

文章编号:1006-2467(2022)01-0081-08

DOI: 10.16183/j.cnki.jsjtu.2020.210

# 车辆路径规划问题的逆向优化方法

陈禹伊, 陈璐

(上海交通大学 机械与动力工程学院, 上海 200240)

**摘要:** 在电商物流的“最后一公里”配送中, 经验丰富的驾驶员(专家)并不总是基于最短路径成本矩阵进行路径规划. 对此, 提出一种逆向优化方法, 通过学习专家的过往路径决策, 得到能够代表专家经验的成本矩阵, 并应用于路径规划模型求解, 使得专家经验能够融入决策算法中. 利用机器学习中的乘性权重更新算法实现对专家经验的学习. 随机算例和电商实际算例的实验结果证明了方法的有效性.

**关键词:** 逆向优化; 车辆路径规划问题; 成本矩阵; 经验学习

**中图分类号:** TP 181

**文献标志码:** A

## An Inverse Optimization Approach of Vehicle Routing Problem

CHEN Yuyi, CHEN Lu

(School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China)

**Abstract:** Generally, experienced drivers or experts do not always follow the shortest path in the last mile delivery of e-commerce. Hence, an inverse optimization approach was proposed to obtain a proper cost matrix by learning from the experts' past experience. Thus, the routing model with respect to the learned cost matrix could provide solutions as good as those given by experts. An algorithm-based multiplicative weights updates algorithm was applied to achieve the experience learning process. The experimental analyses based on the random and real-life instances demonstrate the effectiveness of this approach.

**Key words:** inverse optimization; vehicle routing problem(VRP); cost matrix; experience learning

随着信息技术的不断发展, 电子商务(简称电商)已成为最受欢迎的消费方式之一. 而在线交易量的急剧增加为物流网络带来了巨大压力, 因此高效的商品交付是电商公司提高服务水平的关键因素之一. 电商物流中的“最后一公里”交付问题, 即商品从配送站到客户的配送过程是一个典型的有容量约束的车辆路径规划问题(Capacitated Vehicle Routing Problem, CVRP), 可以描述为由一个配送中心向  $n$  个客户派遣  $K$  辆车配送货物, 每个客户由一辆车服

务且仅被服务一次. 每辆车设有运送容量上限, 而每个客户有特定的货物量需求. 其目的为得到成本最低的客户配送服务路线.

目前, 利用 CVRP 求解算法<sup>[1-2]</sup> 估算现实生活中各条道路的行驶成本较为困难, 得到的优化结果存在较大误差, 实际应用价值往往较低. Bertsimas 等<sup>[3]</sup> 认为, 各条道路的行驶成本之间存在相互依赖性, 因此对基于最短路径或最短行驶时间成本矩阵的 CVRP 模型进行优化难以得到符合实际需求的

收稿日期: 2020-07-08

基金项目: 国家自然科学基金资助项目(51775347)

作者简介: 陈禹伊(1996-), 女, 重庆市人, 硕士生, 研究方向为路径规划.

通信作者: 陈璐, 女, 副教授; E-mail: chenlu@sjtu.edu.cn.

路径.此外,经验丰富的驾驶员(专家)并不总是遵循最短路径成本矩阵选择配送路径,如优先避免易产生拥堵或交通信号灯多的道路.因此,在路径规划中还需要融入驾驶员的经验或偏好. Kovacs 等<sup>[4]</sup> 研究考虑一致性的车辆路径规划问题. 其中,一致性包括人员一致性,即客户希望被所熟悉的驾驶员服务<sup>[5]</sup>; 区域一致性,即让驾驶员在更为熟悉的服务区域进行配送<sup>[6-7]</sup>. 然而,驾驶员的偏好并不总需要完全遵循一致性原则. Liu 等<sup>[8]</sup> 通过分析驾驶员的连续数据轨迹来揭示驾驶员的操作模式. 韦增欣等<sup>[9]</sup> 建立基于多属性决策方法的路径优化模型,并根据驾驶员的偏好调整模型参数. Pahlavani 等<sup>[10]</sup> 利用局部线性神经模糊模型学习驾驶员偏好来预测路线. He 等<sup>[11]</sup> 根据市区出租车司机的经验,利用协作偏好发现算法和智能司机网络生成算法计算在旅行时间变化的情况下更可靠的行驶路线. 黄敏等<sup>[12]</sup> 根据出租车司机的经验,提出约束深度强化学习算法,并在线计算不同时间段内的最快行驶路线. 以上研究多针对特定应用场景进行数据或实验分析,因此方法的应用价值有限.

逆向优化是指已知一个优化问题的专家解(最优解),从而逆向推断优化模型的参数<sup>[13]</sup>. Ahuja 等<sup>[13-14]</sup> 研究了基于最优性条件和对偶理论的逆线性优化问题,并将其扩展到逆向网络流问题. Zhang 等<sup>[15-16]</sup> 利用牛顿法解决了在  $l_\infty$  范数下的逆向组合问题. Heuberger<sup>[17]</sup> 综述了不同逆向组合优化问题的算法. 随着大数据应用的发展,研究者提出在逆向优化中结合数据来设计开发算法. You 等<sup>[18]</sup> 对城市卡车货运量进行建模和预测,并提出利用观测数据对车辆路径规划问题的模型参数进行校准,使得优化结果更符合实际需求. 牟健慧等<sup>[19]</sup> 将逆向优化理论和方法引入车间调度领域,建立考虑调度效率和稳定性的数学模型,以解决多目标流水车间的逆调度问题. Bärmann 等<sup>[20]</sup> 结合逆向优化和在线学习算法的思想,将模型解决方案与专家决策之间的差异设为成本矩阵的梯度,并利用迭代法令梯度逐渐降低并收敛. Aswani 等<sup>[21-22]</sup> 考虑了在凸优化问题最优解的测量结果被噪声破坏情况下的逆向优化问题. Saez-gallego 等<sup>[23]</sup> 利用逆向优化法预测对价格敏感的消费者的电力总需求,建立电力总需求的价格响应模型,并利用广义逆向优化法估算模型参数.

综上分析,提出一种逆向优化方法. 通过学习配送过程中专家的过往路径决策,逆向优化得到能够表示专家对道路选择偏好的成本矩阵,并将该矩阵应用于 CVRP 模型,令配送路径能够融入专家经验.

## 1 逆向优化模型

对“最后一公里”的配送问题进行建模,并根据该模型的对偶模型建立逆向优化数学模型.

### 1.1 CVRP 模型

配送问题描述如下:共有  $K$  辆车为  $n$  个客户进行配送,每个客户仅由一辆车负责服务,每辆车的容量约束为  $Q$ ,第  $i$  个客户的需求量为  $q_i (i = 1, 2, \dots, n)$ ,  $0 \leq q_i \leq Q$ . 定义集合  $V = \{0, 1, \dots, n\}$ , 其中 0 表示配送中心.  $C = \{c_{ij}, i, j \in V\}$  为成本矩阵,其中  $c_{ij}$  为从客户  $i$  到客户  $j$  的行驶成本. 优化目标为在满足车辆容量约束的前提下,得到总行驶成本最小的车辆配送路径.

模型的决策变量包括  $x_{ij} (i, j \in V)$  和车辆到达客户  $i$  时的负载  $u_i (i \in V/\{0\})$ . 如果车辆直接从客户  $i$  行驶到客户  $j$ , 则  $x_{ij} = 1$ , 否则  $x_{ij} = 0$ .

CVRP 的数学模型(模型 L)表示如下:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (1)$$

$$\text{s. t. } \sum_{j \in V} x_{ij} = 1, \quad \forall i \in V/\{0\} \quad (2)$$

$$\sum_{i \in V} x_{ij} = 1, \quad \forall j \in V/\{0\} \quad (3)$$

$$\sum_{j \in V} x_{0j} = K \quad (4)$$

$$u_i - u_j + Q(1 - x_{ij}) \geq q_i \quad (5)$$

$$u_i \leq Q, \quad \forall i \in V/\{0\} \quad (6)$$

$$u_i \geq q_i, \quad \forall i \in V/\{0\} \quad (7)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in V \quad (8)$$

目标函数式(1)使得总行驶成本最小化;式(2)和(3)确保每一位客户仅被服务一次并控制网络流守恒;式(4)确保配送车辆数目为  $K$ ;式(5)为子回路消除约束和车辆容量约束;式(6)和(7)用于定义  $u_i$ ;式(8)为决策变量的属性定义.

### 1.2 逆向优化模型

在模型 L 的对偶模型基础上建立逆向优化模型,目标为给定专家解  $\mathbf{x}^*$ , 在满足约束的前提下,对模型中的成本矩阵  $C$  进行最小化变动,得到一个新的成本矩阵  $\bar{C} = \{\bar{c}_{ij}, i, j \in V\}$ , 使得  $\mathbf{x}^*$  成为模型 L 的最优解.

定义集合  $J = \{(i, j) \mid x_{ij}^* = 1\}$  和  $\bar{J} = \{(i, j) \mid x_{ij}^* = 0\}$ . 为简化运算,假设:

$$\bar{c}_{ij} = c_{ij} + \eta_{ij} - \sigma_{ij}, \quad \forall i, j \in V$$

其中:  $x_{ij}^*$  为  $\mathbf{x}^*$  的决策变量;  $\eta_{ij}$  为  $\bar{c}_{ij}$  的增量;  $\sigma_{ij}$  为  $c_{ij}$  的减量. 且  $\eta_{ij} \geq 0$ ,  $\sigma_{ij} \geq 0$ ,  $\eta_{ij}\sigma_{ij} \geq 0$ .  $\eta_{ij}$  和  $\sigma_{ij}$  即为  $C$  的变动量. 则逆向优化模型(模型 Inv-L)表示为

$$\min \sum_{i \in V} \sum_{j \in V} (\eta_{ij} + \sigma_{ij}) \quad (9)$$

$$\text{s. t. } \sum_{i \in V} \sum_{j \in V} (c_{ij} + \eta_{ij} - \sigma_{ij}) x_{ij}^* - K\delta - \sum_{i \in V} \sum_{j \in V} \varphi_{ij} - \sum_{j \in V/\{0\}} \sum_{i \in V/\{0\}} [\alpha_i + \beta_j + (Q - q_i) \gamma_{ij} + Q\lambda_i + q_i \mu_i] = 0 \quad (10)$$

$$\delta \leq c_{00} + \eta_{00} \quad (11)$$

$$\beta_j + \delta \leq c_{0j} + \eta_{0j}, \quad \forall (0, j) \in \bar{J} \quad (12)$$

$$\beta_j + \delta + \phi_{0j} = c_{0j} + \eta_{0j} - \sigma_{0j} \quad (13)$$

$$\forall (0, j) \in J$$

$$\alpha_i \leq c_{i0} + \eta_{i0}, \quad \forall (i, 0) \in \bar{J} \quad (14)$$

$$\alpha_i + \phi_{i0} = c_{i0} + \eta_{i0} - \sigma_{i0}, \quad \forall (i, 0) \in J \quad (15)$$

$$\alpha_i + \beta_j - Q\gamma_{ij} \leq c_{ij} + \eta_{ij} \quad (16)$$

$$\forall (i, j) \in \bar{J}, i, j \in V/\{0\}$$

$$\alpha_i + \beta_j - Q\gamma_{ij} + \varphi_{ij} = c_{ij} + \eta_{ij} - \sigma_{ij} \quad (17)$$

$$\forall (i, j) \in J, i, j \in V/\{0\}$$

$$\sum_{j \in V/\{0\}} \gamma_{ij} - \sum_{j \in V/\{0\}} \gamma_{ji} + \lambda_i + \mu_i \leq 0 \quad (18)$$

$$\forall i \in V/\{0\}$$

$$\alpha_i, \beta_j, \delta \in R, \quad \forall i, j \in V/\{0\} \quad (19)$$

$$\lambda_i, \phi_{ij} \leq 0, \quad \forall i, j \in V/\{0\} \quad (20)$$

$$\gamma_{ij}, \mu_i, \eta_{ij}, \sigma_{ij} \geq 0, \quad \forall i, j \in V/\{0\} \quad (21)$$

式中:  $\alpha_i, \beta_j, \delta, \gamma_{ij}, \lambda_i, \mu_i$  和  $\phi_{ij}$  分别为模型 L 中式(2)~(8)的对偶变量。目标函数式(9)使  $\mathbf{C}$  的变动量最小化;式(10)确保  $\mathbf{x}^*$  成为模型 L 的最优解;式(11)~(18)为对偶约束;式(19)~(21)为对偶变量的属性定义。

利用模型 Inv-L 的迭代算法(Inv-L Algorithm, ILA),输入过往专家解,迭代多次求解 Inv-L 模型,每次求解时对  $\mathbf{C}$  进行变动,最终得到一个能够代表专家经验的成本矩阵。然而,该算法在每次迭代时均需要对模型 Inv-L 求最优解,因此求解时间较长。对此,开发一种在线学习算法以实现对过往专家解的学习。

## 2 乘性权重更新算法

乘性权重更新(Multiplicative Weights Updates, MWU)算法<sup>[24]</sup>采用迭代更新的方式,在每一次迭代中,通过定义每轮的损失成本,并利用乘法更新规则对决策进行更新,能够使得总损失成本的期望最小化。采用改进的 MWU 算法,将原 MWU 算法中基于列向量的运算扩展至基于二维矩阵的运算,并针对问题属性确定算法学习率,使得改进后的算法具备收敛性。

### 2.1 算法描述

算法中的相关参数如表 1 所示。

表 1 参数定义

Tab. 1 Definition of parameters

符号	含义
$\mathbf{C}_{\text{true}}$	代表专家经验的真实成本矩阵,是未知量
$T$	训练集中专家解的个数
$\mu$	权重更新公式的学习率,与参数 $T$ 和 $n$ 相关
$\mathbf{C}_t$	第 $t$ 轮的成本矩阵
$X(p_t)$	第 $t$ 轮模型 L 的约束条件
$\mathbf{x}_t^*$	第 $t$ 轮的专家解
$\mathbf{x}_t$	第 $t$ 轮基于当前 $\mathbf{C}_t$ 得到的模型 L 最优解
$\mathbf{w}_t$	第 $t$ 轮的决策权重
$\mathbf{P}$	代表专家经验的真实决策概率
$\mathbf{P}_t$	第 $t$ 轮的决策概率,形式上为归一化后的 $\mathbf{w}_t$
$\mathbf{m}_t$	第 $t$ 轮的损失成本

MWU 算法通过学习过往的专家解,使得  $\mathbf{C}_t$  不断趋近于  $\mathbf{C}_{\text{true}}$ 。在每一次(设为第  $t$  轮)迭代中,首先根据  $\mathbf{C}_t$  和  $X(p_t)$ ,正向求得  $\mathbf{x}_t$ 。然后,将  $\mathbf{x}_t^*$  和  $\mathbf{x}_t$  之间的差异  $\frac{\mathbf{x}_t^* - \mathbf{x}_t}{\|\mathbf{x}_t^* - \mathbf{x}_t\|_\infty}$  作为  $\mathbf{m}_t$ 。算法依据  $\mathbf{m}_t$  更新下一轮的决策权重,更新公式如下:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu(\mathbf{w}_t \odot \mathbf{m}_t) \quad (22)$$

式中:  $\mathbf{a} \odot \mathbf{b}$  表示维数相同的两个矩阵对应分量相乘。

给定初始权重  $\mathbf{w}_1 = (\theta_{ij})_{n \times n}$ ,  $\theta_{ij} = 1$ ;  $\mu = \sqrt{\frac{\ln n^2}{T}}$ 。MWU 算法的伪代码如下所示。

MWU 算法伪代码

Input:  $(X(p_t), \mathbf{x}_t^*)$ ,  $t = \{1, 2, \dots, T\}$ ,  $\mathbf{w}_1$ ,  $\mu$   
for  $t = 1, 2, \dots, T$  do

$$\mathbf{C}_t = \frac{\mathbf{w}_t}{\|\mathbf{w}_t\|_1}$$

$$\mathbf{x}_t = \arg \min (\mathbf{C}_t \cdot \mathbf{x}_t \mid \mathbf{x}_t \in X(p_t))$$

$$\text{if } \mathbf{x}_t = \mathbf{x}_t^*$$

$$\mathbf{m}_t = \mathbf{0}$$

else

$$\mathbf{m}_t = \frac{\mathbf{x}_t^* - \mathbf{x}_t}{\|\mathbf{x}_t^* - \mathbf{x}_t\|_\infty}$$

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \mu(\mathbf{w}_t \odot \mathbf{m}_t)$$

end if

end for

return  $\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_T$

对于第  $t$  轮,算法首先通过标准化  $w_t$  得到  $C_t$ ,并基于  $C_t$  和  $X(p_t)$  正向求得  $x_t$ ,然后计算  $m_t$ .最后基于  $m_t$  和式(22)将  $w_t$  更新为  $w_{t+1}$ .经过反复迭代,得到一个成本矩阵序列  $C_1, C_2, \dots, C_T$ ,其中  $C_T$  为最终学习得到的代表专家经验的成本矩阵.由于 MWU 算法具有收敛性,所以可以保证经过足够多次学习迭代后得到的  $C_T$  无限接近于  $C_{\text{true}}$ .

在 MWU 算法中,仅存在一个循环体,在每个循环中均需要调用数学优化技术 CPLEX 执行一次 CVRP 的求解.由于 CPLEX 求解线性规划的内置算法为分支定界算法,所以 MWU 算法的时间复杂度为  $O(n'2^{n'})$ ,其中  $n'$  为输入参数.

## 2.2 算法收敛性

在 MWU 算法中,  $P_t = \frac{w_t}{\|w_t\|_1}$ ,总的损失成本期望为  $\sum_{t=1}^T (m_t \cdot P_t)$ . Arora 等<sup>[25]</sup>证明了 MWU 算法可以确保对于任何概率分布  $P$ ,均有

$$\sum_{t=1}^T (m_t \cdot P_t) \leq \sum_{t=1}^T [(m_t + \mu \text{abs}(m_t)) \cdot P] + \frac{\ln n^2}{\mu} \quad (23)$$

式中:  $\text{abs}(\cdot)$  为求数据绝对值的函数.

定义  $F = \{C \in \mathbf{R}^{n \times n} \mid \|C\|_1 = 1, c_{ij} \geq 0\}$ . 将  $C_{\text{true}}$  和  $C_t$  代入式(23),可以证明当  $\mu = \sqrt{\frac{\ln n^2}{T}}$  时,均有

$$0 \leq \frac{1}{T} \sum_{t=1}^T [(C_{\text{true}} - C_t) \cdot (x_t - x_t^*)] \leq 2\sqrt{\frac{\ln n^2}{T}} \quad (24)$$

即 MWU 算法具有收敛性.

## 3 算例分析

利用不同维度的算法指标评价 MWU 算法的有效性,然后分别利用 VRPLIB 算例库中的随机算例和电商真实算例进行实验分析,所需计算机配置为 Intel® Core™ i7-8500@3.60 GHz,内存为 8 GB. 程序由 MATLAB R2019b 进行编译,模型求解调用 CPLEX IBM 12.9.0.

VRPLIB 算例库中的随机算例包括成本矩阵、车辆容量和客户需求等信息.基于给定的成本矩阵,通过更改客户需求和车辆容量能够生成多个不同算例,求得各算例的最优解并作为专家解.将同一规模下的专家解分成两部分,其中 90% 的专家解为训练集,10% 的专家解为验证集.

### 3.1 算法指标

从算法的有效性、可实现性和收敛性等方面提出以下评价指标.

平均目标函数值偏差:

$$\epsilon_1 = \frac{1}{M} \sum_{i=1}^M \frac{|C_{\text{true}} \cdot x_i^* - C_{\text{true}} \cdot x_i|}{C_{\text{true}} \cdot x_i^*}$$

其中:  $M$  为验证集的数量.该指标用于衡量由逆向优化学习算法得到的最优解与真实专家解目标函数值之间的平均偏差.

平均收敛误差:

$$\epsilon_2 = \frac{1}{T} \sum_{t=1}^T [(C_{\text{true}} - C_t) \cdot (x_t - x_t^*)]$$

该指标用于衡量算法的收敛误差.

平均编辑距离:

$$E = \frac{1}{M} \sum_{i=1}^M \text{Edit}(x_i^*, x_i)$$

其中:  $\text{Edit}$  为求解两个变量编辑距离的函数.编辑距离为将一个字符串转换为另一个字符串所需要的最小操作数.该指标用于衡量验证集中的专家解和由逆向优化学习法得到的最优解之间的平均编辑距离.

平均相似度:

$$S = 1 - \frac{1}{M} \sum_{i=1}^M \frac{\text{Edit}(x_i^*, x_i)}{|H|}$$

其中:  $|H|$  为算例规模.判断两个解的相似度需要同时考虑  $E$  和  $|H|$ .此外,评价指标还包括算法的学习时间(CPU).

### 3.2 基于随机算例的实验分析

**3.2.1 学习效果分析** 分别利用 ILA 和 MWU 算法求解具有 10 个客户点的算例.当训练集数( $N_1$ )和验证集数( $N_2$ )逐渐增大时,两种算法的性能分析如图 1 所示.可知,在指标  $\epsilon_1, \epsilon_2, S$  和 CPU 方面, MWU 算法均优于 ILA.随着训练集数量的增多,两种算法的学习效果均有所提升,提升趋势均为先激增后趋于平缓,且 MWU 算法的提升幅度更显著.因此,在求解大规模算例时,可以考虑权衡训练时间和训练效果.虽然 MWU 算法需要花费较长时间学习  $C_T$ ,但在后续过程中,可以直接利用  $C_T$  求解新问题.而其在求解新问题时的效率等同于求解一个 CVRP 问题时的效率.因此,前期训练完成后, MWU 算法求解新问题的效率较高.

根据上述结论,选择 MWU 算法求解后续算例,结果如表 2 所示.可知,与验证集中已知专家解相比, MWU 算法可以提供  $S = 88.05\%$  和  $\epsilon_1 = 0.16\%$  的配送路径,表明了算法在学习专家经验方

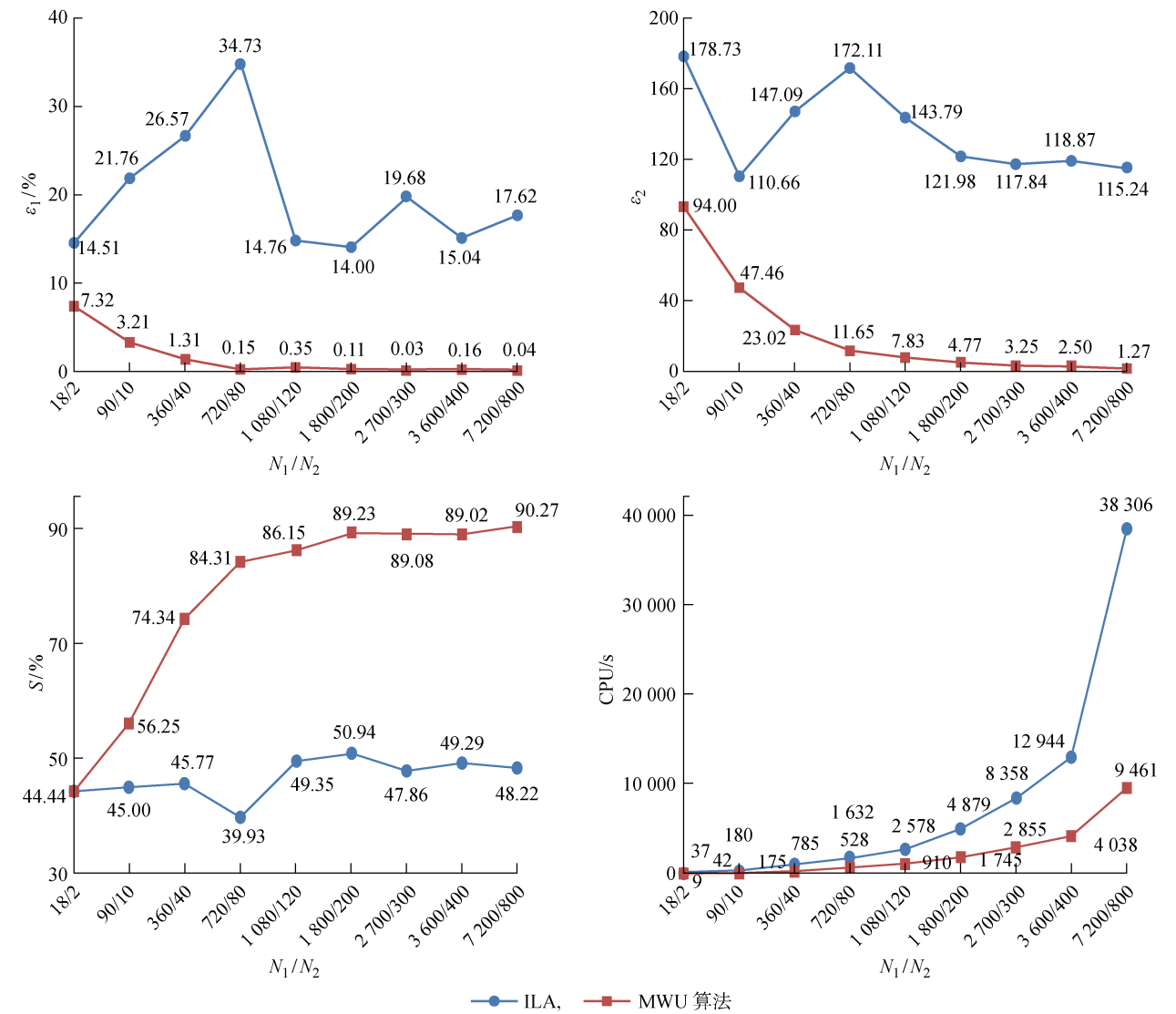


图 1 两种算法的效果对比

Fig. 1 Comparison of two learning algorithms

表 2 MWU 算法求解不同规模算例的结果  
Tab. 2 Performance of MWU algorithm for different size instances

$ H $	$\epsilon_1/\%$	$\epsilon_2$	$E$	$S/\%$	CPU/s
10	0.04	1.27	0.79	90.27	9 460.87
15	0.32	4.25	1.14	91.29	16 528.59
19	0.20	0.20	1.09	93.61	25 344.56
21	0.24	1.02	2.06	89.17	27 636.93
25	0.14	1.11	2.98	87.06	128 741.12
34	0.08	3.26	3.57	88.86	89 862.27
45	0.14	0.04	7.66	82.20	215 451.29
56	0.12	0.03	10.02	81.96	345 601.23
均值	0.16	1.40	3.66	88.05	107 328.36

面的有效性. 此外, MWU 算法的  $\epsilon_2 = 1.40$ , 表明算法的收敛性较好.

**3.2.2 网络图结构的敏感性分析** 上述随机算例均为非对称网络, 而在较多配送场景中, 网络结构呈对称性. 因此, 需要分析 MWU 算法应用于两种不同网络结构时的性能, 结果如图 2 所示. 保留非对称网络成本矩阵的上半部分不变, 可以将其转化为对称成本矩阵.

图 2 中, 与求解非对称网络的算例相比, MWU 算法在求解具有对称网络结构算例时的  $\epsilon_1$  值略低, 而  $\epsilon_2$  值相对一致. 表明算法的收敛性不受网络对称性影响. 由于对称网络中存在多个最优解, 会对算法学习造成干扰, 所以对其进行学习后所得求解结果的相似度较差. 此外, 随着算例规模增大, 对称网络的训练时间呈逐渐增长趋势.

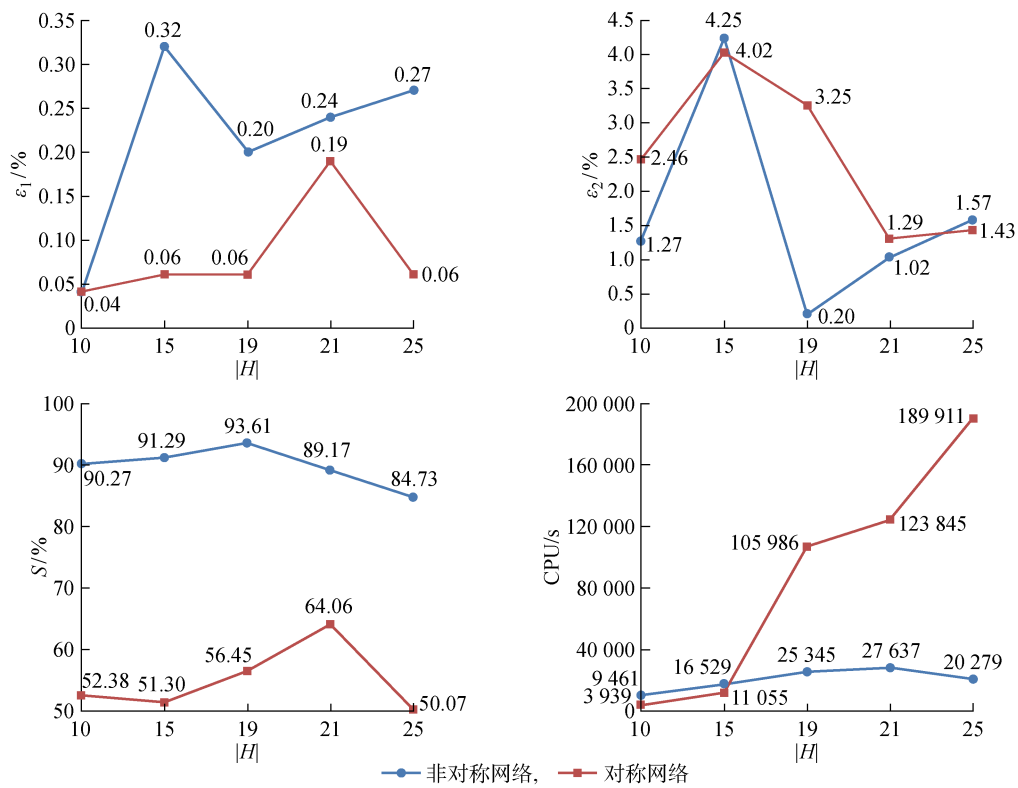


图 2 网络结构对称性敏感性分析  
Fig. 2 Sensitivity analysis of two networks

3.3 基于电商真实数据的算例分析

采用某电商在 2018 年 9 月至 11 月的日常家用电器配送路线进行算例分析. 服务区域为上海市徐汇区, 收集到包括 422 辆厢型货车在内的 20 万条订单的配送路线. 数据集包括配送的车次、时间和地址等. 根据企业反馈, 选择工作效率较高的驾驶员, 收集其配送路径, 并剔除异常解和存在客户时间窗约束等不属于研究范围的解, 但此时所得专家解并不等同于基于最短路径矩阵的 CVRP 模型最优解. 将

驾驶员的配送路径作为专家解进行学习, 以得到能够代表驾驶员经验的成本矩阵, 并用于求解新问题.

**3.3.1 数据预处理** 数据预处理包括缺省值、异常值和地址的经纬度转换处理. 为了增加客户点的重复度, 减少模型训练难度, 根据客户的地理位置采用 K-means 算法进行聚类. 从算法的可实现性、聚类合理性及其聚类指标等方面综合分析, 选择聚类数为 55, 客户聚类情况如图 3 所示. 数据预处理后共得到 185 个真实算例.

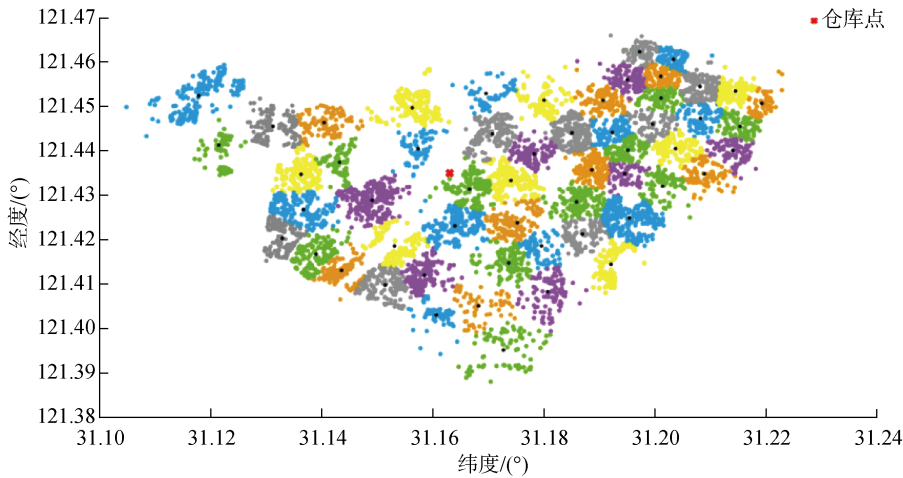


图 3 研究区域的客户聚类分布  
Fig. 3 Clustered customer network of studied district

**3.3.2 结果分析** 各聚类质心的欧氏距离成本矩阵为  $C_1$ , 算法学习过程为从 185 个算例中随机选择 184 个并利用 MWU 算法进行训练, 将剩余的一个算例作为验证算例. 将学习得到的  $C_T$  代入路径规划模型 L, 求解验证算例, 并对比所得解与已知专家解. 按上述过程训练 185 次, 求每一次训练效果的平均值. 由于真实场景下无法得到真实的成本矩阵, 所以本节不考虑指标  $\epsilon_1$  和  $\epsilon_2$ . 此外, 利用  $C_1$  求解算例, 并对比采用算法前后的优化效果, 如表 3 所示.

表 3 真实算例结果对比  
Tab.3 Comparison of real instances

算例	$ H $	$E$	$S/\%$	CPU/s
基于 $C_1$ 求解	56	17.95	57.62	—
基于 $C_T$ 求解	56	11.24	71.84	40 128.13
优化效果	—	37.38	24.67	—

可知, 当基于  $C_T$  求解算例时, 求解结果明显优于基于  $C_1$  的求解结果, 在指标  $E$  和  $S$  上分别优化了 37.38% 和 24.67%. 同时, 所得配送路径与专家解之间的平均相似度为 71.84%, 表明当真实算例的数量增加时, 算法的学习效应也将随之加强.

4 结语与展望

对电商物流中的“最后一公里”交付问题进行研究. 在现实生活中, 经验丰富的驾驶员(专家)并不总是遵循最短路径成本矩阵选择配送路径. 因此, 采用逆向优化法学习专家的过往路径决策方案, 训练得到代表专家对路径选择偏好的成本矩阵, 将其应用到路径规划模型中求解. 其中, 利用 MWU 算法实现对专家经验的学习, 并设计随机算例和电商真实物流算例进行验证. 结果表明: ① MWU 算法在收敛性、有效性和可实现性方面表现优异, 利用学习所得成本矩阵可以得到近似专家的配送路径; ② 算法收敛性不受网络结构非对称性和对称性的影响; ③ 在求解真实算例时, MWU 算法将所得解与专家解的平均相似度提升至 71.84%, 表明该算法能够有效利用过往经验, 提供与专家解相似度极高的配送路径.

在后续研究中, 可以将逆向优化法应用于具有不确定性车辆路径规划等更复杂的优化问题. 同时, 现实生活中的专家解集往往来源于多个专家, 如何对专家解进行分类并提供全局逆向优化方案, 同样值得探讨.

参考文献:

[1] 范厚明, 徐振林, 李阳, 等. 混合遗传算法求解多中

心联合配送路径问题[J]. 上海交通大学学报, 2019, 53(8): 1000-1009.

FAN Houming, XU Zhenlin, LI Yang, *et al.* Hybrid genetic algorithm for solving multi-depot joint distribution routing problem[J]. **Journal of Shanghai Jiao Tong University**, 2019, 53(8): 1000-1009.

[2] 熊浩, 鄢慧丽, 周和平, 等. 多阶段动态车辆路径问题实时优化策略[J]. 上海交通大学学报, 2013, 47(3): 450-453.

XIONG Hao, YAN Huili, ZHOU Heping, *et al.* Real-time optimization strategy of the multi-period dynamic vehicle routing problems [J]. **Journal of Shanghai Jiao Tong University**, 2013, 47(3): 450-453.

[3] BERTSIMAS D, GUPTA V, PASCHALIDIS I C. Data-driven estimation in equilibrium using inverse optimization[J]. **Mathematical Programming**, 2015, 153(2): 595-633.

[4] KOVACS A A, GOLDEN B L, HARTL R F, *et al.* Vehicle routing problems in which consistency considerations are important: A survey [J]. **Networks**, 2014, 64(3): 192-213.

[5] SMILOWITZ K, NOWAK M, JIANG T T. Workforce management in periodic delivery operations[J]. **Transportation Science**, 2013, 47(2): 214-230.

[6] HAUGHTON M A. The efficacy of exclusive territory assignments to delivery vehicle drivers[J]. **European Journal of Operational Research**, 2008, 184(1): 24-38.

[7] ZHONG H S, HALL R W, DESSOUKY M. Territory planning and vehicle dispatching with driver learning[J]. **Transportation Science**, 2007, 41(1): 74-89.

[8] LIU L, ANDRIS C, RATTI C. Uncovering cabdrivers' behavior patterns from their digital traces[J]. **Computers, Environment and Urban Systems**, 2010, 34(6): 541-548.

[9] 韦增欣, 陈进来, 罗朝晖. 基于驾驶员偏好的最优路径选择[J]. 交通运输系统工程与信息, 2010, 10(6): 141-144.

WEI Zengxin, CHEN Jinlai, LUO Chaohui. Optimal path selection based on driver's preference[J]. **Journal of Transportation Systems Engineering and Information Technology**, 2010, 10(6): 141-144.

[10] PAHLAVANI P, DELAVAR M R. Multi-criteria route planning based on a driver's preferences in multi-criteria route selection[J]. **Transportation Research Part C: Emerging Technologies**, 2014, 40: 14-35.

- [11] HE Z C, CHEN K Y, CHEN X Y. A collaborative method for route discovery using taxi drivers' experience and preferences[J]. **IEEE Transactions on Intelligent Transportation Systems**, 2018, 19(8): 2505-2514.
- [12] 黄敏, 毛锋, 钱宇翔. 基于出租车司机经验的约束深度强化学习算法路径挖掘[J]. **计算机应用研究**, 2020, 37(5): 1298-1302.
- HUANG Min, MAO Feng, QIAN Yuxiang. Mining fastest route using taxi drivers' experience via constrained deep reinforcement learning[J]. **Application Research of Computers**, 2020, 37(5): 1298-1302.
- [13] AHUJA R K, ORLIN J B. Combinatorial algorithms for inverse network flow problems[J]. **Networks**, 2002, 40(4): 181-187.
- [14] AHUJA R K, ORLIN J B. Inverse optimization[J]. **Operations Research**, 2001, 49(5): 771-783.
- [15] ZHANG J Z, LIU Z H. Calculating some inverse linear programming problems[J]. **Journal of Computational and Applied Mathematics**, 1996, 72(2): 261-273.
- [16] ZHANG J Z, LIU Z H. A general model of some inverse combinatorial optimization problems and its solution method under  $l_\infty$  norm[J]. **Journal of Combinatorial Optimization**, 2002, 6(2): 207-227.
- [17] HEUBERGER C. Inverse combinatorial optimization: A survey on problems, methods, and results [J]. **Journal of Combinatorial Optimization**, 2004, 8(3): 329-361.
- [18] YOU S I, CHOW J Y J, RITCHIE S G. Inverse vehicle routing for activity-based urban freight forecast modeling and city logistics[J]. **Transportmetrica A: Transport Science**, 2016, 12(7): 650-673.
- [19] 牟健慧, 郭前建, 高亮, 等. 基于混合的多目标遗传算法的多目标流水车间逆调度问题求解方法[J]. **机械工程学报**, 2016, 52(22): 186-197.
- MOU Jianhui, GUO Qianjian, GAO Liang, *et al.* Multi-objective genetic algorithm for solving multi-objective flow-shop inverse scheduling problems[J]. **Journal of Mechanical Engineering**, 2016, 52(22): 186-197.
- [20] BÄRMANN A, MARTIN A, POKUTTA S, *et al.* An online-learning approach to inverse optimization [DB/OL]. (2020-03-29)[2020-07-08]. <https://arxiv.org/abs/1810.12997v2>.
- [21] ASWANI A, SHEN Z J M, SIDDIQ A. Inverse optimization with noisy data[J]. **Operations Research**, 2018, 66(3): 870-892.
- [22] CHAN T C Y, LEE T, TEREKHOV D. Inverse optimization: Closed-form solutions, geometry, and goodness of fit[J]. **Management Science**, 2019, 65(3): 1115-1135.
- [23] SAEZ-GALLEGO J, MORALES J M. Short-term forecasting of price-responsive loads using inverse optimization[J]. **IEEE Transactions on Smart Grid**, 2018, 9(5): 4805-4814.
- [24] ARORA S, HAZAN E, KALE S. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method[C]// **46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)**. Pittsburgh, USA: IEEE, 2005: 339-348.
- [25] ARORA S, HAZAN E, KALE S. The multiplicative weights update method: A meta-algorithm and applications[J]. **Theory of Computing**, 2012, 8(1): 121-164.

(本文编辑:孙伟)