基于强化学习的双层物流选址路径演化优化方法研究

摘　要：为提升双层次物流配送中心的选址和运输路径的整体优化性能，将双层物流选址路径任务看成是两个不同的优化任务，提出一种基于强化学习的双层次物流选址路径演化优化方法。首先，采用强化学习分别估计两层选址-路径问题中的上下节点分配选址概率，然后，设计基于分配概率的多任务交叉策略，采用多因子演化算法协同优化不同层的物流选址路径优化双层物流系统的成本。实验结果表明，提出的算法在求解双层物流选址路径优化问题上具有一定的优越性。

关键词：双层次选址路径优化；强化学习；演化算法；物流配送

中图分类号：　　**文献标识码：**

An Evolutionary Optimization Approach Based on Reinforcement Learning for Two-Echelon Location-routing Problems

Abstract: To improve the overall performance of location selection and transportation path optimization in two-echelon logistics distribution centers, this paper proposes a reinforcement learning-based method for optimizing the evolution of two-echelon logistics location selection and path routing. The two-echelon logistics location selection and path routing tasks are viewed as two separate optimization tasks. First, reinforcement learning is used to estimate the selection probabilities for upper and lower nodes in the two-echelon location-path routing problem. Then, a multi-task crossover strategy based on allocation probabilities is designed to optimize the cost of different layers of logistics location selection and path routing using a multi-factor evolutionary algorithm. Experimental results show that the proposed algorithm has some superiority in solving the two-echelon logistics location routing problem.

Key words: two-echelon location-routing optimization; reinforcement learning; evolutionary algorithm; logistics delivery

1 引　言

选址-路径问题(Location-Routing Problem，LRP)一直是物流管理领域中的常见问题，其主要关注如何在确定的地理区域内选择合适的设施位置，并在设施之间建立最优路径以最小化总成本或最大化运输效率和效益。LRP问题具有NP难性质，主要可以分为基于服务的LRP问题和基于需求的LRP问题。基于服务的LRP主要考虑设施能够服务的区域，以及在服务区域内的需求点，而基于需求的LRP则考虑到每个需求点的需求量和设施的容量限制。现有的研究将LRP分为单目标和多目标问题，单一目标LRP将优化目标限制在单一因素，而多目标LRP则将优化目标扩展到多个方面，实现效益的平衡。

在LRP问题的研究中，经典的方法包括精确式算法和启发式算法。由于LRP问题本身的复杂性，精确式方法在求解大规模问题时效率往往较低，而启发式算法在实践中得到了更广泛的应用。Wang等人采用两阶段启发式算法将带时间窗的多中心路径优化问题简化为单中心路径优化问题，并利用双层规划法建立多目标整数规划模型求解。Ting等人在蚁群算法的基础上进行改进，提出了一种多蚁群优化方法，将LRP问题分为三个子问题进行求解，实现了路径分配的优化。

随着深度学习技术和多因子演化算法[2](Multi-factorial Evolutionary Algorithm，MFEA)的发展，求解LRP问题的效率得到提高。Bello等人[8]使用强化学习演员评论家算法(Actor-Critic)训练指针网络模型，在大规模TSP问题上得到了很好的效果。Nazari 等人也将 Actor-Critic 的强化学习策略训练端到端的简化版指针网络用于求解VRP问题，通过分析公开测试用例在模型上的结果，证明了所提出方法的可行性。Kool等人[9] 提出了一种基于注意力机制的模型，证明了该模型在解决VPR组合优化问题上的有效性。Zhao等人将DRL模型与局部搜索方法相结合，同时设计了一种自适应的critic机制，很大程度上提高了求解效率。

本文首先通过强化学习训练每层的选址模型，得到初始分配概率，进而指导演化算法的交叉操作，实现强化学习和演化算法结合求解双层路径优化问题。

2 问题描述

LRP 问题可以看作是选址定位问题(Location Allocation Problem, LAP)和车辆路径优化问题(Vehicle Routing Problem，VRP)的联合决策问题。在各条件都满足的情况下，实现路径分配总成本的最小化。

双层选址-路径问题可以在一个带权有向图 $G=(V,E)$ 中正式定义，其中 V 和 E 分别表示顶点和边的集合。集合 V 由配送中心节点 V_d 的子集、客户节点 V_c 的子集和中转站节点 V_r 组成。集合 E 分为 N 个部分，表示不同级别的边集。

解决双层选址-路径问题的目的是通过多个设施集中货物，包括位置决策、分配决策和路由决策，从而确定一组车辆路线。其数学模型如下所示：

$$F(x)=\mathrm{Min}\sum_{k=1}^{2}\left(\sum_{u\in U_k}F_u y_u+\sum_{r_j\in R_k}(f_k+q_j^k)z_j\right) \quad (1)$$

公式(1)旨在最小化整个双层选址路径分配的成本，包括开放设施的固定成本、使用车辆的固定成本以及两层路线的运输成本。其中 N 表示路由子问题的层数，上层节点集被定义为 U_k，下层节点集被定义为 L_k。q_j^k 表示每条路线 $r_j, r_j\in R_k$ 的距离成本，而 t_k 是第 k 级路由子问题上的单位运输成本。双层选址-路径问题涉及两个层次的决策，第一层决策选择开放哪些中转站以及需要支付开设中转站的固定成本 F_u；第二层决策为确定车辆路线，最小化整个两层路线中使用车辆的固定成本 F_k。

双层 LRP 问题的主要约束条件如下。

（1）货物平衡约束：上层节点集 U_k 的总货物容量需要满足下层节点集 U_k 的总运输需求。改变当前层级的分配关系，可能或多或少地影响相邻上层层级的的分配关系。在某种意义上，两层节点中分配关系的多样性取决于货物平衡约束。

（2）流量守恒约束：当前层级的上层节点向下层节点的运输取决于相邻下层层级的上下节点之间是否存在分配关系。

（3）顾客访问约束：每个客户节点必须由一辆车辆服务，并且一辆车辆不能为多个客户服务。

（4）车辆容量约束：每个路线的总运输需求不应超过多级路由子问题每辆车的最大容量。

（5）车辆数量约束：车辆的容量必须满足其在整个路线上服务的所有客户的需求，每辆车必须至少分配给一个设施。

（6）路线长度约束：对于双层 LRP 问题，每条路径的长度受到限制。

双层 LRP 问题的示意图如图 1 所示。

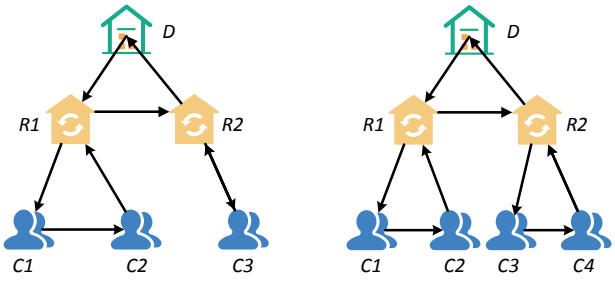

图 1 双层选址-路径问题

Fig. 1 Two-echelon location-routing problem

3 基于强化学习的双层选址路径演化方法

3.1 过程概述

本文基于强化学习方法解决选址问题。该方法采用双层路径，分别对中转站到客户和配送中心到中转站的路径进行模型训练。每一层的模型独立训练，并采用相同的模型结构，区别在于输入和输出。在第一层模型训练中，网络模型学习如何根据客户位置、客

户需求、中转站位置和中转站容量等信息，将每个客户分配给中转站。而在第二层模型训练中，网络模型学习如何根据中转站位置、中转站需求、配送中心位置和配送中心容量等信息，将每个中转站分配给配送中心。该训练方法能够使模型在分配选址时考虑到所有因素，得到最优的选址方案。本文提出的基于强化学习的双层选址路径演化方法整体过程如图 2 所示。

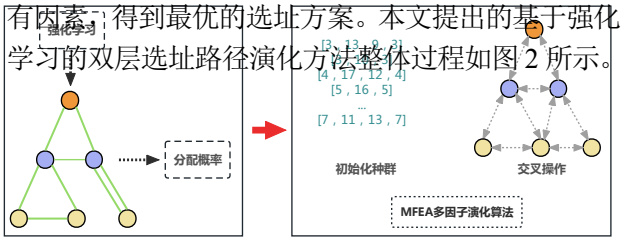

图 2 方法总体过程

Fig. 2 Overall process of our method

同时，我们使用算法流程图展示本文算法的基本流程，如下所示：

算法：基于强化学习的双层选址路径演化方法

输入：元组序列X_t

初始化 actor 和 critic 网络参数θ,ω和指针$z_0(k)$；初始化拥有 N 个个体的种群P_0

while（停止条件不满足）**do**

　重置梯度：$d\theta \leftarrow 0, d\omega \leftarrow 0$

　把样本划分成 N 份

　for $n=0 \rightarrow N-1$ **do**

　$t \leftarrow 0$，$X(k)^t \leftarrow Z_{t+1}(k)$

　　while（停止条件不满足）**do**

　　生成赋值关系序列$Z(k)=\{z_t(k), t=0,...,T_k\}$

　　$t \leftarrow t+1$

　　end while

for 种群P_0中每个个体p_i **do**

　分配一个技能因子τ_i

　end for

　对于种群P_0中所有个体，计算标量适应度

　$t=0$

　while（停止条件不满足）**do**

　　结合强化学习的分配结果，选择性交叉使P_t生成子代种群D_t

for 种群D_t中的每个个体d_i **do**

　垂直文化传播继承一个技能因子τ_i

　end for

　合并父代种群P_t与子代种群D_t构成S_t

　更新种群S_t中所有个体的标量适应度

　在种群S_t中选择 N 个精英个体构造新的父代种群P_{t+1}

end while

3.2 基于强化学习的选址规划

为了估计分层模糊图中不同节点之间的分配关系，我们提出了一种基于强化学习的两层异步指针网络。该网络用于处理两层路由任务，以获得每个节点之间的潜在分配概率。在网络的训练中，我们考虑到了 LRP 的货运平衡和流量守恒约束，并采用异步的训练方法来估计整个分层模糊图的分配关联度。为了捕捉不同节点之间的分配概率，我们利用每个节点的局部信息作为输入，包括上层节点和下层节点的坐标以及上层节点的容量和下层节点的需求。在两层路由任务中，我们只能确定客户节点在第一层路由任务上的需求，而下一节点对其他路由任务的需求是不确定的。因此使用 Actor-Critic 算法对指针网络模型进行训练，以生成相对合理的分配决策。通过这种方法，我们可以有效地估计分层模糊图中不同节点之间的分配关系，并为路由任务的分配提供参考。

在形式上，我们定义了一组在当前状态t下的分层模糊图中的两层分配关联度估计问题，以便更好地进行求解，其中输入为$X_t=\{X(k)^t, k=1,...,N\}$。在该问题中，每个输入$X(k)^t$具体由一个元组序列$X(k)^t=\{x_{ij}(k)^t, i=1,2,...,l(k), j=1,2,...,u(k)\}$表示。$l(k)$为第$k$个路由任务的下节点的个数，$u(k)$则表示第$k$个路由任务的上节点的个数。$x_{ij}(k)^t=(x_i^l(k), y_i^l(k), d_i^l(k,t), x_j^u(k), y_j^u(k), d_j^u(k,t), F_j)$表示下节点$i$和上节点$j$之间的分配特征向量。$x_i^l(k)$和$y_i^l(k)$是分别为下节点$i$的 x 轴和 y 轴位置，而$x_j^u(k)$和$y_j^u(k)$分别为上节点$i$的 x 轴和 y 轴位置。$d_i^l(k,t)$是下层节点$i$的需求，$d_j^u(k,t)$是上层节点$j$的容量。$F_j$为开放上节点$j$的固定成本。对于第$k$梯队路由任务上的每个指针网络，我们从$X_0(k)$中的任意输入开始，并利用指针$z_0(k)$指向该输入。$z_{t+1}(k)$在每个解码状态$t$指向一个可用的输入$X(k)^t$，决定了下一个解码器状态的输入。我们重复这个过程，直到所有可用的下层节点都分配给了上层节点。最后，该过程将会生成一个赋值关系序列$Z(k)=\{z_t(k), t=0,...,T_k\}$，$T_k$为第$k$梯队路由任务的分配关系序列的长度。赋值关系序列$Z(k)$的评估函数可以定义为

$$g(Z(k))=\sum_{j\in\omega}F_j y_j+2\sum_{t=0}^{T_k}(f_k+t_k\varphi(z_t(k))) \quad (2)$$

$$\varphi(z_t(k))=\sqrt{(x_j^u(k)-x_i^l(k))^2+(y_j^u(k)-y_i^l(k))^2} \quad (3)$$

其中$\omega(\omega \subseteq U_k)$表示所生成的赋值关系序列

$Z(k)$所涉及的上层节点集合。如果$z_t(k)$指向一个可用的输入分配$x_{ij}(k)^t$，那么$\varphi(z_t(k))$是上节点j和下节点i之间的分配距离。

在不同的路由任务中，采用 Actor-Critic 算法训练的多级指针网络可以估计不同路由任务中的分配关系概率。我们利用概率链式法则对产生分配关系序列$Z(k)$的概率进行分解，即$P(Z(k)|X_0(k))$，如下所示。

$$P(Z(k)\mid X_0(k))=\prod_{t=0}^{T(k)}\pi\left(z_{t+1}(k)\mid Z_t(k),X(k)^t\right) \quad (4)$$

其中$\pi(z_{t+1}(k)|Z_t(k),X(k)^t)$采用注意力机制计算，$Z_t(k)=\{z_0(k),...,z_t(k)\}$表示到当前状态$t$的解码序列。指针网络中 Actor-Critic 算法训练的目标是找到最优策略π，生成最优分配关系以获得上层节点的容量。actor 网络在任何给定的分配决策中预测下一个行动的概率分布，而 critic 网络从给定的状态t估计单个梯队路由任务的奖励。我们通过观察奖励来验证生成的分配关系序列的可行性：

$$R_n(Z(k),X_0(k))=g(X_0(k))-g(Z(k)) \quad (5)$$

其中$R_n(Z(k),X_0(k))$是奖励信号函数，$g(*)$是公式(2)中分配关系序列的评价函数。

3.3 基于 MFEA 的路径优化

受到多因子遗传模型的启发，本文借助 MFEA 算法的思想，提出了一种新的多因子遗传算法，用于解决双层路径优化问题。第一个任务是优化第一层中转站到客户的路径，第二个任务则是优化第二层配送中心到中转站的路径。在处理这些任务时，该算法对整个种群中每个个体进行适应度评估，同时升序排名，从而得到每个个体在该任务中的因子等级。

我们研究一个包含 k 个分量任务的多任务优化问题，并使用多因子演化算法生成一个大小为 n 的单个种群，同时搜索每个任务的全局最优解。假设个体p_i在第j个任务上进行适应度评估，首先计算整个种群所有在第j个任务上进行评估的个体的适应度，然后对它们进行升序排名r，其中个体p_i的因子等级r_{ij}可以表示为它在r中排名位置。

技能因子τ_i表示个体p_i在所有任务中表现最好的那一个任务。假设个体p_i在第j个任务上的因子等级的排序最靠前，那么个体p_i的技能因子就是j。同时，个体p_i的标量适应度是它的因子排名的倒数，标量适应度=1/因子排名。因子排名根据适应度函数计算得到，所以它的标量适应度间接地跟适应度相关。适应度越高，其因子等级排名越靠前，则标量适应度越大。

基于 MFEA 的路径优化算法具体的求解过程可以分为初始化种群、选择性交叉操作以及变异操作，每个过程的具体细节如下所示。

3.3.1 初始化种群

在使用演化算法迭代寻找最优解时，需要先对优化的任务进行编码。本文研究的是求解双层的配送路径优化问题，为了提高求解的效率和有效性，我们选择更适合的自然数编码方式。为了更好地解释我们的编码方式，下面用一个例子简单说明。

假设有 10 个客户、5 个中转站和 3 个配送中心，一共有 18 个节点，则我们将编码设置为 0~17。对于第一层，有 10 个客户和 5 个中转站，3~7 代表 5 个中转站节点，8~17 代表 10 个客户节点。对于第二层，有 5 个中转站和 3 个配送中心，0~2 代表 3 个配送中心节点，3~7 为 5 个中转站节点。例如在一条染色体个体中，对于第一层可以表示为[[3，13，9，3]，[3，10，3]，[4，17，12，4]，[5，16，5]，[5，14，5]，[6，15，6]，[7，11，13，7]]，从[3，13，9，3]可以看出，它是按某些自然数进行排序组合得到的，表示了一条路线中车辆从 3 号中转站开始先访问 13 号客户，然后访问 9 号客户，最后返回 3 号中转站。[3，10，3]表示了 3 号中转站的第二条行车路线，车辆从 3 号中转站出发，访问了 10 号客户，最后返回 3 号中转站。对于第二层，如[[0，4，0]，[1，3，6，1]，[1，5，1]，[2，7，2]]，其中[0，4，0]表示车辆从 0 号配送中心出发，访问 4 号中转站后返回出发点。

在确定编码方式的前提下，需要对演化算法求解所需要的种群进行初始化，为后续整个算法的迭代做准备。种群中的个体数量需要合理设置，种群太大容易增加算法的求解难度以及迭代所耗费的时间，而种群设置得太小容易使算法陷入局部最优的情况，迭代求解的过程太快而导致收敛。初始化种群产生单个个体的方式具体为：第一层从中转站到客户，将客户随机分配给中转站；第二层从配送中心到中转站，即中转站在满足一定的约束条件随机分配给配送中心。至此，两层的分配已经完成，随后使用节约里程法求得两层路径优化的解作为种群的初始解。重复以上操作即可生成种群中所有个体。

3.3.2 选择性交叉操作

与 MFEA 算法中的交叉操作类似，不同任务之间通过不同个体间的交叉进行隐式的遗传转移。在双层路径优化任务中，两个父代个体的技能因子相

同时才可以进行交叉操作，否则需要满足一定的 *rmp* 才能进行交叉操作，或者只执行变异操作。*Rmp* 概率为算法的交叉操作提供了一种特殊的探索方式。当 *rmp* 接近 1 时，不同技能因子的两个个体可以随意进行交叉操作，这样可以避免算法陷入局部最优解。然而，探索的空间较大可能会使算法难以收敛。当 *rmp* 接近 0 时，不同技能因子几乎不能进行交叉操作，以此着重探索某些重点范围，从而到较好的解。

在交叉操作中，本文借鉴了强化学习训练模型的分配概率。使用强化学习训练的模型在输入一个状态 *state* 时，会生成一个分配的概率关系，本文使用这个概率关系指导交叉操作。第一层和第二层的交叉操作思路一致。为了说明交叉操作的过程，本文以一个小规模客户的例子为例，如图 3 所示。

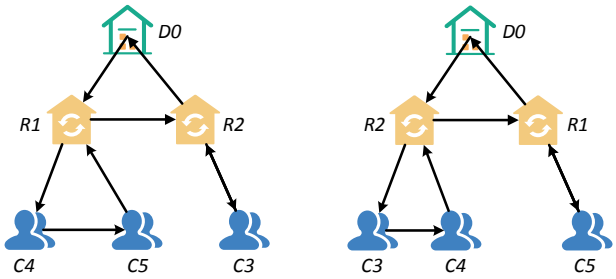

图 3 第一层两条染色体的可视化解

Fig. 3 Visual solution of the first chromosome layer

图 3 所展示的是一个双层路径优化问题，该问题包含 3 个客户、2 个中转站和 2 个配送中心。左右两边分别代表两个染色体(a)和(b)的可视化解。在这个问题中，每个染色体代表两个任务，第一层的路径优化结果代表一个任务，第二层的路径优化结果代表另一个任务。这两个任务可以通过不同个体之间的交叉操作进行隐式的遗传转移。两个解第一层的 1 号中转站进行交叉操作的过程如图 4 所示。

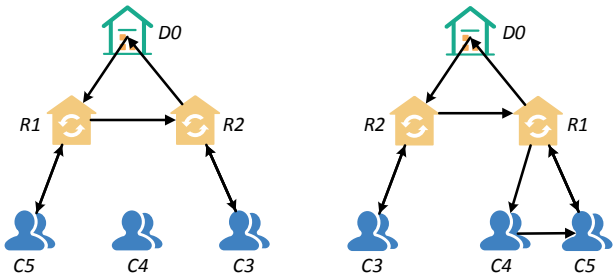

图 4 第一层染色体交叉操作

Fig. 4 First layer chromosome crossing

由图 4 可以看出，左图(a)中的 4 号客户未被分配，这意味着这个解需要进行优化。为了解决这个问题，我们可以将左图的解作为状态信息输入到第一层已经训练好的强化学习模型中，从而得到该客户点在各中转站中被分配的概率。具体而言，我们将左图(a)的解转化为一个状态，并将其作为输入，以得到一个概率分布，该分布表示了 4 号客户被分配到每个中转站的概率。具体可以分解为三个步骤：状态表示、动作选择和奖励反馈。首先，我们将左图(a)的解作为状态信息输入到模型中；接下来，模型会根据这个状态信息选择一个最佳的动作，即预测 4 号客户被分配到每个中转站的概率分布；最后，模型根据该动作的效果给出相应的奖励反馈，更新模型的参数，以期望得到更好的预测结果。

接着，我们可以根据概率值决定将 4 号客户分配到哪个中转站中。在此之前，需要进一步考虑是否有足够的容量满足该客户点的需求。假设 2 号中转站的概率最高，如果该中转站的剩余容量足以满足 4 号客户的需求，则需要比较该客户点与 3 号客户点以及 2 号中转站之间的距离，并将其分配到距离更近的节点。如果 4 号客户到 3 号客户的距离小于 4 号客户到 2 号中转站的距离，则我们需要将 4 号客户插入至 2 号中转站和 3 号客户点的路径中。反之，如果 2 号中转站的剩余容量不足以满足 4 号客户的需求，则需要重新为该客户点分配一条新的路径。根据以上逻辑规则，我们可以利用强化学习模型和距离计算的方法对未被分配的客户点进行重新分配，并得到优化后的路径方案，如图 5 所示。

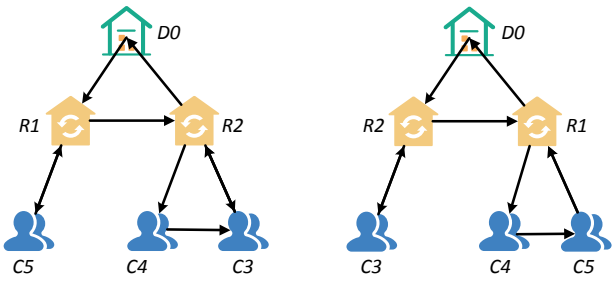

图 5 优化后的交叉结果

Fig. 5 Optimized crossing results

3.3.3 变异操作

本文基于 MFEA 算法进行双层选址-路径问题中的变异操作，以客户、中转站和配送中心作为个体，产生新的个体，对个体的某些基因进行改变以产生新个体。

为确保种群中基因的多样性，我们采用随机选择个体的方法进行变异操作。一般情况下，我们选择适应度较低的个体进行变异，包括客户、中转站和配送中心。对于被选中的个体，我们对其进行基因变异操作，以产生新的个体。

在变异操作中，对于每个个体的某个基因因

子，都有相应的变异概率，若被选中进行变异，则进行相应的基因改变。具体而言，我们会对个体中因因子，我们可以改变其位置和数量；对于中转站和配送中心个体的基因因子，我们可以改变其位置和服务范围。通过以上变异操作，MFEA 算法可以保持种群的多样性，并且以不同的概率进行基因变异，有效地搜索解空间，提高全局搜索能力。

4 实验

4.1 实验设置

本文实验所用的工具是 pycharm，强化学习模型代码编写使用 python 编程语言，第一层和第二层选址的强化学习模型主要通过 pytorch 框架实现。强化学习对训练的数据要求是大量的，因此本文的实验在衡量时间消耗的情况下，考虑使用 300000
组训练数据和 5000 组的测试数据。 循环次数 *epoch* 设置为 20，每批训练 *batch* 的数据是 64。在训练网络时，需要对网络进行参数设置。Actor 网络中 LSTM 隐藏层单元为 128，嵌入层为 128，的某个基因进行随机变换，包括将其取反、随机增加或减少一个值等操作。例如，对于客户个体的基学习率设置为 0.0005。Critic 网络使用与 Actor 网络同样参数设置的嵌入层，随后连接一个输出节点数为 128 的全连接层 ReLU 激活函数，最后连接一个输出节点数为 1 的全连接层。学习率设置为 0.0005，并且使用 Adam 优化器。为了更好地解决模型过拟合问题，在解码器中我们将 dropout 设置为 0.1。

另外，在演化算法实验部分的参数设置中，种群大小设置为 100，最大运行迭代数设置为 500，rmp 设置为 0.3。参数的设置是经过反复调试得到的，在实验中能够表现出较好的效果。

4.2 实验数据

本文的研究对象是双层选址-路径问题，研究的数据集包含两种情况：100 个客户、10 个中转站和 8 个配送中心；以及 50 个客户、10 个中转站和 8 个配送中心。其中，关于客户和中转站的数据来自于 Prins 等人[11] 的标准数据集。不同之处在于仓库被中转站所取代。配送中心的坐标、容量和固定成本等信息见表 1。

表 1 八个配送中心的数据
Tab. 1 Data of the eight distribution centers

	配送中心							
	1	2	3	4	5	6	7	8
x	0	1	33	58	31	6	13	32
y	0	53	60	2	22	10	12	5
K_i	3877	3750	4013	4125	3963	4375	4325	4488
F_i	165000	175000	179000	182000	176000	173000	131000	151000

这些数据来自 Zhuo Dai 等人[7] 公开发表的论文。第一行表示数据的名称，第二行表示序号，第三行和第四行是配送中心的水平坐标 x 和纵向坐标 y，第五行 K_i 表示配送中心的容量，第六行表示配送中心的固定成本，两点之间的距离采用欧氏距离进行计算。本文的第一层和第二层的单位运输成本分别为 100 和 200，车辆的容量分别为 70 和 840，车辆使用的固定成本分别为 1000 和 5000。这些数据将被用于双层选址-路径问题的建模和求解，旨在优化配送方案，同时提高物流效率。

此外，除了配送中心的数据，强化学习训练模型也需要大量的数据集。为了满足我们的研究假设和参考标准，我们根据双层路径的特征，随机生成了具有一定规律的数据集。通过这些数据，我们可以建立模型并进行强化学习训练，以提高模型的准确性和泛化能力。

4.3 结果分析

为了验证我们所提出的方法有效性，我们需要使用一个客观、可度量的指标来评估该方法。因此，我们采用将以双层路径优化成本作为评价指标，通过对比分析得出结论。通过这种评价方法，我们可以更加客观地评估不同方法的优

劣，并从中选择出最优解决方案。实验结果见　　表 2，表 3，表 4。

表 2 100 个客户标准用例算法对比结果
Tab. 2 Comparison results of 100 customer standard use case algorithms

Data	cplex+CW	RL+CW	cplex+EEMTA	**RL+MFEA**
coord100-10-1-2e	457018	514522	454960	**438218**
coord100-10-1b-2e	427832	453807	413692	**410555**
coord100-10-2-2e	414544	482610	404344	428386
coord100-10-2b-2e	383095	424612	382133	**380988**
coord100-10-3-2e	422158	495417	422102	432052
coord100-10-3b-2e	411695	451118	414452	**410334**

表 3 100 个客户生成用例算法对比结果
Tab. 3 Comparison results of 100 customer generated use case algorithms

Data	cplex+CW	RL+CW	cplex+EEMTA	**RL+MFEA**
100-10-8_0	437488	543703	433445	453382
100-10-8_1	418128	470341	411427	**410275**
100-10-8_2	444347	466524	453573	**433570**
100-10-8_3	407387	433286	400268	422362
100-10-8_4	390133	383077	383965	**374761**
100-10-8_5	387895	470145	383749	**379216**

表 4 50 个客户生成用例算法对比结果
Tab. 4 Comparison results of 50 customer generated use case algorithms

Data	cplex+CW	RL+CW	cplex+EEMTA	**RL+MFEA**
50-10-8_0	291645	292236	285290	**280744**
50-10-8_1	260073	290975	253043	256158
50-10-8_2	259416	270513	253258	263967
50-10-8_3	279912	275639	267374	**265544**
50-10-8_4	255893	286456	255623	**241734**
Data	cplex+CW	RL+CW	cplex+EEMTA	**RL+MFEA**

表 2 是 100 个客户、10 个中转站、8 个配送中心在公开测试用例的对比分析。其中，cplex+CW 表示 cplex 工具的求解分配结果，使用节约里程算法求解路径优化阶段；RL+CW 表示强化学习模型的求解分配结果，节约里程算法求解路径优化阶段；cplex+EEMTA 表示 cplex 工具的求解分配结果，路径优化阶段采用显式进化多任务算法(Explicit Evolutionary Multi-tasking Algorithm，EEMTA)，由 L Feng[13] 等人提出的解决多个带容量 VRP 组合优化问题方法。RL+MFEA 是本文提出的方法。表 3 是 100 个客户、10 个中转站、8 个配送中心在生成测试用例的对比分析，表 4 是 50 个客户、10 个中转站、8 个配送中心在生成测试用例的对比分析。

从上述表中可以分析得出，对于 50 个和 100 客户的每个算例，强化学习训练的模型分配结果

有一定的效果，但是整体是要比 cplex 分配效果稍微差一点，这一点可以从 cplex+CW 和 RL+CW 的对比中体现出来。造成这种情况的可能原因有几个：首先，本文使用单智能体强化学习方式用于分配阶段，在输入信息处理方面，多智能体可能更为适合。例如，对于第一层中转站到客户，我们可以把每个中转站看作是一个智能体，然后每个智能体根据环境做出一个动作，也就是代表某个客户被分配到某个中转站，这种处理方式更易于模型的输入数据的处理。其次，由于强化学习需要大量的数据集进行训练，并且本文在综合考虑时间成本的情况下，选用了 30 万的训练数据集，根据以往学者将强化学习用于组合优化的研究，未来可以增加数据集的大小，甚至达到 50 万或者 100 万，这样做可能会增加模型训练的时间消耗。因此，需要考虑对模型进行改进和优化，未来我们的工作会针对这一问题，尝试减少模型训练的时间损耗。

我们使用相同的分配工具 Cplex、节约里程法(CW)和 EEMTA 算法在求解 100 个客户和 50 个客户的路径优化求解。结果显示，我们的实验结果总体上优于 EEMTA 算法和 CW 算法。节约里程法是一种常用的启发式算法，也是一种贪婪的算法，主要求解思路是有顺序地将路径中的两条回路归结成一条回路，每次选择合并归结的目标时使得总路径长度减小的程度最大。当车辆容量超出装载能力时，算法结束，并使用另一辆车继续装载。在求解 VRP 问题时，CW 算法可以快速得到接近最优的满意解，但在求解精度要求较高的情况下，效果不如演化算法等其他算法。

为了更好地展示实验结果，我们将结果可视化为迭代曲线，如图 1 所示。

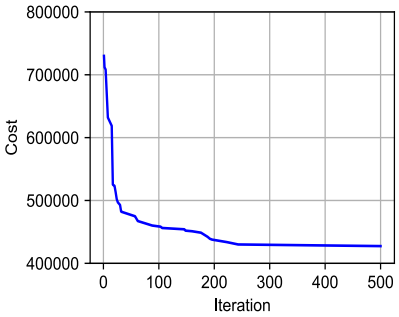

图 6 测试用例的迭代曲线
Fig. 6 Iteration curve of test cases

图 1 展示了演化算法迭代次数与双层路径优化成本之间的关系，其中使用了两个标准测试用例进行测试。图中的纵坐标表示迭代次数，横坐标表示运输成本。可以观察到，在迭代次数为 200 时，算法几乎已经收敛，这表明双层路径优化对演化算法的性能产生了显著影响。因此，使用双层路径优化可以提高演化算法的效率。为了形成对比，我们还选择了迭代次数为 300 时候的情况进行测试，结果同样使用迭代曲线进行可视化，如图 7 所示。

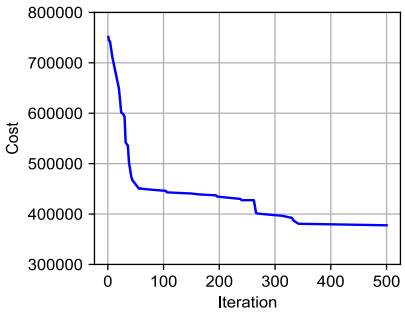

图 7 测试用例为 300 时的迭代曲线
Fig. 7 Iteration curve at 300 test cases

由图 7 可以看出，当测试用例迭代次数为 300 时，曲线趋向收敛状态。

同时，我们在客户规模为 50 的情况下，使用两个生成的测试用例进行演化算法求解。从图中可以明显看出，当迭代次数在 150 到 180 之间时，算法已经接近收敛状态。这表明在处理该问题时，演化算法在迭代 150 次后已经发现了一个较为优秀的解，并且后续迭代中未有显著改进。因此，可以提前终止算法以节省时间和资源。算法的迭代状态如图 8 所示。

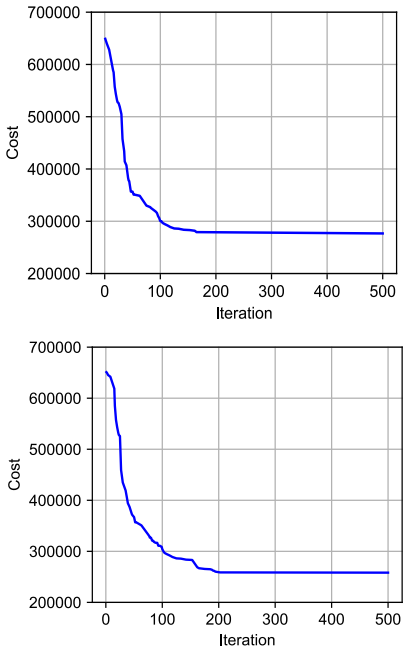

图 8 客户规模为 50 时的迭代曲线
Fig. 8 Iteration curves at a customer size of 50

从上述结果可以分析出，本文提出的强化学习结合演化算法(RL+MFEA)的求解方式，无论在100个客户还是50个客户的双层优化问题，在一半以上的测试用例都要优于其它方法。在演化算法的染色体交叉操作中，本文使用了强化学习训练的分配概率来指导优化过程，从而有效改善了演化算法。

5 结　论

物流配送系统是物流业的重要组成部分，其优化对于降低物流总体成本具有关键性的作用。本文旨在通过求解物流配送中心选址和路径优化问题，实现物流配送系统的优化。为了解决这一问题，我们利用了精确算法和启发式算法，并构建了一个双层选址-路径模型，采用强化学习和演化算法相结合的方法进行求解。实验结果表明，我们提出的方法在实践中是有效的，可以有效提高物流配送系统的效率和降低总体成本。

未来我们将考虑增大强化学习训练数据集的数量，以进一步提高选址问题的求解效果。此外，我们还将探索扩展配送时间窗口、环境绿化以及不同车辆类型等优化目标的多目标优化问题，这些问题还有待进一步研究和实验验证。

参考文献(References)

[1] Zhang X, Zhuang Y, Wang W, et al. Transfer Boosting With Synthetic Instances for Class Imbalanced Object Recognition[J]. IEEE Transactions on Cybernetics, 2016, 46(8): 1823-1836.

[2] Gupta A, Ong Y S, Feng L. Multifactorial Evolution: Toward Evolutionary Multitasking[J]. IEEE Transactions on Evolutionary Computation, 2016, 20(3): 343-357.

[3] Abhishek, Gupta, Yew-Soon, et al. Insights on Transfer Optimization: Because Experience is the Best Teacher[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2017, 2(1): 51-64.

[4] Nagy G, Sallhi S. Location-routing: issues, models and methods[J]. European Journal of Operational Research，2007，177(2): 649-672.

[5] Yu X, Zhou Y, Liu X F. A novel hybrid genetic algorithm for the location routing problem with tight capacity constraints[J]. Applied Soft Computing Journal, 2019, 76: 504-515.

[6] Kechmane L, Nsivi B, Baalal A. A hybrid particle swarm optimization algorithm for the capacitated location routing problem[J]. International Journal of Intelligent Computing and Cybernetics, 2018, 11(1): 106-120.

[7] Zhuo Dai，Fa B，Kg C，et al. A two-phase method for multi-echelon location-routing problems in supply chains[J]. Expert Systems with Applications, 2019，115：618-634.

[8] Hopfield JJ, Tank DW. "Neural" computation of decisions in optimization problems. Biological Cybernetics, 1985, 52(3): 141-152.

[9] Bello I, Pham H, Le Quoc V, et al. Neural combinatorial optimization with reinforcement learning[J]. arXiv preprint arXiv:1611.09940, 2016.

[10] Kool W, Van Hoof H, Welling M. Attention, learn to solve routing problems!. arXiv preprint arXiv:1803.08475, 2018.

[11] Nazari M, Oroojlooy A, Snyder L, et al. Reinforcement Learning for Solving the Vehicle Routing Problem[J]. Advances in Neural Information Processing Systems (NIPS), 2018, 31: 9839-9849.

[12] Prins C, Prodhon C, Calvo R W. Solving the capacitated location-routing problem by a GRASP complemented by a learning process and a path relinking. 4OR, 2006, 4(3): 221-238.

[13] FeFeng L, Huang Y, Zhou L, et al. Explicit Evolutionary Multitasking for Combinatorial Optimization: A Case Study on Capacitated Vehicle Routing Problem. IEEE Transactions on Cybernetics, 2021, PP(99): 1-14.