第3章

优化算法基本常识

3.1 深度 Q 网络简介

深度 Q 网络（DQN）是一类重要的强化学习方法。因此，在介绍深度 Q 网络之前，有必要简单介绍一下强化学习中的一些基本概念。

强化学习是一类指导智能体求解需要在与环境交互的过程中求得最大回报的决策问题的机器学习方法。使用强化学习方法求解优化问题时，存在四个决定性的基本要素：状态（state）、策略（policy）、行动（action）、奖励（reward）。其中，状态包含了做出决策时环境所包含的信息；策略是指用于指导决策行为的方法；行动是指智能体在当前状态下能够做出的行为；奖励则是指智能体做出相应的行动后得到的收益。一个强化学习四要素的简要介绍如图 3.1 所示。

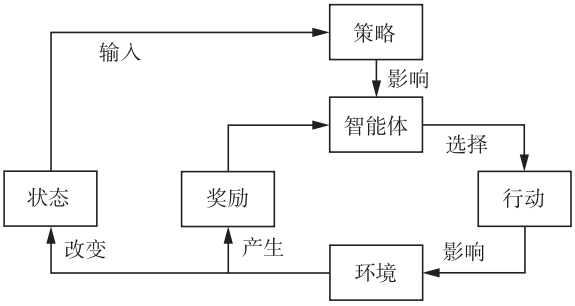

图 3.1 强化学习

以一个简单的旅行商问题为例，在一个旅行商问题中，状态可定义为全部客户节点的位置、已访问的客户节点的集合以及旅行商当前所在的城市三部分信

息的集合。策略可定义为选择下一个待访问客户节点的方法。行动可定义为旅行商从当前城市出发，到达下一个待访问城市的行为。奖励则可定义为城市间的距离。

从图 3.1 不难看出，强化学习求解优化问题的主要目的，即是学习得到一组合理的策略，该策略能够通过输入的环境状态，得到对应的最优行动。在强化学习中，一种经典的策略表示方式是采用 Q 表格（Q-table）。

图 3.2 展示了一个典型的 Q 表格。在该表格中，存在 3 种不同的输入状态，每种输入状态下均存在 4 种不同的动作。强化学习方法对 Q 表格中不同状态下行动的评分进行学习，从而最终得到一组针对当前环境的最优动作选择策略。

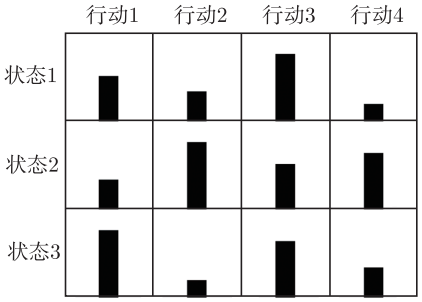

图 3.2 Q 表格

然而，Q 表格方法表示的策略具有一个显著的缺陷：当问题的状态和行动的组合极多或不可穷尽时（这一点在组合优化问题上很常见），采用 Q 表格方式将无法获得最优的“状态-行动”策略。为解决这一问题，Mnih 等[115] 提出了深度 Q 网络的基本概念。

深度 Q 网络采用一个深度神经网络（DNN）来代替强化学习方法中经典的 Q 表格。该网络的输入为当前状态转化而成的向量，输出则为在该状态下对应的不同行动的评分。在得到不同行动对应的评分后，只需要采用简单的贪婪原则选取最高评分所对应的行动，即可完成对优化问题的求解。一个简单的深度 Q 网络如图 3.3 所示。

由图 3.3 可以看出，采用深度 Q 网络方法求解优化问题，实质上是将优化问题转化为一个采用深度神经网络拟合行动与状态的组合所能够获得的奖励的问题。在此基础上，只需要在问题的每一步选择深度神经网络预测能够获得最大奖励值的行动，即可完成问题的求解。

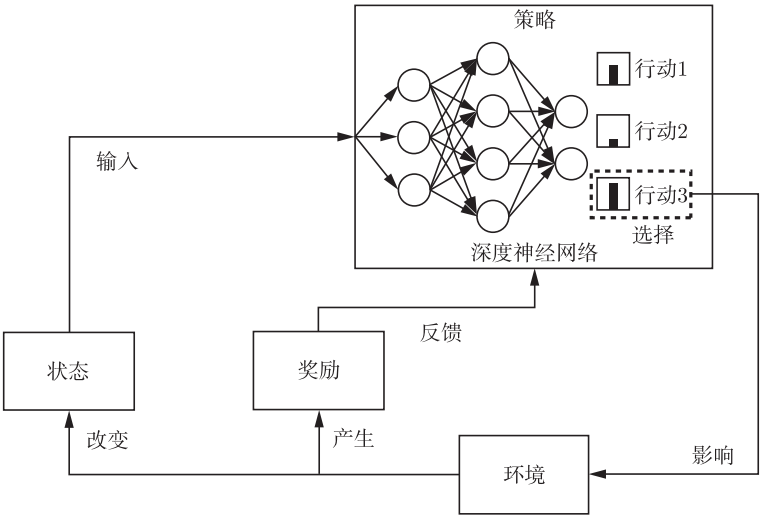

图 3.3　深度 Q 网络

3.2　蚁群算法简介

蚁群算法是一种模拟蚂蚁的觅食行为设计的优化算法。该算法首先在意大利学者 Dorigo 博士的博士论文[79] 中提出。在该论文中，Dorigo 博士提出了蚂蚁觅食行为中的一个有趣现象：作为一种仅具有极低的有限智能的生物，蚂蚁能够通过极为简单的随机探索和信息交互，在不熟悉的环境中快速找到一条通往食物源的最佳通路。一个典型的案例如图 3.4 所示。

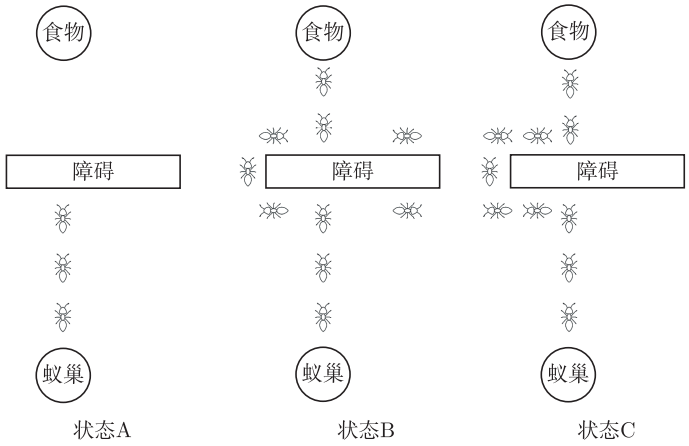

图 3.4　蚁群算法案例

在图 3.4 中，状态 A 为初始状态。蚂蚁从蚁巢出发并向可能存在食物的方向前进。当遇到障碍时（如图 3.4的状态 B 所示），蚂蚁需要选择从何方向绕过障碍。蚂蚁选择绕过障碍的方向的主要依据是路径上的信息素（pheromone）浓度。在搜索的开始阶段，由于路径上未遗留信息素，因此蚂蚁会随机选择一个方向进行探索。在搜索的过程中，蚂蚁会在经过的路径上遗留下一定数量的信息素。随着搜索过程的进行，蚁巢和食物之间最短的路径上会留下更多的信息素，其原因是最短路径上的蚂蚁往返速度更快，因此累积的信息素数量更多。在搜索的完成阶段，全部蚂蚁会逐渐聚集到一条信息素浓度最高的路径上，从而完成了最短路径的搜索（如图 3.4 状态 C 所示）。

蚁群算法通过模拟以上过程在优化问题的问题空间中搜索并逐步改进可行解，具体是指：首先，设存在一个包含 n 个决策节点的问题 Q，每个决策节点至多能够采取 m 个动作。设动作 a_j^i 代表在决策节点 i 采取动作 j，则一个针对问题 Q 的可行解 S 可表述为 $S=\left\{a_j^1,a_j^2,\cdots,a_j^n\right\}$。与可行解 S 相关的决策收益可定义为 C。在此基础上，蚁群算法通过如下过程对问题进行求解。

在蚁群中，通常包含 AN 只人工蚂蚁。在问题求解的开始阶段，将一只人工蚂蚁 ant 的状态进行初始化，并开始逐步构建解。在构建解的每一步，人工蚂蚁均依照概率公式 (3.1) 选择在本决策节点应采取的动作。

$$P\left(sk\right)=\frac{\left(\tau_{ijsk}\right)^{\alpha}\left(\eta_{ijsk}\right)^{\beta}}{\sum\limits_{sk}^{m}\left(\tau_{ijsk}\right)^{\alpha}\left(\eta_{ijsk}\right)^{\beta}} \tag{3.1}$$

其中，s 为当前决策阶段；i 为 s 的上一个决策阶段；j 代表在决策阶段 i 选择了动作 j；$P\left(sk\right)$ 为在决策阶段 s 选择动作 k 的概率；τ_{ijsk} 为积累在决策路径 $\left\langle a_j^i,a_k^s\right\rangle$ 上的信息素的数量；η_{ijsk} 为决策路径 $\left\langle a_j^i,a_k^s\right\rangle$ 上的启发式信息，通常由算法设计者根据经验设计；$\{\alpha,\beta\}$ 为控制参数，分别代表了启发式信息和信息素信息的重要性程度。

经过 n 步决策，人工蚂蚁 ant 完成了解的构建，并得到了可行解 S_{ant} 和与该可行解对应的收益 $\Delta\eta_{ijsk}^t$。

在全部 AN 只人工蚂蚁均完成解的构建后，蚁群算法按照式 (3.2) 更新其信息素矩阵。

$$\eta_{ijsk}^{t+1}=\left(1-\rho\right)\eta_{ijsk}^t+\Delta\eta_{ijsk}^t\rho \tag{3.2}$$

其中，η_{ijsk}^t 为第 t 代时在决策路径 $\left\langle a_j^i,a_k^s\right\rangle$ 上积累的信息素的量；ρ 为信息素蒸发系数，代表每次迭代，积累在决策路径 $\left\langle a_j^i,a_k^s\right\rangle$ 上的信息素均进行等比例衰

减；$\Delta\eta_{ijsk}^{t}$ 为第 t 代时决策路径 $\left\langle a_j^i, a_k^s \right\rangle$ 上的信息素增量，$\Delta\eta_{ijsk}^{t}$ 可通过式 (3.3) 计算。

$$\Delta\eta_{ijsk}^{t} = \sum_{\text{ant}}^{\text{AS}} f\left(C_{\text{ant}}\right) x_{ijsk}^{\text{ant}} \tag{3.3}$$

其中，$f\left(C_{\text{ant}}\right)$ 为根据决策收益 C_{ant} 计算得到的信息素残留的量；x_{ijsk}^{ant} 为一个 $\{0,1\}$ 变量，若 S_{ant} 中包含决策路径 $\left\langle a_j^i, a_k^s \right\rangle$，则 $x_{ijsk}^{\text{ant}} = 1$，否则 $x_{ijsk}^{\text{ant}} = 0$。

在经过 T 步迭代后，蚁群算法收敛，并输出在 T 步迭代后找到的最优解。蚁群算法的简明流程图如图 3.5 所示。

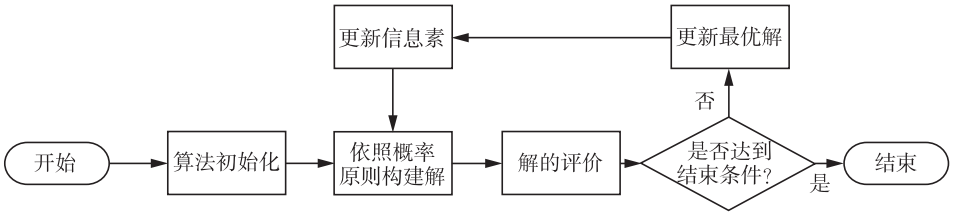

图 3.5　蚁群算法流程图

3.3　模拟退火算法简介

模拟退火算法是模拟自然界中金属物理退火过程的一类启发式优化方法。在自然界的金属退火过程中，首先将固体金属加热到较高温度，使金属获得一定的内能。在该温度下，金属中的粒子趋于无序状态。然后使金属缓慢降温，在降温的过程中，金属中的粒子逐渐失去内能，同时状态趋于有序。最终，金属在常温下达到基态，内能减为最小。在合理控制降温速度的情况下，金属固体内部的粒子会由最初的较无序状态逐渐转化为内能最低的结晶态。一个简明退火过程可由图 3.6 表示。

模拟退火这一概念最早由美国物理学家 Metropolis 等人于 1953 年提出，用于模拟计算多分子系统中的能量分布。1983 年，Kirkpatrick 等[99] 首次将该概念应用于组合优化领域，并采用模拟退火算法（simulated annealing, SA）求解了旅行商问题。下面简要描述模拟退火算法的主要流程。

设存在一个优化问题 Q，针对优化问题 Q 的一个可行解被定义为 S，与可行解 S 相关的决策收益可定义为 C。设在本问题中，决策收益越小越好，则模拟退火算法通过如下流程对问题进行求解：

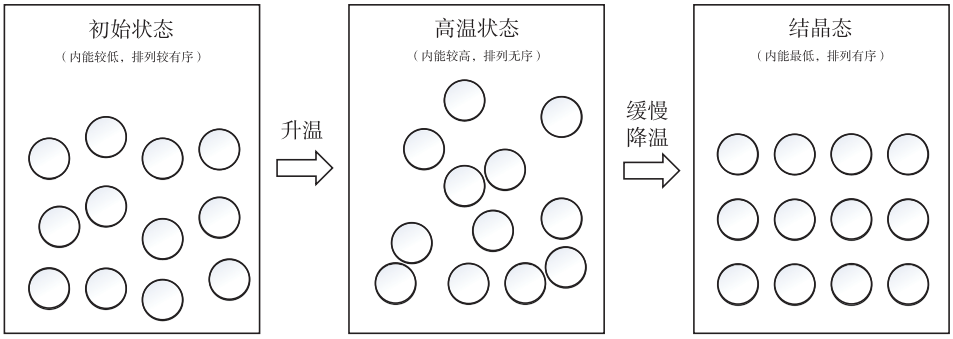

图 3.6 物理退火过程

首先，算法初始化并产生一个初始解 S_0，初始解一般通过启发式算法或者随机算法产生。其次，对该解进行评价，得到该解的决策收益 C_0。再次，采用邻域搜索算法对解 S_0 进行改进，得到新可行解 S_1，并计算针对解 S_1 的决策收益 C_1。最后，通过式 (3.4) 计算解的改进质量。

$$\Delta C = C_0 - C_1 \tag{3.4}$$

若 $\Delta C \leqslant 0$，即解 S_1 优于解 S_0，则接受解 S_1 作为当前解，并继续对解 S_1 进行改进。若 $\Delta C > 0$，即解 S_1 劣于解 S_0，则采用式 (3.5) 计算解 S_1 能够被接受的概率。

$$P\left(S_1\right) = \exp\left(-\Delta C \setminus t\right) \tag{3.5}$$

其中，t 为当前温度。该原则一般也被称为 Metropolis 原则。模拟退火算法每迭代一次，则当前温度 t 根据式 (3.6) 衰减。

$$t' = t \cdot k \tag{3.6}$$

其中，k 为降温系数。当当前温度 t 达到预先设定的最低温度 T 时，算法结束，并输出搜索过程中找到的最优解。图 3.7 给出了模拟退火算法的简明流程说明。

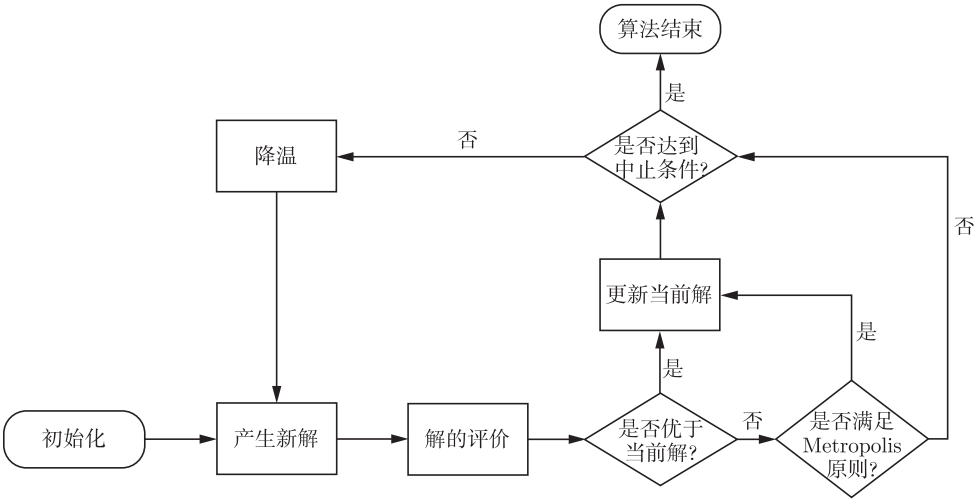

图 3.7　模拟退火算法简明流程

3.4　本章小结

本章针对本书所采用的三类优化算法：深度 Q 学习方法、蚁群算法以及模拟退火算法给出了基本的流程和原理的介绍，并描述了这三类算法中的关键步骤。本章内容有助于读者在后续章节中快速理解求解相关空天资源任务规划问题所设计的算法的基本设计逻辑。