强化学习动态路径规划降低建筑构件运输的研究与应用

朱 帅[1]，林士勉[1,*]
1. 潍坊科技学院，建筑工程学院，山东，潍坊, 262700

摘要：建筑构件运输过程的容量约束与路径优化问题相互交织，形成一个复杂的耦合难题。针对这一难题，本文深入研究基于 *Q*-learning 算法动态路径规划在建筑构件运输的应用。为了引导算法朝着更优的路径决策方向学习，设计基于距离惩罚的奖励函数。通过这种方式，奖励算法在不断的学习过程中，逐渐探索出成本更低、效率更高的运输路线，并以仿真验证该方法的有效性和优越性。仿真结果显示，与传统算法相比，强化学习 *Q*-learning 算法能降低 19.49% 的运输距离，缩短 0.11%运输成本。这使得建筑构件能够以成本更低，速度更快的方式送达施工地点，有效提高了施工效率，实现运输路径的优化，为建筑构件运输领域提供了一种切实可行且高效的解决方案，具有广阔的应用前景和推广价值。
关键词：建筑构件；运输路径；容量约束；强化学习

Research and Application of Machine Learning Dynamic Path Planning to Reduce the Transportation of Building Components

ZHU Shuai[1]，LIN Shi-Yang[1,*]
1. School of Architecture Engineering, Weifang University of Science and Technology, Weifang 262700, China.

Abstract: The capacity constraints and path optimization problems in the transportation process of building components are intertwined, forming a complex coupling problem. In response to this challenge, this article delves into the application of Q-learning algorithm based dynamic path planning in the transportation of building components. In order to guide the algorithm towards learning in a more optimal path decision direction, a reward function based on distance penalty is designed. In this way, the reward algorithm gradually explores transportation routes with lower costs and higher efficiency through continuous learning, and verifies the effectiveness and superiority of this method through simulation. The simulation results show that compared with traditional algorithms, the reinforcement learning *Q*-learning algorithm can reduce transportation distance by 19.49% and shorten transportation costs by 0.11%. This enables building components to be delivered to the construction site at a lower cost and faster speed, effectively improving construction efficiency, optimizing transportation routes, and providing a practical and efficient solution for the transportation of building components. It has broad application prospects and promotion value.
Keywords: Building components; Transportation route; Capacity constraints; Reinforcement learning

在建筑行业中，随着建筑项目规模的扩大和复杂性的增加，如何优化建筑构件的运输路径，尤其是在面临容量约束的情况下，有效地运输建筑构件是确保项目按时完成并控制成本的关键环节[1]。传统的路径规划方法，如遗传算法、A*算法、Dijkstra 算法等，虽然在单纯的路网环境中可以找到最短路径，但较难有效处理复杂的现实条件。因建筑构件种类的繁多，包括钢、混凝土、砖等，需要从不同的生产工厂运输到建筑现场，由于每个部件的体积、重量和形状各不相同，运输车辆在容量限制要求下必须考虑到这些限制。优化容量约束的建筑构件不仅可以降低运输成本，提高物流效率，还可以减少对环境的影响，提高整体运输效率。强化学习作为一种在动态环境中通过试错学习最优

策略的方法，为这种复杂的路径提供了新的思路优化问题。

随着建筑行业的快速发展，建筑构件的运输需求不断增加，当前建筑构件运输面临着诸多挑战，如运输成本高、运输效率低、运输过程中的容量约束难以满足等。建筑构件形状和尺寸各异，对运输车辆的装载和固定提出了特殊要求，例如大型预制梁需要特殊的装载支架和固定方式，以确保运输过程中的安全与稳定。其次，建筑构件重量和体积较大，要适当、合理的分配车辆，以应对不同状况。同时，施工现场环境复杂，可能存在道路狭窄、场地拥挤、施工障碍物等情况，对运输车辆的通行和卸货操作提出了特殊要求，运输车辆需要能够在有限的空间内安全、准确地完成卸货任务。这些挑战不仅影响了建筑项目的进度和成本控制，也制约了建筑行业的可持续发展。运输过程中的容量约束是一个关键问题，如何在满足容量约束的条件下优化运输路径，是当前建筑运输行业面临的主要挑战之一[2]。

强化学习[3]作为一种先进的机器学习方法，在路径规划领域[4]已经显示出了广泛的应用前景。通过对主体和环境的互动学习，强化学习可以找到最优路径策略，具有较强的适应性和机动性，在交通运输路径优化问题中的应用为解决这一领域的难题提供新的思路和方法。强化学习可以根据反馈的奖励信号不断调整策略。并且，根据环境条件，最终找到最优路径。这种方法不仅适用于处理复杂的约束，而且可以适应动态变化的环境。因此，强化学习在建筑构件运输路径优化中的应用前景非常广阔。

当前，国内外建筑构件的运输路径优化的研究成果不计其数，传统的优化方法，如遗传算法、A*算法[5]、Dijkstra 算法[6]在运输路径优化中取得了一定成果，但在处理容量约束问题时仍可在进行优化。例如，蚁群算法[7]和遗传算法[8]虽然能够处理复杂的约束条件，但计算复杂度较高，容易陷入局部最优解[9]。因此，需要寻找一种更有效的优化方法来解决建筑构件运输路径优化问题。国内学者在强化学习和智能优化算法领域进行了深入研究，为解决有能力约束的车辆路径问题（Capacitated Vehicle Routing Problem，CVRP）[10]提供了新的思路，并在实际数据集上取得了较好的效果，以及基于大数据和云计算平台的数据挖掘和分析，致力于提升建筑物流的信息化和智能化管理水平。相比之下，国际研究更侧重于智能化算法的广泛应用，特别是强化学习和深度学习技术在运输路径优化中的角色，注重运输过程的动态变化和实时优化，并将绿色物流和可持续发展纳入优化目标。强化学习在物流与运输领域的应用主要集中在路径规划、车辆调度、库存管理等方面。在路径规划方面，强化学习能够根据环境反馈的奖励信号，自动学习最优的路径策略，具有较强的适应性和鲁棒性。其中，*Q*-learning 算法在车辆路径规划问题中取得了较好的效果，能够有效的降低运输成本，提高整体的运输效率。因此，强化学习在物流与建筑构件运输领域的应用十分广阔。

本文首先对建筑构件运输系统进行详细分析，明确运输过程中的关键因素和约束条件。然后基于 *Q*-learning 算法构建运输路径优化模型[11]，设计合理的状态空间、动作空间和奖励函数。接着，利用 Python 软件进行模拟仿真，验证算法的有效性和可行性。最后，通过模拟仿真分析，评估该算法在仿真仿真中的性能，从而为建筑构件运输路径优化问题提供一个新的思路和方法。利用强化学习的运输路径优化模型[12]，设计合理的状态空间、动作空间和奖励函数[13]。利用 *Q*-learning 算法对其进行验证算法的可行性。最后，根据模拟仿真建立的环境，进行综合的分析。通过将这一算法优化，为容量约束的建筑构件运输路径优问题提供了新的思路和方法[14]，提高运输效率和降低成本。本文第一章提出研究问题，第二章探讨相关理论基础，第三章构建模型，第四章建置仿真分析与验

证，内容包括容量约束处理等关键研究节点，突出整个研究的关键方向。

1 理论基础与相关技术

1.1 强化学习理论基础

1.1.1 强化学习的基本概念

强化学习（Reinforcement Learning，RL）[15]作为机器学习领域中一种独特的方法，基本组成要素包括智能体、环境、状态、动作和奖励。智能体作为决策的执行者，通过对环境状态的感知来决定动作；环境提供智能体活动的背景和条件；状态是对环境在某一时刻的具体描述；动作是智能体在状态下的行为选择；奖励则是衡量智能体动作效果的量化指标，指导智能体优化策略。其核心在于智能体与环境之间的交互学习过程，目的是达成特定的目标。在这个过程中，智能体根据当前所处的状态自主选择并执行相应的动作，而环境会根据智能体的动作给予反馈，即返回下一个状态以及奖励。智能体的核心任务便是通过不断的持续学习，探索出一种能长期累积奖励最大化的强化学习策略。

1.1.2 基于马尔可夫决策过程

马尔可夫决策过程是构建强化学习系统的基本框架，用于准确地描述主体与环境之间的交互过程。马尔可夫决策过程由几个关键的元素组成。状态空间（S）涵盖了所有可能的状态集，全面反映了主体在不同时间可能面临的各种环境条件。动作空间（A）包含了智能体在任意状态下能够采取的所有动作，明确了智能体的行为选择范围。状态转移概率（P）指的是从状态 s 采取动作 a 转移到状态 s'的概率，记为 $P(s'|s,\ a)$，体现了环境对智能体动作的响应规律。奖励函数（R）用于衡量在状态 s 采取动作 a 时，智能体所获得的即时奖励，记为 $R(s, a)$，为智能体的决策提供直接的反馈信号。折扣因子（γ）取值范围为 0 到 1，用于衡量未来奖励对当前决策的影响程度，γ 越接近 1，表明智能体越重视对未来的奖励。越接近 0，则更关注即时奖励。马尔可夫决策过程的目标是找到一个最优策略 π，使得从初始状态开始，智能体在与环境的持续交互中获得最大的长期累积奖励。事实上，绝大多数的强化学习算法都是基于马尔可夫决策过程这一框架进行设计与实现的，马尔可夫决策过程为强化学习算法提供了清晰的数学模型和坚实的理论支撑。

1.1.3 基于强化学习的动态路径规划

基于强化学习的动态路径规划[16] [17]是一种能够适应动态环境[18]变化的路径规划方法。在动态路径规划[19]中，环境的状态和奖励函数可能随时间发生变化。例如，运输过程中交通状况实时变化、施工现场需求临时调整等。强化学习算法通过不断地与环境交互学习，能够实时感知这些变化，并根据新的环境信息调整路径规划策略，找到最优路径。这种方法在自动驾驶、机器人控制等领域得到广泛应用，该方法已广泛应用于自动驾驶、机器人控制等领域。例如，自动驾驶汽车可以通过强化学习根据实时-及时学习时间道路状况和交通信号调整驾驶路径，确保驾驶的高效和安全。

1.2 相关技术与算法

1.2.1 经典路径规划算法

在路径优化领域中，经典路径规划算法发挥着重要作用，其中 A*算法和 Dijkstra 算法应用广泛，且 A*算法和 Dijkstra 算法存在着明显的差异，如表 1 所示。

表 1 经典路径规划算法对比

Table 1 Comparison of classic path planning algorithms

区别	A*算法	Dijkstra 算法
性能	通过参数控制路径，更倾向于向终点方向探索。	实质是广度优先搜索，空间和时间复杂度较高。
品质	使用估算值寻路，得到的路径不一定是最优的。	使用实际的 cost 值，总能找到最优解。
启发式函数	使用启发式函数（如欧几里得距离或曼哈顿距离）来估计从当前点到目标点的距离。	不使用启发式函数，没有方向性。
应用场景	适用于静态路网中寻找最短路径。	适用于解决有向图中最短路径问题。

1.2.2 容量约束车辆路径问题

容量约束车辆路径问题（CVRP）[20]是车辆路径问题（Vehicle Routing Problem，VRP）的一个特例，主要研究满足容量约束[21]条件下的车辆路径优化问题。CVRP 旨在尽量减少对运输的能力限制，它的数学模型也是与建筑构件运输路径优化问题的数学模型类似，其中包括决策变量、目标函数和约束条件等要素。通过对这些要素的合理定义和求解，可以找到满足容量约束的车辆路径方案和最优运输成本，在实际应用中具有重要的应用价值。

1.2.3 *Q*-learning 算法原理

Q-learning[22]是由 Watkins 提出的一种模型无关的强化学习算法，又称为离策略 TD 学习（off-policy TD）。*Q*-learning 算法是基于值函数的强化学习算法，其核心是通过学习状态-动作值函数 $Q(s,a)$ 来指导智能体的行为决策。$Q(s, a)$ 表示在状态 s 下采取动作 a 所能获得的长期累积奖励的期望值。在学习过程中，*Q*-learning 算法依据更新公式表示为：

$$Q(s,a) = Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right] \quad (1)$$

其中 α 为学习率，γ 为折扣因子，r 为即时奖励，s'为下一个状态。通过不断调整 Q 值，智能体在不同状态下做出更优决策，随着学习的深入，Q 值逐渐收敛到最优状态，智能体从而得到最优行为策略。

1.2.4 强化学习与传统算法的对比分析

在路径规划这一关键领域中，强化学习算法[23]与传统算法各有所长。传统算法中的 A*、Dijkstra 算法等，在处理静态路径规划问题时相比其他算遥遥领先。以城市交通导航为例，当城市地图中的道路布局、路况信息等环境条件固定不变时，这些传统算法能够凭借其严谨的数学逻辑与高效的搜索机制，迅速且精准地在众多可能路径中找到最优解，为出行者规划出最短或最省时的路线。然而，一旦场景切换至动态路径规划问题，传统算法的缺点便显露出来。现实世界的交通状况瞬息万变，道路可能突发的交通事故导致拥堵，或者是因临时施工而禁行，面对此类实时变化的环境因素，传统算法由于其固有的设计思路，难以实时调整规划策略，往往无法及时给出最优路径。例如，在上述城市交通场景中，当道路突然出现拥堵状况时，基于静态信息计算出的原有最优路径不再适用，而传统算法却难以迅速做出反应，无法实时为出行者重新规划出避开拥堵路段的新路线。反观强化学习算法，其独特的优势在于能够适应动态环境变化，具备极强的适应性和鲁棒性。仍以交通场景为例，搭载强化学习算法的智能导航系统如同一位经验丰富且反应敏捷的驾驶员，它可以通过与实

时变化的交通环境进行持续交互，不断收集路况信息、车辆行驶状态等数据。当遇到道路拥堵、临时管制等突发状况时，能够及时依据环境反馈的信息调整自身策略，为出行者重新规划出可行的替代路线。这种实时应变能力使得强化学习在动态路径规划方面展现出巨大潜力。但强化学习算法并非完美，当在处理大规模问题时，其计算复杂度较高的问题较为突出。当面对如超大城市的复杂交通网络，或是大型物流配送系统中涉及众多配送点和车辆的路径规划任务时，强化学习算法需要处理海量的状态信息、动作组合以及环境反馈数据。这就意味着需要消耗大量的计算资源，不仅对硬件设备的性能要求极高，而且计算过程耗时较长。因此，在实际应用场景中，无论是规划机器人在复杂工厂车间的移动路径，还是为城市中的出行者提供导航服务，亦或是安排物流车辆的配送路线，都需要根据具体问题的特点，全面综合地考虑算法性能、可获取的计算资源以及实际应用对时间和精度的要求等多方面因素。只有这样，才能从众多算法中挑选出最契合特定场景的路径规划算法，实现高效、精准且经济的路径规划方案，为相关行业和领域的发展提供有力支持。

2 基于强化学习优化容量约束运输的模型构建方法

2.1 建筑构件运输路径优化问题描述

2.1.1 问题的数学模型

建筑构件运输路径优化问题[24]本质上可归类为带容量约束的车辆路径问题（Capacitated Vehicle Routing Problem，CVRP）[25]。在构建其数学模型时，首先定义决策变量，若车辆从节点 i 行驶到节点 j，设$x_{\mathrm{ij}} = 1$。若不行驶，则$x_{\mathrm{ij}} = 0$。通过这种方式，能够清晰地描述车辆在运输网络中的行驶路径选择。目标函数通常设定为使运输成本达到最低可表示为：

$$Z_{min} = \sum\sum c_{\mathrm{ij}}\, x_{ij} \tag{2}$$

其中 c_{ij} 表示从节点 i 到节点 j 的运输成本，通过对所有可能的行驶路径成本进行求和，从而确定整个运输过程的总运输成本。约束条件主要包括：每个节点只能被访问一次，用$\sum x_{\mathrm{ij}} = 1(\forall\, i)$表示，以此确保运输过程的有序性和完整性。车辆的容量约束可表示为：

$$\sum w_{\mathrm{i}}\, x_{\mathrm{ij}} \le C(\forall\, j) \tag{3}$$

其中 w_i 表示节点 i 的需求量，C 表示车辆的容量，防止车辆超载，并且保证在每个节点处，流入和流出的车辆数量保持平衡，维持运输网络的稳定运行。

2.1.2 容量约束的定义与影响

容量约束问题[26]是建筑构件运输路径优化问题[27]中一个至关重要的约束条件。运输车辆的容量包括载重量和载货空间两个方面。由于建筑构件的重量和体积的相同，在运输过程中需要合理安排运输车辆的车型和数量，以达到最佳合理标准。容量约束的数学表达式为：

$$\sum w_{\mathrm{i}}\, x_{\mathrm{i}} \le C, \sum v_{\mathrm{i}}\, x_{\mathrm{i}} \le V \tag{4}$$

其中 w_i 和 v_i 分别表示第 i 个建筑构件的重量和体积，x_i 为第 i 个建筑构件的装载数量，C 和 V 分别表示运输车辆的载重量和载货空间容量。超出容量的限制都会产生许多负面影响。在运输成本方面，它可能会导致运输车辆超载，从而面临罚款和公司额外成本。并且，增加了车辆损耗期和运输成本，运输效率也将降低。同时，超载运输也会增加运输过程中的安全风险，如车辆制动性能下降、控制稳定性差等，这将会引起交通事故造成一系列不可控的风险。因此，为了满足容量的限制，可以适当增加运输车辆的数量或合理调配车辆的型号，这将会节约更多的时间和资源，运输安全也将会受到保障。

2.1.3 目标函数与约束条件

优化容量的建筑构件运输路径问题的目标是在满足各类约束条件的前提下，实现运输成本的最小化。目标函数的数学表达式可表示为：

$$Z_{min} = C_1D + C_2T + C_3F \tag{5}$$

其中 C_1、C_2和 C_3分别表示单位距离成本、单位时间成本和单位燃油消耗成本，D 表示运输路径的总距离，T 表示运输时间，F 表示燃油消耗量，全面考虑了运输过程中的各种成本因素。约束条件除了容量约束（$\sum w_i x_i \le C$，$\sum v_i x_i \le V$）外，还包括时间约束，即运输时间为：

$$T \le T_{max} \tag{6}$$

其中 T_{max} 为最大允许运输时间，以确保建筑构件能按时送达，满足施工进度要求。车辆数量约束，运输车辆的数量为：

$$N \le N_{max} \tag{7}$$

其中 N_{max} 为最大允许车辆数量，有助于合理控制运输资源的投入，提高运输效益。

2.2 强化学习框架设计

2.2.1 状态空间的定义

在强化学习的框架内，状态空间代表着运输车辆在运输过程中可能出现的各种状态[28]。状态空间的定义需综合考虑运输车辆的位置、剩余容量、当前时间和已访问节点等信息。可将状态分别定义为四元组（s_i, c_i, t_i, v_i）其中 s_i 表示车辆当前所处节点，c_i 代表车辆剩余容量，t_i 为当前时间，v_i 是已访问的节点集合。合理的定义状态空间，可确保智能体能够精准感知运输环境的变化。

2.2.2 动作空间的定义

动作空间表示运输车辆在每个状态下可采取的所有可能动作。在建筑构件运输路径优化问题中，动作主要包括选择下一个访问节点、调整运输速度、改变运输路线等。可将动作定义分为三元组（a_1, a_2, a_3），其中 a_1 表示下一个访问的节点，a_2 为运输速度，a_3 是运输路线。合理定义动作空间，能使智能体灵活应对运输过程中的各种状况。

2.2.3 奖励函数的设计

奖励函数用于评估智能体在每个状态下所采取动作的优劣。在本次建筑构件运输路径优化问题中，将奖励函数设计为考虑单位运输成本、运输距离的函数。如下函数所示：

$$R(s,a) = -\alpha \cdot d_{ij}(1 + \beta \cdot c_k) \tag{8}$$

其中，d_{ij} 表示表示节点间的欧氏距离，计算公式为：

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{9}$$

其中（x_i, y_i）和（x_j, y_j）分别为节点 i 和 j 的坐标。c_k 表示车辆 k 的单位距离运输成本（元/km）。α 为基础距离成本权重，默认值为 1，该参数体现了运输距离对成本的影响。β 为车辆成本调节因子，代码中取值为 0.1。该参数平衡了车辆类型对总成本的影响比例。通过合理设计奖励函数，能够引导智能体学习到最优运输路径。

2.2.4 策略与价值函数的更新机制

在强化学习中，策略与价值函数的更新机制至关重要。策略体现智能体在每个状态下采取动作的概率分布，价值函数则表示在每个状态下采取最优策略的预期累积奖励。通过不断与环境交互，智能体能够更新策略与价值函数，逐步学习到最优运输路径。以 Q-learning 算法更新 Q 值：

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max_a Q(s',a') - Q(s,a)] \quad (10)$$

其中，α 为学习率，γ 是折扣因子，r 为即时奖励，s'是下一个状态。

2.3 容量约束的处理方法

2.3.1 容量约束的量化与约束条件

容量约束是建筑构件运输路径优化问题中的关键约束条件。由于运输车辆容量有限，需合理安排运输任务，确保不超出车辆容量限制。容量约束可量化为：

$$\sum_{i=1}^{n} w_i x_i \le Q \quad (11)$$

其中，w_i表示第 i 个建筑构件的重量，x_i表示是否选择第 i 个建筑构件（取值为 0 或 1），Q 代表运输车辆的容量。合理量化容量约束，可保障运输任务的可行性。

2.3.2 强化学习中的约束处理策略对比

在强化学习中，可利用不同的约束处理策略来处理容量约束。以下是比较常用的三种方法比较，如表 2。

表 2 强化学习的约束处理策略对比

Table 2 Comparison of Constraint Handling Strategies in Reinforcement Learning

策略名称	原理	应用	优点	缺点
惩罚函数法	将约束条件转化为惩罚项，加入到奖励函数中。当智能体采取的行动违反约束时，会受到相应的惩罚，从而减少违反约束的行为。	在建筑构件运输路径优化中，可以将超过容量限制的运输行为视为违反约束，并在奖励函数中加入相应的惩罚项。例如，如果运输路径的容量超过限制，可以给予负奖励，从而引导智能体选择符合容量约束的路径。	实现简单，易于与其他强化学习算法结合。	需要仔细设计惩罚项的权重，否则可能导致学习过程不稳定或难以收敛。
约束满足优先策略	在智能体采取行动时，优先考虑满足约束条件，然后再优化目标函数。即在每一步决策中，先筛选出满足约束的可行行动，再从中选择最优的行动。	在建筑构件运输路径优化中，可以在每一步决策时，先筛选出不违反容量约束的路径，然后从这些路径中选择运输成本最低或运输时间最短的路径。	能够确保约束条件始终得到满足。	可能会限制智能体的探索空间，导致学习效率降低。
模型预测控制法	利用模型预测未来的状态和约束条件，提前规划满足约束的行动。通过多步预测，智能体可以提前规避可能导致约束违反的行动。	在建筑构件运输路径优化中，可以利用模型预测未来运输路径的容量需求和供应情况，提前规划满足容量约束的运输路径。	能够提前规避约束违反，具有较好的鲁棒性。	需要准确的模型预测，否则可能导致规划失误。

根据以上对比，在容量约束的建筑构件运输路径优化中采用惩罚函数法[29]较优。因为，它能将车辆容量限制等约束条件转化为惩罚项融入奖励函数，当智能体出现超容量运输等违反约束的行为时，通过负奖励实施惩罚，促使智能体在学习过程中主动规避违规操作，确保运输行为符合现实约束条件，提升方案可行性。同时，这种惩罚机制对违反约束的行为形成“反向引导”，让智能体为追求奖励最大化，会主动调整策略，选择最为符合容量约束的路径，进而得到满足实际运输规则的最优策略。此外，惩罚函数法实现逻辑简洁，只需在奖励函数中添加惩罚项，无需大幅改动强化学习

算法框架，还能轻松与 *Q*-learning、深度强化学习等其他强化学习算法结合，适配不同运输优化场景的算法需求，从而进一步提升方法的普适性与应用灵活性。

2.3.3 容量约束的编码

容量约束在运输过程中不断发生变化，需对这些变化进行处理。通过利用编码技术，可将容量约束的变化编码为状态的一部分，使智能体依据新状态及时调整动作，确保运输任务的可行性，从而将容量约束的变化反映到强化学习框架中。利用以下方法构建：

1. CapacityConstrainedTransportEnv 类继承自 gym.Env，定义运输环境。
2. 状态由当前容量 current_capacity 和每种货物的数量 goods_amount 组成。
3. 动作空间是一个多维离散空间，每个维度表示每种货物要运输的数量，最大不能超过车辆的最大容量。
4. step（单步模拟）方法中，智能体采取一个动作，如果动作导致运输量超过当前容量，则给予惩罚；否则，更新容量和货物数量，并给予奖励。
5. reset（重置状态）方法用于重置环境到初始状态，用于强化学习训练中重新开始一个周期。
6. init（初始化）用于初始化模型的参数和初始状态，例如设置运输网络的节点容量、初始货物分布等。

上述方案展示容量受限运输环境（Capacity Constrained Transport Environment，CCTE）的核心运作机制，其设计目的是通过结构化流程帮助强化学习的智能体在资源受限的场景下优化运输决策。整个过程从初始化参数开始，包括运输的基本容量、货物类型等，为后续操作提供基准数据。随后，系统定义了动作空间和观察空间。动作空间定义了智能体可以执行的运输指令的范围（路径选择、装载顺序），而观察空间定义了智能体可以感知的环境状态（剩余容量、货物体积）。

在初始化的货物信息阶段，系统将根据预设的参数生成运输任务清单，包括货物数量、优先级等属性。在每次决定之前，系统都会验证行动的合法性，以确保指令不超过容量限制。系统在完成有效行动后，通过奖励函数计算反馈值，该过程始终进行，直到触发终止条件（交货结束、超过时间限制或系统崩溃）为止。最后，为运输的状况生成报告（成本、耗时等关键指标）。在结束完整个运行环境后，就可以打开一个新的训练周期。这种闭环设计使智能体通过反复迭代，逐步掌握复杂约束下的最优运输策略。其该设计架构如图 1。

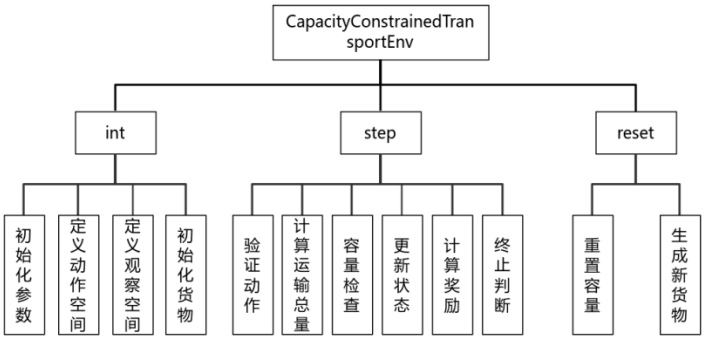

图 1 容量约束与运输环境系统架构

Fig. 1 Capacity constraints and transportation environment system architecture

3 仿真设计与结果分析

3.1 仿真环境搭建

3.1.1 Python 仿真平台架构

在本研究中，我们使用 Python 作为主要的仿真平台[30]。Python 提供了强大的数值计算和可视化功能，适合本次进行复杂路径优化问题的建模和求解。其中，该仿真平台主要由环境模拟模块、智能体模块、*Q*-learning 算法模块和可视化模块构成。

环境模拟模块负责模拟建筑构件运输的实际场景，包含节点信息、车辆容量约束等。其中，reset 方法用于初始化环境状态，step 方法依据智能体的动作更新环境状态并返回奖励和下一个状态。

智能体模块负责根据环境状态选择动作，并更新 *Q* 表。（依据环境状态做出决策），而 get_action 方法依据探索率决定是随机选择动作还是依据 *Q* 表选择最优动作，update_q_table 方法根据 *Q*-learning 算法更新 *Q* 表。

Q-learning 算法模块执行核心算法，通过多个回合的训练，让智能体与环境不断交互，更新 *Q* 表并记录每回合的总奖励。即实现强化学习算法以更新智能体的策略

可视化模块使用 matplotlib 库将训练过程中的总奖励进行可视化展示，方便观察智能体的学习效果。

具体操作步骤如下：在设计的主程序中，满足所需要求设定节点数量、车辆容量和各节点需求，创建环境和智能体，进行 6000 个回合的训练，并将训练奖励进行可视化展示。

3.1.2 模型环境搭建（节点、坐标）

为进一步确保能够在建设项目中正常运行，且同时为了全面提高建筑构件运输的效率，构建一个全面的数据统计平台。平台的总体结构，涵盖了 10 个建筑构件预制厂的详细协调数据，如表 3。作为项目的重要客户，这些预制位置的精确位置对整个运输环境的布局[31]非常重要。通过准确标注各预制厂的坐标，可以直观地呈现出其在该区域内的分布情况，为后续的运输路线规划提供了不可或缺的基础信息。

表 3 各建筑构件预制厂坐标数据（客户）

Table 3 Coordinate of each prefabricated building component factory (customer)

编号	X 坐标	Y 坐标	构件需求量（t）	总构件体积（m^3）
1	22	34	12	50
2	35	67	7	20
3	45	94	14	39
4	50	48	9	19
5	53	87	14	30
6	60	54	5	11
7	38	45	15	30
8	90	83	20	41
9	76	11	11	28
10	68	2	31	70

同时，该平台还包含了一个详细载货车的容量和数量的统计，如表 4。在载货车的容量方面，充分考虑了不同类型建筑部件的尺寸和重量上的显着差异，并仔细梳理了各种类型载货车的实际载

货能力。每个型号的容量都被清晰的表示出来，从适合小型部件的轻型载货车到能够运输大型预制部件的大重型载货车。在此环境运行时要充分考虑载货车数量、施工项目的总进度、各预制厂的生产节奏以及交货频率等因素并且要密切地结合起来。从而准确统计所需的卡车数量，以实现合理调度和有效使用车辆，避免出现闲置或产能不足的情况。

表 4 运输车辆数据

Table 4 Transport truck data

车辆编号	载重能力（t）	容量体积（m^3）	运行成本（元/公里）	车辆数（辆）
V1	20	100	5	3
V2	27	100	6	4
V3	32	100	7	6
V4	33	70	5	3
V5	14	50	3	2

3.2 仿真对比与参数配置

3.2.1 对比算法选择（传统算法、强化学习算法）

将 Dijkstra 算法、A*算法和强化学习算法相互比较，进行研究和分析，对不同算法描述、仿真结果和应用案例进行梳理，如表 5。

表 5 传统算法与强化学习算法比较

Table 5 The Comparison of Traditional Algorithms and Reinforcement Learning Algorithms

算法名称	算法基础	主要特点	优势	劣势	应用场景举例
Dijkstra 算法	基于贪婪策略，不断选择最接近源点的节点，逐步构建从源点到所有其他节点的最短路径。	在非负图中能保证找到全局最优解，精度高。	适用于需要精确最优解且图规模较小的场景，如小型网络路由规划。	计算复杂度为 O（V^2），处理大规模图时效率低、运行时间长。	小型地图的路径规划（如校园地图内的导航）。
A*算法	在 Dijkstra 算法基础上引入启发式函数，通过预测节点到目标节点的距离引导搜索方向。	通常时间复杂度较低，在许多场景中找最优路径速度更快。	适用于需要快速找到最优路径的场景，如游戏角色寻路。	性能高度依赖启发式函数设计，启发式函数选择不当会影响结果或降低搜索效率。	游戏中虚拟角色在复杂地形中的移动路径规划。
强化学习算法	通过代理与环境不断交互，依据环境反馈的奖励信号调整行为策略。	具有很强的适应性和灵活性，能在动态变化环境中表现良好。	适合动态、不确定性高的环境，如自动驾驶、机器人动态避障等。	训练需要大量样本数据和计算资源，学习过程不稳定，易陷入局部最优解。	自动驾驶过程中根据实时路况调整驾驶策略。

在仿真过程中，考虑算法的适用场景和性能评估指标。对于路径规划问题，构建多种不同规模和复杂程度的仿真结构，以模拟不同的实际场景。在每个图结构中，设定单个起点和多个终点组合，确保仿真结果的普遍性和可靠性。

3.2.2 对比结果分析

在仿真的过程中，按照上述所提供的仿真环境与仿真条件，分别对强化学习（*Q*-learning 算法）

以及传统算法（Dijkstra 算法、A*算法）进行编码仿真，并在相同的环境下运行仿真过程。同时，记录每种算法在相同场景下的运行方案、找到运输成本最低或者路径最优等关键指标。

强化学习（*Q*-learning 算法）如图 2 所示，令建筑单位的项目部中心位于坐标原点（0，0），以星形标记呈现，它作为所有运输路线的起点和终点。工厂节点由圆点标记，标注了工厂编号、重量及体积，如示例：F1: 12t, 50m^3。在车辆路径可视化上，5 种车辆类型分别以不同颜色区分，如 V1 为红色、V2 为橙色等，路径线型（如实线、虚线等）用于区分同一车辆的不同运输批次。坐标系单位为公里，通过 X 和 Y 坐标明确显示。每条路径都是从物流中心出发，经过工厂节点序列后再返回物流中心的闭环。

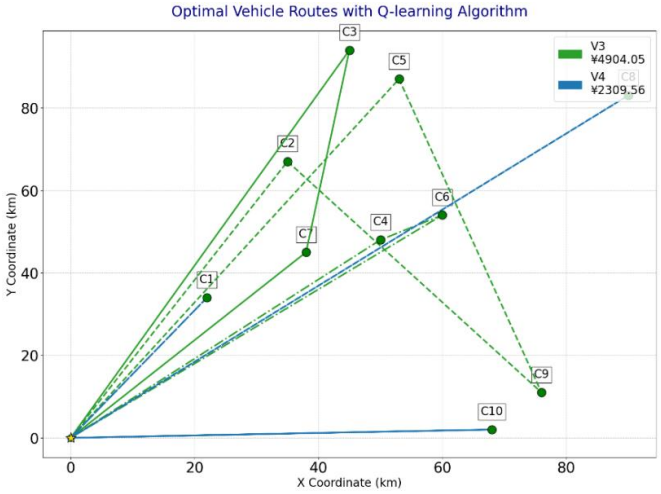

图 2 强化学习算法

Fig. 2 Example of path planning based on the reinforcement learning algorithm

Dijkstra 算法如图 3 所示，在多车辆路径可视化中，以不同颜色线条分别代表不同车辆（V1-V5）的配送路线，通过路径可直观看到其包含往返仓库的闭环。建筑单位的项目部中心以星形标记，位于坐标(0, 0)，客户点用圆形标记，标记为 C1-C10。此算法核心逻辑运用 Dijkstra 算法通过函数生成邻接矩阵，按车辆成本优先原则分配路线，同时进行负载约束校验，涵盖重量和体积的双重限制。

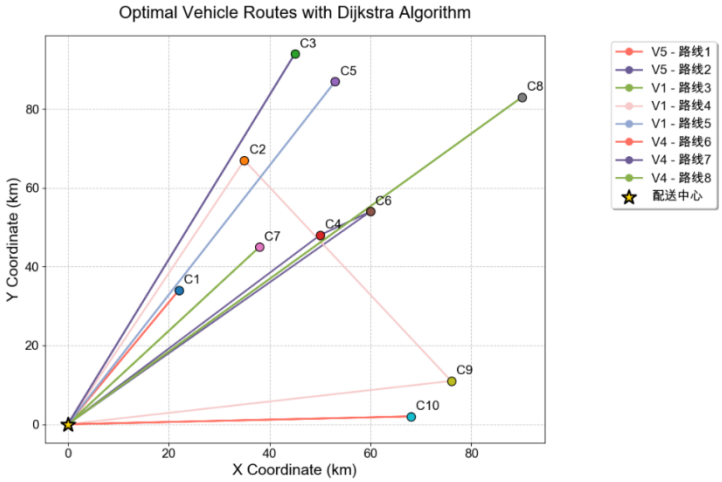

图 3 Dijkstra 算法

Fig. 3 Example of path planning based on the Dijkstra algorithm

A*算法如图 4 所示，其中 10 个客户位置由黑色散点表示，图中紫色线条为 A*算法下采用 V2 车型的最优行驶路径，红色线条为 A*算法下采用 V4 车型行驶路径，表明 A*算法依不同车辆特性进行差异化路径规划，且路径均从物流中心出发并返回，依 A*核心公式为：

$$f(n) = g(n) + h(n) \tag{12}$$

其中，*g(n)*为实际代价，即从起点到当前节点 *n* 的实际累积成本（如行驶距离、时间、能耗）。*h(n)*启发函数，即从当前节点 *n* 到终点的预估最小成本（需满足可纳性）。*f(n)*为评估函数，即节点 *n* 的综合优先级评分（值越小优先级越高），其数值点间线条连接体现算法经优先级队列不断选取当前的最优节点扩展的过程。

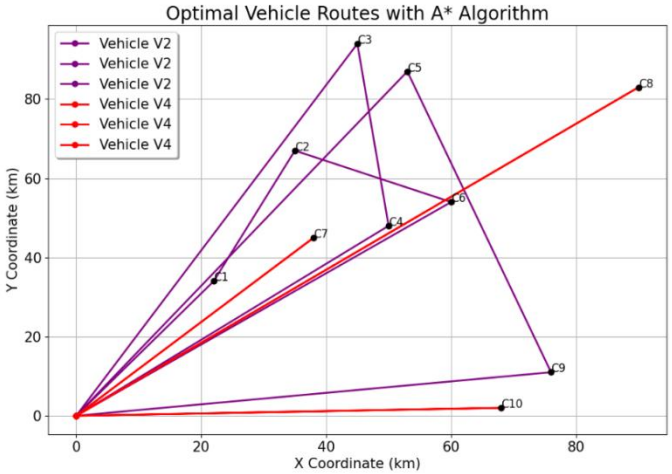

图 4 A* 算法

Fig. 4 Example of path planning based on the A* algorithm

3.3 仿真分析

3.3.1 强化学习算法的分析

强化学习（*Q*-learning 算法）通过提取客户坐标、需求量、体积及车辆载重、容量、成本、数量等数据构建运输环境，将当前车辆、剩余载重和容量、已服务客户、当前位置等定义为状态，在 step 方法中明确动作、奖励及环境状态转移逻辑，利用结合 *Q*-learning 奖励机制 ε-greedy 策略[32]实现探索与利用，更新 *Q* 表选择动作并更新 *Q* 值，逐步学习最优运输策略。由于状态空间包含车辆选择、容量限制、客户访问状态等维度，其规模计算如下：

$$S = 5 \times \left(\sum_{k=1}^{5} v_w\right) \times \left(\sum_{k=1}^{5} v_v\right) \times 2^{10} \tag{13}$$

其中，*S* 为状态空间规模，v_w 各个车辆的载重上限，v_v 各个车辆的体积上限。2^{10} 为 10 个客户的二进制访问状态组合。经计算，最少需约 3000 次训练才能覆盖关键状态-动作组合。为确保方案的准确性，对强化学习算法进行 6000 次训练迭代。首先，设置初始化参数，设置如学习率、折扣因子等初始值。接着创建运输环境，生成工厂、货车等相关数据。然后构建执行决策的主体，进入训练循环后，通过判断是否达到足够次数来控制循环。若未达到，继续收集经验数据，更新策略，持续迭代训练。若已达到次数，则保存训练模型，并做训练奖励，最终结束流程。该流程完整呈现了强化学习从准备、训练到结束的过程，通过不断收集数据更新策略，直至完成指定训练回合，保存模

型并展示训练效果。其具体流程按照图 5 执行。

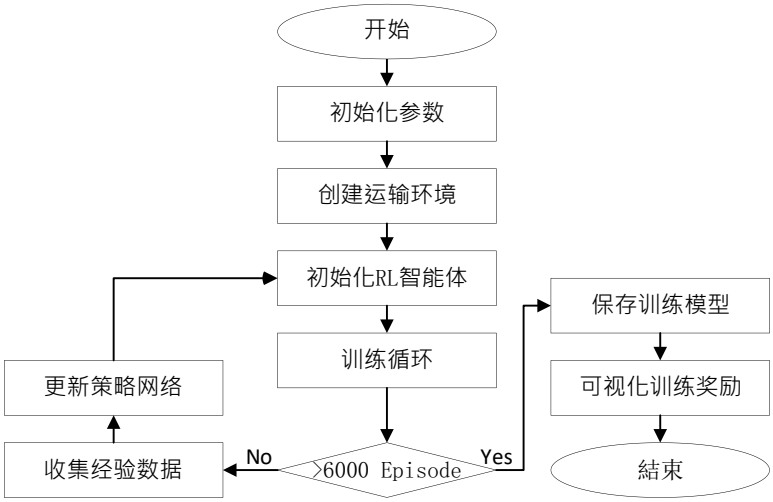

图 5 主程序流程图

Fig. 5 Main program flowchart

图 6 为训练奖励图的多次仿真结果，说明在训练初期（前 1000 轮），总奖励波动剧烈且为负值。这是因为智能体处于探索环境阶段，尚未掌握有效策略，频繁采取非最优行动（如选择超出车辆容量的客户），导致获得负奖励或惩罚。随着训练轮次增加（2000 轮之后），总奖励逐渐趋于平稳，且向 0 靠近。这表明智能体通过不断与环境交互，学习到了更优的路径规划策略，减少了无效动作，累积奖励提升，模型逐渐收敛，体现了强化学习从“盲目试错”到“优化决策”的学习过程。

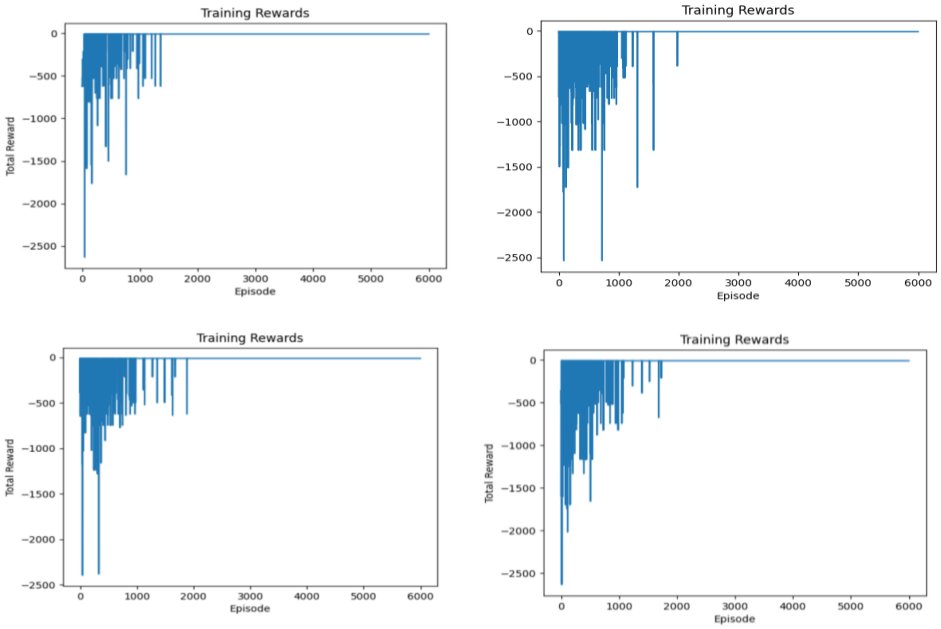

图 6 训练奖励图

Fig. 6 Reward

3.3.2 运输成本分析

通过前述三种算法的分析比较，利用公式 14 计算运输成本 C_1：

$$C_1 = C_2\, S \tag{14}$$

其中 C_1代表单次运输成本，C_2表示车辆单位成本（参考表 ），S 作为路径总距离，直观地反映运输路线的长短。具体而言，在模拟运输场景中，强化学习算法能够更好的根据所对应的环境调出相应的策略，有效降低综合成本。依照仿真计算分析，得到表 6、表 7 与表 8，以清晰直观的形式呈现不同算法在各类指标上的对比数据。

表 6 强化学习算法车辆路径汇总表

Table 6 The Summary Table of Reinforcement Learning Algorithm

车辆编号	服务客户	总距离（km）	总成本（元）	路径详情
V3	3, 7	212.62	1488.34	(0, 0)→(45, 94)→(38, 45)→(0, 0)
V3	5, 9, 2	326.01	2282.08	(0, 0)→(76, 11)→(35, 67)→(53, 87)→(0, 0)
V3	4, 6	161.95	1133.65	(0, 0)→(50, 48)→(60, 54)→(0, 0)
V4	10	136.06	680.29	(0, 0)→(68, 2)→(0, 0)
V4	8	244.86	1224.3	(0, 0)→(90, 83)→(0, 0)
V4	1	80.99	404.95	(0, 0)→(22, 34)→(0, 0)
Totla		1162.49	7213.61	

表 7 Dijkstra 算法车辆路径汇总表

Table 7 The Summary Table of Dijkstra Algorithm

车辆编号	服务客户	总距离（km）	总成本（元）	路径详情
V5	1	80.99	242.98	(0, 0)→(22, 34)→(0, 0)
V5	4, 6	161.95	900.2	(0, 0)→(50, 48)→(60, 54)→(0, 0)
V1	7	117.8	588.98	(0, 0)→(38, 45)→(0, 0)
V1	2, 9	289.99	1523.83	(0, 0)→(35, 67)→(76, 11)→(0, 0)
V1	5	203.74	1018.72	(0, 0)→(53, 87)→(0, 0)
V4	10	136.06	680.29	(0, 0)→(68, 2)→(0, 0)
V4	3	208.43	1042.16	(0, 0)→(45, 94)→(0, 0)
V4	8	244.86	1224.3	(0, 0)→(90, 83)→(0, 0)
Total		1443.82	7221.46	

表 8 A*算法车辆路径汇总表

Table 8 The Summary Table of A* Algorithm

车辆编号	服务客户	总距离（km）	总成本（元）	路径详情
V2	1, 2,6	184.86	1109.19	(0, 0)→(22, 34)→(35, 67)→(60, 54)→(0, 0)
V2	3, 4	219.8	1318.79	(0, 0)→(45, 94)→(50, 48)→(0, 0)
V2	5, 9	258.07	1548.41	(0, 0)→(53, 87)→(76, 11)→(0, 0)
V4	7	117.8	706.78	(0, 0)→(38, 45)→(0, 0)
V4	8	244.86	1714.01	(0, 0)→(90, 83)→(0, 0)
V4	10	136.06	952.41	(0, 0)→(68, 2)→(0, 0)
Total		1161.45	7349.59	

通过对强化学习（*Q*-learning 算法）与 Dijkstra 算法和 A*算法进行数据结果的分析，能够清晰地看出强化学习算法在相同场景下的优势，相较于传统的 Dijkstra 算法和 A*算法，*Q*-learning 算法通过不断与环境交互、试错学习及策略迭代，能更高效地探索出更优的运输路径（缩短距离 19.49%）

以及运输成本（对 Dijkstra 下降 0.11%、对 A*下降 1.9%）。体现了强化学习在处理复杂决策问题时的灵活性与优越性。它不拘泥于传统算法的静态规则，能在复杂场景中自适应地优化策略。

3.3.3 容量约束对路径优化的影响分析

在建筑工程领域，高效的建筑构件运输路径规划对于保障项目进度和控制成本具有重要意义。强化学习中的 *Q*-learning 方法为解决这一复杂问题提供了有力的技术支持。容量约束主要体现在运输车辆的载货能力限制上，包括重量限制和体积限制。这一约束条件深刻地改变了运输路径规划的决策空间。在无容量约束的理想情况下，运输路径的规划可能仅围绕路程最短或时间最短等单一目标展开，算法只需在地理信息构成的网络中搜索最优路径即可。然而，当引入容量约束后，情况变得复杂。车辆每次装载建筑构件时，都必须考虑剩余容量是否足以容纳下一个待装载的构件。这种考虑使得运输决策不再仅基于地理距离或时间，还涉及对车辆容量资源的动态管理。

在利用 *Q*-learning 方法进行路径优化时，容量约束首先反映在状态空间的构建上。状态空间不再仅包含车辆的位置信息，还需将车辆的实时剩余容量纳入其中。例如，当车辆从一个装载点前往另一个装载点或交付点时，其剩余容量随着构件的装载和卸载不断变化，这一变化过程构成了状态转移的重要维度。这种更丰富的状态描述为智能体提供了更全面的环境信息，使其能够根据容量状况做出更合理的路径选择决策。

从动作空间来看，容量约束限制了智能体的可选动作。在选择装载哪些建筑构件时，智能体必须确保所选构件的总体积和总重量不超过车辆的剩余容量。这就要求智能体在众多可能的装载组合中进行筛选，摒弃那些会导致超载的组合。例如，在面对多种不同尺寸和重量的建筑构件时，智能体需要综合考虑它们对车辆容量的占用情况，优先选择那些既能满足运输需求又能充分利用车辆容量的组合。

奖励函数的设计也因容量约束而变得更为复杂。为了引导智能体学习到合理利用容量的策略，奖励函数中需要加入与容量利用率相关的项。当车辆的容量利用率较高时，给予智能体正向奖励，鼓励其在后续决策中继续保持这种高效的容量利用方式；反之，若容量利用率过低，如出现大量空载或严重超载的情况，则给予惩罚。同时，奖励函数还需兼顾运输成本、按时交付等其他目标，通过合理设置权重系数来平衡这些目标之间的关系。

在实际应用中，容量约束对路径优化的影响还体现在算法的收敛速度和稳定性上。由于容量约束增加了问题的复杂性，使得状态空间和动作空间急剧增大，这可能导致 *Q*-learning 算法的收敛速度变慢。为了克服这一问题，需要对算法进行优化，如采用更高效的探索-利用策略，或者引入经验回放机制，提高样本的利用率。此外，容量约束的动态变化，如在运输过程中可能出现的临时增加运输任务或调整构件需求的情况，也对算法的稳定性提出了挑战。算法需要能够实时适应这些变化，重新规划路径，以确保在满足容量约束的前提下，实现运输成本的最小化。

4 结论与未来展望

本文在强化学习在容量约束下的建筑构件运输路径优化领域展开深入研究，选取典型装配式建筑项目作为研究案例，剖析项目背景、运输需求以及在容量约束下的运输场景。通过理论和实际成果综合性，搭建起基于强化学习的建筑构件运输路径优化模型。在模型构造的过程中设计状态、状态空间、动作空间和奖励函数，使模型能够有效地克服容量约束问题。为了使模型的性能更好，对学习率、折扣因子、探索率等超参数进行仔细调整，使模型在训练过程中快速收敛，实现理想的优

化效果。仿真结果表明在相同场景的环境下，相比较与传统算法（Dijkstra 算法），强化学习（*Q*-learning 算法）提高运输效率，缩短运输距离 19.49%，减少运输成本 0.11%；与传统算法（A*算法）相比，运输成本也下降 1.9%，资源的利用更加科学合理。在路径长度方面，*Q*-learning 算法能够依据复杂多变的运输环境，如动态的交通状况、各建筑构件交付点的位置分布，以及车辆的容量限制，智能地规划出更为精简的运输路线，有效减少了不必要的行驶里程。在运输距离上，该算法充分考虑路况信息，通过合理安排运输顺序和路径，减少车辆的运输次数，使得整体运输成本大幅缩短。其中，运输成本不仅作为衡量运输效率的综合指标，而且涵盖了油耗、车辆损耗以及人力成本等多个方面。*Q*-learning 算法凭借对路径和时间的优化，成功降低了运输过程中的各项成本支出，在运输成本控制上表现卓越。相比之下，其他对比算法由于难以全面、实时地处理运输过程中的复杂因素，在路径长度、运输时间和运输成本等方面综合考虑低于 *Q*-learning 算法带来的效益。

此外，*Q*-learning 算法算法具有显着的适应性。在建筑构件运输过程中，可能会出现各种不可预见的情况。例如，当道路因自然灾害、严重的交通事故或紧急基础设施维修等意外事件而暂停通行时，交通计划就会严重中断，这会为建筑构件的运输带来了重大挑战。而 *Q*-learning 算法的优点在于它能够应对这些挑战。它基于强化学习机制。在运输过程中的每个决策点，该算法评估当前状态（如车辆的位置、剩余容量和周围的道路状况）。通过不断地与环境互动，并根据其行为的结果获得奖励或惩罚（选择一个特定的路径），算法不断更新其 q 值，从而达到最佳。

针对上述的不足，未来研究可向多智能体协同运输发展，探索多智能体协同运输模式，通过多个智能体的协作，进一步提升运输效率和资源利用率。比如，研究多个运输车辆间的协同调度，实现运输任务动态分配与优化，提升整体运输系统性能。不断探索更有效的算法和模型，以提高强化学习模型的训练效率和泛化能力。而且，要探索多智能体强化学习在车联网、智能交通管理平台这些分布式系统中的应用，这样就能实现跨区域、跨部门的交通协同管理，让交通管理更高效。

参考文献

[1] Zhou H, Li Y, Ma C, et al. Modular vehicle routing problem: Applications in logistics[J]. Transportation Research Part E, 2025, 197104022-104022.

[2] 徐伟华, 邱龙龙, 张根瑞, 等. 求解带容量约束车辆路径问题的改进遗传算法[J]. 计算机工程与设计, 2024, 45(03): 785-792.

[3] Khallaf N, Rouf E A O, Algarni D A, et al. Enhanced vehicle routing for medical waste management via hybrid deep reinforcement learning and optimization algorithms[J]. Frontiers in Artificial Intelligence, 2025, 81496653-1496653.

[4] Meng W, He Y, Zhou Y. Q-Learning-Driven Butterfly Optimization Algorithm for Green Vehicle Routing Problem Considering Customer Preference[J]. Biomimetics, 2025, 10(1): 57-57.

[5] 张传伟, 芦思颜, 秦沛霖, 等. 融合简化可视图和 A*算法的矿用车辆全局路径规划算法[J]. 工矿自动化, 2024, 50(10): 12-20.

[6] 王景存, 张晓彤, 陈彬, 等. 一种基于 Dijkstra 算法的启发式最优路径搜索算法[J]. 北京科技大学学报, 2007, (03): 346-350.

[7] 辜勇, 刘迪. 自适应混合蚁群算法求解带容量约束车辆路径问题[J]. 东北大学学报(自然科学版), 2023, 44(12): 1686-1695+1704.

[8] 刘祥坤, 李万龙, 李东升, 等. 基于改进遗传算法求解容量约束车辆路径问题[J]. 长春工业大学学报, 2023, 44(03): 254-261.

[9] 付梦印, 李杰, 邓志红. 限制搜索区域的距离最短路径规划算法[J]. 北京理工大学学报, 2004, (10): 881-884.

[10] Rahman H M, Menezes C B, Amin A M. Determination of optimal depot location for a capacitated vehicle routing problem (CVRP) based on gross vehicle weight[J]. International Journal of Systems Science: Operations & Logistics, 2024, 11(1).
[11] Chen J, Jiang Y, Pan H, et al. Path Planning in Complex Environments Using Attention-Based Deep Deterministic Policy Gradient[J]. Electronics, 2024,13(18): 3746-3746.
[12] 何封. 基于强化学习的运输路径优化问题求解[D]. 西安电子科技大学, 2023.
[13] Bäumler A, Benterki A, Meng J, et al. Energy management strategies based on soft actor critic reinforcement learning with a proper reward function design based on battery state of charge constraints[J]. Journal of Energy Storage, 2024, 90 (PA).
[14] 江明, 何韬. 基于深度强化学习的带容量约束车辆路径问题求解[J/OL]. 系统仿真学报, 1-10 [2025-03-16].
[15] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(01): 1-27.
[16] 刘明洋. 大规模车辆路径问题的深度强化学习算法研究[D]. 大连海事大学, 2022.
[17] 牛鹏飞, 王晓峰, 芦磊, 等. 强化学习在车辆路径问题中的研究综述[J]. 计算机工程与应用, 2022, 58(01): 41-55.
[18] 周鲜成, 王莉, 周开军, 等. 动态车辆路径问题的研究进展及发展趋势[J]. 控制与决策, 2019, 34(03): 449-458.
[19] 杨丹. 动态车辆路径问题的算法设计与系统实现[D]. 哈尔滨工业大学, 2016.
[20] 靳康飞, 闫军, 梁云涛. 容量约束的车辆路径问题研究现状综述[J]. 甘肃科技纵横, 2022, 51(10): 52-56+16.
[21] 张景玲, 冯勤炳, 赵燕伟, 等. 基于强化学习的超启发算法求解有容量车辆路径问题[J]. 计算机集成制造系统, 2020, 26 (04): 1118-1129.
[22] 马朋委. Qlearning 强化学习算法的改进及应用研究[D]. 安徽理工大学, 2016.
[23] 高阳, 陈世福, 陆鑫. 强化学习研究综述[J]. 自动化学报, 2004, (01): 86-100.
[24] 赵英男. 基于强化学习的路径规划问题研究[D]. 哈尔滨工业大学, 2017.
[25] Niu Y, Wang S, He J, et al. A novel membrane algorithm for capacitated vehicle routing problem[J]. Soft Computing, 2015, 19(2): 471-482.
[26] 丁伟（Sefa Vidinlioglu）. 用遗传算法求解应急物流中有容量约束的车辆路径问题[D]. 华中科技大学, 2013.
[27] Wang C, Jin C, Han J. A multistage algorithm for multi-objective joint optimization of loading problem and capacitated vehicle routing problem[J]. ICIC Express Letters, Part B: Applications, 2014, 5(5): 1453-1459.
[28] 朱加园. 物流配送车辆路径问题建模及多目标优化算法研究[D]. 沈阳建筑大学, 2014.
[29] 蔡海鸾. 惩罚函数法在约束最优化问题中的研究与应用[D]. 华东师范大学, 2015.
[30] 杨娟, 郭海湘, 杨文霞, 等. 基于 MATLAB 的 GUI 设计车辆路径问题的仿真优化平台[J]. 系统仿真学报, 2012, 24(03): 722-727.
[31] Laporte G. What you should know about the vehicle routing problem[J]. Naval Research Logistics (NRL), 2007, 54(8): 811-819.
[32] 李琛, 李茂军, 杜佳佳. 一种强化学习行动策略 ε-greedy 的改进方法[J]. 计算技术与自动化, 2019, 38(02): 141-145.

基金项目: 新一代 5G/6G 智能通信效能研究, 潍坊科技学院（KJRC2023029）

[1] **第 1 作者简介:** 朱帅（2002），男，学士，潍坊科技学院，研究方向：智能交通建筑运输路径、交通规划。 E-mail: 17606467067@163.com

***通讯作者简介:** 林士飏（1979），男，博士，台湾成功大学（中国台湾），研究方向：智能建造、智能网联车辆通讯技术、多接入边缘计算、无线通信网络、通讯协议。 E-mail: shihyang.lin@wfust.edu.cn