



## 基于 DQN 的动态深度多分支搜索自动配载算法

杨 奔, 王炜晔, 赵婉婷, 谢瑾奎

(华东师范大学 计算机科学与技术学院, 上海 200062)

**摘 要:** 自动配载是自动化码头运营的重要环节之一, 往往需要考虑多种因素, 限制条件复杂, 是一个 NP 完全性问题。传统的配载算法更关注配载结果而忽视箱区调度对作业效率的影响, 为提高堆场设备的利用率和配载结果的合理性, 根据桥机计划安排的配载任务, 提出一种深度优先且动态深度多分支搜索的配载算法。在线下学习阶段中通过历史数据学习得到箱区状态值函数, 线上配载选箱时综合值函数与各项约束条件通过动态深度分支搜索的方式得到最佳决策。在上海洋山港四期自动化集装箱码头进行真实船舶数据仿真测试, 结果表明, 与传统的贪心策略相比, 该算法可使翻箱率和双小车拼车率均降低 2% ~ 5%, 堆场设备利用率稳定在 90% ~ 96%。

**关键词:** 自动化码头; 自动配载; 翻箱率; 强化学习; 贪心算法

开放科学(资源服务)标志码(OSID):



**中文引用格式:** 杨奔, 王炜晔, 赵婉婷, 等. 基于 DQN 的动态深度多分支搜索自动配载算法[J]. 计算机工程, 2020, 46(8): 313-320.

**英文引用格式:** YANG Ben, WANG Weiye, ZHAO Wanting, et al. DQN-based automatic stowage planning algorithm using dynamic depth multi-branch search[J]. Computer Engineering, 2020, 46(8): 313-320.

## DQN-based Automatic Stowage Planning Algorithm Using Dynamic Depth Multi-branch Search

YANG Ben, WANG Weiye, ZHAO Wanting, XIE Jinkui

(School of Computer Science and Technology, East China Normal University, Shanghai 200062, China)

**[Abstract]** Automatic stowage planning is an important part of automatic terminal operation. As an NP-complete problem, it requires consideration of multiple factors and complex restrictions. Traditional stowage planning algorithms focus more on planning results and ignore the impact of container area scheduling on task efficiency. To improve the utilization rate of yard equipment and the rationality of stowage planning results, this paper proposes a stowage planning strategy using dynamic depth-first multi-branch search based on the stowage planning tasks arranged by the Crane Work Plan (CWP). In the offline learning phase, the state value function of the container area is obtained through learning of historical data. In the online planning phase, considering the value function and various constraints, the optimal decision for container selection is obtained by using dynamic deep multi-branch search. A simulation experiment is carried out using data of real ship in the fourth phase of the automatic container terminal project of Shanghai Ocean Port. Experimental results show that compared with the traditional greedy strategy, the proposed algorithm can reduce the turnover rate of containers and the sharing rate of double trolley by 2% to 5%, the utilization rate of yard equipment stabilized at 90% ~ 96%.

**[Key words]** automated terminal; automatic stowage planning; turn-over rate; reinforcement learning; greedy algorithm

**DOI:** 10.19678/j.issn.1000-3428.0055667

### 0 概述

随着经济全球化的不断加深, 国际航运发展迅速, 集装箱码头朝着自动化、智能化的方向发展。目前, 全球范围内已建成多个自动化码头, 而一个合理

的自动配载算法往往关系到集装箱船营运的经济性和安全性以及码头的整体工作效率。

洋山港四期工程作为全球规模最大的自动化码头, 在提高自动化程度、实现无人化操作的同时, 对船舶的装卸效率有极高的要求, 整个码头的装卸工

**基金项目:** 国家自然科学基金(11871222)。

**作者简介:** 杨奔(1994—), 男, 硕士研究生, 主研方向为计算智能控制; 王炜晔、赵婉婷, 硕士研究生; 谢瑾奎(通信作者), 副教授、博士。

**收稿日期:** 2019-08-05 **修回日期:** 2019-09-18 **E-mail:** 51174506096@stu.ecnu.edu.cn

艺采用双小车岸桥 + 自动引导小车 (AGV) + 自动化轨道吊 (ASC)<sup>[1]</sup>。自动配载是指在船公司给出预配船图和堆场箱区的情况下,根据桥机计划 (Crane Work Plan, CWP) 给出的作业要求将堆场上需要配载的集装箱在船舶上找一个合适位置的过程,而每个预配位对应数十个或数百个满足配载条件的集装箱,从满足配载条件的集装箱中选择一个合适的位置进行配载是一个研究重点。

自动配载是港口的一个核心问题,是一个 NP 完全性问题<sup>[2-4]</sup>。自动配载算法的主流方法包括启发式算法<sup>[5]</sup>、数学规划模型<sup>[6]</sup>、遗传算法<sup>[7]</sup>和蚁群算法<sup>[8]</sup>等。文献[9]研究线性规划优化算法,通过实例测试实验,有效提高船舶运输和港口运作效率。文献[10]针对配载种 Bay 优选问题,以倒箱量、重心横向偏移和初稳性高为目标,提出一种多目标离散粒子群算法 (MODPSO),为船舶配载提供多种方案以供备选。文献[11]提出一种集装箱配载自动化解决方案,该方案的推导是通过组合优化原则来完成的,主要是采用禁忌搜索的启发式算法来产生一个合理解。文献[12]提出一种禁忌搜索算法,该算法能够避免局部最优解,并且通过仿真实验证明了其配载的有效性。文献[13]利用遗传算法求得一个优化的集装箱装船序列,优化目标是满足客户的交货期限并最小化集装箱船舶的在港时长。文献[14]假设工作顺序已知的前提下,设计了一个三段式的启发式算法,通过该算法能够明显缩短桥机工作时长。文献[15]建立了一个动态规划模型确定预配箱位置以及最小化倒箱次数,同时利用决策树模型来得到实时决策信息。文献[16]为简化配载问题,在不考虑船舶稳性以及其它一些限制的情况下以最小化倒箱量为目标制定出一个 0-1 二元线性规划模型,并用启发式算法来寻找最佳配载方案。通常的集装箱船舶配载量为 2 000 箱 ~ 3 000 箱,主流方法对于这样大规模的算例求解效率不高。传统的算法往往更关注配载结果,而忽视了箱区调度效率因素,导致装船时箱区设备利用率低,使得整个码头的效率低下。

箱区设备利用率是提高自动化码头效率的重要因素,为提高箱区设备利用率,本文将配载的过程建模为顺序决策问题,通过无标签的历史数据,在线下阶段通过强化学习<sup>[17]</sup>得到箱区状态值函数,从而对不同的箱区状态给出一个量化的评估。在线上选箱时为了能够得到更合理的配载结果,提出一种深度优先且动态深度多分支搜索的配载策略。

## 1 研究背景及自动配载问题

自动配载是现代化港口的一项重要举措,关系到整个码头的运营效率。洋山港四期自动化码头将整个装船过程分为桥机工作计划 (CWP) 和自动配载模块。其中,CWP 安排了船舶上每个预配位作

业时间,该作业时间是自动配载的重要数据输入。CWP 计划中,包含每一个桥机的工作起始时间、结束时间、桥机位置、桥机作业顺等信息,桥机计划关联了船舶上对应位置被作业的先后顺序。因此,本文需要在遵循桥机作业的前提下进行集装箱的选取,才能找到符合预配条件要求的集装箱。每个预配位对应多个候选集装箱,而选取集装箱应考虑以下因素:

1) 船舶重量稳定性和船公司的重量要求。国际集装箱运输船的航海距离远,航行时间长,因此对船舶的重量稳定性要求很高,配载结果应遵循下重上轻的原则。不同的配载结果可能导致船舶的重心改变,稳定性改变,甚至吃水差发生变化。

2) 箱区任务分配。岸桥的作业效率往往要高于堆场轨道吊的作业效率,因此要根据 CWP 的作业顺序均衡堆场各个箱区的出箱量。为了最大化码头的整体效率,提高箱区设备利用率,减少闲置时间和阻塞时长是本文优化目标。

3) 翻箱因素<sup>[18]</sup>。翻箱率通常会作为评判一个配载结果好坏的重要指标,往往一次翻箱会造成更多的二次翻箱,使得堆场箱区效率低下,合理的配载结果能有效减少翻箱次数。

4) 双小车拼车。为了提高装卸效率,洋山四期轨道吊采用双箱吊作业工艺,即岸桥可以一次抓取 2 个小箱同时装载作业。为保证 AGV 运输的效率,双箱工艺的 2 个小箱应该严格地控制在同一个箱区发箱。

### 1.1 预处理

在配载时,一艘船一般有上千个集装箱需要配载,但对于多目标约束条件的优化问题,算法的时间代价较大。在洋山港四期码头的预配图信息中,船公司给每个预配位指定了对应的集装箱目的地、尺寸大小、箱型和危险品备注。因此,本文根据集装箱的目的地、尺寸、箱型、危险品等信息将其分为不同的属性组。这样,对于每一个预配位,都对应地有相应的属性组分类,只需要在对应属性组的候选集装箱中进行选择,大大减少了解的搜索空间,提高了算法的效率。

为了满足船舶稳定性和重量要求,配载结果遵循下重上轻、船舶重心不可发生偏移的原则。本文提出重量等级概念,例如将 0 t ~ 5 t、5 t ~ 10 t、10 t ~ 15 t……每 5 吨的重量差设为一个等级。根据预配船图和堆场预配集装箱的重量分布,将每一个预配位划分到相应的重量等级。重量等级划分的思路是根据重量分布趋势由重到轻、先下后上、从中间到两边的原则遍历船舶所有的预配位,在保证属性组匹配的前提下,为每一个预配位划分到一个重量等级。该阶段的预处理为后续的选取集装箱提供了极大的便利。

### 1.2 自动配载问题

自动配载是为了找到预配位和集装箱的最佳匹配。通过属性组分类和重量等级划分,一方面减少算法的搜索空间,另一方面保证了船舶重量稳定性和船公司的要求。通过数据预处理,得到带有重量等级信息的预配图,根据 CWP 安排的预配位作业顺序,可以得到配载作业列表,在列表中标记每个预配位的作业时间和预配的信息,包括船箱位、属性组、重量等级等。将列表中的信息依据作业时间进行排序,然后通过属性组和重量等级逐一地为每个预配位匹配集装箱。预处理阶段虽然减少了解的搜索空间,但是还未做到一一匹配,因此需提出一个合理的集装箱评价函数。通过评价函数对每个候选集装箱进行评价,从而匹配到最合适的集装箱。预处理阶段解决了船舶重量稳定性,其余3个因素通过图 1 中的线上和线下策略解决。

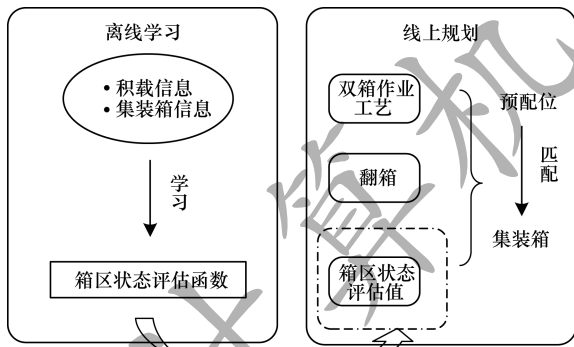


图 1 自动配载算法框架

Fig.1 Framework of automatic loading algorithm

## 2 离线学习阶段

配载阶段箱区设备的利用率是评价配载算法的优劣指标之一,同时也是最难控制的因素。当前配载选择哪一个箱区的集装箱进行配载会对后续一系列作业任务造成影响,这个决策有可能造成箱区阻塞,也有可能造成在后续任务中该箱区一直处于闲置状态。若要最大化码头的效率,不仅要关注当前的状况,而且更需要考虑到当前决策对未来的影响。因此,本文将配载过程构建为马尔可夫决策过程(MDP),将学习得到的评估函数用于评判箱区是否适合出箱,即将箱区的状态进行量化,从而选择更好的箱区中的箱子进行配载。

### 2.1 问题描述

对于顺序决策问题可以通过马尔可夫决策过程建模。在马尔可夫决策过程中,agent 通过决策选择一个动作,然后跳转到下一个状态。agent 的目标是

最大化增益  $G_t = \sum_{i=t}^T R_{i+1}$ ,即从时间  $t$  开始未来的预期收益。

MDP 定义:在此阶段,本文将箱区状态构建为一个 agent,虽然会造成多个 agent 的问题,但由于不存在特殊箱区,这些 agent 本质都是一样,因此不需要考虑多 agent 之间的交互问题。同时模型的构建对于状态转化、行动及奖励的定义会大大简化。

状态:为了使箱区的效率值最大化,需要充分利用设备资源,因此最大化堆场的整体效率是本文的目的。本文在状态定义方面不仅关注单独的一个箱区,同样考虑堆场其他箱区的情况,此外,本文将箱区的状态表示划分为如下 5 个阶段:

- 1) ASC 从 AGV 交互区移动至目标集装箱所在贝位。
- 2) 提取目标集装箱阶段。
- 3) 从目标集装箱所在贝位移至交互区。
- 4) 翻箱(若存在翻箱)。
- 5) ASC 闲置。

其中,第 1 阶段和第 3 阶段操作的预估平均时间长度都是 1 min30 s,目标集装箱的贝位距离交互区越远,则这 2 个阶段的时间越久。第 2 阶段预估执行时间长度也是 1 min,翻箱阶段中同贝位执行翻箱操作的时间长度是 1 min。由此,任何一个箱区的状态都可以通过一个 5 维向量来表示该箱区的状态。那么对于一个箱区集合  $G = \{g\}$ ,任意一个箱区的状态表示如下:

$$s_{g_i} = (g_{i_1}, g_{i_2}, g_{i_3}, g_{i_4}, g_{i_5}) \tag{1}$$

$$g_{i_j} = \begin{cases} 1, & g_i \text{ 箱区处于第 } j \text{ 阶段} \\ 0, & g_i \text{ 箱区处于其他阶段(非第 } j \text{ 阶段)} \end{cases} \tag{2}$$

$$s. t. \sum_{j=1}^5 g_{i_j} = 1, j \in [1, 2, 3, 4, 5] \tag{3}$$

其中,  $i \in [1, 2, \dots, b]$  代表  $b$  个箱区,用来区分代表不同的箱区,  $s_{g_i}$  虽然能够表示  $g_i$  箱区的状态,但是要使整个堆场的效率最大化,同时还需要兼顾除了目标箱区的其他箱区状态,故给出统筹其他箱区的综合表示:

$$s_{G'} = ( \sum_{i=1, i \neq k}^b g_{i_1}, \sum_{i=1, i \neq k}^b g_{i_2}, \sum_{i=1, i \neq k}^b g_{i_3}, \sum_{i=1, i \neq k}^b g_{i_4}, \sum_{i=1, i \neq k}^b g_{i_5} ) \tag{4}$$

其中,  $k$  表示目标箱区,则可以将目标箱区的状态表示为:

$$S_{g_k} = \begin{pmatrix} s_{g_k} \\ s_{G'} \end{pmatrix} \tag{5}$$

动作:本文中的模型有 2 种动作类型,第 1 种是 agent 选择出箱,进而更改箱区状态的矩阵表示,第 2 种是 agent 不选择出箱,有 2 种情况下会出现这种情况,一种是因为这个箱区不合适出箱,另一种是因为这次预配位需求的箱子在这个箱区不存在候选箱。

奖励:奖励的定义决定了模型优化的目标。自动配载算法的目的是为了提高码头的工作效率,那么如何在最短的时间内完成装船任务将作为算法优化的目标。作为箱区状态评估模型,本文希望 agent 能够给出一个评判箱区状态优劣的优势值,优势值越好,表示这个箱区在接下来的作业中越适合出箱。目标箱区进行出箱操作和箱区的状态息息相关,若目标箱区还处于上次配载作业的某一阶段,那么本次配载的预配箱需等待上次作业结束,ASC 才可以进行提箱任务。若目标箱区处于闲置状态,上述情况将不存在,可以直接进行提箱任务。本文将延缓时间和配载时间作为奖罚值,延缓时间是指目标箱区在当前时间窗口完成上次配载作业的预估时间长度,若目标箱区处于闲置状态,则延缓时间为 0。配载时间指目标箱区完成本次配载的时间长度,即阶段 1~阶段 3 的时间总和,若存在翻箱,则将翻箱时间损耗加入配载时间。对于某次配载,若总共预估需要  $T$  个阶段,延缓时间需要  $T'$  个阶段,其中  $T' < T$ ,则奖惩公式如下:

$$R = \sigma \frac{\sum_{m=T'+1}^T t_m}{\sum_{n=1}^T t_n} \quad (6)$$

其中,  $\sigma$  表示 ASC 和 QC 效率的比值,不同的码头设备,效率会有所差别,效率比值越高,奖励值相对更大,如洋山港四期的比值为 2。

## 2.2 策略评估

基于 2.1 节模型的构建,该阶段模型求解通过历史预配数据训练得到,这里使用的历史预配数据是指没有配载结果的初始数据,本文通过文献[19-20]中的算法框架来求解强化学习模型。DQN 通过建立 2 个相同结构的神经网络来求解,包括当前网络  $Q$  和目标网络  $Q'$ ,当前网络用来评判箱区状态优劣,实时更新模型参数,目标网络  $Q'$  的参数不需要迭代更新,而是每隔一定轮数从  $Q$  网络中复制过来,即延时更新,这样减少了目标  $Q'$  和当前  $Q$  的相关性,同时可以得到损失函数。算法中通过建立经验池机制,将  $(S, S', R)$  三元组存入经验池。由于模型是针对于箱区状态进行评估,如图 2 所示,将箱区状态序列作为输入,评估值作为神经网络的输出。因此,经验池中并没有动作因素,agent 在进行箱

区选择时,都要根据之前的任务来更新每个箱区的状态矩阵,得到当前时间窗口的状态矩阵。

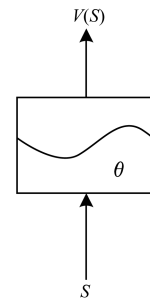


图 2 模型结构

Fig. 2 Model structure

目标神经网络  $Q'$  的状态价值函数如下:

$$V' = R + \gamma \max Q'(S') \quad (7)$$

损失函数定义如下:

$$\text{loss} = \frac{1}{k} \sum_{q=1}^k (V' - Q(S))^2 \quad (8)$$

线下强化学习算法的伪代码如算法 1 所示。 $\hat{S}$  表示所有的箱区状态集合,  $CWP$  集合中包含了每个预配位的作业时间点,在第 4 行中,根据  $CWP$  集合中预配位的时间点先后顺序,依次为每个预配位选取合适的集装箱,在第 5 行中,选取到合适箱区后,原则上选取翻箱最少的箱子作为目标箱,在第 9 行中,  $is\_end$  若为 true,则表示本轮最后一个预配位进行配载,没有下一次的状况跳转;若为 false,则相反。

### 算法 1 基于 DQN 的箱区状态值评估算法

输入  $CWP$  作业顺序集合  $T$ , 需要配载的集装箱集合

$C$ , 箱区状态集合  $\hat{S}$

输出 箱区状态值函数  $v(s)$

1. 初始化神经网络  $Q$  并随机化参数  $\theta$ , 初始化神经网络  $Q'$  和随机化参数  $\theta' = \theta$ ; 建立经验池  $D$ 。
2. For episode = 1,  $M$  do
3. 初始化箱区状态  $\hat{S}$ ;
4. For  $t = 1, T$  do
5. 根据  $\epsilon$ -贪婪策略, 从箱区状态集合中通过  $Q(S)$  选择合适的箱区发箱;
6. 获得奖励  $R$ , 更新箱区状态集合  $\hat{S} \leftarrow \hat{S}$ ;
7.  $(S, S', R, is\_end)$  存入经验池  $D$ ;
8. 经验池中随机采样 minibatch 样本  $(S, S', R, is\_end)$ ;
9.  $V' = \begin{cases} R, is\_end \text{ is true} \\ R + \gamma \max Q'(S'), is\_end \text{ is false} \end{cases}$
10. 使用均方误差  $(V' - Q(S))^2$  做梯度下降更新网络参数  $\theta$ ;
11. 每隔  $\alpha$  步更新  $\theta' = \theta$ ;

12. End for

13. End for

### 3 线上规划阶段

线下通过大规模的原始数据训练得到了箱区状态评估函数,那么在线上配载时,通过箱区评估函数给出的评估值再结合候选集装箱的实际位置情况进行一个统一的评价,从而得到一个合理的配载方案。

#### 3.1 翻箱与双小车拼车

洋山港四期的堆场由多个并排的箱区组成,在箱区中,竖直堆放的一列成为排,排列在一起的多个排组成一个贝,从底部到顶部又分为若干个层。每一个集装箱都会放在其中具体的一个位置上,而其对应的箱区号、贝位号、排号、层号标记了该集装箱在堆场中唯一的位置,这也是配载过程中判断集装箱具体位置的重要依据。在配载中,倒箱是影响配载效率的重要因素之一。倒箱是指在集装箱提取过程中,当目标集装箱所在位置上方有其他阻碍箱压在上面时,只能将阻碍箱移动到其他位置后才能取到目标集装箱,移开阻碍箱的过程叫倒箱。

目前双箱吊具已普遍应用于现代化港口,如图 3 所示,针对 20 英尺的小箱,岸桥可以通过一次性抓取两个同时进行装载。所以,针对双箱作业工艺,每次 AGV 从堆场提取集装箱时,应严格控制从一个箱区发箱,若将双箱工艺的作业安排到两个箱区进行发箱,AGV 需要前往不同的箱区提取集装箱,这样严重降低了装卸效率。

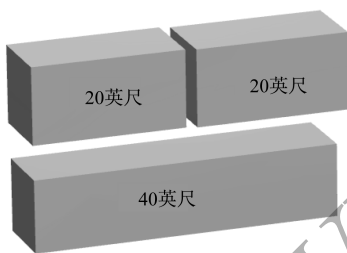


图 3 20 英尺小箱和 40 英尺大箱

Fig. 3 Twenty foot small box and forty foot large box

#### 3.2 集装箱评价函数

在线上配载具体某一个预配位时,对于所有的候选集装箱,需要指定一个合理评价策略,整个决策过程根据评价策略的指标来进行,每一个预配位的决策过程都希望选到最优解或者次优解,进而获得整体的优解。因此,评价函数的制定对预配位选择的结果起到决定性的作用。

虽然线下通过大规模原始数据训练得到了箱区状态评价函数,在线上配载时, $V(S)$ 最有优势值的

箱区并不一定可以选到最优的箱子,如在众多箱区中,此次配载 1 号箱区的优势值最高,但是该箱区的预配集装箱都需要翻箱才能够得到,因此这并不是一个较好的策略,同时还需要控制双箱工艺作业同一个箱区发箱。

综合考虑翻箱和双箱工艺的线上集装箱配载评价函数如式(10)所示:

$$f(c) = \theta^{\Delta h_c} (V(S_c) + \delta), 0 < \theta < 1 \quad (9)$$

其中,  $\delta = \begin{cases} 1, & \text{双箱作业工艺且, } n_c \geq 2 \\ 0, & \text{单箱作业工艺或, } n_c < 2 \end{cases}$ ,  $\Delta h_c$  表示目标集装箱  $c$  的翻箱数目,翻箱数目越多,对应的评价值越小。针对双箱作业工艺,20 英尺小箱数目越多的箱区在式(9)中通过  $\delta$  条件赋予了更多的优势值, $n_c$  表示目标集装箱  $c$  所在箱区中与  $c$  同属性组同重量等级集装箱的数目。同类 20 英尺小箱数目越多,更易避免双箱作业的拼车现象。评价值  $f(c)$  值越大,集装箱  $c$  更有机会被选中。基于以上设计的评价函数,本文提出一种深度优先、动态深度多分支搜索的配载策略,算法流程如算法 2 所示。

#### 算法 2 动态深度多分支搜索算法

输入 CWP 作业顺序集合  $T$ , 需要配载的集装箱集合

$C$ , 箱区状态集合  $\hat{S}$

输出 每个船舶预配位对应的堆场集装箱信息

1. 初始化箱区状态  $\hat{S}$ ;

2. For  $t = 1, T$  do

3. Find  $C'$  from  $C$  // 根据重量等级和属性组约束

4. For  $c \in C'$  do

5.  $f(c) = \theta^{\Delta h_c} (V(S_c) + \delta), 0 < \theta < 1$

6. End for

7. 根据评价函数选择最好的前  $W$  个集装箱作为候选箱;

8. 根据船舶预配位来搜索当前预配位后面的  $(D-1)$  个预配位;

9. 假设将当前评价最好的前  $W$  个集装箱中的每个集装箱作为目标集装箱,分别求这  $(D-1)$  个预配位最合适的一个集装箱;

10. 根据加权每条分支的结果来为当前预配位选择最佳的一个集装箱;

11. 更新箱区状态集合  $\hat{S}$ ;

12. End for

动态深度多分支搜索算法示意图如图 4 所示,算法中的搜索宽度  $W$  和深度  $D$  作为可调节的输入参数,通过设置不同的参数可以使算法适用于不同规模的配载问题,也能够使算法的计算时间在可控的范围内结束。此外,算法深度  $D$  在同一次配载过

程中是会出现动态调整的,因为每一个预配位的后续不一定有 $(D-1)$ 个未配载预配位(如配载结束的预配位),或者后续预配位中可能会遇到没有符合条件候选箱的情况,这时需要将当前预配位的搜索深度调整为可计算的深度。

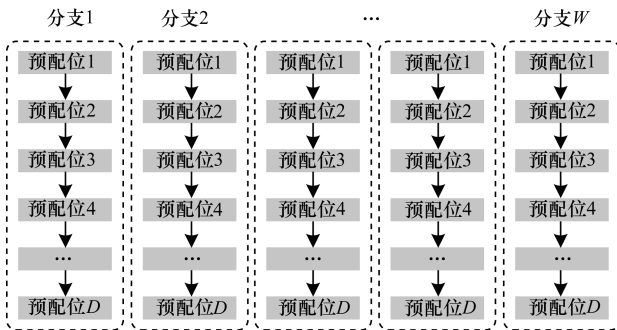


图4 动态深度多分支搜索算法示意图

Fig.4 Schematic diagram of dynamic deep multi-branch search algorithm

#### 4 实验结果与分析

本文通过洋山港四期真实船舶数据来验证算法的有效性。在离线训练阶段,本文通过2019年1月至3月的洋山港四期的历史船舶配载数据来训练,其中包括100箱~400箱的沿海驳船和1000箱~3000箱的集装箱船。通过这些大规模的数据训练,得到最终的 $V(S)$ 函数。堆场集装箱的进箱结果对配载结果也会有部分影响,本文堆场集装箱的进箱是通过DMDP算法<sup>[21]</sup>生成,集装箱均匀分布在不同箱区。

##### 4.1 自动配载指标分析

自动配载算法的有效性除了码头工人的判断外,一般通过以下几个指标来判断配载方案的优劣,即线上算法的计算时间、翻箱率、箱区设备利用率、双小箱拼车率。

本文通过一艘积载量为1572箱量的集装箱船来测试分析动态深度多分支策略的宽度和深度参数,如表1所示,通过设置不同的搜索宽度 $W$ 和深度 $D$ ,分析算法的计算时间、翻箱率和拼车率。随着宽度和深度的增加,算法的执行时间明显增加,但是对应的翻箱率和拼车率都有相应的下降,尤其是在宽度为5、深度为10时可以得到一个较好的结果。虽然之后随着深度和宽度的增加,翻箱和拼车的指标略有下降,但是算法的执行时间却相对增加。为方便分析,本文后面的测试对比分析均采用宽度为5、深度为10的参数设置。

表1 宽度与深度参数分析比较

Table 1 Analysis and comparison of width and depth parameters

参数	计算时间/s	翻箱率/%	拼车率/%
$W=5, D=1$	23	11.87	16.43
$W=5, D=5$	81	10.04	16.17
$W=5, D=10$	146	8.87	13.23
$W=5, D=15$	208	8.23	13.25
$W=5, D=20$	281	8.34	13.73
$W=5, D=25$	409	8.12	13.23
$W=10, D=5$	72	10.25	15.92
$W=10, D=10$	223	8.47	13.19
$W=10, D=15$	328	8.29	13.45
$W=10, D=20$	428	8.09	13.23
$W=10, D=25$	588	8.10	12.92

为测试分析不同规模船舶的效果,本文通过4组不同规模的数据进行测试,如表2所示,这4个航次对应的积载箱量分别是389个、1239个、1980个、2750个。其中,航次1为沿海驳船,积载量相对较少,算法翻箱率指标可以控制在12%以下。堆场上箱子的堆放会影响翻箱数目,在做堆场计划时,往往并不知道配载船舶的信息和后期的积载需求,堆场进口的箱子只能根据其重量、大小等因素的不同分配指定到合理的箱区中,所以部分翻箱仍然难免。在拼车率的指标上,可以直观地看到本文算法可以控制在20%左右。造成拼车的原因主要有2个方面:1)由于翻箱约束导致,候选的集装箱为了避免翻箱不得不去其他箱区拼车;2)堆场计划影响导致,部分箱区中候选的20英尺小箱数目是奇数,往往导致最后一个小箱不得不与其他箱区的小箱拼车,所以,部分拼车情况是不可避免的。

表2 箱区测试结果

Table 2 Results of box area test

航次	积载量/个	计算时间/s	翻箱率/%	拼车率/%
航次1	389	56	5.32	12.53
航次2	1239	127	8.34	14.34
航次3	1980	351	11.37	23.87
航次4	2750	688	9.87	17.85

表3展示了航次2的箱区作业统计结果。箱区的工作方式与数据结构中的队列相似,每次需要把旧的任务处理后才能开始下一次的任務。如果把每个箱区看作是一个队列,当目标箱区每添加一个新的任务时,查看当前箱区队列是否为空,若为空则代表本次作业的阻塞时长是0,若队列不为空则统计队列中待做任务的工作时长。最后将每次作业的阻塞

时长累加得到阻塞总时长。航次2 总共的装箱量是1 239 个, 这些箱子分布在堆场的10 个箱区。表3 中的总作业时长代表这个箱区整个作业过程的模拟预估时长, 此外统计了每个箱区的空闲总时长和阻塞总时长。通过表3 可以看出, 经过强化学习得到的模型能将空闲时长和阻塞时长多数控制在15 min 左右。其中, 45 箱区和51 箱区的阻塞时长甚至可以控制到0 min, 47 箱区的箱子数目远远多于其他箱区, 因此造成了阻塞时长高达31 min, 这种情况是可接受的。实验结果表明, 本文算法能够合理地分配箱区任务以及充分利用设备资源。

表3 箱区作业情况分析  
Table 3 Analysis of box area task

箱区号	箱量/个	总作业 时长/min	闲置 时长/min	阻塞 时长/min	设备利用率 /%
43	127	470	15	9	94.89
44	91	422	24	15	90.75
45	96	464	17	0	96.33
46	131	496	20	11	93.75
47	160	522	12	34	91.76
48	120	489	19	3	95.50
49	139	491	14	12	94.70
50	128	499	16	10	94.78
51	112	445	16	0	96.40
52	135	491	21	15	92.66

## 4.2 对比算法

目前对于自动配载算法的研究大多使用整数规划、遗传算法, 算例规模较大时算法的求解时间也较长。因此本文使用以下3 个算法进行对比实验。

**算法1** 通过文献[22]的算法思想和洋山四期的数据性质进行扩充修改的自动配载算法。在实验过程中, 训练期间采用了原文算法中的9 个隐含层网络, 通过迭代训练不同的配载数据得到算法模型。为了方便比较双小车拼车率的指标, 本文在网络结构的输入节点的第9 维上进行了修改, 将判断从同一个贝位修改为同一个箱区, 同时加上双箱作业的约束条件。

**算法2** 基于时间差值的贪心配载策略。根据CWP 安排时间点和配载过程中预估的时间点做差值, 其中翻箱通过时间罚值体现, 实验中每翻一个箱子, 时间罚值是150 s。双小车拼车情况通过目标箱区20 英尺小箱占比率转化为时间罚值来计算, 实验过程中每个20 英尺小箱转化罚值为1 min 的罚分时长, 最终通过贪心策略得到配载结果。

**算法3** 本文算法。

3 个对比算法分别根据4.1 节中的4 个航次数据进行比较, 通过分析翻箱率和双小车拼车率来验

证本文算法。通过对比实验结果可以看出, 在航次1 的沿海驳船上, 3 个算法在翻箱率和拼车率上的指标基本一致, 但是在大型集装箱船舶的数据上, 本文算法相对于算法1 和算法2, 在这2 个指标上得到了很大程度的优化, 尤其是在航次4, 本文算法相比于算法1 在翻箱率上下降了3%, 与算法2 相比下降了4%。在双小车拼车率方面, 航次3 的数据表现更佳, 相比算法1 和算法2 分别下降了6% 和11%, 效果比较明显。

通过以上实验的分析和对比可以看出, 本文算法在不同规模的船舶上的指标均优于现有的传统算法。在求解时间上, 即使采用宽度为5、深度为10 的参数设置, 2 000 积载量的船舶也可以在5 min 之内得到配载方案。本文算法在减少翻箱率和双小车拼车率的同时, 提高了箱区设备的利用率, 提升了整个码头的工作效率。翻箱率与双小箱拼车率对比如图5、图6 所示。

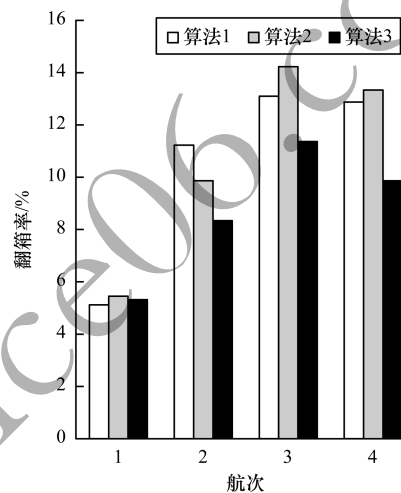


图5 翻箱率对比

Fig. 5 Comparison of turn-over rate

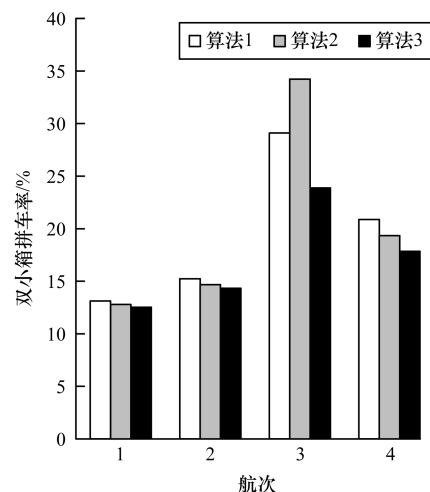


图6 双小箱拼车率对比

Fig. 6 Comparison of car sharing rates of double small boxes

## 5 结束语

本文主要研究自动化码头集装箱出口流程中的自动配载问题。通过将配载过程建模为顺序决策问题,运用强化学习通过大规模数据训练得到箱区状态值函数,以均衡箱区工作能力,并使堆场箱区设备的利用率稳定在90%~96%。为得到合理的配载结果,提出一种深度优先且动态深度多分支搜索的线上策略。实验结果表明,相比于传统贪心算法,该算法翻箱率和双小车拼车率下降了2%~5%。下一步将考虑实船配载作业的容错度,以加强对实时问题的处理能力。

感谢洋山港四期项目团队的数据支持!

### 参考文献

- [1] LIU Guanghong. General layout of fully automated container terminal in phase IV of Yangshan project[J]. Port & Waterway Engineering, 2016(9): 46-51. (in Chinese)  
刘广红. 洋山四期工程全自动化集装箱码头总体布置[J]. 水运工程, 2016(9): 46-51.
- [2] LI Hu, HUO Jiazhen, CHEN Yao. A hybrid genetic algorithm for the ship stowage planning problem[J]. Industrial Engineering and Management, 2006, 11(3): 27-31. (in Chinese)  
李虎, 霍佳震, 陈瑶. 一种求解船舶配载问题的混合遗传算法[J]. 工业工程与管理, 2006, 11(3): 27-31.
- [3] ZHENG Feifeng, MEI Qihuang, LIU Ming, et al. Research on genetic algorithm and greedy method of stowage planning in multiple ports [J]. Operations Research and Management, 2018, 27(5): 5-11. (in Chinese)  
郑斐峰, 梅启煌, 刘明, 等. 基于遗传算法与贪婪策略的多港口集装箱配载研究[J]. 运筹与管理, 2018, 27(5): 5-11.
- [4] KEMP S E, ROACH P A, WARE A J, et al. Artificial intelligence for automatic container stowage planning optimization[J]. Hansa, 2003, 50(4): 151-156.
- [5] AMBROSINO D, SCIOMACHEN A, TANFANI E. A decomposition heuristics for the container ship stowage problem[J]. Journal of Heuristics, 2006, 12(3): 211-233.
- [6] AMBROSINO D, SCIOMACHEN A, TANFANI E. Stowing a containership: the master bay plan problem [J]. Transportation Research, Part A: Policy and Practice, 2004, 38(2): 81-99.
- [7] TODD D S, SEN P. A multiple criteria genetic algorithm for containership loading [C]//Proceedings of IEEE International Conference on Genetic Algorithms. Washington D. C., USA: IEEE Press, 1997: 135-146.
- [8] WEI Jiajun. Research on intelligent stowage of container ships[D]. Dalian: Dalian Maritime University, 2012. (in Chinese)  
卫家骏. 集装箱船智能配载研究[D]. 大连: 大连海事大学, 2012.
- [9] HOU Fangbo. Using linear programming to optimize container ship loading[J]. Ship Science and Technology, 2017, 39(2A): 148-150. (in Chinese)  
侯方博. 利用线性规划优化集装箱船舶的配载[J]. 舰船科学技术, 2017, 39(2A): 148-150.
- [10] WANG Yuanyuan, CHEN Shunhui. Particle swarm optimization for container ship loading [J]. Ship Engineering, 2017, 39(S1): 255-258, 291. (in Chinese)  
汪圆圆, 陈顺怀. 集装箱船配载的粒子群算法[J]. 船舶工程, 2017, 39(S1): 255-258, 291.
- [11] WILSON I D, ROACH P A. Principles of combinatorial optimization applied to container-ship stowage planning[J]. Journal of Heuristics, 1999, 5(4): 403-418.
- [12] BORTFELDT A, GEHRING H, MACK D. A parallel tabu search algorithm for solving the container loading problem[J]. Parallel Computing, 2003, 29(5): 641-662.
- [13] AYACHI I, KAMMARTI R, BORNE M K E P. A genetic algorithm to solve the container storage space allocation problem [C]//Proceedings of IEEE International Conference on Computational Intelligence and Vehicular System. Washington D. C., USA: IEEE Press, 2010: 132-143.
- [14] LEE Y, HSU N Y. An optimization model for the container pre-marshalling problem [J]. Computers and Operations Research, 2007, 34(11): 3295-3313.
- [15] KIM K H, PARK Y M, RYU K R. Deriving decision rules to locate export containers in container yards[J]. European Journal of Operational Research, 2000, 124(1): 89-101.
- [16] AVRIEL M, PENN M, SHPIRER N, et al. Stowage planning for container ships to reduce the number of shifts [J]. Annals of Operations Research, 1998, 76: 55-71.
- [17] SUTTON R S, BARTO A G. Reinforcement learning: an introduction [J]. IEEE Transactions on Neural Networks, 1998, 9(5): 1054-1054.
- [18] CASERTA M, STEFAN V, SNIEDOVICH M. Applying the corridor method to a blocks relocation problem[J]. Operations Research-Spektrum, 2011, 33(4): 915-929.
- [19] WATKINS C, DAYAN J P. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.
- [20] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning [EB/OL]. [2019-07-12]. <https://www.openai.com/paper/4042798#.Xt-4ZG5uI2w>.
- [21] LI Suikai, LI Yitao, SUN Weiwei, et al. A storage space allocation algorithm for export containers in automated container terminals [J]. Computer Engineering, 2019, 45(5): 272-278, 284. (in Chinese)  
李隋凯, 励益韬, 孙未未, 等. 一种自动化集装箱码头出口箱进箱选位算法[J]. 计算机工程, 2019, 45(5): 272-278, 284.
- [22] SHEN Yifan, ZHAO Ning, XIA Mengjue, et al. A deep Q-learning network for ship stowage planning problem [J]. Polish Maritime Research, 2017, 24(3): 102-109.