计算机科学
COMPUTER SCIENCE

基于双重指针网络的车货匹配双重序列决策研究

蔡岳, 王恩良, 孙哲, 孙知信

引用本文

蔡岳, 王恩良, 孙哲, 孙知信.基于双重指针网络的车货匹配双重序列决策研究[J]. 计算机科学, 2022, 49(11A): 210800257-9.

CAI Yue, WANG En-liang, SUN Zhe, SUN Zhi-xin. Study on Dual Sequence Decision-making for Trucks and Cargo Matching Based on Dual Pointer Network [J]. Computer Science, 2022, 49(11A): 210800257-9.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800257

基于双重指针网络的车货匹配双重序列决策研究

蔡 岳 王恩良 孙 哲 孙知信
南京邮电大学江苏省邮政大数据技术与应用工程研究中心 南京 210023
南京邮电大学国家邮政局邮政行业技术研发中心(物联网技术) 南京 210023
(caiyue_china@outlook.com)

摘 要 由于我国对公路运输资源利用不均,车货供需问题成为如今的热点问题。车货供需匹配平台为最大化总体运力资源利用率,需要整合运输需求和运力,降低成本并提高效率。大部分平台通常采用启发式算法求解车货匹配问题,此类算法面对大规模的问题时存在寻优瓶颈。针对上述问题,首次将车货供需匹配问题转变成一种双重序列决策问题,据此研究适用于当今车货供需匹配环节的一种高效算法。首先,提出了一种车货匹配的数学模型,并将该模型抽象为双重序列决策问题,再创新性地提出双重指针网络算法求解该问题。本实验使用 Actor-Critic 算法作为模型的训练框架来训练双重指针网络,并评估了模型。经实验得,双重指针网络的车货匹配求解方法的寻优能力在小问题规模中与传统启发式算法相当,在大问题规模中超越启发式算法,同时时间消耗都大大下降。

关键词:双重指针网络;双重序列决策问题;深度强化学习;组合优化;车货匹配;critic 网络
中图法分类号 TP311

Study on Dual Sequence Decision-making for Trucks and Cargo Matching Based on Dual Pointer Network

CAI Yue, WANG En-liang, SUN Zhe and SUN Zhi-xin
Post Big Data Technology and Application Engineering Research Center of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
Post Industry Technology Research and Development Center of the State Posts Bureau(Internet of Things Technology), Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Abstract Due to the uneven utilization of road transportation resources in my country, the supply and demand of trucks and cargo become a hot issue today. In order to maximize the utilization of overall transportation resources, the truck and cargo supply-demand matching platform needs to integrate transportation demand and capacity, reduce costs and improve efficiency. The algorithms used by most platforms are usually heuristic algorithms to solve the problem of trucks-cargo matching. Such algorithms have a bottleneck in optimizing when faced with large-scale problems. In response to the above-mentioned problems, this paper transforms the supply-demand matching problem of vehicles and goods into a double sequence decision-making problem for the first time. Based on this, we study an efficient algorithm that is suitable for today's vehicle and goods supply-demand matching links. First, a mathematical model of trucks-cargo matching is proposed and the model is abstracted as a double sequence decision problem, and then a dual-pointer-network algorithm is innovatively proposed to solve this problem. The experiment uses the Actor-Critic algorithm as the model training framework to train the dual-pointer-network and evaluates the model. Experiments show that the dual-pointer-network's vehicle-to-cargo matching solution method is equivalent to traditional heuristic algorithms in small problem scales, and surpasses heuristic algorithms in large problem scales. At the same time, the time consumption is greatly reduced.

Keywords Dual pointer network, Double sequence decision-making problem, Deep reinforcement learning, Combinatorial optimization, Trucks and cargo matching, Critic network

1 引言

组合优化(Combinatorial Optimization)是运筹学[1]、工程学[2]、生物学[3]等众多领域都涉及的课题,其本质上是一种离散空间中的优化问题,在许多实际工程类项目中具有重要应用[4]。因此,一个多世纪以来,该问题得到研究界的广泛关注。在研究初期,许多常见问题会采用启发式算法[5]进行搜索求解,而如今研究者更倾向于将问题描述为一个顺序决策

基金项目:国家自然科学基金(61972208)
This work was supported by the National Natural Science Foundation of China(61972208).
通信作者:孙知信(sunzx@njupt.edu.cn)

过程并构建数学模型[6]进行解决。

车货供需匹配问题是一种集 TSP(旅行商问题)[7]、VRP(车辆路由问题)[8]、背包问题[9]、装箱问题[10]、带时间窗[11]等于一体的复杂组合优化问题,具有极高的复杂性,传统的问题建模方法以及启发式算法均存在求解耗时长、大问题规模出现瓶颈的局限性。在车货匹配问题中考虑了货物的质量、体积、价值等;车辆的载重、运费、容积、车速等;路线的距离、优先级等;还有违约金、司机费用、车辆成本或租金等许多复杂约束,需要对其约束条件求解,可见其可被视为在离散状态的约束条件下求极值的最优化运筹问题。此时若以收益最大化为例,则收益函数 $F(x)$ 可看作如下数学模型:

$$\max F(x) \tag{1}$$

$$\text{s.t. } G(x)\geqslant 0 \tag{2}$$

$$x\in D \tag{3}$$

其中,x 为决策变量,$F(x)$ 为目标函数,$G(x)$ 为约束条件,D 表示离散的决策空间即有限个 x 取值范围。

由于我国正处于物流行业的发展阶段,许多新型物流模式尚处不成熟,在源目地址不确定的离散条件下的物流体系还处于探索阶段,大量车货供需平台没有实现较为高效的方法,从而导致车货供需不均衡、资源利用率低、受众群体更倾向于私人承包等,进而导致缺乏智能化的供需匹配体系,缺乏对交易的控制力及难监管等弊端。

为了促进车货匹配领域智能化、标准化,本文对上述业务进行数学化的约束和规范,采用了一种深度强化学习方法来求解上述车货匹配问题的数学模型。受车货匹配问题的启发,提出了一种带约束的双重序列决策问题(Constrained Double Sequence Decision-making Problem,CDSDP)。传统的单个序列的决策问题可以描述为:

$$\max \sum_{seq} Reward(seq)p(seq) \tag{4}$$

$$\text{s.t. } seq\in D \tag{5}$$

其中,$Reward$ 表示回报函数,seq 表示一个动作序列,$p(seq)$ 表示该序列出现的概率,D 为可行的解空间,目标是最大化回报的期望。而两个相关的序列决策问题可以描述为:

$$\max \sum_{seq_2}\sum_{seq_1} Reward(seq_1,seq_2)p(seq_1|seq_2)p(seq_2) \tag{6}$$

$$\text{s.t. } seq_1,seq_2\in D \tag{7}$$

其中,$seq1$ 与 $seq2$ 是两个序列,$Reward$ 函数与两个序列均有关,且 $seq1$ 与 $seq2$ 不独立,即 $p(seq_1|seq_2)p(seq_2)=p(seq_1,seq_2)\neq p(seq_1)p(seq_2)$,优化目标为最大化总体回报的期望。可见该种问题相比传统序列决策问题的特点在于:$Reward$ 与两个序列相关,而且两个序列本身也是相关的。传统的网络结构可以解决单序列决策问题,但是无法解决两个相关序列的决策问题。

本文将车货匹配处理为上述以 seq2seq 为模型[29]所衍生出的 CDSDP。传统的 seq2seq 模型无法解决输出序列的词汇表会随着输入序列长度的改变而改变的问题,如寻找凸包等,其输出往往是输入集合的子集。指针网络可以解决该类单序列决策问题,但不适用于两个分属不同问题但相关的 CDSDP。于是,本文使用了 Actor-Critic[27]作为模型的训练框架,设计了 Pointer Network[28]的双网络结构以适应两个相关序列的优化问题,提出了双重指针网络。对比传统算法,本文算法的泛化性、速度和匹配率更优。基于双重指针网络,可以有效改善车货匹配中存在的缺乏智能化供需匹配体系及匹配效率不高等问题,在促进车货匹配领域智能化、标准化等方面具有重要意义。

本文第 2 节描述了车货匹配的问题并建模;第 3 节介绍了算法架构,提出双重指针网络;第 4 节进行数据预处理和网络设计;第 5 节进行实验并对实验结果进行分析;最后总结全文。

2 车货组合优化研究现状

车货供需匹配作为组合优化的一种特殊问题,需要结合路径规划、装箱等决策问题的研究,近几年于国内形成一个新的研究领域。本文以国内车货匹配研究方法与国外组合优化研究方法进行辩证分析。

在早期研究的 TSP 与 VRP 问题中,模拟退火算法(Simulate Anneal Arithmetic, SAA)[12]、粒子群算法(Particle Swarm Optimization,PSO)[13]、遗传算法(Genetic Algorithm,GA)[14]等是常见的启发式算法;而背包问题与装箱问题以贪心算法[15]求解居多。

Bello 等[16]首次尝试将梯度策略算法应用于组合优化算法,并使用神经网络和强化学习同时求解 TSP 问题与背包问题,使用循环长度的负值为奖励信号,采用神经网络的策略梯度法对循环参数进行优化,取得了满意的结果。如今,深度强化学习[36](Deep Reinforcement Learning,DRL)在组合优化领域被广泛研究与应用。

国内早期研究车货匹配问题中,以评价策略方法研究为主。Guo[17]使用三角模糊数建立车源方与货源方相互的多指标语言评价体系,提出基于车源方与货源方整体相互满意度最高的模糊群决策方法;Li[18]为配载型物流信息服务平台的车货供需匹配模块建立以货源方、车源方为主的车货两层筛选匹配指标体系;Xiong[19]针对车货匹配问题中匹配指标的权重问题,采用“反馈式竞争法”的权重改进方法,实现权重的帕累托最优改进;在匹配排序的方法中,Wu[20]采用模糊综合评价法构建多属性决策理论、双边匹配理论,构建一对多供需匹配排序模型。

在车货匹配问题研究方面,现阶段涌现了建立带约束条件的数学模型并采取各类算法对其目标函数求解的方法。如 Hu 等[21]采用禁忌搜索算法,以求解匹配成本最小的目标函数;Zhang[22]采用量子进化算法实现满载率和托运期望最大;Mou 等[23]将目标函数设为匹配率最大、匹配成本最小,使用了改进量子进化算法、有约束惩罚的适应度衰减方法求解;Zhao[24]采用了遗传算法进行求解;Yu 等[25]基于改进 Balance 算法求解最大匹配收益问题。此类算法在小规模的情况下具有效率高、匹配结果良好的优点,但随着问题规模增加,算法性能会出现下降。

与早期启发算法相比,Bello 等[16]采用的深度强化学习方法充分利用了历史样本。当任何优化算法都需要寻找一个下降或者以一定概率下降的方向时,遗传、粒子群和量子进化等算法通过与最优解随机杂交得到高概率下降方向。该类启发式算法只利用了当前最优样本来更新,而强化学习基于评价函数得到下降方向,由于评价函数是基于所有已有样本的,因此强化学习对信息的利用更充分,并且对存在样本饥荒现象的问题有着更优结果[26]。此外,粒子群算法等启发式算法

对不同规模的问题存在“瓶颈问题”，处理更大规模的问题时，对算法优化效果并不明显，且需要耗费更长的时间与资源。

综上所述，现有文献以评价策略体系、启发式搜索算法、深度学习、强化学习等方法对车货供需匹配问题进行研究。其中建立评价策略体系通过构建匹配评价体系或者匹配排序模型两种途径实现，其指标需要人为设定，存在的主观性会较大程度地影响解的质量。传统启发式搜索算法在小规模的车货匹配问题上具有很好的效果，但是随着车货源信息增多，问题规模变大，算法的性能与解的质量会逐渐下降，难以求解现实情况的大规模问题。同时，由于启发式算法的特点，其在处理序列决策问题时易陷入局部最优，需要进行大量的改进。如今，在大规模的车货供需匹配问题上，深度学习与强化学习的结合可以更好地应对信息爆炸时代的庞大数据量，更适应车货供需平台。本文提出双重指针网络并采用深度强化学习的方法求解大规模情况下的车货供需匹配问题，研究成果不仅可对车货匹配问题的现有理论进行补充，而且可为企业实际运营决策提供有价值的指导建议。

3　问题描述与建模

本节考虑了一种包含车辆容积、载重以及时效性的车货匹配场景，分别以货物或者车辆的 6 种属性构建向量序列，并针对该场景进行建模，将车货匹配问题转化为解决马尔可夫决策过程的问题，为下文采用强化学习求解车货供需匹配问题做铺垫。

3.1　问题描述

本文的车货匹配场景为城市与城市之间的物流，关注一个城市内部的车货匹配。设一个边长为 1 000 km 的正方形区域内有一定数量的城市，每个城市采用数字编号构成二维坐标。某一城市有一定数量的货物和车辆。该城市不同的货物有不同的目的城市，并且具有时间限制，如果超出时限送达则需要赔付违约金。本文的车货匹配场景是针对一个城市中的货物和车辆进行匹配。每个货物最多由一辆车服务，也可以没有车辆为其提供服务，一辆车可以服务多个货物。城市中有一定数量的车辆，这些车辆分属不同的车型，不同的车型具有不同的载重和容积。

定义 1　将货物属性设定为六维向量$[m_c, v_c, p_c, d_c, f_c, c_c]$，对应单位为(t，$m^3$，元，/，h，元)。其中，$m_c$为货物质量，$v_c$为货物体积，$p_c$为货物运费，$d_c$为目城市编号，$f_c$为最晚到达时间，$c_c$为违约金。

定义 2　将车辆属性设定为六维向量$[m_t, v_t, p_t, s_t, a_t, b_t]$，对应单位为(t，$m^3$，元/千米，km/h，t，$m^3$)。其中，$m_t$为车辆载重上限，$v_t$为车辆容积上限，$p_t$为车辆每千米单价，$s_t$为车辆平均速度，$a_t$为车辆载重的使用情况，$b_t$为车辆容积的使用情况，而且 $a_t \leqslant m_t$，$b_t \leqslant v_t$。

货物属性中的目的地编号为所有城市编号中的一个。货物的最晚到达时间以车辆出发为 0 点计算，若一个货物的最晚到达时间为 4 h，意味着车辆需要在发车后的 4 h 之内到达货物的目的城市，如果没有在 4 h 内到达，则需要赔付违约金。车辆属性中的载重和容积用量用于记录信息，为下文车货匹配模型的构建服务。

一个车辆可以装载多个货物，可以有多个目的地，车辆依照各个城市的到达时限的轻重缓急依次访问多个目的地，对于每一辆车的每一个目的地存在一个由其装载的货物所确定的访问时限。另外，完成配送后，车辆不返回出发城市。

3.2　问题建模

已知每个订单的始发地、目的地、重量、方位、发出时间和到达时间(可以接受的最晚到达时间)。

已知每个车辆的车型，可跑范围或线路，价格(按整条线路，或每公里费用，或线路＋里程综合报价)，运行时速。已知订单延误时会产生违约金(罚款)，违约金将导致总成本的提高。

综上，设货物数量为 n，车辆数量为 e，则车辆序列 $Tseq=[truck_1, truck_2, \cdots, truck_e]$，其中 $truck_i$ 为定义 2 中所述车辆的属性向量；货物序列 $Cseq=[cargo_1, cargo_2, \cdots, cargo_n]$，其中 $cargo_i$ 为定义 1 中所述货物的属性向量。设有判别变量 d_{ij} 和 o_i，两变量定义如下：

$$d_{ij}=\begin{cases}1, & \text{第 } j \text{ 号货物分配给第 } i \text{ 号车}\\ 0, & \text{其他}\end{cases} \tag{8}$$

$$o_i=\begin{cases}1, & \text{第 } i \text{ 个货物超出时限}\\ 0, & \text{第 } i \text{ 个货物未超时限}\end{cases} \tag{9}$$

由 d_{ij} 组成的 e 行 n 列的矩阵为车货匹配矩阵 $\boldsymbol{D}$；由 o_i 组成的 n 维向量为超时判别向量，记为列向量 $\boldsymbol{o}$。设第 i 个货物超时的违约金为 w_i，由 w_i 组成的 n 维向量，为违约金列向量记为 $\boldsymbol{w}$，第 i 辆车的单价为 pi 元/千米，第 i 辆车需要行驶的距离为 li，车辆单价和车辆行驶距离分别组成维度为 e 的列向量 $\boldsymbol{l}$ 和 $\boldsymbol{c}$。$\vec{1}$为一个维度与相乘的向量或者矩阵相容的元素全为 1 的列向量，$\boldsymbol{p}$ 为一个由货物运费组成的 n 维列向量，其中第 i 个元素为第 i 个货物的运费。类似地，可以定义货物质量列向量 $\boldsymbol{m}$ 和货物体积列向量 $\boldsymbol{v}$，车辆载重列向量 $\boldsymbol{m}'$和车辆容积列向量 $\boldsymbol{v}'$。

则该场景的优化问题可以公式化描述为如下优化问题：

$$\max\ \vec{1}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{p}-\boldsymbol{l}^{\mathrm{T}}\boldsymbol{c}-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{o} \tag{10}$$

$$\text{s.t. } \vec{1}^{\mathrm{T}}\boldsymbol{D}\vec{1}\leqslant n \tag{11}$$

$$\boldsymbol{D}\boldsymbol{m}\leqslant\boldsymbol{m}' \tag{12}$$

$$\boldsymbol{D}\boldsymbol{v}\leqslant\boldsymbol{v}' \tag{13}$$

式(1)是优化目标函数，其代表了利润，第一项为所有匹配货物的运费收入，第二项为车辆的成本，第三项为需要赔付的违约金；式(2)为实现约束匹配结果中每一个货物最多可以由一辆车服务，一辆车可以服务多个货物；式(3)的含义为车辆不超载；式(4)的含义为给车辆匹配到的货物不超过车辆的容积。

货物与车辆均有一种排序方法，本文所提出的方法是寻找一种车辆和货物的序列使得最终得到的利润最大。本文的方法首先将车辆序列与货物序列映射为一种车货匹配结果，然后将该匹配结果映射为利润，最终的优化目标为利润最大化。将货物序列映射为匹配结果的实现方法见 4.1.3 节。

车辆路线由车辆所装载的货物所决定，若车辆上装载了 A，B，C 3 个目的地的货物，其中时间限制最紧急货物的时间窗分别为 9 h，6 h，8 h，则车辆的路径为 $B-C-A$，然后依照该路径，可以计算出各个城市的到达时间，即可以统计超时的货物。经过上述序列到匹配结果的转化，本节最开始提出的

优化目标转变成了 max $R(\tau_c,\tau_t)$，约束条件保持不变，其中 τ_c 和 τ_t 分别为货物序列与车辆序列，$R(\tau_c,\tau_t)$ 为该种序列情况下的利润，即式(6)。

4 算法设计

本节分为两个部分：算法概述与双重指针网络。算法概述部分说明算法的框架与一些外围工作。双重指针网络部分介绍算法核心：双重指针网络的架构。

本算法主要包括预处理、特征 Embedding、双重指针网络及其训练方法，以及一种将序列映射为匹配结果的方法，算法各部分的流程如图 1 所示。

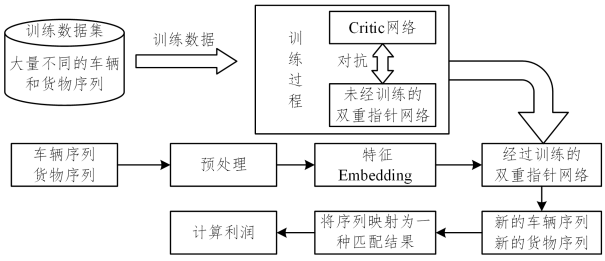

图 1 算法流程

Fig. 1 Algorithm flow

4.1 预处理

一般的数据标准化和归一化无法反映车货匹配问题中各个属性的关系，并且车辆之间、货物之间存在相互关系，因此需要一些特殊的预处理方法。本小节将介绍适用于本文车货匹配的预处理方法。

4.1.1 归一化

预处理是将货物属性信息转化为神经网络可处理的无单位数据，另外货物属性信息需要添加一些其他属性，车辆属性信息同理但是需要去除一些属性。本文将用于输入神经网络进行训练的货物和车辆的描述方式分别称为货物特征和车辆特征。货物特征和车辆特征分别用一个向量 $\vec{t}$ 和 $\vec{c}$ 表示，$\vec{t}$ 和 $\vec{c}$ 的定义如下：

$$\vec{c}=[m_c,v_c,p_c,x,y,f_c,c_c] \tag{14}$$

$$\vec{t}=[m_t,v_t,p_t,s] \tag{15}$$

其中，m_c,v_c,p_c,x,y,f_c,c_c 分别表示货物的质量(单位：吨)、体积(单位：m^3)、运费(单位：元)、目的坐标 1、目的坐标 2、最晚到达时间(单位：h)、违约金(单位：元)。m_t,v_t,p_t,s 分别表示车辆的载重(单位：吨)、容积(单位：m^3)、单价(单位：元每千米)、车速(单位：km/h)。

为了使得网络可以学习到货物目的地的信息，需要将货物信息中的目的地编号换成目的城市坐标。在将样本输入网络之前需要进行归一化，但是一般的归一化方法不适用本问题。如果使用常用的最大最小归一化，或者标准化，则会破坏货物和车辆之间数据的关联性，破坏其所具有的物理意义。因此本文采用了不同的预处理方法，将属性按照最大值进行缩放，将位置、金钱、载重等缩放为无单位数据，变换方法如下：

(1) $m_c=m_c/\mathrm{Max}(m_t)$

(2) $v_c=v_c/\mathrm{Max}(v_t)$

(3) $p_c=p_c/\mathrm{Max}(p_c)$

(4) $x=x/1\,000$

(5) $y=y/1\,000$

(6) $f_c=f_c/\mathrm{Max}(f_c)$

(7) $c_c=c_c/\mathrm{Max}(p_c)$

(8) $m_t=m_t/\mathrm{Max}(m_t)$

(9) $v_t=v_t/\mathrm{Max}(v_t)$

(10) $p_t=p_t/\mathrm{Max}(p_c)$

(11) $s=s/1\,000$

4.1.2 特征 Embedding

预处理的后一步是特征 Embedding，车辆特征的 Embedding 采用一个全相连实现，货物特征的 Embedding 使用了 Self Attention[30] 和 Multi-Head Attention[30] 机制以及一个全相连层，如图 2 所示。

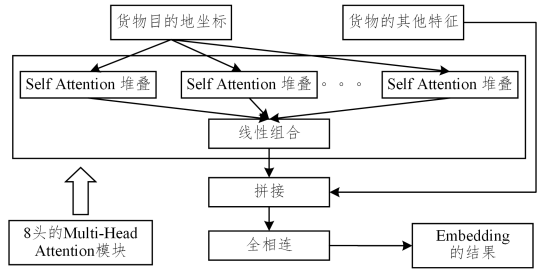

图 2 货物 Embedding 示意图

Fig. 2 Schematic diagram of cargo Embedding

不同货物之间坐标是具有关联性的，其定量反映了货物之间相对的远近关系。因此对于货物的坐标，本文采用了文献[30]中的 Self Attention 和 Multi-Head Attention 机制，用通过这两种机制得到的坐标 embedding 值替换原先的两个坐标值后得到新的货物特征；然后将新的货物特征与车辆特征分别放入两个全相连层，最终得到货物和车辆特征的 embedding 结果。对坐标的 Embedding，首先通过货物的坐标生成多个 Attention 三元组 QKV[30]。Self Attention 的计算过程如图 3 所示。

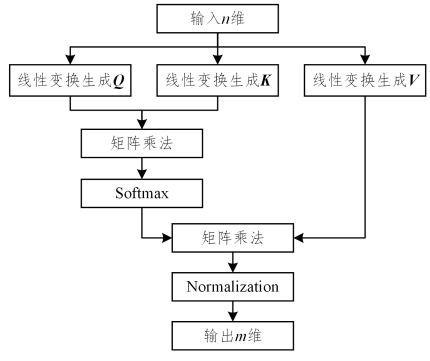

图 3 Self Attention 计算过程

Fig. 3 Self Attention calculation process

Self Attentions 模块首先生成 QKV 三元组，然后进行矩阵乘法和 Softmax，输出 Attention 结果。图 3 在 Normalization 层之前的计算过程公示化描述如下：

$$\boldsymbol{V}=W_3\boldsymbol{C} \tag{16}$$

$$\boldsymbol{K}=W_2\boldsymbol{C} \tag{17}$$

$$\boldsymbol{Q}=W_1\boldsymbol{C} \tag{18}$$

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{m}}\right)\boldsymbol{V} \tag{19}$$

其中，$\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}$ 均为矩阵，通过对货物坐标线性变换得到；m 为

输出维度；C 为货物坐标组成的矩阵；W_1，W_2，W_3 为模型参数。将上述 Attention 结果经过一个 Normalization 层[31]得到最终的输出结果。本文的 Multi-Head Attention 与文献[30]有较大不同，其结构如图 4 所示。

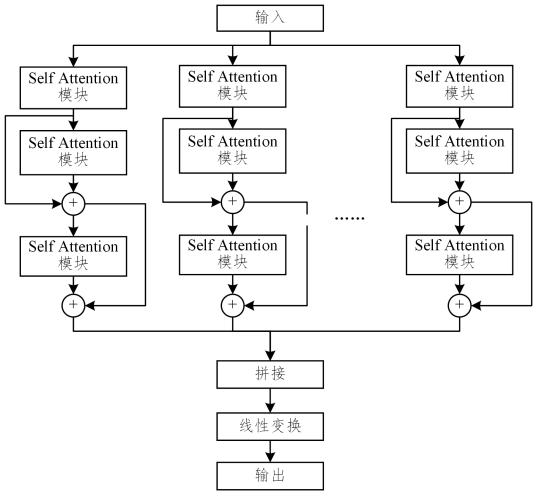

图 4　Multi-Head Attention 计算过程

Fig. 4　Multi-Head Attention calculation process

每一头是多个 Self Attentions 模块的堆叠，中间采用残差连接[32]的方式。设 A_i 为第 i 个 Attention 头的计算结果，图 4 体现了该过程，其中拼接和线性变换部分的计算公式化表述如下：

$$Output = Concat(A_1, A_2, \cdots, A_n)W \tag{20}$$

4.1.3　序列映射与匹配结果的映射

将序列转换为匹配结果主要是采用贪心的思想将两个序列转换为匹配结果，实现方法如算法 1 所示。

算法 1　序列映射算法

输入：车辆序列、货物序列

输出：匹配矩阵 **D**

1. 初始化一个 n×e 的全零矩阵 **D**；
2. 遍历货物序列 $Cseq=[cargo_1, cargo_2, \cdots, cargo_n]$，取出 Cseq 中的元素 $cargo_i$，其中 $1 \leqslant i \leqslant n$；
3. 遍历车辆序列 $Tseq=[truck_1, truck_2, \cdots, truck_e]$，取出 Tseq 中的元素 $truck_j$，其中 $1 \leqslant j \leqslant n$；
4. 提取元素 $cargo_i$ 的六维向量 $[m_{ci}, v_{ci}, p_{ci}, d_{ci}, f_{ci}, c_{ci}]$ 中的 (m_{ci}, v_{ci}) 属性；提取元素 $truck_j$ 六维向量 $[m_{tj}, v_{tj}, p_{tj}, s_{tj}, a_{tj}, b_{tj}]$ 中的 $(m_{tj}, v_{tj}, a_{tj}, b_{tj})$ 属性；
5. 判断 m_{ci} 是否小于 $m_{tj}-a_{tj}$ 且 v_{ci} 是否小于 $v_{tj}-b_{tj}$，如符合要求，则 $D_{ij}=1$ 并跳转步骤 2，如果不能分配给所述车辆元素 $truck_i$ 则跳转步骤 3。

该算法首先遍历货物序列，为当前遍历到的货物在车辆序列中依次寻找可以装载的车辆。当寻找到第一辆可以装载的车辆时停止，最终形成一个匹配结果。

4.2　双重指针网络

第 2 节中将车货匹配问题映射为针对两个分属不同问题但相关的序列决策问题，本节中提出双重指针网络解决上述问题，并在车货匹配的应用中证明了该网络的有效性。双重指针网络是一个使用了两个序列生成模块的网络结构，针对车货匹配问题，第一个序列生成模块生成车辆序列，第二个序列生成模块生成货物序列，流程如图 5 所示。

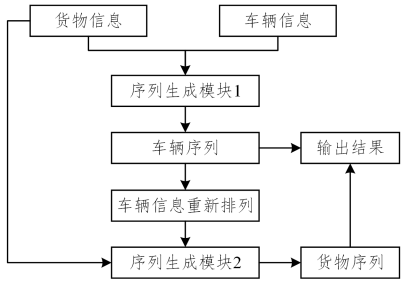

图 5　双重指针网络流程图

Fig. 5　Flowchart of dual-pointer-network

双重指针网络，输入两个序列的信息，在车货匹配中分别对应车辆信息和货物信息。输出两个序列，在车货匹配问题中分别为车辆序列和货物序列。

图 5 中的序列生成模块输入两个序列的信息，输出一个序列。该模块的作用是整合两个原始序列的信息，规划新的序列，从而获取最大回报。序列生成模块的数据流如图 6 所示。

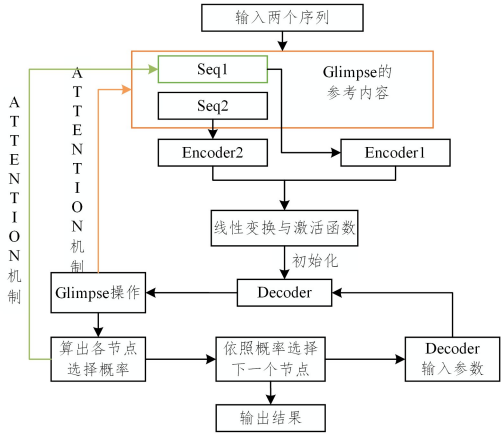

图 6　序列生成模块数据流

Fig. 6　Data flow of sequence generation module

序列生成模块的输入为两个序列 Seq1 与 Seq2。该模块会根据两个序列的信息生成 Seq1 的重排序结果。该网络中采用 LSTM[33]作为 Encoder 和 Decoder 单元，通过不断迭代 Decoder 解码最终得到一个 Seq1 的重排序结果。序列生成模块主要通过两个操作聚合两个序列的信息，第一个操作采用两个序列 Encoder 的末态输出作为 Decoder 的初始化[34]。第二个操作采用了 Glimpse 操作[16]，该操作参考了两个序列的输入。首先通过 Glimpse 操作对 Decoder 的输出向量变换整合两个序列的信息，然后使用指针网络的 Attention 机制对下一个节点的选择概率进行计算。Glimpse 操作的公式化表述如下：

$$Glimpse(\boldsymbol{q}, \boldsymbol{R}, \boldsymbol{v}) = (W_1\boldsymbol{R})softmax[\boldsymbol{v}^{\mathrm{T}} \tanh(W_1\boldsymbol{Q} + W_2\boldsymbol{R})] \tag{21}$$

在车货匹配中，$\boldsymbol{R}$ 为参考矩阵 $\boldsymbol{R}_{.g}=[c_1, c_2, \cdots, c_n, t_1, t_2, \cdots, t_e]$，该矩阵为车辆序列与货物序列的连接，$n$ 为货物数量，e 为车辆数量，$c_1, c_2, \cdots, c_n$ 为货物预处理后得到的向量，$t_1, t_2, \cdots, t_e$ 为车辆预处理后得到的向量。其中，$\boldsymbol{Q}$ 为查询向量 q（Decoder 的输出）复制组成的矩阵 $\boldsymbol{Q}_{.(n+e)}=[\boldsymbol{q}, \boldsymbol{q}, \cdots, \boldsymbol{q}]$，列数为车辆数量与货物数量的和，此处 $\boldsymbol{q}$ 为 Decoder 的输出。W_1，W_2 和向量 $\boldsymbol{v}$ 为参与梯度下降的模型参数。本发明采用

的 Pointer Network[28]的 Attention 机制实现了节点选择。计算选择节点概率的方法如下：

$$Pointer(\boldsymbol{q},\boldsymbol{R},\boldsymbol{v})=softmax[\boldsymbol{v}^{\mathrm{T}}\tanh(W_1\boldsymbol{Q}+W_2\boldsymbol{R})] \quad (22)$$

其中，$\boldsymbol{R}$ 为参考矩阵 $\boldsymbol{R}_{\cdot p}=[c_1,c_2,\cdots,c_n]$ 货物序列构成的矩阵，n 为货物数量，$c_1,c_2,\cdots,c_n$ 为货物预处理后得到的向量。上式中的矩阵 $\boldsymbol{Q}$ 同 Glimpse 操作中的 $\boldsymbol{Q}$。W_1,W_2 及向量 $\boldsymbol{v}$ 为参与梯度下降的模型参数。

4.3 训练方法

本文采用 Actor-Critic 框架[27]进行训练。Actor-Critic 框架是策略梯度法[35]的一种改进形式，使用 Critic 网络与双重指针网络进行对抗，将其作为基线指导双重指针网络的训练。Critic 实现了对问题求解的预测，由于对数据的预处理存在缩放，Critic 网络输出的结果为货物运费最大值的倍率。设 Critic 网络输出的结果为“4”，即期望的回报为货物运费最大值的 4 倍。Critic 网络结构与 Actor 网络结构类似，均为先进行 Embedding 操作再通过 Encoder-Decoder 框架实现。Critic 网络中的数据流如图 7 所示。

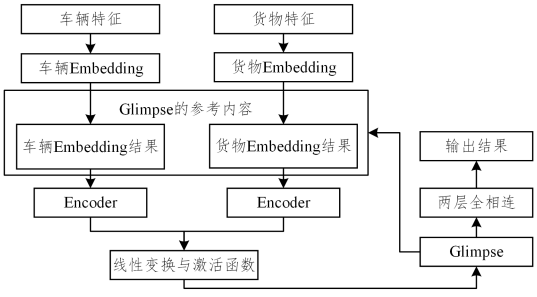

图 7 Critic 网络数据流

Fig. 7 Data flow of Critic network

Critic 网络首先对货物和车辆信息进行 Embedding，接着使用 Encoder 将其转换为查询向量 $\boldsymbol{q}$，然后使用 Glimpse 操作整合两个序列的信息，并最终使用全相连输出结果。本文的训练使用的 loss 函数如下：

$$loss=-E[R(\tau_c,\tau_t)-b] \quad (23)$$

其中，E 为求期望，$R(\tau_c,\tau_t)$ 为该种序列情况下的利润。上式中的 b 为一个基线，由 Critic 网络提供。由于对实验数据进行了一定程度上的缩放，Critic 网络的输出结果需要乘以一定的倍率才能作为上式中的 b。若在某种 Actor 网络参数集合 θ 的情况下出现 τ_c,τ_t 的概率为 $P(\tau_c,\tau_t|\theta)$，则 loss 函数的梯度如下：

$$\nabla_\theta loss=E[\nabla_\theta(\log P(\tau_c|\tau_t,\theta)+\log P(\tau_t|\theta))(R(\tau_c,\tau_t)-b)] \quad (24)$$

双重指针网络首先计算得到车辆序列 τ_t，在此基础上计算 τ_c，所以使用两个序列概率的对数相加来计算梯度，并反向传播。

4.4 时间复杂度分析

本文所提出的适用于车货匹配预处理的方法主要使用 Multi-Head Attention 机制和全相连层实现，即预处理部分的时间复杂度应该主要考虑 Attention 机制的时间复杂度。

如图 3 以及式(16)－式(19)所示，一层 self attention 的主要时间复杂度为 4 次矩阵运算。设有维度为 i_s 的向量，有 n_s 个输入 self attention 模块向量，输出向量的维度为 o_s，该模块的时间复杂度记为 f_s，则：

$$f_s=O(i_s n_s o_s+o_s n_s^2) \quad (25)$$

给定一个模型，则 o_s,i_s 为常数，代入上式得：

$$f_s=O(n_s^2) \quad (26)$$

如图 4 所示，Multi-Head Attention 机制为 self atteion 机制的堆叠，设堆叠层数为 s_m，attention 头数为 c_m，给定一个模型后 s_m 和 c_m 为常数，记 Multi-Head Attention 的时间复杂度为 f_m，则：

$$f_m=O(c_m(f_s+s_m f_s)+c_m i_s n_s o_s) \quad (27)$$

$$f_m=f_s=O(n_s^2) \quad (28)$$

根据 3.2 节中的记号，货物数量为 n，货物的坐标采用了 Multi-Head Attention，即 $n_s=n$，得到货物的预处理时间复杂度为：

$$f_{cp}=O(n^2) \quad (29)$$

根据式(21)，设 Glimpse 操作的输入向量 $\boldsymbol{q}$ 的维度为 i_g，矩阵 $\boldsymbol{R}$ 维度为 $n_{gr}\times d_{gr}$，$n_{gr}=n+e$，其中 n 为货物数量，e 为车辆数量，d_{gr} 为车辆和货物 Embedding 后的维度，记 Glimpse 操作的时间复杂度为 f_g，当模型确定后 d_{gr} 和 i_g 为常数，则：

$$f_g=O(n_{gr}i_g^2+n_{gr}d_{gr}i_g) \quad (30)$$

$$f_g=O((n+e)i_g^2+(n+e)d_{gr}i_g) \quad (31)$$

$$f_g=O(n+e) \quad (32)$$

设序列生成模块中的计算选择下一个节点概率的时间复杂度为 f_p，该步骤的计算式(22)与 Glimpse 操作类似，同理可得 $f_p=O(n_i)$，其中 n_i 为输入计算下一个节点选择概率模块的元素个数。综合上述结论，若图 6 中 Seq1 的长度为 e，Seq2 长度为 n，记这样的一个序列生成模块的时间复杂度为 f_{seq}，则：

$$f_{\mathrm{seq}}=O(e\,f_g)=O(ne+e^2) \quad (33)$$

双重指针网络使用了两个序列生成模块，第一个序列生成模块的 Seq1 为车辆，Seq2 为货物，第二个序列生成模块相反，记双重指针网络的时间复杂度为 f_{double}，则综上所述：

$$f_{\mathrm{double}}=O(n^2+2ne+e^2)=O((n+e)^2) \quad (34)$$

根据 4.1.3 节的序列到匹配结果的映射算法描述可知该算法的时间复杂度 $f_{\mathrm{match}}=O(ne)$。假设计算利润的算法时间复杂度为 $f_r=O(f(n,e,z))$，其中 z 为城市个数，f 为一个关于车辆数量、货物数量、城市个数的函数。设遗传算法的种群数量为 p_{size}，迭代次数为 $loop$。遗传算法每得到一个解，需要计算其适应度，假设遗传算法的交叉和变异时间复杂度为 $O(1)$，记遗传算法的时间复杂度为 f_{ga}，则：

$$f_{\mathrm{ga}}=O(p_{\mathrm{size}}*loop(f_{\mathrm{match}}+f_r)) \quad (35)$$

$$f_{\mathrm{ga}}=O(p_{\mathrm{size}}*loop(ne+f_r)) \quad (36)$$

蚁群算法、模拟退火算法与遗传算法类似，存在两个循环，每一个循环体内部均需要运行序列到匹配结果的映射并且需要计算利润，从而指导下一步的选择。因此，蚁群算法和模拟退火算法的时间复杂度与遗传算法的时间复杂度相同。

根据上述分析，传统的蚁群算法、模拟退火、遗传算法这样的启发式算法，在每一次迭代和局部搜索时均需要计算一次目标函数(在车货匹配问题中为利润)，存在大量的目标函数计算过程，而双重指针网络不是基于随机搜索的，因此不需要大量的计算目标函数，可以大大降低算法的时间复杂度。此外，当问题规模逐渐扩大时，原先解决小规模所使用的启发式算法参数需要调整，才能得到一个较优的解，例如遗传算法需要更多的种群数量和迭代次数。这就导致了当面临超大问题规模的时候，传统启发式算法的时间复杂度将远远大于

f_{ga}。而双重指针网络则不需要调整模型参数，这大大减小了在处理大规模问题情况下的时间复杂度。

5　实验设计及结果分析

本节将采用第 4 节所述算法对第 3 节中的车货匹配模型进行求解实验。第 3 节中的车货匹配场景是针对一个城市中的货物和车辆进行匹配，进行车货匹配的城市编号设为 0 号，所有货物和车辆的始发地均为 0 号城市。

本实验的硬件环境：CPU 使用 Ryzen 7 5800H，内存 32 GB，GPU 使用 RTX 3060 Laptop GPU。软件环境为 Python 3.8.9；使用了 CUDA 11.1 与 cudnn 8.2.1 作为 GPU 计算环境。使用了 numpy 1.19.5 与 numba 0.53.1 作为 Python 的加速库，神经网络库使用了 pytorch 1.9.0。

实验中首先采用 5 个城市、250 个货物和 50 辆车的问题规模训练双重指针网络。然后在该问题规模下，保持 5 个城市不变重新生成 100 个车货匹配问题，即重新生成 100 组相同数量的车辆与货物。在多个指标上比较双重指针网络得到的结果与传统启发式算法得到的结果。然后逐步扩大问题规模，提升货物数量与车辆数量，观察模型的泛化能力、求解的质量以及求解耗时。基于上述实验流程，首先编写了用于实验数据生成的源码来生成实验数据，详细参考 5.1 节；然后编写了用于对比的遗传算法、模拟退火算法以及最新的自适应蚁群算法[37]，基于上述实验的软硬件环境运行 3 种算法并与双重指针网络的求解结果进行对比。

5.1　实验数据生成

本文的车货匹配场景是针对一个城市中的货物和车辆进行匹配，进行车货匹配的城市编号设为 0 号，所有货物和车辆的始发地均为 0 号城市。实验数据生成的流程如图 8 所示。

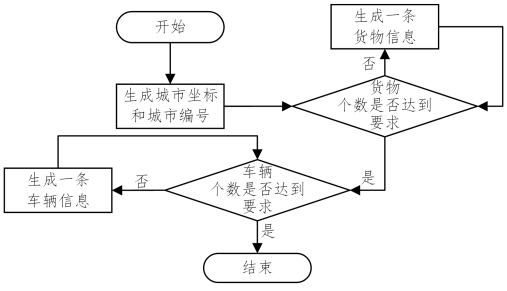

图 8　实验数据生成流程图

Fig. 8　Flow chart of experimental data generation

流程的第二步即生成城市坐标以及编号，依照 0 中的场景事先指定区域边长为 1 000 km 和城市个数 n，然后随机在正方形中选取 n 个点作为城市。上述流程中的第 1 个和第 2 个判断中货物数量和车辆数量需要提前指定。该流程生成了一条实验数据，在生成货物和车辆数据时有一些参数需要指定，参数如表 1 所列。

表 1　数据生成参数

Table 1　Data generation parameters

序号	参数名	单位
1	平均货物质量	吨
2	平均货物体积	m^3
3	平均货物运费	元
4	平均车辆单价	元每千米
5	平均车速	km/h

根据 3.2 节的建模与问题场景中规定的车辆属性和货物属性，一条货物信息的生成方法如算法 2 所示。

算法 2　货物信息生成方法

输入：平均货物质量、平均货物体积、均货物运费、平均车速

输出：货物信息

1. 货物质量＝random(0，2 * 平均货物质量)
2. 货物体积＝random(0，2 * 平均货物体积)
3. 目的地编号＝choice(1，2，3，…，n)
4. distance＝欧氏距离(0 号城市坐标，目的地坐标)
5. baseTime＝distance/平均车速
6. 最晚到达时间＝random(baseTime，2 * baseTime)
7. basePrice＝distance * 平均货物运费
8. 运费＝random(0.75 * basePrice，2.25 * basePrice)

一条车辆信息的生成方法如算法 3 所示。

算法 3　车辆信息生成方法

输入：平均车辆单价、平均车速

输出：车辆信息

1. 车型编号＝choice(1，2，3，4)
2. 载重＝车型编号对应的车辆载重
3. 容积＝车型编号对应的车辆容积
4. 单价＝random(0.5 * 平均车辆单价，2.5 * 平均车辆单价)
5. 车速＝random(0.75 * 平均车速，2.25 * 平均车速)
6. 载重用量，容积用量＝0，0

上述算法中 *random* 为随机取数，例如 *random*(a，b)的含义为从 a 到 b 之间按照均匀分布随机取数；*choice* 为从给定的候选项中随机选取。车辆信息生成方法中首先选取车型编号，然后整合该型号车辆的容积与载重。

上述两个算法实现了图 8 中的“生成一条货物信息”和“生成一条车辆信息”的工作，当然在实际强化学习训练和测试中不可能只生成一条实验数据，可以循环多次执行图 8 的步骤，生成多条实验数据也就是多个问题样本。

5.2　求解能力实验

采用 5 个城市、250 个货物和 50 辆车的问题规模训练双重指针网络，再以相同的问题规模生成 100 个问题，作为评估集用于训练过程中对模型进行评估。然后以相同的问题规模生成 60 个问题作为测试集，测试双重指针网络的性能以及启发式算法的求解性能。

图 9 中的利润、收入、违约金和货车成本为测试集中 100 个问题求解结果的平均值。由图可以看出双重指针网络在求解 5 个城市、250 个货物和 50 辆车的问题规模时与传统算法性能相差无几。3 种方法求解的时间消耗如图 10 所示。可以看到在该问题规模下，4 种算法的寻优能力相当，但是双重指针网络的求解耗时明显小于启发式算法。如 4.4 节所述，由于无需计算车货匹配的利润(目标函数)，所以双重指针网络相比启发式算法求解耗时大大减少。

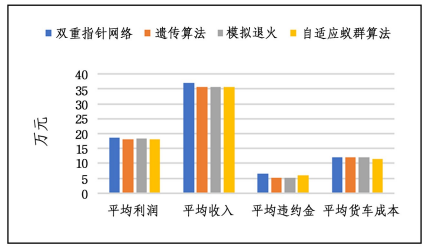

图 9　不同算法寻优能力的对比

Fig. 9　Comparison of optimization capabilities of different algorithms

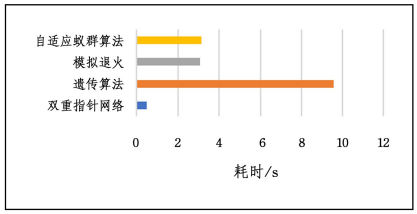

图 10 不同算法平均耗时的对比

Fig. 10 Comparison of the average time-consuming of different algorithms

5.3 模型泛化实验

该部分将扩大问题的规模，保持城市数量为 5 个不变，以 50 为步长增加货物数量，以 10 为步长增加车辆数量。为了实验结果的一致性，保持遗传算法种群数量 100、繁殖代数 100 不变，保持模拟退火算法初温 10 0000 度、迭代次数 1 000 次不变。在 250 辆车和 50 件货物的问题规模时，两种启发式算法在该参数情况下与双重指针网络求解得到的利润相近。

保持启发式算法使用相同的参数，并且 3 种算法求解相同问题的情况下，在不同问题规模时求解得到的利润如图 11 所示。

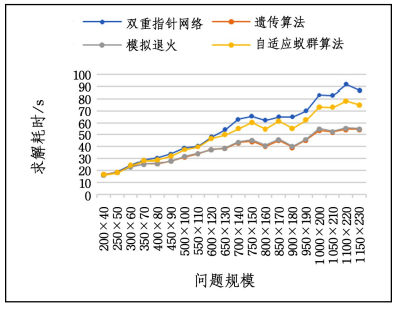

图 11 不同问题规模的利润对比

Fig. 11 Profit comparison of different problem scales

图 11 中横坐标为问题规模，“×”号前为货物数量，“×”号后为货车数量。“250×50”的含义为该问题待匹配的车辆有 50 辆，货物有 250 件。泛化性能实验中两个启发式算法的参数是经过精心设计的，其求解的结果与双重指针网络求解质量相似。从图 11 中可以看到，在最小的问题规模时求解质量相似。随着问题规模逐渐变大，自适应的蚁群算法[38]明显好于传统的遗传算法和模拟退火算法，但还是劣于双重指针网络在 4.4 节中的结论，当问题规模逐渐扩大时，遗传算法等启发式算法需要更大的规模才能得到较优的结果。在同样情况下，3 种方法的求解耗时对比如图 12 所示。

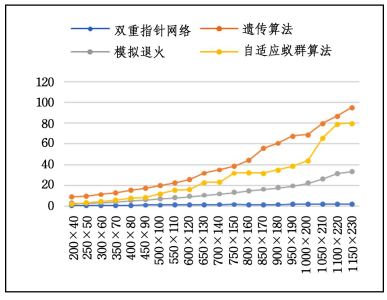

图 12 不同问题规模的耗时对比

Fig. 12 Time-consuming comparison of different problem scales

由图 12 可以看出，在问题规模较小时求解耗时均较小，但是随着问题规模的扩大，新老启发式算法的求解耗时迅速上升，但是使用双重指针网络的求解耗时仍然不超过 1 s。由于无需计算车货匹配利润，双重指针网络的求解速度明显快于遗传算法等启发式算法。根据 4.4 节中对 f_{ga} 的分析可得，如果增大启发式算法的规模，如增加蚂蚁数量或者增加迭代次数，时间消耗将大大增加。

5.4 模型收敛性实验

上文中使用一个训练集对双重指针网络进行优化训练，使用一个评估集评估模型的性能，在训练过程中双重指针网络对评估集的求解得到的平均利润如图 13 所示。

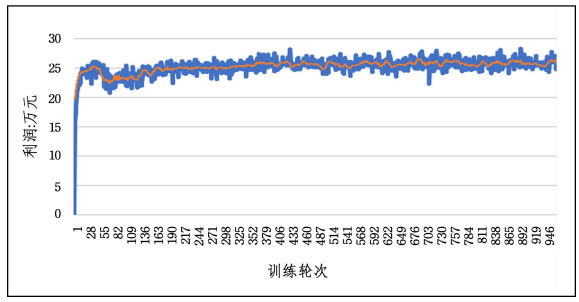

图 13 双重指针网络不同轮次的训练利润

Fig. 13 Different rounds of training profit of dual-pointer-network

在训练前期，评估集求解的质量迅速上升，在 32 轮后经过一系列震荡，随着训练逐渐趋于平稳，大约在 300 轮的时候基本达到稳定。可见模型的训练速度是比较快的。

结束语 本文针对车货供需匹配问题，确立目标函数及其约束建立了数学建模。通过贪心的算法思想将车货匹配求解问题模型映射为序列的匹配策略，并采用双重指针网络求解。这是一种高效的，具有较强泛化性的方法。与启发式算法在大规模问题下的“瓶颈”状态不同，可以在问题规模较大的情况下迅速求解车货匹配问题，并得到一个相对较优的解。

双重指针网络可以作为处理车货匹配问题较为高效的创新性的算法，其求解问题范围可以推广到一切序列决策问题。通过整合两个原始序列的信息，规划新的序列，可以在问题规模较大时求得较高质量的 reward。同时，双重指针网络算法作为一种基础算法，具有丰富的改进前景，我们的下一步计划是从机制入手，通过尝试改进求解更为常规的 CVRP(带约束的车辆路径问题)，并与现如今主流处理 CVRP 问题的算法进行对比，以此验证其推广性。

参 考 文 献

[1] WANG C, NI Y, YANG X. The Production Routing Problem Under Uncertain Environment[J]. IEEE Access, 2021, PP(99): 1-1.

[2] ODILI J B. Combinatorial optimization in science and engineering[J]. Current science, 2017, 113(12): 2268-2274.

[3] MARATHE M V, PERCUS A G, TORNEY D C. Combinatorial Optimization in Biology[Z]. 1999.

[4] CHEN C, LI C. Process Synthesis and Design Problems Based on a Global Particle Swarm Optimization Algorithm[J]. IEEE Access, 2021, PP(99): 1-1.

[5] MELNIKOV, TSYGANOV, BULYCHOV. A Multi-heuristic Algorithmic Skeleton for Hard Combinatorial Optimization Problems[C]// 2nd International Joint Conference on Computational Sciences and Optimization(CSO). 2009.

[6] GREBENNIK I, DUPAS R, URNIAIEVA I, et al. Mathematical

Model of Containers Placement in Rail Terminal Operations Problem[C]// International Conference on Advanced Computer Information Technologies. Dept. of System Engineering, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine; Laboratory IMS, University of Bordeaux, Bordeaux, France; Dept. of System Engineering, Kharkiv National University of Radioelectronics, Kharkiv, Ukraine; Dept. of, 2019.

[7] SAHANA S K, JAIN A. High Performance Ant Colony Optimizer (HPACO) for Travelling Salesman Problem (TSP) [M]. Springer International Publishing, 2014.

[8] KALAKANTI A K, VERMA S, PAUL T, et al. RL SolVeR Pro: Reinforcement Learning for Solving Vehicle Routing Problem[C]// 2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS). 2019.

[9] HIFIM, YOUSSOUF A M, SAADI T, et al. A Cooperative Swarm Optimization-Based Algorithm for the Quadratic Multiple Knapsack Problem[C]// 2020 7th International Conference on Control, Decision and Information Technologies (CoDIT). 2020.

[10] MUNIEN C, MAHABEER S, DZITIRO E, et al. Metaheuristic Approaches for One-Dimensional Bin Packing Problem: A Comparative Performance Study[J]. IEEE Access, 2020, PP(99).

[11] CHEN B H. A variable neighborhood search algorithm with reinforcement learning for a real-life periodic vehicle routing problem with time windows and open routes[J]. RAIRO-Operations Research, 2020, 54(5): 1467-1494.

[12] HAO X. Optimization Models and Heuristic Method Based on Simulated Annealing Strategy for Traveling Salesman Problem [J]. Applied Mechanics and Materials, 2010, 34-35(4): 1180-1184.

[13] OUYANG A J, ZHOU Y Q. An improved PSO-ACO algorithm for solving large-scale TSP[J]. Advanced Materials Research, 2011, 143-144: 1154-1158.

[14] BAI D L, GUO Q P. An Optimized Genetic Algorithm for TSP [C]// Proceedings of 2008 International Symposium on Distributed Computing and Applications for Business Engineering and Science. 2008.

[15] KAYSTHA S, AGARWAL S. Greedy genetic algorithm to Bounded Knapsack Problem [C] // IEEE International Conference on Computer Science & Information Technology. IEEE, 2010: 301-305.

[16] BELLO I, PHAM H, LE Q V, et al. Neural combinatorial optimization with reinforcement learning [J]. arXiv: 1611. 09940, 2016.

[17] GUO J N. Vehicle-Cargo Matching Using a Fuzzy Group Decision-Making Approach[J]. Journal of Transportation Engineering and Information, 2017, 15(4): 141-146.

[18] LI H. Research on Supply and Demand Matching of Vehiclesand Cargos Based on Stowage Logistics Information Service Platform [D]. Beijing: Beijing Jiaotong University, 2015.

[19] XIONG Y Q. LogisticsPublic Information Platform Freight Forwarding Matching and Credibility Motivation Mechanism[D]. Beijing: Tsinghua University, 2015.

[20] WU G S. Research on Matching Model of Vehicle Cargo Considering Trading Party's Preference[D]. Nanjing: Nanjing University, 2017.

[21] HU J L, BING C, HAN S G. Study on vehicles and goods matching of arterial road freight platform based on TS algorithm [J]. Journal of Zhejiang Sci-Tech University (Social Science Edition), 2018, 40(5): 478-486.

[22] ZHANG Q J. Research on Combination Matching Model Based on Logistics Supply and Demand Information[D]. Xi'an: Xidian University, 2017.

[23] MU X W, CHEN Y, GAO S J, et al. Vehicleand Cargo Matching Method Based on Improved Quantum Evolutionary Algorithm [J]. Chinese Journal of Management Science, 2016, 24(12): 166-176.

[24] ZHAO C Y. Research on Vehicles and Cargos Matching Problem for Virtual Social Reserve Platform in M area[D]. Beijing: Beijing Jiaotong University, 2019.

[25] YU Y S, LIU X Y. Research on Vehicles and Cargos Matching Based on Improved Balance Algorithm [J]. Journal of Wuhan University of Technology, 2016, 38(10): 47-54.

[26] AFFAN M, JAWAID J, AHMED S U, et al. Solving Combinatorial Problems through Off-Policy Reinforcement Learning Methods[C]// 2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE). 2020.

[27] HAARNOJA T, ZHOU A, ABBEEL P, et al. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor [C] // International Conference on Machine Learning. PMLR, 2018: 1861-1870.

[28] VINYALS O, FORTUNATO M, JAITLY N. Pointer networks [J]. arXiv: 1506. 03134, 2015.

[29] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[J]. arXiv: 1409. 3215, 2014.

[30] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv: 1706. 03762, 2017.

[31] LEIBA J, KIROS J R, HINTON G E. Layer normalization[J]. arXiv: 1607. 06450, 2016.

[32] HE K M, ZHANG X Y, REN S Q, et al. . Deep residual learning for image recognition[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[33] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780.

[34] SUN F, JIANG P, SUN H, et al. Multi-source pointer network for product title summarization[C]// Proceedings of the 27th ACM International Conference on Information and Knowledge Management. 2018: 7-16.

[35] SUTTON R S, MCALLESTER D A, SINGH S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]// NIPs. 1999: 1057-1063.

[36] MAZYAVKINA N, SVIRIDOV S, IVANOV S, et al. Reinforcement learning for combinatorial optimization: A survey[J]. arXiv: 2003. 03600, 2020.

[37] LING H, FU Y, HUA M, et al. An Adaptive Parameter Controlled Ant Colony Optimization Approach for Peer-to-Peer Vehicle and Cargo Matching[J]. IEEE Access, 2021, 9: 15764-15777.

CAI Yue, born in 1997, postgraduate. His main research interests include information networks, reinforcement learning, and deep neural networks.

SUN Zhi-xin, born in 1964, doctor, professor, doctoral supervisor. His main research interests include the theory and technology of network communication, computer network and security.