系统来说，每一个时刻的状态可以用所有机组对应的七个变量来描述，因此此时状态变量矩阵的形状为 $[N_{gen}, 7]$，该矩阵构成了智能体的所有输入。对于智能体的输出来说，应该输出当前时刻所有机组的启动（或保持启动）的概率。因此，对于每一个时刻来说，应该输出长度为 N_{gen} 的列向量，该列向量构成了智能体的所有输出。

3.4.2 智能体网络

3.4.2.1 网络选择

机器学习中的学习理论表明足够大的一层神经网络可以逼近任意函数[24]，为机器学习打下了重要的理论基础。但是在实践中，仅仅使用一层神经网络不能达到比较好的效果。参数过多、过拟合等等都是浅层网络进行函数逼近中不得不面对的问题。为了克服这些困难，人们选择使用更加深层的网络进行函数拟合[25]，以期得到更好的拟合效果。但是仅仅靠堆叠网络层数并不能得到很好的结果。在使用深层神经网络的时候，由于梯度消失（或梯度爆炸）、退化等问题使得深层网络也会难以训练。在某些任务中，深层网络的表现甚至不如浅层网络。

总之，浅层的网络会面临参数过多、过拟合等问题；深层的网络会面临梯度消失（或梯度爆炸）、退化等问题。在当时，浅层网络和深层网络的问题都限制了神经网络在图像分类上的发展。ResNet 网络是深层神经网络的一次巨大创新，其使用残差学习的方法出色地解决了深度卷积神经网络中存在的退化问题[26]，使得神经网络层数相比以前可以大幅度提升，从而更好地使用深层网络进行特征识别与分析。通过引入” 单位短路” 连接，ResNet 完成了对深层神经网络残差的引入，并因此解决了深层网络退化问题，从而可以很好地完成了图像分类任务。因此，本文选用了 ResNet 网络作为智能体中的核心网络。

3.4.2.2 智能体结构

在3.4.1节中，本文将该问题的输入总结为一个大小为 $[N_{gen}, 7]$ 的矩阵。如果将该矩阵看为一张等大的图片，那么本问题的目标是将该” 图片“分类至 $2^{N_{gen}}$ 个可能的系统开关组合。因此，本节使用 python 中 PyTorch 包的预训练好的 ResNet18 模型[26]作为智能体，并通过对其输入输出端进行修改从而完成该模型从 ImageNet

数据集到机组组合问题数据集上的迁移。

接下来将介绍智能体的结构。首先，考虑到静态变量和动态变量在训练过程中的区别：静态变量的值不发生改变，而动态变量的值会随着时间步骤变化而变化。本文将静态变量和动态变量分别使用两个不同的嵌入层进行嵌入，将低维的静态变量与动态变量向更高的维度中投影。则在每一天 24 小时的求解过程中，静态变量只需要进行一次嵌入，而动态变量需要在每个时段都进行嵌入。然后，将两个输入进行合并，并分别使用三个不同的全连接层将其进行转换。从而得到类似图像的三维结构，其中每一维的图层的大小与原来合并数据的大小保持一致。最后，使用 ResNet18 网络对该”图片“进行分类，并将最终结果用一个 $1000 \rightarrow N_{gen}$ 的全连接层输出每一个机组开机的概率①。智能体的结构如图3.2所示。

也就是说该智能体接受每一个时刻的系统状态变量，并且给出该时刻对应的机组启停变量作为输出。由于输入和输出都是针对某一个时刻而言的，智能体实际上实现的是在每一个时间断面上的机组组合问题，并没有完成在整个时序上的串联。考虑到机组组合问题中存在着时序约束（最短开停机时间约束、最大启停次数约束），还需要强化学习来实现时序上的约束，从而完成整个机组组合问题。

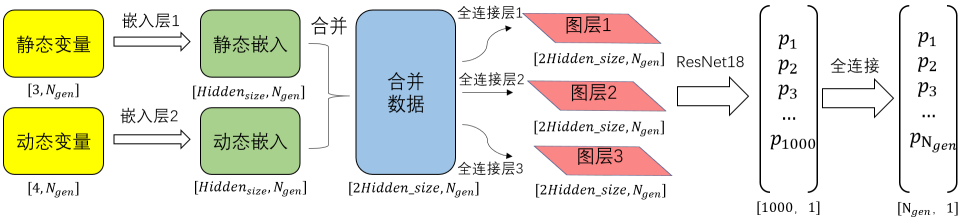


图 3.2　智能体结构示意图

在该网络中，按照如下方式进行参数的初始化。对于 ResNet 中的卷积层，使用 kaiming_normal 方法[27]进行初始化，即参数在 $N(0,\sigma)$ 中进行采样，其中 σ 使用式(3.1)进行计算；且其 bias 取为 0。ResNet 中的 BatchNorm 和 GroupNorm 的参数初始化为 1，bias 初始化为 0。

$$\sigma^2 = \frac{2}{f\_out}$$
$$f\_out = out\_channels \times kernel\_size^2 \tag{3.1}$$

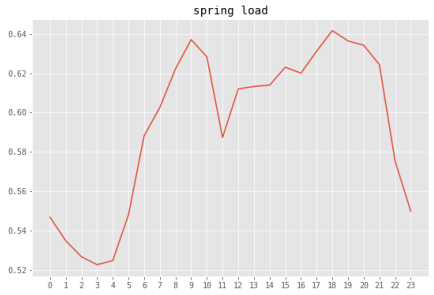
① 在强化学习部分会进一步对该概率使用 Sigmoid 进行激活，使之有概率的意义

3.5 算例分析

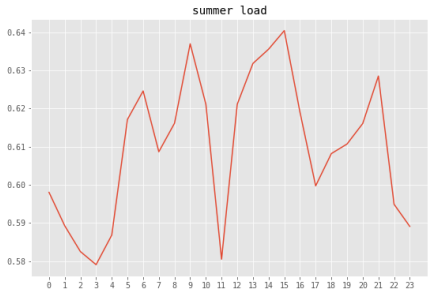
本节中我们使用一个实际算例来验证本章中提出的针对机组组合的模仿学习算法。该算例使用的是电气电子工程师学会可靠性小组委员会在 1979 年提出的 IEEE 可靠性测试系统[28]，该系统是用于测试稳定性分析方法的基准。在该系统中，包含负荷模型、发电机模型以及传输线模型。通过给定每一天的负荷曲线，我们对机组组合问题进行运行模拟。

3.5.1 实际负荷数据

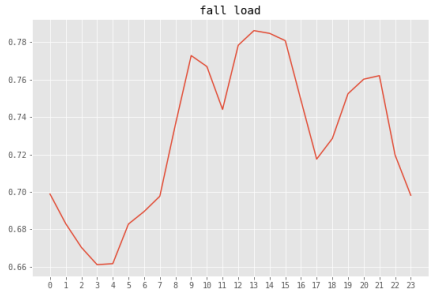
本文中所使用的数据来源于江苏省 2016 年全年共 365 天全天共 24 小时的实际负荷曲线，并且对其进行标幺化处理①。在这 365 条曲线中，分别选取在每一个季度中的负荷数据，并对其做时序上的平均并做标幺化，得到负荷曲线如3.3所示。从图中可以看出在不同季度中负荷的峰出现时间、高度都有较大区别，表现出一定的季节性。而在日内，表现出凌晨用电量少，而白天用电量大的现象，这可能是由于人们的作息规律导致。



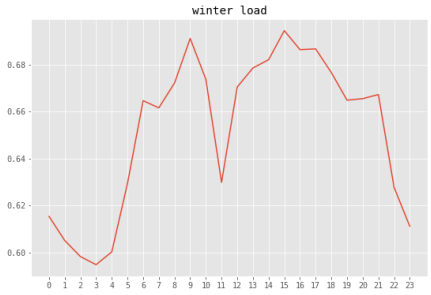
(a) 春季负荷



(b) 夏季负荷



(c) 秋季负荷



(d) 冬季负荷

图 3.3 典型负荷数据

将每一个季度中的每一天中的负荷取平均值，再对其进行统计分析，可得表3.1。

① 将全年每一时刻的负荷除以全年最高负荷

从表格中可以看到在秋冬季节平均负荷会更高，这可能是由于在2016年江苏省尚未实行供暖，而人们在冬天更多的使用电暖设备进行供暖，造成负荷较高。

表3.1　日均负荷统计表

季节	总数	平均值	标准差	最小值	25%	50%	75%	最大值
春季	89.0	0.593536	0.104571	0.306140	0.559048	0.620056	0.659635	0.733315
夏季	93.0	0.609814	0.039677	0.470162	0.598637	0.610766	0.622503	0.720589
秋季	93.0	0.728457	0.089740	0.540684	0.659149	0.700634	0.810085	0.906161
冬季	92.0	0.652009	0.048198	0.482186	0.630495	0.648479	0.680145	0.739434

3.5.2　生成负荷数据

在模仿学习和强化学习过程中，需要大量的数据才能让智能体比较好地学习该问题，并最终得出一个比较好的结果。目前对于江苏省的真实负荷数据仅有365条负荷曲线显然是不够的。因此，本节对江苏省2016年共365条负荷数据进行扩展。

对于每一天的负荷曲线，首先计算当天负荷的标准差表示当天负荷的波动情况。然后，基于该标准差生成50条长度为24的白噪声序列，该白噪声序列的标准差即为当天负荷的标准差。然后，对生成的负荷数据进行截断处理：由于此处为负荷的标幺值，因此应只保留值在[0,1]范围内的数据，否则截断为0或1。最后将这50条数据，随机平均划分为两个部分各25条数据，分别用于模仿学习与强化学习。在图3.4中，展现了对于2016年1月1日这一天进行扩展产生的三条负荷曲线与原负荷曲线的对比。

3.5.3　参数设置

本节说明在模仿训练中所设置的参数情况。

1. **智能体**

   在智能体中需要确定 N_{gen} 和Hidden_size两个参数。N_{gen} 是系统中机组的个数，对于RTS[28]系统来说，该系统一共包含33个机组。因此有 $N_{gen}=33$。其次Hidden_size是静态变量和动态变量在嵌入操作过后投影到的更大的隐藏层中。为了让该嵌入操作能够更加充分、更加透彻地解析机组组合问题的数据，选择一个足够大的隐藏层，选择Hidden_size=64

2. **优化器**

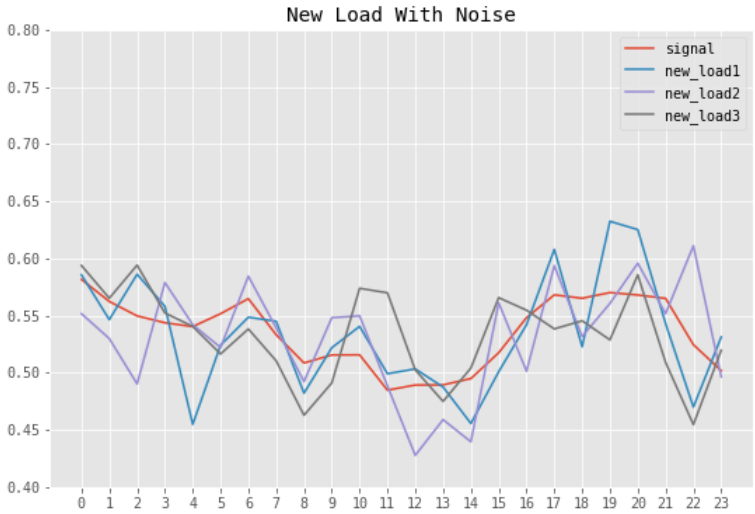


图 3.4　负荷数据生成示意图

在模仿学习中，采用 Adam 优化器[29]来实现对智能体的优化过程。Adam 优化器使用随机优化方法，基于对梯度的估计结果给出不同参数的自适应学习率，使得其能够实现优化过程中的高效计算和低内存占用。为了选择更优的学习率参数，本文对 lr=0.0001 和 lr=0.0005 分别进行了测试，最终选择 lr=0.0005。

3. **训练数据**

在3.5.2节中，我们通过加入白噪声的方法对 365 条负荷曲线中的每一条都额外构造了 25 条生成曲线作为模仿学习的数据。为了在这个监督学习过程中检查学习效果，将这总共 $365 \times 25 = 9125$ 个数据按照 9：1 分为训练集和测试集。

4. **训练过程**

在训练过程中，需要确定训练循环次数 epoch 以及 batch_size 两个参数。epoch 决定进行多少轮对于全部数据的训练，为了得到更好的训练效果，设置为 epoch=30。batch_size 决定在进行 mini-batch 训练[30]中，每一个 batch 的大小，设置为 batch_size=128 。

3.5.4　训练结果

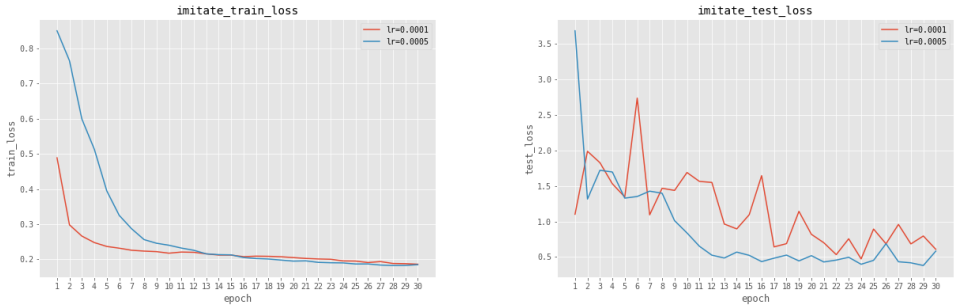
本文选择行为克隆作为模拟学习的算法，将专家的决策经验划分为 (s, a) 状态动作对，并分别将状态和动作作为监督学习的输入和标签进行监督学习。本节对

lr=0.0001 和 lr=0.0005 进行测试，并报告不同学习率下的模型的表现差别。

图3.5给出了模型在训练集和测试集上的误差。从图3.5(a)可以看出，使用 lr=0.0001 时训练集上的误差能够快速下降，而使用 lr=0.0005 的时候训练集上的误差下降较慢。但是当 epoch 超过 14 之后, 两者在训练集上的误差就不大了。说明当训练次数上升后，小学习率带来的快速下降的效果就会消失。

从图3.5(b)可以看出，使用 lr=0.0001 时在测试集上的表现不太稳定，虽然在初始值优于 lr=0.0005，但是很快就会被 lr=0.0005 的模型所超越。对于 lr=0.0005 的模型来说，其也能以比较快的速度达到收敛，并且整体表现比较稳定，不会产生较大的起伏。

因此，选用 lr=0.0005 的模型作为模仿学习的结果，并以之为下一步强化学习的初始模型。



(a) 训练集误差　　(b) 测试集误差

图 3.5　不同参数下模仿学习训练结果

3.6　情景分析

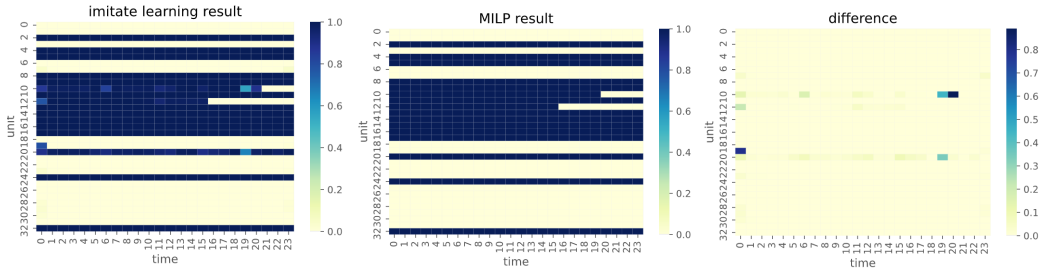
为了检验模仿学习的成果，使用真实的负荷曲线作为测试集来进行检测。

定义误差为 MILP 方法给出的机组启停 0/1 变量与智能体给出的启停概率 P（P∈[0,1]）之差的绝对值。则对于每一条负荷曲线来说，其误差为全天内全部机组的误差之和，计算公式如(3.2)所示。

$$L = \sum_{i=1}^{N_{gen}} \sum_{j=1}^{N_T} |p_{ij} - U_{ij}| \tag{3.2}$$

根据上述误差定义，选择其中误差最小的一条曲线得到其在全天时间内的模仿学习机组启停调度情况，与 MILP 算法得到的结果进行比较，得到图3.6。从图

中可以看出，在最优的结果下模仿学习给出的智能体能较好地完成对 MILP 方法的模仿，仅仅在少数几次决策中出现了与 MILP 算法不相同的情况。



(a) 模仿学习调度计划　(b) MILP 调度计划　(c) 模仿学习误差

图 3.6　模仿学习最优结果

给出在测试集上的误差统计数据如表3.2所示。其中，误差平均值为 34.1006，这说明在全天 24 小时对于 33 个机组的决策过程中，产生的平均总误差为 34.1006。则对于每一个位置上的平均误差为 $34.1006 \div (33 \times 24) = 0.043$。因此，在每个位置上产生的平均误差较小，说明该模型较好地学习了 MILP 方法。

表 3.2　模仿学习误差统计表

mean	std	min	25%	50%	75%	max
34.1006	16.6298	4.6840	22.0262	31.8629	45.3592	89.9311

3.7　本章小结

基于上一章中引入的 MDP 模型，本章引入了模仿学习中的行为克隆算法，给出了一个基于 ResNet 网络的智能体。行为克隆是指让智能体使用基于 MILP 优化方法给出的某些场景下的状态-动作决策对作为输入和标签进行强化学习。然后，本章基于在深层网络中有良好表现的 ResNet 网络设计了一个用于机组组合问题的网络结构，作为单时段下的机组组合决策模型。由于实际负荷曲线仅有 365 条，本章通过加入白噪声的方法将每一条负荷曲线扩展成 50 条新的生成负荷曲线，并将其按照每天平均分成两部分作为模仿学习和强化学习的训练数据。通过训练后，该智能体能够在单时段机组组合决策问题中表现良好。

第 4 章　机组组合强化学习

4.1　概述

在第3章中，我们使用了模仿学习中的行为克隆算法使得智能体获得了对机组组合问题的先验知识。但是由于在模仿学习的行为克隆算法中，给出的是针对一个时间截面的状态动作对，使得智能体只能学习在某个时刻机组组合结果，并不能对一天内的连续时间内生成完整的调度计划。考虑到机组组合问题中除了线路有功潮流约束、网络功率平衡约束等横截面上的约束外，同时还存在着日内最大启停次数约束与最小开停机时间约束等时序上的约束，仅仅依靠模仿学习的结果是不足够的。因此，本章引入强化学习来实现在该问题上的时序上的约束。

4.2　主要符号对照表

G_t	在时刻 t 后的累计奖励
R_t	在时刻 t 执行动作后得到的奖励
$V_\pi(s)$	使用策略 π，状态 s 的价值
$Q_\pi(s,a)$	使用策略 π，在状态 s 采取动作 a 的价值
V_*, Q_*, π_*	最优状态价值函数，最优状态动作价值函数，最优策略
γ	未来奖励折现系数,$\gamma \in [0,1]$
$P^a_{ss'}$	从状态 s 采用动作 a 达到状态 s' 的概率
θ	策略梯度下的策略函数的参数
$J(\theta)$	策略函数基于参数 θ 所得到的期望奖励
α	θ 的更新步长
ϵ	Actor 网络中的贪心系数，ϵ 越小代表 Actor 自由探索的程度越大。$\epsilon \in [0,1]$
P	Actor 网络给出的机组启动概率向量
$\tilde{P}$	调整后的机组启动概率向量
P_+, P_-	机组违反正备用约束和负备用约束的惩罚项
U_+, U_-	机组违反正负备用约束的哑变量，1 表示违反，0 表示没有违反
$R_{forward}$	策略模型每前进一步给予的奖励，目的是使得策略模型能给出更长的决策序列

4.3 主要构成

在强化学习问题中，智能体需要与陌生的环境进行互动并且在互动中获得奖励。智能体需要在这个过程中采取动作使自己可以获得最大的累计奖励。而强化学习算法的目标则是通过智能体与陌生环境互动的过程中的状态、动作与奖励等信息来学习该问题下的最优策略，让智能体可以学习环境并采取最优策略。

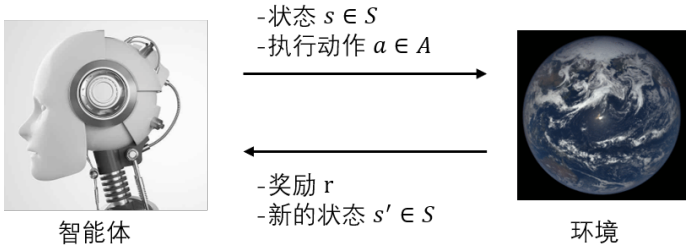


图 4.1　智能体与环境互动并采取策略来最大化累计奖励

在强化学习中，存在以下几个关键组成部分：问题模型、策略、价值函数以及最优价值函数和最优策略。接下来的几个小节中，我们展开讨论这几个组成部分。

4.3.1 问题模型

强化学习中的模型是描述环境的关键要素，应当根据实际问题来决定。在模型中存在着两个关键内容：转移概率函数 P 和奖励函数 R，其描述了智能体是如何与环境进行交互的，以及交互最终产生的结果如何。例如，当智能体在状态 s 采取了动作 a 达到了下一个状态 s'，并且得到了奖励 r。那么这就构成了一个完整的状态转移的一步，可以用元组 (s, a, s', r) 表示。记 $\mathbb{P}$ 为概率，定义转移函数、状态转移函数以及奖励函数。

- 转移函数

$$P(s', r|s, a) = \mathbb{P}[S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a] \tag{4.1}$$

- 状态转移函数

$$P_{ss'}^{a} = P(s'|s, a) = \mathbb{P}[S_{t+1} = s'|S_t = s, A_t = a] = \sum_{r \in \mathcal{R}} P(s', r|s, a) \tag{4.2}$$

- 奖励函数

$$R(s,a)=\mathbb{E}[R_{t+1}|S_t=s,A_t=a]=\sum_{r\in\mathcal{R}}r\sum_{s'\in\mathcal{S}}P(s',r|s,a) \tag{4.3}$$

在机组组合问题中，问题模型是已知的。实际上，在第2章中已经对 MDP 进行了建模，动作空间即为机组在当前的动作，状态空间为相关的动静态变量。因此，机组组合问题中下一刻状态在动作给出后是确定的：若动作满足安全约束，则下一刻状态完全由该动作确定；若动作不满足安全约束，则固定违反约束的机组状态不变，其他机组按照给定的动作执行，此时下一刻状态也是确定的。因此，转移函数是已知的。而奖励函数 $\boldsymbol{R}$ 的意义比较直观，即为每一时刻产生的机组成本的相反数。此时最大化累计奖励的过程相当于最小化累计成本，则可以满足我们的目标。

4.3.2 策略

强化学习中，策略 $\boldsymbol{\pi}$ 是智能体根据状态给出的动作。在本文中，动作是部分随机的：在智能体给出了机组启动的概率后加入一定的随机因素，使得智能体不完全遵循给定策略，而是提供了一定探索的空间。在强化学习中，对环境的探索和对已有经验的利用哪一样更加重要是人们需要权衡的问题。因此，本文中的策略可以写作式4.4，其中 Q 为引入的随机因素，并要保证 $\pi(a|s)\leqslant 1$。

$$\pi(a|s)=\mathbb{P}_{\pi}[A=a|S=s]+Q \tag{4.4}$$

4.3.3 价值函数

价值函数衡量一个状态或者价值有多“好”的指标。这里的“好”指在到达这个状态或采取这个动作后，能够获得尽可能高的预期累计奖励。通常来说，预期累计奖励是对未来预期奖励使用折现因子 $\gamma(\gamma\in[0,1])$ 折现后的结果。因此，预期累计奖励有以下形式。

$$G_t=R_{t+1}+\gamma R_{t+2}+\cdots=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1} \tag{4.5}$$

状态价值函数 V 则是指在时间 t 下，该状态能够提供的预期累计奖励。

$$V_{\pi}(s)=\mathbb{E}_{\pi}[G_t|S_t=s] \tag{4.6}$$

状态动作价值函数 Q 则是指在某状态下采取某个动作得到的预期累计奖励。

$$Q_{\pi}(s,a)=\mathbb{E}_{\pi}[G_t|S_t=s,A_t=a] \tag{4.7}$$

当给定策略 π 的时候，由于在每个状态下各个动作的概率都已经给出，所以可以得到状态价值函数 V 的另一个形式。

$$V_{\pi}(s)=\sum_{a\in\mathcal{A}}Q_{\pi}(s,a)\pi(a|s) \tag{4.8}$$

在机组组合问题中，同样能满足当前负荷要求的动作可能会产生不一样的价值。例如，负荷曲线在某一刻后会已知保持很高的负荷，而在当前时刻仅开小机组就能满足要求。由于可以预期需要较大的出力，当前时刻就把大机组打开会是一个更加明智的选择。

4.3.4 最优价值函数和最优策略

最优价值函数能够产生最大的累计奖励。

$$V_{*}(s)=max_{\pi}V\pi(s),Q_{*}=max_{\pi}Q_{\pi}(s,a) \tag{4.9}$$

最优策略能够产生最优价值函数。

$$\pi_{*}=argmax_{\pi}Q_{\pi}(s,a) \tag{4.10}$$

显然，$V_{\pi_{*}}(s)=V_{*}(s)$ 与 $Q_{\pi_{*}}(s,a)=Q_{*}(s,a)$ 本文的目标即是找出最有价值函数与最优策略，使得总的累计奖励最大化（累计成本最小化）。

4.4 贝尔曼方程

4.4.1 一般贝尔曼方程

贝尔曼方程是强化学习中的基石，其将价值函数分解为即期奖励和未来折现价值两部分。由于该方程把某个状态价值（状态动作价值）进行了分解，通过即期奖励和未来折现价值来表示，能够极大地化简强化学习中的优化问题。因此，其在动态规划、蒙特卡洛估计以及时序差分学习中都有着很重要的作用。

$$\begin{aligned}V(s)&=\mathbb{E}[G_t|S_t=s]\\&=\mathbb{E}[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots|S_t=s]\\&=\mathbb{E}[R_{t+1}+\gamma(R_{t+2}+\gamma R_{t+3}+\cdots)|S_t=s]\\&=\mathbb{E}[R_{t+1}+\gamma G_{t+1}|S_t=s]\\&=\mathbb{E}[R_{t+1}+\gamma V(S_{t+1})|S_t=s]\end{aligned} \tag{4.11}$$

对于状态动作价值函数来说，其贝尔曼方程为以下形式。

$$\begin{aligned} Q(s,a) &= \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) | S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} + \gamma \mathbb{E}_{a\ \pi} Q(S_{t+1}, a) | S_t = s, A_t = a] \end{aligned} \tag{4.12}$$

4.4.2 期望贝尔曼方程

当采用策略 π 的时候，如果将该迭代过程再前进一步的话，可以得到期望贝尔曼方程。期望贝尔曼方程同时基于状态价值函数与状态价值函数。

$$\begin{aligned} V_\pi(s) &= \sum_{a\in\mathcal{A}} \pi(a|s)(R(s,a) + \gamma \sum_{s'\in\mathcal{S}} \in \mathcal{S} P^a_{ss'} V_\pi(s')) \\ Q_\pi(s,a) &= R(s,a) + \gamma \sum_{s'\in\mathcal{S}} P^a_{ss'} \sum_{a'\in\mathcal{A}} \pi(a'|s') Q_\pi(s',a') \end{aligned} \tag{4.13}$$

4.4.3 最优贝尔曼方程

如果我们有最优价值函数，则由此可以直接根据最优价值来选择动作，而不用依赖于策略。此时的最优贝尔曼方程如下所示。

$$\begin{aligned} V_*(s) &= max_{a\in\mathcal{A}}(R(s,a) + \gamma \sum_{s'\in\mathcal{S}} \in \mathcal{S} P^a_{ss'} V_\pi(s')) \\ Q_\pi(s,a) &= R(s,a) + \gamma \sum_{s'\in\mathcal{S}} P^a_{ss'} max_{a'\in\mathcal{A}} Q_\pi(s',a') \end{aligned} \tag{4.14}$$

4.5 强化学习算法

求解强化学习有很多类方法，包括：动态规划、蒙特卡洛模拟方法、时序差分法以及策略梯度法，其中的每一类方法也能衍生出多种细分方法。其中，动态规划法、蒙特卡洛模拟法以及时序差分法都是通过与环境互动来学习状态价值函数以及状态动作价值函数，并依此对动作进行选择。而策略梯度法则是通过学习一个带有参数 θ 的策略函数 $\pi(a|s;\theta)$，其中的 θ 为需要学习的参数。

通常来说，策略梯度法更加适合连续空间下的强化学习问题。考虑到上述其他方法都依赖于状态价值函数和状态动作价值函数，在连续空间下这些状态的个数是无穷的，则策略在基于价值函数进行选择 $argmax_{a\in\mathcal{A}} Q^\pi(s,a)$ 的时候则会遇到维度灾难的问题。策略梯度法则只需要对策略函数中的 θ 进行优化即可解决该问题，因此策略梯度法更加适合连续空间下的问题。基于以上讨论，本文选择使用策略梯度法作为强化学习的求解方法。

4.5.1 策略梯度法

在4.1节中，我们已经讨论了强化学习的目标为找到令智能体能够得到最大的累计奖励。为了实现对策略的建模和优化的目的，策略梯度法通常对带有参数 θ 的策略 $\pi(a|s;\theta)$ 进行学习。

该策略函数基于参数 θ 以及当前所状态 s 给出对于动作 a 的价值估计。在该方法下，奖励函数如式(4.15)所示。其中，$d^{\pi}(s)$ 是使用策略 π_{θ} 下的马尔科夫链稳定分布。

$$J(\theta)=\sum_{s\in\mathcal{S}}d^{\pi}(s)V^{\pi}(s)=\sum_{s\in\mathcal{S}}d^{\pi}(s)\sum_{a\in\mathcal{A}}\pi_{\theta}(a|s)Q^{\pi}(s,a) \tag{4.15}$$

为了实现最终价值的最大化，策略梯度法应该实现 $J(\theta)$ 的最大化。因此，在策略梯度法的每一步中，都应该使得 θ 往使得 $J(\theta)$ 梯度上升的方向前进，如式(4.16)所示，从而获得最大的累计奖励。

$$\theta_{t+1}=\theta_t+\alpha\nabla J(\theta_t) \tag{4.16}$$

由于计算策略梯度需要根据策略选择的动作以及基于该策略下最终稳定状态，在环境未知的情况下计算策略梯度将会十分困难。为了方便求解 $\nabla_{\theta}J(\theta)$，在此引入策略梯度定理。策略梯度定理给出了一个简易的计算 $\nabla_{\theta}J(\theta)$ 的方法，在这篇文章中[31]给出了策略梯度定理的具体证明。

$$\nabla_{\theta}J(\theta)=\mathbb{E}_{\pi}[Q^{\pi}(s,a)\nabla_{\theta}\ln\pi_{\theta}(a|s)] \tag{4.17}$$

该定理告诉我们，在计算策略梯度的过程中我们即使没有状态的遍历分布以及环境就能计算策略梯度。在大部分实际场景下，由于对这两个变量进行建模有一定难度，策略梯度定理则为我们移除了该困难。策略梯度定理使得我们可以十分轻松地计算策略梯度，其成为策略梯度法中的各种算法的理论基础。

4.5.2 Actor-Critic 算法

Actor-Critic 算法是时序差分法的一种，其分别使用两个网络 Actor 和 Critc 来学习策略模型以及价值函数。其中，Actor 是策略网络，用于在给定状态下选择一个动作。这个网络的主要任务是在 Critc 网络的指导下优化策略，为 $\pi_{\theta}(a|s)$ 更新参数 θ。Critic 是价值网络，用于给出状态价值，从而指导 Actor 优化的方向。Critic 会在每一次 Actor 给出动作后对新的状态进行评价，告诉 Actor 在采用该动作之后

相比预期问题是向更优发展或是向更差发展了，并且通过时序差分误差（TD error）来进行表述。

$$\delta_t = r_{t+1} + \gamma V\left(s_{t+1}\right) - V\left(s_t\right) \tag{4.18}$$

其中 V 是当前的 Critic 的价值函数，用于评判状态的价值。而 TD error 则决定了 Actor 网络在状态 s_t 下决策出的动作 a_t 的价值及 Actor 网络更新的方向。从 TD error 的形式来看，当该动作产生的状态比较好的时候，$\delta_t > 0$，此时 Actor 网络应该更加倾向于选择该动作以得到更高的预期奖励。当该动作产生的状态比较差的时候，$\delta_t < 0$，此时 Actor 网络应该避免采取该动作，否则将会降低预期奖励。如果以 $p(s_t, a_t)$ 来作为在状态 s_t 下 Actor 采取动作 a_t 的概率，那么可以根据以下方法来做每一步的更新过程。对于 Critic 网络来说，目标是令其能够更准确地估计状态的价值，因此应该最小化 TD error。

$$p\left(s_t, a_t\right) \leftarrow p\left(s_t, a_t\right) + \alpha\delta_t \tag{4.19}$$

Actor-Critic 具体算法如下所示。

算法 4.1 Actor-Critic 算法

随机初始化状态 s，策略网络参数 θ 以及价值函数参数 ω。对动作进行采样 $a \sim \pi_\theta(a|s)$
for $t = 1, \cdots, T$ ：**do**
    对奖励采样 $r_t \sim R(s, a)$，对下一状态进行采样 $s' \sim P(s'|s, a)$
    对下一个动作进行采样 $a' \sim \pi_\theta\left(a' \mid s'\right)$
    更新策略网络参数 $\theta \leftarrow \theta + \alpha_\theta Q_w(s, a)\nabla_\theta \ln \pi_\theta(a \mid s)$
    计算在时刻 t 的时序 TD error，$\delta_t = r_t + \gamma Q_w\left(s', a'\right) - Q_w(s, a)$，并且用该误差更新价值函数 $w \leftarrow w + \alpha_w \delta_t \nabla_w Q_w(s, a)$
    更新动作 $a \leftarrow a'$ 和状态 $s \leftarrow s'$
end for

综上所述，Actor-Critic 算法能够在连续空间的强化学习问题中有着较好的表现，并且有着以下优点。

1. Actor-Critic 算法选择动作时计算量最小。由于 Actor-Critic 算法不像强化学习的其他算法一样只是储存状态价值，而是使用 Actor 网络直接给出各动作的概率，因而在其进行状态选择时不需要对状态空间内的所有状态进行一遍搜索，节省了大量的计算量。Actor-Critic 算法也因此能够处理连续状态空间的强化学习问题。
2. Actor-Critic 算法能够显式地学习一个策略网络，从而给出在每种状态下的每种动作执行的概率。

在机组组合问题中的状态空间中同时包括连续变量和离散变量，连续变量包括机组出力、未来负荷等，离散变量包括上一时刻机组启停状态、状态转变冷却时间等。鉴于该状态空间存在连续变量，因此本文选择方便求解连续状态空间的Actor-Critic 算法进行强化学习问题求解。

4.6 网络设计

4.6.1 安全约束实现

机组组合问题是在要求解在满足电力系统安全约束的前提下在全天二十四小时时间跨度内满足负荷需求的机组启停调度规划，而在以上的内容中都未涉及对于电力系统安全约束的保证。在本节中，提出使用优化限制、屏蔽函数、惩罚函数相结合的方法来实现在第2.4节中所定义的电力系统安全约束。

4.6.1.1 OPF 优化限制

在安全约束中，线路有功潮流约束与网络功率平衡约束是在给定机组启停动作后进行最优潮流（Optimal Power Flow，OPF）优化计算中需要保证的约束。线路有功潮流约束描述在各节点上的潮流情况。网络功率平衡约束要求整个电力系统网络中总的出力应等于总的负荷要求。这两个约束是在给定机组的启停状态后，在计算 OPF 的时候需要得到满足。

除了线路有功潮流潮流与网络功率平衡约束外，还有机组爬坡约束与机组出力上下限约束。由于火电机组的能量转换过程需要从燃料与水蒸气的内能转换为机械能，最后再转换为电能。这是一个比较缓慢、惯性较大的过程，也就决定了火电机组在相邻时刻的出力不能发生突变，于是就有了爬坡约束。同理，火电机组的出力峰值是有限的，取决于机组本身设计的最大容量。同时，由于靠燃煤来进行能量发电，出力一定会高于一个大于零的最低阈值。因此机组出力存在着上下限约束。爬坡约束与机组出力上下限约束同样是在给定机组的启停状态后，在计算 OPF 的过程中，对机组出力本身的约束条件。

在本文计算 OPF 过程中，使用 pandapower[22] 包实现 OPF 的计算。在该软件包中，可以在经典的电力系统网络中自定义机组的启停状态、出力上下限、负荷大小等参数，然后再计算该状态下的最优潮流。因此，上述四个约束可以通过以下过程来实现：首先通过上一时刻机组出力与爬坡约束及机组出力上下限约束给出在当前时刻机组的出力上下限，然后再使用 pandapower 包求解该状态下的 OPF 问题。这样即可以通过计算 OPF 优化问题的方法同时满足四个约束条件。

4.6.1.2 屏蔽函数

在安全约束中，最小开停机时间约束与日内最大启停次数约束是在全天二十四小时内的序贯决策过程中需要满足的时间尺度上的约束。最小开停机时间约束指传统机组在机组的状态转换之间需要有一定时间间隔。日内最大启停次数则是为了保护机组，延长机组使用寿命而在一天之内不宜过多次开停机。这两个约束都是为了保证机组本身安全而在时间尺度上增加的约束。

为了满足在时序上的约束，本节引入屏蔽函数来对 Actor 网络给出的动作进行合法性检查。该屏蔽函数实际上起到了 MDP 问题中的转移核的效果，对于 Actor 网络给出的动作，如果合法则直接影响下一时刻的状态；否则控制违反最小开停机时间约束与日内最大启停次数约束的个别机组状态不能改变，再影响下一时刻的状态。

屏蔽函数的实现方法如下所示。在每一时刻 Actor 给出动作后，根据目前所记录的机组的启停次数变量与距离上一次开停机时间变量来维护一个长度为 N_{gen} 的描述机组是否满足安全约束的数组 Mask。在 Mask 中，满足安全约束的机组对应元素值为 1，否则对应元素值为 0。在得到 Mask 函数后，将其对数值加入 Actor 给出的概率后再使用 Sigmoid 函数转化为 [0,1] 之间的概率。屏蔽函数实现流程图如图4.2所示。

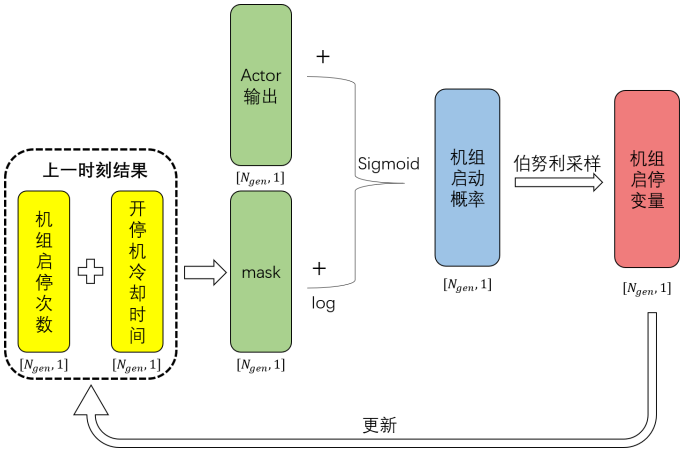


图 4.2 屏蔽函数实现机制

4.6.1.3 惩罚函数

在安全约束中，正负备用约束是指为了保证电力系统运行安全，除了要求线路上功率与负荷相匹配，还要求开机机组保留正负备用容量，以保证在某些突发事故中（如 n-1 事故）电力系统仍然能够保证满足负荷端的要求。但是，考虑到正

负备用约束是针对整个电力系统的出力的约束而并非是针对单独机组出力的约束，因此不能使用屏蔽函数的方法来实现该约束。

由于强化学习的目标是获得最大的累计奖励，因此可以通过影响强化学习中每一步的奖励（成本的相反数）来影响策略模型的偏好。因此，本节使用在每一步的成本中加入惩罚函数的方法来指导策略模型尽量避免违反正负备用约束。当 Actor 给出的动作导致正负备用约束不能满足的时候，则在成本上加上惩罚项 P_+ 或者 P_-。同时，我们希望策略模型能够给出长度为二十四小时的决策计划，则需要让策略模型每前进一步都能获得正的奖励才能激励其继续向前。因此，我们加入前进奖励 $R_{forward}$ 以使得模型每次向前进都能获得正的奖励。

同时考虑违反正负备用约束的惩罚和前进奖励，可以得到新的成本函数式(4.20)。其中 C 表示运行成本，P_+ 与 P_- 表示惩罚项的大小，U_+ 与 U_- 是表示是否违反正负备用约束的哑变量，$R_{forward}$ 是前进奖励的大小。需要注意的是，此处的 $P_-, P_+, R_{forward}$ 都是超参数，而且 $R_{forward}$ 的值应该足够大，使得 $\tilde{R}$ 是正数。当 $\tilde{R}$ 是正数则策略模型每次成功的决策都能让其获得正的奖励，从而让其学习一个给出更长的决策计划的策略。

$$\tilde{R} = -C_{sys} - P_+U_+ - P_-U_- + R_{forward} \tag{4.20}$$

4.6.2 Actor 网络

在第3章中，我们采用了模仿学习算法，利用行为克隆方法训练得到一个基于 ResNet 网络的能根据给定的状态输出每个机组启停状态的神经网络。该网络接受某一时刻的状态变量，先后经过数据嵌入、维数扩展以及 ResNet 网络，最终输出使用全连接网络转换得到的每个机组的启动的概率。但是，该神经网络只能够处理某一个时刻的截面上的机组组合问题，不能处理全天二十四小时的时序的机组组合决策问题，因此仍然需要使用强化学习解决时序上的要求。

在 Actor-Critc 算法中，需要使用 Actor 网络给出在给定状态下的动作，需要一个能够处理截面问题的网络，则可以使用上述模仿学习得到的神经网络。通过使用强化学习，则可以将只能处理截面问题的策略模型扩展到能处理时序决策问题的策略模型。回顾使用模仿学习训练的网络，其结构如图所示。该网络接受某一个时刻的静态变量和动态变量，并最终输出一个列向量 P，列向量 P 中每一个元素表示某一个机组启动的概率。

在强化学习中需要平衡对于环境的探索以及对于已有经验的利用。在 Actor 网络中，在不同任务下有着不同的处理方法。当模型在训练的时候，为了让模型能

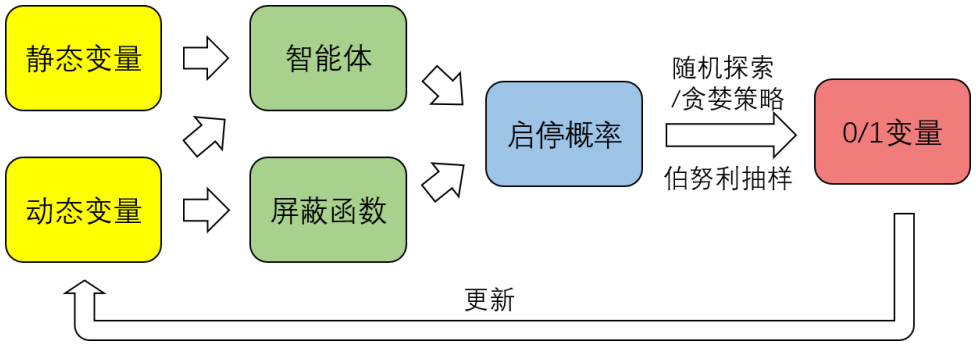


图 4.3　Actor 网络结构示意图

够兼顾对环境的探索以及对已有经验的利用，使用下式对概率进行调整。其中 $\boldsymbol{P}$ 为模型给出的概率向量，而 ϵ 是一个平衡探索与利用的系数。该式子的含义是进一步扩大每台机组开机的可能性，从而使算法可以探索其他可能的调度计划。该方法类似于强化学习中的 ε-greedy 优化方法[32]。当模型训练结束后处于使用模式时，则应完全根据算法给出的动作概率进行动作的选择，从而达到最大化累计奖励的目标。

$$\tilde{P} = \begin{cases} \varepsilon P + (1-\varepsilon) & ,if \quad train \\ P & ,if \quad eval \end{cases} \tag{4.21}$$

由于机组的启停状态应该是 0/1 整形变量，因此基于上述给出的启动概率向量，还应该进一步据此进行抽样，进而得到每一个机组的启停 0/1 整型变量。得到调整后概率向量 $\tilde{\boldsymbol{P}}$ 后，再对其进行伯努利抽样。伯努利抽样是指依据伯努利分布进行抽样的方法，接受随机变量取 1 的概率，并最终随机抽样返回 0/1 结果作为下一时刻机组的启停 0/1 变量。

4.6.3　Critic 网络

按照4.5.2节的定义，Critic 网络需要在给定状态下给出对该状态的价值估计，并根据该价值估计指导 Actor 网络更新的方向。因此，将 Critic 网络设计成接收一个时间截面上的各个机组的状态变量，并且输出价值估计标量的网络。

在 Critic 网络中，与 Actor 网络相似，首先分别使用两个嵌入层将静态变量与动态变量投影到更加高维度的空间里，从而可以更好地发掘该状态下的信息。然后，将静态嵌入数据与动态嵌入数据进行合并得到合并数据。合并数据通过三层由一维卷积层与 ReLU 激活函数构成的网络后最终得到一个行向量。将该行向量求和则得到了最终对该状态的价值估计。

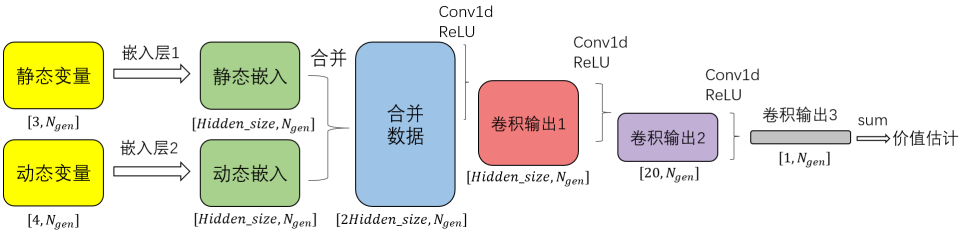


图 4.4　Critic 网络结构示意图

4.6.4　训练过程

根据 Actor 网络和 Critic 网络以及基于 OPF 求解的环境，则构成了如图4.5的训练过程。在 Actor 网络中，将静态变量和动态变量通过智能体和屏蔽函数给出了单时段内每个机组的启停状态 0/1 变量。该变量作为满足安全约束的机组动作，即可用于更新下一时刻的动态变量，此处的更新应同时更新对 Critic 网络的动态变量。同时，将该动作与环境进行交互，求解 OPF 得到该时段的成本。将该成本与 Critic 网络估计得到的状态价值估计进行比较，得到 TD error，从而对 Actor 网络和 Critic 网络进行更新，便完成了对于一个时刻的训练。将上述过程重复 24 次，便能得到一个 24 小时的机组调度计划，完成了一条负荷曲线的训练过程。

本文同时采用 mini-batch 的训练方法，每次对一个批的数据进行训练。使用 mini-batch 训练方法，可以加快训练速度，有效地提升训练后算法模型的表现[33]。在采用 mini-batch 训练的时候，当 batch 中某一部分负荷曲线在求解过程中给出了违反 OPF 的约束条件的机组动作，导致该 OPF 不能求解的时候，则对该曲线进行标记，并且对该曲线的求解在当前训练中退出。在最终进行误差回馈的时候，也只考虑前面成功求解部分的误差。

4.7　算例分析

4.7.1　数据

在3.5.2中使用给真实数据加上白噪声的方法扩展江苏省的 365 条真实负荷数据，对每天的数据都生成 50 条加白噪声的生成负荷曲线，最终得到了 18250 条生成的负荷数据。这 18250 条负荷数据是对每一天的负荷数据加入了 50 次白噪声得到的，因此将这 50 条构造的生成负荷曲线随机分成大小相等的两个部分，并且分别用于模仿学习和强化学习。强化学习中的训练数据来自上述过程，共得到 $365 \times 25 = 9125$ 条负荷曲线。

在每一轮训练的测试过程中，则直接使用真实的负荷数据作为测试数据。使

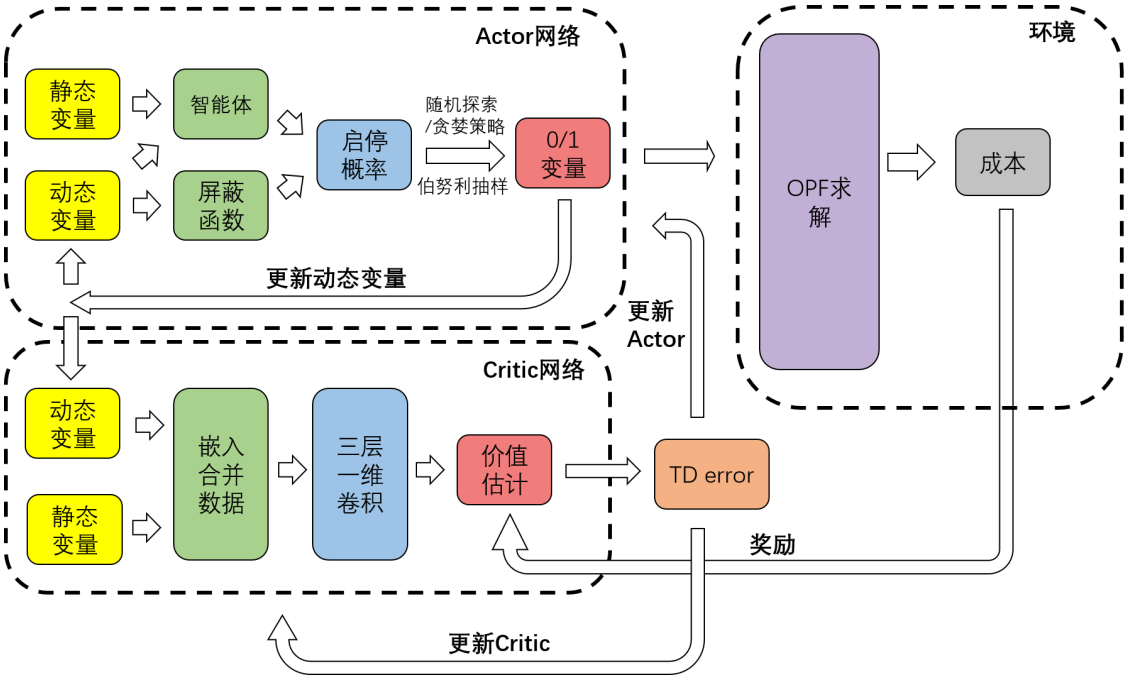


图 4.5　强化学习算法训练过程

用真实的负荷数据作为测试数据，可以检验算法从生成的数据中学习的成果，从而检验该算法在实际生产中的效果。

4.7.2　参数设置

1. **Actor 网络**

Actor 网络是基于第3章中使用模仿学习行为克隆算法得到的智能体进一步改造得到的网络。在该智能体的基础之上，本节中加入了屏蔽函数对安全约束进行保障；同时加入了对于环境的随机探索，用于让 Actor 能够在局部最优的情况下也有几率继续探索找到全局最优解。此处，在进行安全约束限制的时候，给定机组的一天内最大启停次数为 4 次，启停冷却最短时间为 2 小时，正负预备率都为 5%，爬坡速率为该机组最大输出功率的 40%。

对 Actor 网络使用的优化器是 Adam 优化器，使用的学习率从 0.0005 和 0.001 中进行选择。

2. **Critic 网络**

Critic 网络中，输入的动静态变量数据首先分别通过两个嵌入层得到嵌入数据，将嵌入数据合并后在经过三个一维的卷积层最终得到 Critic 网络对于状态的价值估计。本节记隐藏层的大小为 Hidden_size，并且设置 Hidden_size=64。则在嵌入层中，将动静态数据都映射成 [Hidden_size,N_{gen}] 大小

的数据，然后将其合并成 [2Hidden_size,N_{gen}] 的合并数据。接着使用三个连续的一维卷积网络，最终变成一个向量，其长度为 N_{gen}，代表对每个机组的价值估计。在这三个连续的一维卷积网络中，其大小变化按照如下模式进行：2Hidden_size→Hidden_size→20→1。将该向量进行求和，则得到对该状态的价值估计。

对 Critic 网络使用的优化器是 Adam 优化器，使用的学习率从 0.0005 和 0.001 中进行选择。

3. **训练过程参数**

本文使用 mini-batch 的方法进行强化训练。在训练过程中，需要确定训练循环次数 epoch 以及 batch_size 两个参数。epoch 决定进行多少轮对于全部数据的训练，为了得到更好的训练效果，设置为 epoch=30。batch_size 决定在进行 mini-batch 训练中，每一个 batch 的大小，设置为 batch_size=128 。

4.7.3 训练结果

由于对于 Actor 网络和 Critic 网络来说都需要调整学习率，因此本文对 lr_a 和 lr_b[①]进行测试。取 lr_a 和 lr_b 进行组合，则可以一共得到四组超参，使用网格搜索法，找到其中表现最好的一组超参。

使用不同超参数进行测试的结果如图4.6所示。

图4.6(a)描述了 Critic 所给出的状态价值估计与环境所给出的奖励之间的差距，该差距越小说明 Critic 对状态的估计越准确。从图中可以看出在经历 15 个 epoch 之后，基本上所有的超参数组合都能得到一个十分接近 0 的 Critic 状态价值估计误差，这说明当训练次数足够多的时候，Critic 网络能够精准地对机组所在的状态价值进行评估。

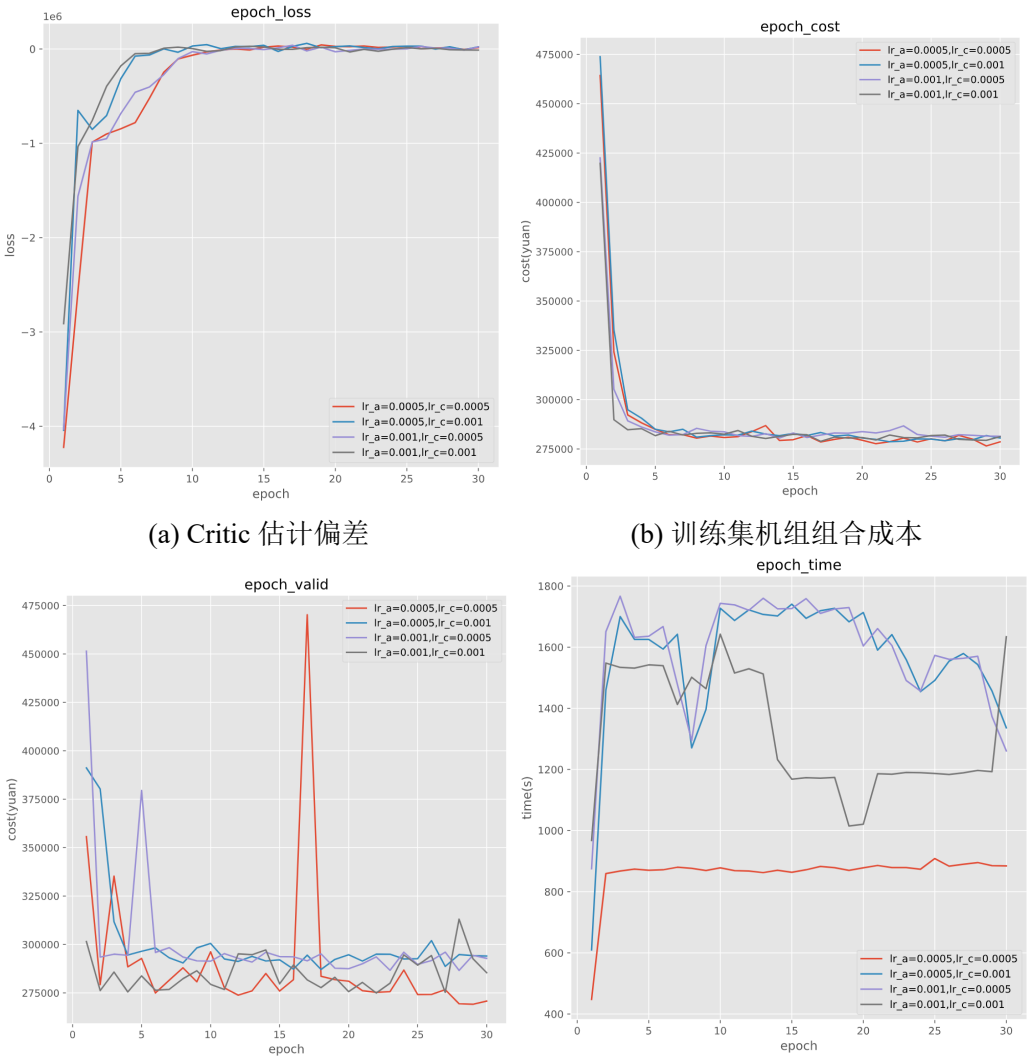
图4.6(b)描述了每次训练过程中，Actor 网络所给出的动作在训练集上产生的调度计划的平均成本。从图中可以看出，在训练刚开始的时候，所有的超参数组合对应算法产生的调度计划的成本都在 400000 以上，并且在前几轮训练中就能使得成本迅速下降。在以后的训练过程中，成本也有所下降但是并不明显。

图4.6(c)描述了每次训练过程中，Actor 网络所给出的动作在测试集上产生的调度计划的平均成本。测试集的数据完全来自真实负荷数据，能更加真实地反映算法模型的求解能力。从图中可以看出，与在训练集上的结果类似，在测试集中刚开始前几轮训练中机组组合成本会迅速下降，并且之后的下降幅度就不明显了。

① lr_a 和 lr_b 分别为 lr_actor 和 lr_critic 的缩写，代表 Actor 网络的学习率和 Critic 网络的学习率

此外，还存在着当超参组合选取 lr_a=0.0005,lr_c=0.0005,epoch=17 的时候会有一次巨大的波动，这说明在该超参组合下算法会不太稳定。

图4.6(d)描述了在每次训练过程中所花费的时间，可以看出在 lr_a=0.0005,lr_c=0.0005 这个超参组合下的平均花费时间最短。而其他超参组合下花费时间偏长。



(a) Critic 估计偏差　　(b) 训练集机组组合成本

(c) 测试集机组组合成本　　(d) 求解时间

图 4.6　不同参数下强化学习训练结果

在四种不同的超参组合中，得到的在测试集上的最低成本的记录如表4.1，该表中的索引按照 [lr_a,lr_c] 的形式给出。从表中可以看出，当 Critic 网络的学习率较大时（lr_c=0.001），算法能够在较短的训练次数中就达到最低的测试集成本。当超参组合选用 [0.0005,0.0005] 的时候，能够获得四组超参下的最低测试集成本。因此，当 Actor 网络和 Critic 网络的学习率较小时，在每一步迭代更新的过程中的步

表 4.1　不同超参组合下测试集最优解

超参组合	最优训练次数	最优测试集成本
[0.0005,0.0005]	28	269409.6798
[0.0005,0.001]	17	294500.0779
[0.001,0.0005]	27	296062.3793
[0.001,0.001]	21	280426.1092

长更短，因此也更加准确地导向最优解，避免了在最优解附近震荡。

4.8　本章小结

由于模仿学习给出的智能体仅能给出在单时段内的决策结果，而机组组合问题是对全天的调度进化进行研究。因此，本章使用强化学习的策略梯度方法将该智能体进一步训练成可以在满足电力系统安全约束的条件下解决机组组合问题的模型。首先，本章介绍了强化学习、策略梯度算法与 Actor-Critic 算法。然后，给出了基于 OPF 优化限制、屏蔽函数以及惩罚函数的三种方法解决机组组合问题中的电力系统安全约束限制。根据机组组合的时序决策模型，为策略模型进行每一步训练设计了训练过程，兼顾“探索”和“利用”两方面，实现了二十四小时内的决策过程。

第 5 章　总结与展望

在全球气候变暖与能源安全问题越来越受重视的当下，风电和光伏等可再生能源的发电技术越来越受到重视，我国能源结构也从传统的极大程度依赖火电机组逐渐向清洁能源方向发展。然而，引入清洁能源的同时也给电力系统带来了更大的随机性与不稳定性，这就为目前的电力行业市场化带来更大的挑战。在电力市场的背景下，要电网能够快速地对需求端的负荷做出反应。在某些情况下，要求在五分钟之内要完成出清，则需要在较短时间内求解机组组合问题，给出机组调度计划。但是，传统的将机组组合问题建模为混合整数线性优化问题的方法并不能很好地适应电力市场化的要求。一方面，混合整数线性优化问题的复杂度会随着电力系统规模的升高而急剧升高，导致对于大系统来说难以在短时间之内完成机组组合问题求解。另一方面，由于在每一天的机组组合问题之中，面对的负荷曲线有着相似的形态。如果每次使用优化方法进行求解的话，则难以对以前的求解结果加以利用，造成计算资源上的浪费。

为了解决优化问题的上述缺点，本文提出了使用强化学习与模仿学习相结合的方法来求解机组组合问题。强化学习是机器学习的一个分支，其通过让智能体与环境互动的方法来学习得到一个能够最大化累计奖励的策略。使用强化学习求解机组组合问题，在完成前期的训练之后，能够快速地完成对于新问题的求解，从而节省大量的计算时间。

本文首先分别将机组组合问题建模为混合整数线性优化问题以及 MDP。由于在这机组组合优化问题中的决策变量包含机组状态以及各机组出力，所以同时决策变量中同时包含整型变量与连续变量，构成了混合整数线性规划问题。在该优化问题中，还包含着电力系统中的各种安全约束。同时，本文还将机组组合问题建模为 MDP，将问题建模为马尔可夫决策过程是使用强化学习算法的前提条件。本文根据机组组合问题的特点，给出了 MDP 中的状态空间、动作空间、转移概率以及奖励函数的四元元组，完成了对于机组组合的 MDP 建模。

然后，本文为了降低强化学习的搜索难度引入了模仿学习。本文采用模仿学习中的行为克隆方法，先对部分情境下的优化问题进行求解从而给出一系列状态动作对 $< s, a >$，然而分别将状态和动作作为监督学习中的输入和标签来训练了一个基于 ResNet 网络的神经网络。该神经网络接受某个时刻的状态变量，然后给出该时刻的各机组启动概率，完成了在一个时间截面上的调度问题。

对于机组组合问题来说，需要完成的是在全天二十四小时内的序贯决策过程，因此仅能给出一个时间截面上的解答并没有解决该问题。此外，这样的解也没有考虑到时间上的约束。因此，本文引入强化学习完成对该问题的求解，使用策略梯度法中的 Actor-Critic 算法进行训练。使用强化学习，能够将模仿学习中得到的模型进行进一步的训练从而满足机组组合问题在时间上的约束。为了保证能够满足安全约束，本文分别引入了 OPF 优化限制、屏蔽函数以及惩罚函数来保证个体机组的约束以及对整个系统的约束。完成前期训练后，在面对新的机组组合问题的时候，仅需要提前估计当天的负荷曲线，该算法就能快速给出当天的机组调度计划。

综上所述，本文对传统机组的机组组合问题的快速求解进行研究。分别将机组组合问题建模为混合整数线性规划问题及 MDP 问题，然后使用模仿学习先训练出一个能解决时间截面上问题的模型，然后再用强化学习进一步训练该模型，以实现全天的机组组合问题求解。

在我国能源结构逐步清洁化的大背景下，清洁能源的引入为电力系统引入了更多的随机性，使得火电机组面对的负荷要求的变化更加剧烈。同时，在电力市场化的要求下，市场出清时间进一步缩短，对于机组组合问题的求解速度要求进一步加大。因此，快速求解机组组合问题成为一个重要的研究课题。传统的优化求解方法需要花费较长的时间，不能满足发展的需要。因此，使用数据驱动的人工智能算法来进行机组组合问题求解，可以快速、准确地完成对机组组合问题的求解。

插图索引

表格索引

参考文献

[1] 周孝信, 鲁宗相, 刘应梅, 等. 中国未来电网的发展模式和关键技术[J]. 中国电机工程学报, 2014, 34(29): 4999-5008.

[2] 杨涛, 朱琳. 大数据分析技术在电力期货交易市场中的应用研究[J]. 电气技术与经济, 2020, 03: 57-61.

[3] 刘羽霄, 张宁, 康重庆. 数据驱动的电力网络分析与优化研究综述[J]. 电力系统自动化, 2018, 42(6): 157-167.

[4] Lee F N. Short-term thermal unit commitment-a new method[J]. IEEE Transactions on Power Systems, 1988, 3(2): 421-428.

[5] Senjyu T, Miyagi T, Yousuf S A, et al. A technique for unit commitment with energy storage system[J]. International Journal of Electrical Power & Energy Systems, 2007, 29(1): 91-98.

[6] Cohen A I, Yoshimura M. A branch-and-bound algorithm for unit commitment[J]. IEEE Transactions on Power Apparatus and Systems, 1983(2): 444-451.

[7] Snyder W L, Powell H D, Rayburn J C. Dynamic programming approach to unit commitment [J]. IEEE Transactions on Power Systems, 1987, 2(2): 339-348.

[8] Virmani S, Adrian E C, Imhof K, et al. Implementation of a lagrangian relaxation based unit commitment problem[J]. IEEE Transactions on Power Systems, 1989, 4(4): 1373-1380.

[9] Carrión M, Arroyo J M. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem[J]. IEEE Transactions on power systems, 2006, 21(3): 1371-1378.

[10] Ostrowski J, Anjos M F, Vannelli A. Tight mixed integer linear programming formulations for the unit commitment problem[J]. IEEE Transactions on Power Systems, 2011, 27(1): 39-46.

[11] Cheng C P, Liu C W, Liu C C. Unit commitment by lagrangian relaxation and genetic algorithms [J]. IEEE transactions on power systems, 2000, 15(2): 707-714.

[12] Ouyang Z, Shahidehpour S. Short-term unit commitment expert system[J]. Electric power systems research, 1990, 20(1): 1-13.

[13] Kumar S S, Palanisamy V. A dynamic programming based fast computation hopfield neural network for unit commitment and economic dispatch[J]. Electric power systems research, 2007, 77(8): 917-925.

[14] Saber A Y, Senjyu T, Yona A, et al. Fuzzy unit commitment solution—a novel twofold simulated annealing approach[J]. Electric Power Systems Research, 2007, 77(12): 1699-1712.

[15] Dudek G. Unit commitment by genetic algorithm with specialized search operators[J]. Electric Power Systems Research, 2004, 72(3): 299-308.

[16] Dudek G. Adaptive simulated annealing schedule to the unit commitment problem[J]. Electric Power Systems Research, 2010, 80(4): 465-472.

[17] Dalal G, Mannor S. Reinforcement learning for the unit commitment problem[C]//2015 IEEE Eindhoven PowerTech. IEEE, 2015: 1-6.

[18] 张钦, 王锡凡, 王建学, 等. 电力市场下需求响应研究综述[J]. 电力系统自动化, 2008.

[19] Gurobi Optimization L. Gurobi optimizer reference manual[EB/OL]. 2021. http://www.gurobi.com.

[20] Kaelbling L P, Littman M L, Moore A W. Reinforcement learning: A survey[J]. Journal of artificial intelligence research, 1996, 4: 237-285.

[21] Van Otterlo M, Wiering M. Reinforcement learning and markov decision processes[M]// Reinforcement learning. Springer, 2012: 3-42.

[22] Thurner L, Scheidler A, Schafer F, et al. pandapower - an open source python tool for convenient modeling, analysis and optimization of electric power systems[J/OL]. IEEE Transactions on Power Systems, 2018. https://arxiv.org/abs/1709.06743. DOI: 10.1109/TPWRS.2018.2829021.

[23] Hussein A, Gaber M M, Elyan E, et al. Imitation learning: A survey of learning methods[J]. ACM Computing Surveys (CSUR), 2017, 50(2): 1-35.

[24] Csáji B C, et al. Approximation with artificial neural networks[J]. Faculty of Sciences, Etvs Lornd University, Hungary, 2001, 24(48): 7.

[25] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv:1409.1556, 2014.

[26] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[27] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1026-1034.

[28] Subcommittee P M. Ieee reliability test system[J/OL]. IEEE Transactions on Power Apparatus and Systems, 1979, PAS-98(6): 2047-2054. DOI: 10.1109/TPAS.1979.319398.

[29] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[30] Goyal P, Dollár P, Girshick R, et al. Accurate, large minibatch sgd: Training imagenet in 1 hour [J]. arXiv preprint arXiv:1706.02677, 2017.

[31] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. MIT press, 2018.

[32] Tokic M, Palm G. Value-difference based exploration: adaptive control between epsilon-greedy and softmax[C]//Annual conference on artificial intelligence. Springer, 2011: 335-346.

[33] Li M, Zhang T, Chen Y, et al. Efficient mini-batch training for stochastic optimization[C]// Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 661-670.

致　　谢

衷心感谢导师张宁副教授对本人的精心指导。在研究过程中，张老师非常关心我的研究情况。在我迷茫、没有思路的时候，张老师多次为我解答问题，点明思路，解决了研究过程中遇到的难题，使得研究能够顺利进行。我不仅在研究时从张老师身上获得很大的帮助，而且也感受到张老师对科研的认真与热情。张老师的教导将使我受益终生。

同时，我也衷心感谢侯庆春博士对本人的耐心指导与鼓励。在这段研究中，无论是前期对基本知识的学习，还是中期对模型的设计、软件的使用、算法的理解，抑或是最终的论文写作及终期答辩，侯学长一直用心地指导细节内容。在最终论文前难熬的时刻，侯学长也从精神上安慰我，让我能够顺利地完成毕业设计。对侯学长的帮助，我不胜感激。

我还要感谢我的父母。在本研究进行期间，他们十分关心我的情况，给予我精神上的支持。我从小到大取得的成绩及最终本科毕业论文的完成离不开他们的关心与默默付出。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：公仲泽　日　期：2021年6月21日

附录 A　外文资料的书面翻译

用于机组组合问题的强化学习算法

摘要：在这篇文章中，我们完成了将机组组合问题建模为马尔可夫决策过程且借此为发电调度寻找对应的低成本策略的工作，从而完成求解日前的机组组合问题。本文展示了已有的两种强化学习算法，并且设计了第三种方法。本文将该结果与之前其他文献中使用模拟退火算法的结果进行对比，结果显示能够有效地将运行成本缩减 27%，且运行时间为 2.5 分钟（相比之下，现有最先进的算法为 2.5 小时）。

目录

A.1 简介

机组组合问题（Unit Commitment）是指在电力系统中为了满足预测的负荷以及储备要求，且同时遵守发电机及传输线约束[1]的要求下的决定最具有成本效益的发电机组组合及其发电量水平的过程。这是一个非线性的混合整数的组合优化问题[2]。机组组合问题的低成本解决方案能够直接降低电力公司的生产成本。随着问题规模的增加，机组组合问题将会变成一个十分复杂，难以解决的问题[3]。

在过去的几年里，多种优化方法被采用来求解机组组合问题，例如优先排序法[4-5]，动态规划法[6]，拉格朗日松弛法[7]，分支约束法[8]，以及整数和混合整数规划法[9-10]。其他较新的方法主要来自人工智能领域，如专家系统法[11]，神经网络法[12]，模糊逻辑法[13]，遗传算法[14]和模拟退火法[15]。

这些方法中要么是纯粹的启发式方法（如优先排序法），要么是半启发式方法（如模拟退火法）。因此，这些方法往往对架构的选择，参数的人为调整以及不同的成本函数非常敏感。另一方面，分析方法也会引入严重的缺点。例如，分支约束法的执行时间会随着机组组合问题的规模增长而呈现指数级增长[8,16]。此外，使用近似值使其对大规模系统具有可操作性将会令解决方案变得高度次优。

因此，在本文中我们采用分析方法来解决该问题，同时保证其不会在大规模系统中变得难以求解，也不会变得高度次优。我们使用马尔可夫决策过程（MDP）框架。MDP 被用于描述许多科学领域的许多现象[17]。这样的模型旨在描述一个决策过程，该过程的结果一部分是随机的，同时另一部分由决策者控制。

在本文中，我们假设决策者知道不同电机的发电成本函数。我们注意到，在欧洲的竞争性的电力市场中，成本信息是不能得到的，因此欧洲的系统运营商通常不能获知上述函数。但是，在许多其他情况下，这个信息是确实可以得到的，比如在一些北美的 TSO，以及拥有多台发电机组的发电公司（这样的公司不会知道电力系统的特性，然而这并不是问题，因为他们在我们的表述中并不发挥作用）。此外，机组组合问题可以很容易地再竞争的市场环境中扩展到发电生产计划[18]。另一篇论文展示了一个框架，其中一个传统的基于成本的机组组合问题求解工具可以用于协助日前电力集合市场的投标策略决策[19]。一般来说，欧洲的 TSO 可以根据历史数据（包括他们从发电机收到的过去和现在的出价）和市场模拟的方法来估算发电成本。另外，在未来的工作中，我们的 MDP 模型可以自然地表达这些近似值中的不确定性。

本文的其余部分组织如下。第二节解释了机组组合问题。然后，我们在第三

节展示了机组组合问题的 MDP 模型，并且在第四节中介绍了强化学习。第五节介绍了我们使用的算法。然后，在第六节中我们展示了我们方法的数值测试。最后，在第七节中，我们对本文进行总结。

A.2 机组组合问题建模

机组组合问题能够被建模为以下的带约束的优化过程。

A.2.1 目标函数

该问题的目标是找到一个可行的发电机调度计划，使得在能够满足用户需求的前提下达到最低的成本。

$$\min_{\alpha_i(t),P_i(t),\forall i,t} \sum_{t=1}^{T}\sum_{i=1}^{N}[\alpha_i(t)C_i(P_i(t)) + \alpha_i(t)[1-\alpha_i(t-1)]SC_i(t_{off_i})]. \tag{A.1}$$

其中，

- 当机组 i 在 t_i 时刻启动时，有 $\alpha_i(t) = 1$，否则有 $\alpha_i(t) = 0$
- $P_(t)$ 是机组 i 在时刻 t 的注入功率 [MW]
- $C_i(P)$ 是机组 i 注入功率 P 的成本 [$]
- $SC_i(t_{off_i})$ 是机组 i 在已经停机 t_{off_i} 时间后的启动成本

A.2.2 约束条件

任何可行的解都应该满足以下的约束。

- 负载平衡

$$\forall t : \sum_{i=1}^{N}(\alpha_i(t)P_i(t)) = D(t). \tag{A.2}$$

- 发电机出力限制

$$\forall i,t : \alpha_i(t)P_{min_i} \leqslant P_i(t) \leqslant \alpha_i(t)P_{max_i}. \tag{A.3}$$

- 备用率约束

$$\forall t : (\sum_{i=1}^{N}\alpha_i(t)P_{min_i} \leqslant D(t), (\sum_{i=1}^{N}\alpha_i(t)P_{max_i} \geqslant D(t) + R(t))). \tag{A.4}$$

- 最短启停时间约束

$$\forall i : t_{off_i} \geqslant t_{down_i}, t_{on_i} \geqslant t_{up_i}. \tag{A.5}$$

其中，

- D(t) 是在时刻 t 的需求。
- R(t) 是在时刻 t 需要的预备功率。
- P_{min_i}, P_{max_i} 是机组 i 的最小出力以及最大出力。
- t_{off_i}, t_{on_i} 是在改变机组 i 状态前距离上一次改变状态的最短等待时间。

A.2.3 成本函数

- 出力成本——该机组的二次发电成本函数：

$$C_i(P_i) = \alpha_i P_i^2 + b_i P_i + c_i. \tag{A.6}$$

- 启动成本——一个自变量为机组已经停止时间的指数形式函数：

$$SC_i(t_{off_i}) = e_i exp(-g_i t_{off_i}) + f_i exp(-h_i t_{off_i}). \tag{A.7}$$

对于发电成本（A）和启动成本（B）的一个特定发电机的例子如图A.1所示。

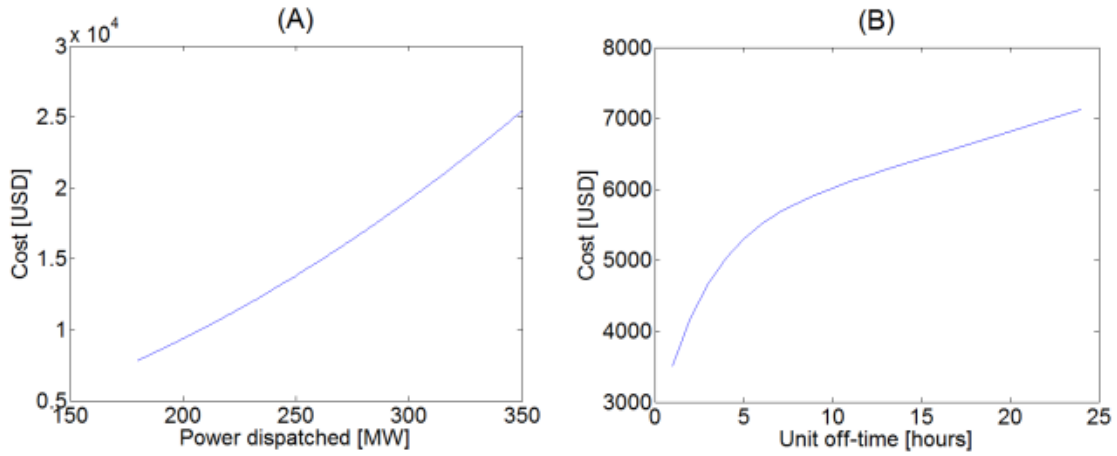


图 A.1 (A) 展示了一个特定的发电机的发电成本，(B) 展示了该发电机的启动成本，是该发电机已经停止的时间的函数

A.3 马尔可夫决策过程

对于这个非凸的混合整数二次问题，寻找全局最优解是难以实现的。因此，与[15]使用在解空间直接搜索的方法不同，我们建议使用另一种方法：将目标分解

成一个连续的决策过程。我们使用马尔可夫决策过程 4 元组 (S, A, P, R)[17]来对系统的动态过程建模。简要地说，在这个模型中的每个时间步骤中，这个过程处于某个状态 s，采取了动作 a，根据转移核 $P(s, a, s')$ 转移到状态 s'，并且获得奖励 $R(s, a, s')$。因此，下一状态 s' 仅仅取决于当前的状态 s 以及决策者的动作 a。

A.3.1 状态空间

系统的状态可以由 N 个发电机中每一个的开/关时间（负数表示关停时间）和一天中的时刻来完全描述。

$$S = \{-24, -23, ..., -1, 1, 2, ..., 24\}^N \times \{1, ..., 24\}.$$

A.3.2 动作空间

每一个机组能够打开/保持开的状态，或者关闭/保持关闭。

$$A = \{0, 1\}^N.$$

A.3.3 奖励

在每个时间步骤中，奖励是 N 台发电机的运行成本（负数）。

$$R(s, a, s') = -\sum_{i=1}^{N} [I_{[s_i'>0]} C_i(P_i) + I_{[s_i'>0]} I_{[s_i<0]} SC_i(s_i)].$$

注入功率 P_i 通过求解有约束的二次规划问题（发电成本是二次的）。通过最大化 MDP 的非折现的累计奖励，我们实现最小化目标函数。

A.3.4 转移核

转移过程是确定的：$f(s, a) = s'$。转移函数限制了该过程必须满足优化问题的约束。

图A.2给出了转移核的一个例子。

A.4 强化学习

强化学习中，策略是指状态空间 S 与动作空间 A 之间的映射。给定一个策略，我们就知道在系统的每个状态下应该执行什么动作。对于定义好的 MDP 来说，我们的目标如下：找到一个最优策略 $\pi^*: S \to A, s.t:$

$$\pi^* = argmax_{\pi \in \Pi} \sum_{t=1}^{T} R(s_t, \pi(s_t), f(s_t, \pi(s_t))). \tag{A.8}$$

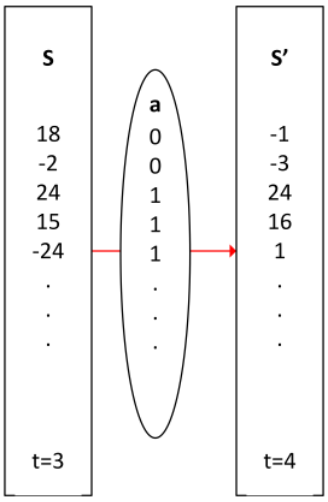


图 A.2　状态核 $f(s,a)=s'$ 的一个例子。当动作取 1 或 0 的时候，发电机打开/保持打开或关闭/保持关闭。时间也表示在状态中

其中 Π 是所有可能的策略空间。

强化学习 (RL) 是机器学习中的一个领域[20]，研究通过与环境互动来进行学习的算法。强化学习是通过状态和动作进行学习的。对于每一个状态 s，给定一个策略 π，其状态价值 $v^{\pi}(s)$ 定义为：

$$v^{\pi}(s)=\sum_{t=1}^{T} R(s_t,\pi(s_t),f(s_t,\pi(s_t)))for s_1=s. \tag{A.9}$$

A.5　强化学习算法解决方案

在本节中，我们介绍了三种不同的强化学习算法来求解A.8

A.5.1　算法 1-近似策略迭代分类法 (Approximate Policy Ieration)

上面定义的状态价值 $V^{\pi}(s)$ 的拓展是状态-动作价值函数 $Q^{\pi}(s,a)$，该函数表明了执行动作 a 的价值（不管策略 π 如何），在此之后再遵循策略（π）。在我们介绍的第一个算法中，我们使用状态价值函数。这个函数被定义为所有 (s,a) 对的集合，其大小为 $|S|\cdot|A|$。

(1) 近似：我们的状态空间随着 N 增长而指数增长：$|S|=24\cdot 48^{N}$，动作空间也是如此：$|A|=2^{N}$。对于 $N\geqslant 4$，已经几乎不可能找到每一个状态动作对的确切价值。因此，我们使用近似方法来评估状态动作价值函数 $Q^{\pi}(s,a)$。我们使用基于特征的回归，这大大地将 $Q^{\pi}(\cdot,\cdot)$ 的维度从 $|S|\cdot|A|$ 降低到 $dim(\phi(s,a))$,

也即特征向量 $\phi(s,a)$ 的维度数。我们对每个发电机 i 使用 4 个二元特征来表示它可能处于的每一个“我们感兴趣的”区域。

$$s_i < -t_{off_i},$$

$$-t_{off_i} \leqslant s_i < 0,$$

$$0 < s_i \leqslant t_{on_i},$$

$$t_{on_i} < s_i$$

然后，这些特征被复制 N 次，并在动作向量为 0 的位置归零。结果又被复制成两份，用来区分会导致问题（过渡到不可行状态）的 (s,a) 对。我们最终得到特征向量 $\phi(s,a)$，且其维度仅仅是 N 的平方级别：$dim(\phi(s,a)) = 2 \cdot 4 \cdot N^2$

(2) 策略迭代算法：基本的算法是策略迭代法[20]。这个著名的算法在两步之间进行迭代：根据一个固定的策略评估状态价格，以及使用学到的值改进策略。我们使用 SARSA 算法[20]（带有 ε-贪心探索）进行评估，并且简单地用以下的最大化方法（对于第 k 步）来提升算法表现：

$$\pi_k(s) = \underset{a \in A}{argmax}\, Q_k(s,a) = \underset{a \in A}{argmax}\, \phi(s,a)^T w_k \tag{A.10}$$

(3) 策略表示：在机组组合问题中使用策略迭代方法的最大的挑战是策略表示的选择。一方面，策略应该对所有状态 $s \in S$ 定义。另一方面，该策略能够用这个巨大的状态空间中的一小部分来训练。此外，它的输出是在从一个巨大的动作空间中进行选择。

为了解决上述困难，我们将策略选定为一个分类器，来将状态分类为动作。这就产生了一个大规模的多分类问题（2^N 个可选类别），这本身就被认为是一个困难的问题。我们通过使用一个基于树状结构的分层分类器来解决这个问题：每个节点对一个机组的动作进行分类，并分成两个节点进行下一机组动作的分类。分类在特征空间完成。感知机算法[21]被用来作为基本的二元分类器（可以通过在线更新来节约内存）。每个时间步骤都会存储不同的树。

值得注意的是这并不是一个决策树分类器，而是多个二元的基于超平面的分类器，这些分类器正在以一种顺序的方式被便利。叶子决定最终的行动预测（封装路径）。

使用分类方法的近似策略迭代法

初始化:
α-SARSA 步长
ϵ-探索参数
N_{pi}-迭代次数
π_0-初始策略
ϕ-基函数
for $k = 1$ to $k = N_{pi}$ **do**
  $w_k = SARSA(\pi_k, \alpha, \epsilon)$
  for all $s \in S$ do **do**
    $\alpha^* = \underset{a \in A}{argmax}\, \phi(s,a)^T w_k$
    $\pi_k = updateClassifier(a^*, \pi_k)$
  end for
end for
return $\pi_{N_{pi}}$

A.5.2 算法 2-树形搜索 (Tree Search)

一个 MDP 可以用一棵树来表示，其中每一个节点代表一个状态且每一条边代表一个动作。在机组组合问题中，我们可以理论上明确地表达这棵树，因为状态转移和奖励是确定的，其中的每一条边可以包括每一步转移的奖励。让我们用 s_j^t 来表示在时间步骤 t 的第 j 个状态。在这种表述下，找到一个最优策略 π^* 对应于找到从根节点出发的最大的累计奖励（初始状态 s_0^0）

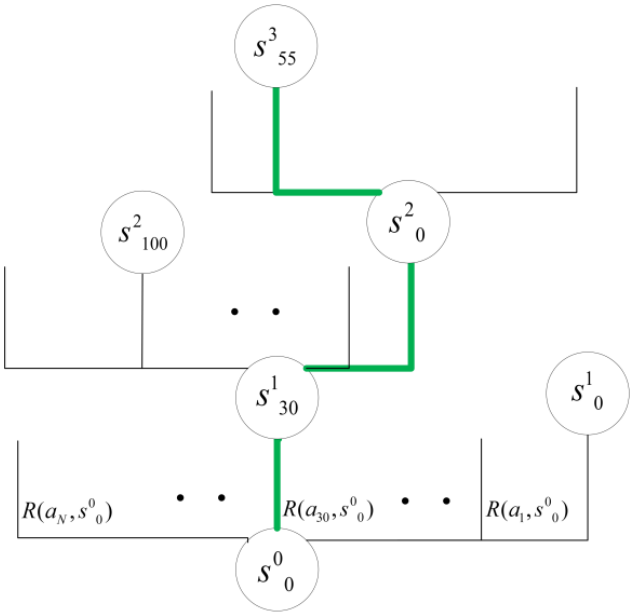


图 A.3　算法 2 的可视化——树形搜索。节点代表状态，边代表状态转移及其对应的奖励

然而，因为在这棵树中可能的道路数是 $|A|^T = 2^{N \cdot T}$，对于机组组合这样规模的问题来说，简单地搜索整棵树来找到一个最优路径是不可实现的。因此，在我们的树形搜索算法中我们将时间范围限制为 H(H<T)。也就是说，树形搜索是通过在有限的前瞻期 H 中迭代所有可能的结果来寻找一个前瞻策略。我们的算法主要包括以下两个部分：

(1) 算法 2.1：算法的第一部分是 $findBestAction(s^t, H)$，通过对该状态 s^t 下所有可能的动作中进行迭代，找到该状态下能执行的最优动作。在这种情况下，最优是指在 H 个时间步长的前瞻范围内的所有坑你路径。也就是说，这一部分在寻找 a^t，即在向量 $\alpha = (a^t, a^{t+1}, ..., a^{t+H})$) 中的第一个分量，其中满足：

$$\alpha = \underset{a' \in A^H}{argmax} \sum_{t'=t}^{t+H} R(s^{t'}, a^{t'}, f(s^{t'}, a^{t'})) \tag{A.11}$$

$(a_m, v_m) = findBestAction(s^t, H)$

if H==0 or t==T-1 **then**
  return (0,0)
end if
$v_m = -\inf, a_m = \bar{0}$
for $alla \in A$ **do**
  $s^{t+1} = f(s^t, a)$
  $(a^{t+1}, v^{t+1}) = findBestAction(s^{t+1}, H-1)$
  $v^t = R(s^t, a, s^{t+1}) + v^{t+1}$
  if $v^t \geqslant v_m$ **then**
    $a_m = a, v_m = v$
  end if
end for
return (a_m, v_m)

(2) 算法 2.2：我们的树形搜索算法的第二步是通过使用 $findBestPolicy$ 的方法在 $t = 0$ 到 $t = T - 1$ 的每个时间步骤中找到最佳前瞻策略。

$\pi = treeSearch(s_t, H)$

for $t = 0$ to $t = T - 1$ **do**
  $(a_m, v_m) = findBestAction(s^t, H)$
  $\pi(s^t) = a_m$
  $s^{t+1} = f(s^t, a_m)$
end for
return π

(3) 通过子抽样改进：我们可以利用这个问题的如下属性：在好的路径（高奖励）中，后续动作不太可能彼此之间有很大不同。这是由于发电机的高启动成本以及最短启停时间限制（快速让不同的机器打开或关断将会导致进入不可行的状态，即没有足够的可用发电机来满足需求）。

利用这一特性，我们为我们的算法加了一项改进。在寻找时间 t+1 的最佳动作时，我们不是在所有的动作中进行迭代，而是只对时间 t 的上一个最佳动作（记为 a^{t*}）进行小的偏移。我们使用一个关于动作 a^{t+1} 的概率密度函数进行抽样，其与 $||a^{t*} - a^{t+1}||_2$ 成反比。这一改进大大地减少了运行时间，并且使得我们可以设置更大的 H。在实验部分，我们测试了我们的改进的可用性，并与原始方法进行了比较。

A.5.3 算法 3-回扫法 (Back Sweep)

我们的”回扫法“是一个创新的算法，其灵感来自于动态规划中从结束时刻开始回溯并且向前推移的概念。这样，我们就能对未来状态的价值有一个可靠的估计，并能根据对未来价值的了解做出正确的决策。主要的创新之处在于对状态空间的”有趣“（潜在的有益）的状态进行采样，并在贝尔曼更新公式（定义见下文）中使用最近邻近似（Nearest Neighbor Approximation）。

1. 算法 3.1：算法的第一部分是估计每个采样状态的最优值 $v^*(s)$。最优值值是指使用A.8定义的最佳策略时的状态价值。

$$v^*(s) = \underset{a \in A}{max}[R(s, a, s') + v^*(s')]. \tag{A.12}$$

$D = evaluateStates(N_s, s_0)$

$InitializeD = \emptyset, \tilde{s} = s^T$
for t=T-1 to 0 **do**
  $\underline{S}^t = sampleEnvironments(\tilde{s}, N_s)$
  for $i = 1 to N_s$ **do**
    $\hat{v}^{*t}(s_i^t) = max_{a \in A}[R(s_i^t, a, f(s_i^t, a)) + \hat{v}^{*t+1}(NN(f(s_i^t, a), D))]$
  end for
  $D = D \cup (\underline{S}^t, \underline{\hat{V}}^{*t})$
  $\tilde{s} = argmax_s \underline{\hat{V}}^{*t}(s)$
end for
return D

- $sampleEnvironment(\tilde{s}, N_s)$ 返回 N_s 个与 $\tilde{s}$ 接近的状态样本。接近性是我们定义的一个量化性指标。

- $NN(s, D)$ 返回 D 中 (s, v) 对的所有状态中的最近邻状态。

2. 算法 3.2：算法的第二部分将通过一次从初始状态 s^0 开始的快速的前扫产产生一个贪婪策略。由于在每一步中我们都会选择最好的动作，因此这个策略是贪婪的。而且已经证明，对于精确的 v^* 值，它也将会是最优策略[20]。

$\pi = findGreedyPolicy(D)$

1: **for** t=0 to T-1 **do**
2: $a^t = argmax_{a \in A}[R(s^t, a, f(s^t, a)) + \hat{v}^{*t+1}(NN(f(s_i^t, a), D))]$
3: $\pi(s^t) = a^t)$
4: $s^{t+1} = f(s^t, a^t)$
5: **end for**
6: **return** π

A.6 实验

为了测试提出的三种算法的表现，我们使用 Matlab[22] 实现了上述方法。问题设置为 N=12 个发电机组，在 24 小时内的规划问题（T=24），参数取自这篇文章[15]。这篇文章[15]中，使用了自适应模拟退火技术，并实现了 644,951 美元的最小目标。

A.6.1 算法 1

算法 1 仅仅在小规模问题上表现良好（N=8，T=12），因此没有被列入表表A.1中。尽管如此，我们还是在本文中介绍算法 1 作为底线。近似策略迭代法是强化学习文献中非常常用的一个算法。除此之外，我们设计的策略结构使得该算法能够从旨在 N=4，T=8 的情况下执行，跃升到 N=8，T=12 的情况。

我们也从理解该方法弱点中学到了更多——由于其前瞻性机制，它不能处理更大规模的问题。由于它从一个随机的策略开始，状态评估在一开始就很差（相比于最优策略），而且在整个迭代过程中，提升变得缓慢和低效。与无限前瞻期的不同的是，该问题被放大，不同的政策被用于不同的时间步长。

A.6.2 算法 2

算法 2 使用了两组不同的前瞻期 H=1 和 H=3 进行测试。在 H=1 的情况下，尽管目标成本稍微有所提升，但是非常低的运行时间让其成为了对该问题最可取的方法。

两种情况下，运行时间的巨大差别源自于 H 的指数性复杂度。

包括了对动作进行子抽样的算法 2 的改进版使运行时间减少，同时在成本上

的提升可以忽略不计。

A.6.3 算法 3

算法 2 的解的最终状态作为初始状态传递给算法 3，且设置采样数为 $N_s = 50$。

A.6.4 结果对比

表 A.1 不同算法的实验结果

算法	目标成本（美元）	运行时间（分钟）
模拟退火[15]	702,379	N/A
适应性模拟退火[15]	644,951	145
树形搜索, H=1	**512,850**	**2.5**
树形搜索, H=3	512,217	240
子采样树形搜索, H=3	512,850	85
回扫法	511,500	6

表A.1总结了实验的结果。我们得到的解互相之间十分相近，都是在 512,000 美元左右的运行成本。

与[15]中提出的先进算法相比，算法 2 在目标值上产生了 27% 的提升，达到了 644,951 美元的最小值，并且运行时间仅仅为 2.5 分钟，而在[15]为 2.5 小时。

A.7 总结

在本文中，我们介绍了强化学习领域中的三个算法，其中有一个是新颖的。我们将机组组合问题建模为 MDP 过程，并且使用上述三个算法对其进行求解（其中两种算法获得成功）。

实验的优秀结果让我们相信将机组组合问题建模为 MDP 过程相比其他现有方法有着很大的优点，这些方法在介绍部分已经有所提及。

另外一个重要的提升在于对随即环境的直接扩展，包括对不确定性的考虑。通过我们现有的 MDP 模型中设置适当的概率转移核以及奖励函数可以轻易地对需求，发电容量以及发电成本的随机性进行建模。本文介绍的算法不需要为获得这种概率版本的解而做出改变。在使用其他优化算法的时候，这种向不确定性模型的转化可能非常具有挑战性[23-24]。

我们打算在这种不确定性的条件下测试我们的算法，并可能改变建模方法来获得机组组合问题的风险规避策略。这可以通过在目标函数中加入风险指标来实现。该标准将会考虑到突发事件，停产和减荷成本，并且将会考虑这些事情发生的概率。

参考文献

[1] Narayana Prasad Padhy. Unit commitment-a bibliographical survey. *IEEE Transactions on power systems*, 19(2):1196–1205, 2004.

[2] Gerald B Sheble and George N Fahd. Unit commitment literature synopsis. *IEEE Transactions on Power Systems*, 9(1):128–135, 1994.

[3] SO Orero and MR Irving. Large scale unit commitment using a hybrid genetic algorithm. *International Journal of Electrical Power & Energy Systems*, 19(1):45–55, 1997.

[4] Fred N Lee. Short-term thermal unit commitment-a new method. *IEEE Transactions on Power Systems*, 3(2):421–428, 1988.

[5] Tomonobu Senjyu, Tsukasa Miyagi, Saber Ahmed Yousuf, Naomitsu Urasaki, and Toshihisa Funabashi. A technique for unit commitment with energy storage system. *International Journal of Electrical Power & Energy Systems*, 29(1):91–98, 2007.

[6] Walter L Snyder, H David Powell, and John C Rayburn. Dynamic programming approach to unit commitment. *IEEE Transactions on Power Systems*, 2(2):339–348, 1987.

[7] Sudhir Virmani, Eugene C Adrian, Karl Imhof, and Shishir Mukherjee. Implementation of a lagrangian relaxation based unit commitment problem. *IEEE Transactions on Power Systems*, 4(4):1373–1380, 1989.

[8] Arthur I Cohen and Miki Yoshimura. A branch-and-bound algorithm for unit commitment. *IEEE Transactions on Power Apparatus and Systems*, (2):444–451, 1983.

[9] Miguel Carrión and José M Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on power systems*, 21(3): 1371–1378, 2006.

[10] James Ostrowski, Miguel F Anjos, and Anthony Vannelli. Tight mixed integer linear programming formulations for the unit commitment problem. *IEEE Transactions on Power Systems*, 27 (1):39–46, 2011.

[11] Z Ouyang and SM Shahidehpour. Short-term unit commitment expert system. *Electric power systems research*, 20(1):1–13, 1990.

[12] S Senthil Kumar and V Palanisamy. A dynamic programming based fast computation hopfield neural network for unit commitment and economic dispatch. *Electric power systems research*, 77(8):917–925, 2007.

[13] Ahmed Yousuf Saber, Tomonobu Senjyu, Atsushi Yona, Naomitsu Urasaki, and Toshihisa Funabashi. Fuzzy unit commitment solution—a novel twofold simulated annealing approach. *Electric Power Systems Research*, 77(12):1699–1712, 2007.

[14] Grzegorz Dudek. Unit commitment by genetic algorithm with specialized search operators. *Electric Power Systems Research*, 72(3):299–308, 2004.

[15] Grzegorz Dudek. Adaptive simulated annealing schedule to the unit commitment problem. *Electric Power Systems Research*, 80(4):465–472, 2010.

[16] Chuan-Ping Cheng, Chih-Wen Liu, and Chun-Chang Liu. Unit commitment by lagrangian relaxation and genetic algorithms. *IEEE transactions on power systems*, 15(2):707–714, 2000.

[17] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[18] José M Arroyo and Antonio J Conejo. Optimal response of a thermal unit to an electricity spot market. *IEEE Transactions on power systems*, 15(3):1098–1104, 2000.

[19] A Borghetti, Antonio Frangioni, F Lacalandra, CA Nucci, and Paolo Pelacchi. Using of a cost-based unit commitment algorithm to assist bidding strategy decisions. In *2003 IEEE Bologna Power Tech Conference Proceedings,*, volume 2, pages 8–pp. IEEE, 2003.

[20] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[21] Yoav Freund and Robert E Schapire. Large margin classification using the perceptron algorithm. *Machine learning*, 37(3):277–296, 1999.

[22] URL www.matlab.com.

[23] Ruiwei Jiang, Jianhui Wang, and Yongpei Guan. Robust unit commitment with wind power and pumped storage hydro. *IEEE Transactions on Power Systems*, 27(2):800–810, 2011.

[24] Farrokh Aminifar, Amin Khodaei, Mahmud Fotuhi-Firuzabad, and Mohammad Shahidehpour. Contingency-constrained pmu placement in power networks. *IEEE Transactions on Power Systems*, 25(1):516–523, 2009.

书面翻译对应的原文索引

[1] Gal Dalal and Shie Mannor. Reinforcement learning for the unit commitment problem. In *2015 IEEE Eindhoven PowerTech*, pages 1–6. IEEE, 2015.

在学期间参加课题的研究成果

个人简历

1999 年 7 月 27 日出生于广东省中山市。

2017 年 9 月考入清华大学电机系电气工程及其自动化专业，攻读学士学位至今。

综合论文训练记录表

学生姓名	公仲泽	学号	2017011036	班级	电 73
论文题目	基于强化学习的机组组合问题求解				

主要内容以及进度安排

主要内容：

1. 机组组合问题建模
   MILP 问题建模，决策变量与约束条件设计
   MDP 过程建模，状态空间与动作空间设计
2. 模仿学习
   在现实数据的基础上加入高斯噪声构造训练数据
   使用 MILP 方法求出最优解后，让 agent 模仿学习，提供一个好的初值
3. 强化学习问题求解
   根据 MDP 过程模型，设计并完成合适的 Actor Critic 网络
   执行强化学习任务并且进行调参优化
4. 传统方法效果对比
   在相同算例下，对优化方法和强化学习方法进行运行成本、求解时间、机组出力的比较

进度安排：

2020.10-2021.1：文献调研，完成主要内容 1，开题报告。
2021.1-2021.4：完成主要内容 2 以及 3 的第一部分，中期报告
2021.4-2021.5：完成主要内容 3 的第二部分和主要内容 4
2021.5-2021.6：总结工作，撰写论文，结题答辩

指导教师签字：

考核组组长签字：

2021 年 1 月 8 日

中期考核意见

85分

考核组组长签字：

2021 年 4 月 22 日

指导教师评语	该生对电力系统优化调度中的机组组合问题展开了研究，使用了MDP对机组组合问题进行建模，使用了强化学习方法与模仿学习方法完成了对问题的求解。论文工作量大，取得较为丰富的研究成果。 指导教师签字： 2021 年 6 月 10 日
评阅教师评语	论文研究了基于强化学习的机组组合问题求解方法，对快速求解满足安全约束的日前优化调度有实际应用价值。 评阅教师签字： 2021 年 6 月 10 日
答辩小组评语	88分 答辩小组组长签字： 2021年 6 月 11 日

总成绩：86.8

教学负责人签字：

2021年 6 月 15 日