

# 基于改进蚁群算法在集装箱码头 泊位优化中的应用

解瑞红<sup>1</sup>, 邢红光<sup>1</sup>, 张洪军<sup>2</sup>

(1. 齐齐哈尔大学, 黑龙江 齐齐哈尔 161006; 2. 岭南师范学院 机电工程研究所, 广东 湛江 524048)

**摘要:** 探讨了蚁群算法在码头泊位优化中的应用问题。以一定时间内所有船舶在港总时间最短作为优化目标, 建立泊位优化配置模型。在求解中, 利用改进蚁群算法确定了停靠在每个泊位上船舶的作业顺序。然后, 在实例场景与最初的先到先服务算法相比, 改进的多态蚁群算法在停港时间上有较大的提高。

**关键词:** 港口物流; 码头泊位; 蚁群算法; 最优化

中图分类号: U673.2

文献标志码: A

文章编号: 1007-984X(2019)06-0059-04

港口的服务系统是一个十分复杂的系统, 需要各个环节高度协调发展, 各种有效资源都应该被合理高效地利用。港口船舶调度优化可以对船舶资源进行合理配置, 提高港口营运效率, 改善船舶营运的经济效益。张新宇等人<sup>[1]</sup>提出基于模拟退火算法和群遗传算法相结合的港口船舶调度优化模型, 这种调度优化模型可以使港口船舶调度优化问题求解速度明显加快, 但是对港口船舶调度优化问题进行简化, 只能从航道容量角度进行研究, 港口航道中经常出现船舶拥挤现象, 使得港口船舶调度成本比较高<sup>[2]</sup>。晏榆洋等人<sup>[3]</sup>基于蚁群算法对多约束优化目标函数进行寻优, 求得港口船舶调度优化问题的解, 进行港口船舶调度优化仿真模拟实验, 以验证蚁群算法的港口船舶调度优化模型的可行性和优越性。

泊位利用率是港口物流调度的一个重要参数, 能够客观揭示港口规划的合理性, 其分配问题直接决定了码头的服务水平。所谓泊位分配, 就是在有限的时间内为更多的船只提供泊位, 工期靠泊作业, 减少在港时间, 以提高作业效率、创造更好的经济价值。集装箱码头的配置流程一般是船舶到港后, 如果有满足停靠条件的空闲泊位, 则直接进入泊位, 如果没有合适的泊位, 先在锚地排队候泊, 等到有合适的泊位停靠。靠泊后, 装卸机械就绪后开始集装箱的装卸作业, 直至装卸完成, 船舶离港。而泊位配置问题核心就是通过对到港船舶分配合理的停靠泊位, 确定到港船舶的停靠位置、顺序和开始作业时间等, 使得所有到港船舶的总成本最小。韩晓龙等人<sup>[4]</sup>讨论了集装箱港口中的泊位配置问题, 给出了连续泊位下同时考虑泊位资源和桥吊资源的泊位-桥吊配置优化模型。船舶到港后, 能够在最短时间内靠港卸货, 对泊位占用时间越短, 货物的周转量也越大, 港口的物流效率也会大大增加。为了不使港口的泊位分配成为整个港口系统的“瓶颈”环节, 优化港口泊位调度, 缩短船舶接受服务的等待时间<sup>[5-6]</sup>。而蚁群算法是一种新型的模拟进化算法<sup>[7]</sup>, 具有正反馈性、并行性等优点, 在求解组合优化问题时具有一定的优越性, 已用于解决多种组合优化问题。但是, 蚁群算法也存在收敛速度慢等问题。Marco Dorigo 等<sup>[8]</sup>针对蚁群算法的不足之处做了大量的研究工作, 为能够更有效地解决不同领域不同特征的优化问题, 提出了精英蚁群优化算法、最大最小蚂蚁系统等多种改进策略<sup>[9]</sup>。本文将使用改进后的蚁群算法应用于集装箱码头泊位优化问题, 以期取得较好的效果。

## 1 集装箱码头泊位问题

### 1.1 问题描述

收稿日期: 2019-05-23

基金项目: 齐齐哈尔市科学技术计划项目(NYGG-201702); 湛江市科技发展专项资金竞争性分配项目(2018A02016); 齐齐哈尔大学大学生创新创业训练计划项目(201810232214); 齐齐哈尔大学大学生创新创业训练计划项目(201810232234)

作者简介: 解瑞红(1991-), 女, 黑龙江齐齐哈尔人, 助教, 硕士, 主要从事新材料应用研究, 909262526@qq.com。

假设集装箱码头有  $M$  个泊位,  $\bar{L}_j, \bar{D}_j$  分别是第  $j$  ( $j=1,2,3,\dots,M$ ) 个泊位的长度和深度, 先在一定时间段内有  $N$  艘到港的船舶等待停靠, 第  $i$  ( $i=1,2,3,\dots,N$ ) 艘船舶的长度和吃水深度分别为  $L_i, D_i$ , 到港和约定离岗时间为  $A_i, T_i$ ,  $p_{ij}$  为第  $i$  艘船在第  $j$  泊位的装卸作业时间,  $S_i$  表示船舶  $i$  开始作业时间,  $r_j$  表示一定时间内停靠泊位  $j$  的船舶数量, 另为方便计算, 做如下假设:

- 每条船舶只能停靠一次 (并不影响实际应用, 因为可将多次停泊的船视为两艘);
- 船舶到达港口后才能被服务;
- 泊位水深充足, 大于船只吃水深度;
- 同一泊位同一时刻只能停泊一条船。

目标问题: 确定一个船舶靠泊的调度, 是在港船舶总成本最小。

## 1.2 模型建立

设  $x_{ij}^s = \begin{cases} 1, & \text{如果第 } i \text{ 艘船停泊在第 } j \text{ 个泊位上第 } s \text{ 个被作业} \\ 0, & \text{否则} \end{cases}$ , 那么该问题的数学模型如下:

首先界定目标函数, 要表示在港船舶总成本最小。总成本包括了船舶的等待停泊成本、作业成本; 同时为了加强管理, 对于非不可抗力造成延迟离港的船舶应该予以惩罚, 惩罚的成本也应该算进在港成本里。于是, 有

$$\min [\sum_{i=1}^N \sum_{j=1}^M \sum_{s=1}^{r_j} w_{1i} (S_i - A_i) x_{ij}^s + \sum_{i=1}^N \sum_{j=1}^M \sum_{s=1}^{r_j} w_{2i} p_{ij} x_{ij}^s + \sum_{i=1}^N \sum_{j=1}^M \sum_{s=1}^{r_j} w_{3i} (S_i + p_{ij} - T_i)^+ x_{ij}^s] \quad (1)$$

3 部分分别为单位候泊成本、单位作业成本和单位延迟成本。

然后需要约束船舶只能停靠一次, 故此, 必须保证

$$s.t. \sum_{j=1}^M \sum_{s=1}^{r_j} x_{ij}^s = 1, \quad i = 1, 2, \dots, N \quad (2)$$

接着需要保证同意泊位同意时间只能服务一艘船舶, 因而有下式

$$\sum_{i_1=1}^N S_{i_1} x_{i_1 j}^{s+1} - \sum_{i_2=1}^N (S_{i_2} + p_{i_2 j}) x_{i_2 j}^s \geq 0, \quad j = 1, 2, \dots, M; s = 1, 2, \dots, r_j - 1 \quad (3)$$

随后要确定泊位长度和水深满足待停泊船只的要求, 分别为式 (4) 和式 (5)

$$(\bar{L}_j - L_i) x_{ij}^s \geq 0, \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M; s = 1, 2, \dots, r_j \quad (4)$$

$$(\bar{D}_j - D_i) x_{ij}^s \geq 0, \quad i = 1, 2, \dots, N; j = 1, 2, \dots, M; s = 1, 2, \dots, r_j \quad (5)$$

随后要确保船舶在到达后才能被服务, 故而模型必须满足

$$S_i \geq A_i, \quad i = 1, 2, \dots, N \quad (6)$$

最后, 定义决策变量的取值范围

$$x_{ij}^s = 0, 1; i = 1, 2, \dots, N; j = 1, 2, \dots, M; s = 1, 2, \dots, r_j \quad (7)$$

## 2 配置问题的改进蚁群算法

算法的基本思路为首先根据船舶的长度和吃水深度以及在泊位上的作业成本确定船舶要停靠的泊位, 再利用改进蚁群算法确定停靠在每个泊位上船舶的作业顺序。在利用改进蚁群算法时, 应该向船舶等候时间和船舶的延迟时间这两个目标优化。

设在泊位  $j$  ( $j=1,2,\dots,M$ ) 上, 若船舶  $i$  完成后假设紧接着船舶  $l$  作业, 那么船舶  $l$  的等候时间和延迟时间分别为  $(A_l - S - p_{ij})^+$  和  $(S_i + p_{ij} + p_{lj} - T_l)^+$ 。

首先定义选择概率。针对每个泊位  $j$  ( $j=1,2,\dots,M$ ), 改进蚁群算法的节点为安排在泊位  $j$  上作业的船舶,  $p_{il}^j(t)$  为路径选择概率, 表示若船舶  $i$  在  $t$  时刻在泊位  $j$  上作业, 则选择下一个在本泊位上作业的是船舶  $l$

的可能性； $\tau_{il}^j(t)$  表示若船舶  $i$  在  $t$  时刻在泊位  $j$  上作业，则选择下一个在本泊位上作业的是船舶  $l$  的这条路径上的信息素； $\eta_{il}^j$  表示若船舶  $i$  在  $t$  时刻在泊位  $j$  上的作业，则选择下一个在本泊位上作业的是船舶  $l$  这条路径上的能见度。设  $\eta_{il}^j = \frac{1}{[(A_l - S_i - p_{ij})^+ + (S_i + p_{ij} + p_{lj} - T_l)^+ + c]}$ ,  $c$  为非0正整数，那么选择概率为

$$p_{il}^j(t) = \begin{cases} \frac{[\tau_{il}^j(t)]^\alpha [\eta_{il}^j(t)]^\beta}{\sum_{q \in \text{allowedk}} [\tau_{il}^q(t)]^\alpha [\eta_{il}^q(t)]^\beta}, & l \in \text{allowedk} \\ 0, & \text{其他} \end{cases} \quad (8)$$

然后在每次循环结束后，定义信息素更新规则如下：

$$\tau_{il}^j(t+1) = (1 - \rho)\tau_{il}^j(t) + \Delta\tau_{il}^j(t, t+1) \quad (9)$$

$$\Delta\tau_{il}^j(t, t+1) = \sum_{k=1}^m \Delta_{il(k)}^j(t, t+1) \quad (10)$$

$$\Delta\tau_{il(k)}^j(t, t+1) = \begin{cases} \frac{Q}{C_{jk}}, & \text{蚂蚁 } k \text{ 在船舶 } j \text{ 上确定的排序调度} \\ 0, & \text{否则} \end{cases} \quad (11)$$

算法步骤如下：

Step1：设  $I = \{1, 2, \dots, N\}$  为船舶集， $J_i = \{j | \bar{L}_j - L_i \geq 0 \text{ 且 } D_j - D_i \geq 0\}$ ,  $i = 1, 2, \dots, N$ ，其中  $J_i$  为满足船舶  $i$  吃水深度和长度的泊位集合；

Step2：从  $I$  中选一条船舶  $I$ ，选取泊位  $j'$ ，使得  $w_{2i} p_{ij'} = \min\{w_{2i} p_{ij}\}$ ，数量不唯一时任意选择；

Step3：将船舶  $i$  分配给泊位  $j'$ ,  $I = I - \{i\}$ ；

Step4：若  $I \neq \emptyset$ ，转 Step2；

Step5：初始化，针对每个泊位  $j$  ( $j=1, 2, \dots, M$ )，设置迭代次数  $t=0$ ，最大迭代次数  $t_{\max}$ ，信息素初始值  $\tau_0$ ，每个蚂蚁的禁忌表  $\text{tabu}$  和可访问节点表  $\text{allowed}$ ；

Step6： $t=t+1$ ；

Step7：对每个泊位  $j$  ( $j=1, 2, \dots, M$ )，将每个蚂蚁放置在若干个船舶上；

Step8：根据式 (8) 计算各待选节点的选择概率，按照概率分布选择下一个节点；

Step9：当所有蚂蚁走遍各自的路径，记录该次循环后的最好目标值函数值，否则转到 Step8；

Step10：如果  $t$  小于  $t+1$ ，调转至 Step6；

Step11：结束搜索，输出最好解。

### 3 计算实例

以威海港集团第二港区为例，改港区一共有 1#和 2#两个停泊位置，两个泊位限于通过能力、吊桥等资源的限制，同一时间段只能处理一艘船；1#位只能处理船长度不超过 80 m 的货船；2#为只能处理 80 m 到 95 m 之间的船长。由 2016 年 7 月 13 日到 7 月 16 日的靠、离港船舶情况如表 1 所示。

表 1 船舶到港数据

编号	船名	船舶到港时间	船舶离港时间	船长/m
1	锦乐 156	2016-7-13 13:45	2016-7-15 05:00	85.00
2	宝宏 2	2016-7-14 20:00	2016-7-16 20:00	77.70
3	新生生	2016-7-14 09:30	2016-7-16 06:00	61.84
4	生生 1 号	2016-7-14 20:00	2016-7-15 18:00	81.00
5	珍珠岛号	2016-7-15 08:00	2016-7-14 18:00	63.00
6	新锦乐	2016-7-14 17:00	2016-7-14 23:00	87.00
7	海润	2016-7-14 04:20	2016-7-14 15:00	90.00
8	金舵手	2016-7-14 10:00	2016-7-15 05:00	77.50

有上面数据可以发现,威海港码头采取的是简单的先到先服务算法,很容易推得,船舶 2,3,5,8 在 1# 位上进行作业,而船舶 1,4,6,7 在 2#位置上进行作业。现在利用改进蚁群算法进行确定,得到结果见表 2 所示。

表 2 实验结果比较

算法	1#位配置方案	2#位配置方案	在港总时间/min
先到先服务	3,2,8,5	1,7,6,4	63528
改进蚁群算法	5,2,8,3	1,6,4,7	30416

由此清晰可见,采用改进蚁群算法得到的结果比最初的先到先服务原则得到更理想的结果。

#### 4 结束语

本文讨论了集装箱码头的泊位问题,将一定时间内所有船舶在港总时间最短作为优化目标,建立泊位优化配置模型。利用改进蚁群算法确定了停靠在每个泊位上船舶的作业顺序。最终将算法在实际应用实例中进行比较,对比原始的先到先服务算法,新算法的时间得到大幅提高。

参考文献:

- [1] 张新宇,林俊,郭子坚,等.基于模拟退火多种群遗传算法的港口船舶调度优化[J].中国航海,2016,39(01):21
- [2] 张新宇,李瑞杰,林俊,等.双向通航港口船舶调度优化模型与算法[J].中国航海,2018,41(02):36-40
- [3] 晏榆洋,帅培.基于蚁群算法的港口船舶调度优化问题研究[J].船舶科学技术,2019,1(3A):49-51
- [4] 韩晓龙,丁以中.集装箱港口泊位配置优化[J].系统工程理论方法应用,2006,15(03):275-278
- [5] 靳凯文,李春葆,秦前清.基于蚁群算法的最短路径搜索方法研究[J].公路交通科技,2006,23(03):128-130
- [6] 董升伟,贾元华,赵雪静.基于改进蚁群算法的集装箱装卸顺序优化研究[J].山东科学,2012,25(05):23-25
- [7] 肖艳秋,焦建强,乔东平,等.蚁群算法的基本原理及应用综述[J].轻工科技,2018,34(03):69-72
- [8] Dorigo M,Maniezzo V,Colormi A.Positive Feedback as a Search Strategy[J].Technical Report,1991:91-016
- [9] Maniezzo V,Gambardella L M, Luigi F D. Ant colony optimization. New Optimization Techniques in Engineering[M]. Springer Berlin:Heidelberg,2004:422-423

Application of the berth optimization of container terminal with the modified ant colony algorithm

JIE Rui-hong<sup>1</sup>, XING Hong-guang<sup>1</sup>, ZHANG Hong-jun<sup>2</sup>

(1.College of Mechanical and Electrical Engineering, Qiqihar University, Heilongjiang Qiqihar 161006, China;

2. School of Mechatronic Engineering,Lingnan Normal University, Guangdong Zhanjiang 524048,China)

**Abstract:**This paper discusses the application of ant colony algorithm(ACO) in berth optimization of wharf. Taking the shortest total time of all ships in port as the optimization objective, the berth optimal allocation model is established. In the solution, the modified ACO is used to determine the operation sequence of ships berthing in each berth. Subsequently, the modified polymorphic ACO is proved to have a greater improvement, which compared with the original first-come-first-serve algorithm in the stopping time.

**Key words:** port logistics; quay berth; ant colony algorithm; optimization