

集装箱船舶装箱排序问题的两阶段算法

李俊, 张煜*, 计三有, 田维

(武汉理工大学 物流工程学院, 武汉 430063)

摘要: 集装箱船舶装箱排序问题综合集装箱装船顺序与船舶贝内排箱, 是一类典型的组合优化问题。考虑堆场发箱顺序和船舶配载之间的关联, 构建集装箱船舶装箱排序问题的数学模型。针对构建的优化模型, 鉴于已有方法难以直接对其进行求解, 基于两阶段分层求解思想设计 SWO-HES 两阶段算法。算法第 1 阶段利用吱呀轮算法(SWO)大邻域导向式搜索的特点优化集装箱装船顺序问题; 第 2 阶段将 SWO 的结果作为输入, 基于启发式规则和演化策略算法(ES)构造混合演化策略算法(HES)优化船舶贝内排箱问题。最后, 通过不同规模算例的研究, 将 SWO-HES 与常见智能算法、基于实际装船规则的启发式算法进行对比分析, 验证模型与算法的有效性。

关键词: 水路运输; 装箱排序; 两阶段算法; 集装箱码头; 吱呀轮算法; 演化策略算法

Two-phase Algorithm for Containership Sequencing and Bin Packing Problem

LI Jun, ZHANG Yu, JI San-you, TIAN Wei

(School of Logistics Engineering, Wuhan University of Technology, Wuhan 430063, China)

Abstract: Sequencing and bin packing problem (SBPP) is a kind of typical combinatorial optimization problem including containership loading sequence problem (CLSP) and slot plan problem (SPP). The mathematical model of SBPP is established considering the relationship between CLSP and SPP. As the existing methods are hardly used to optimize SBPP, SWO-HES two-phase algorithm is designed based on the two-phase hierarchical solving method. CLSP is optimized based on the large neighborhood guiding search characteristic of squeaky wheel algorithm (SWO). Then the heuristic evolutionary strategy algorithm (HES) is designed based on evolutionary strategy (ES) and heuristic rules to optimize SPP with the result of SWO as input. Finally, through the research of different scale examples, SWO-HES is contrasted with common optimization algorithm and heuristic algorithm based on actual loading rule to verify the model and algorithm.

Keywords: waterway transportation; sequencing and bin packing; two-phase algorithm; container terminal; squeaky wheel algorithm; evolutionary strategy algorithm

0 引言

船舶大型化和全航线配载对集装箱动态配载

决策提出了更高要求, 需要综合考虑堆场发箱与船舶配载之间的关联。不同的堆场集装箱堆存状态

收稿日期: 2016-07-12

修回日期: 2016-08-25

录用日期: 2016-09-13

基金项目: 国家自然科学基金/ National Natural Science Foundation of China(71372202)。

作者简介: 李俊(1989-), 男, 湖北钟祥人, 博士生。

*通信作者: sanli@whut.edu.cn

及装船发箱顺序,会直接导致船舶贝内阻塞箱的形成,进而引起后续港口卸载时的翻倒箱,对全航线配载产生不良影响.在现实作业中,翻倒箱不仅会大大增加码头额外操作成本,也会增加船舶在港作业时间.

目前,许多学者分别对集装箱装船顺序问题(Containership Loading Sequence Problem, CLSP)^[1-4]或船舶贝内排箱问题(Slot Plan Problem, SPP)^[5-8]展开研究.针对CLSP,目前的研究大都是基于已知的堆场堆存状态和集装箱船配载图,以最小倒箱量为目标设计启发式算法^[1]、遗传算法^[2-3]进行求解.也有学者同时考虑堆场翻箱,设计粒子群算法^[4]进行优化.针对SPP,DELGADO等^[5]构建了约束规划和整数规划模型,利用Gecode和CPLEX进行求解.MONACO等^[6]以最小化装船时间和翻箱为目标设计两阶段启发式算法优化.DING等^[7]设计启发式算法优化翻倒箱数量.祝慧灵等^[8]研究全航线多目标配载,基于遗传算法求解.

综上,目前已有相关研究大多都从单一角度出发,在研究SPP时未考虑CLSP,与码头的实际配载情况存在一定差异,决策结果不能满足实际需求.为此,本文综合考虑CLSP与SPP,提出集装箱船舶装箱排序问题(Sequencing and Bin Packing Problem, SBPP),其优化目标及约束条件非线性的

复杂特性使得计算复杂度与单一问题相比大幅上升,已有的方法很难直接优化,在求解效率和结果方面都不太理想.

因此,本文提出SWO-HES两阶段算法来求解SBPP.第1阶段:吱呀轮算法(Squeaky Wheel Algorithm, SWO)已被很多研究证明在求解排序调度类问题时具有良好性能^[9-10].第2阶段:演化策略算法(Evolutionary Strategy Algorithm, ES)由于直接在解空间上操作,且同时有多个解并行搜索,因而比常见智能算法具有更好的寻优性能.为了保证寻优时解的可行性,同时提高算法搜索效率,加入启发式规则构造启发式演化策略算法(Heuristic Evolutionary Strategy Algorithm, HES).结果表明,该算法与常见的智能算法及基于实际装船规则的启发式算法相比,在优化SBPP问题时性能更优.

1 SBPP建模分析

SBPP需要确定堆场集装箱的装船发箱顺序,以及集装箱在船舶贝(Bay)内的具体箱位,要满足船舶积载强度、横倾力矩、集装箱重不压轻及各目的港集装箱远不压近等约束,目标是 minimized 堆场翻箱次数、船舶贝内翻倒箱数量及横倾力矩.SBPP问题描述如图1所示.

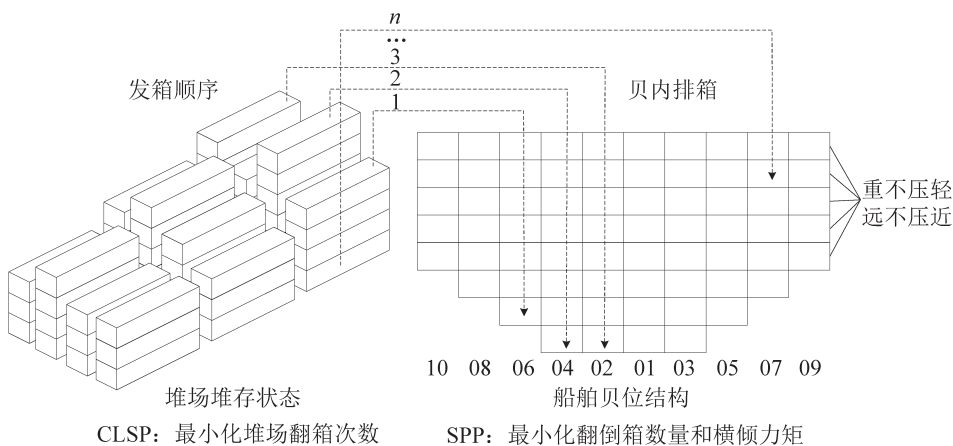


图1 SBPP问题描述

Fig.1 Description of SBPP

1.1 假设条件与现实约束

为了便于模型的建立与求解,考虑现实约束,做出以下假设:

(1) 仅考虑同一尺寸的普通箱;

(2) 不考虑舱盖板对制定船舶配载计划的影响;

(3) 在现实作业中,堆场会按照目的港—箱型—重量等级等对集装箱进行分列集中堆放以方便后续装船作业.

1.2 模型构建

(1) 基本参数.

C 是堆场贝位内待装船集装箱集合; D 是待装船集装箱的目的港集合; k 是待装船集装箱的编号($k \in C$); d_k 是集装箱 k 的目的港($d_k \in D$); w_k 是集装箱 k 的重量(t); θ_k 是集装箱 k 的重量等级; $\pi(k)$ 是堆场中集装箱 k 上方的集装箱集合; k' 是堆场中集装箱 k 上方的集装箱($k' \in \pi(k)$); S 是船舶贝内的箱位集合; LS 、 RS 是船舶贝内左边与右边的箱位集合($S=LS \cup RS$); SN 是船舶贝内列集合; ST 是船舶贝内层集合; $s=(i,j)$ 是船舶贝内箱位,其中 i 表示列, j 表示层($s \in S, i \in SN, j \in ST$); $s'=(i,j+1)$ 是船舶贝内箱位 s 正下方相邻的箱位($s' \in S, i \in SN, j+1 \in ST$); Q_i

是船舶贝内第 i 列箱位集合($Q_i \subset S$); SW_i 是船舶贝内第 i 列的积载强度(t); α 、 β 是集装箱宽度(m)及船舶贝内两相邻列集装箱之间间隙(m).

(2) 决策变量.

$x_{kk'}$ 表示集装箱 k 装船时,集装箱 k' 是否需要翻箱,为0-1决策变量; a_k 表示堆场集装箱 k 的装船顺序,越小表示越早装船; y_{ks} 表示堆场集装箱 k 是否占用船舶贝内的箱位 s ,为0-1决策变量; z_{ks} 表示船舶贝内占用箱位 s 的集装箱是否为其下方箱位 s 上集装箱的阻塞箱,为0-1决策变量.

优化模型:

$$\min f_1 = \min \left(\sum_{k \in C} \sum_{k' \in \pi(k)} x_{kk'} \right) \quad (1)$$

$$\min f_2 = \min \left(\sum_{k \in C} \sum_{s \in S} y_{ks} z_{ks} \right) \quad (2)$$

$$\min f_3 = \min \left| \sum_{k \in C} \sum_{s \in LS} w_k y_{ks} \left(\frac{\|SN\| - 2i + 1}{2} \right) (\alpha + \beta) - \sum_{k \in C} \sum_{s \in RS} w_k y_{ks} \left(\frac{2i - \|SN\| - 1}{2} \right) (\alpha + \beta) \right| \quad (3)$$

约束条件:

$$\sum_{k \in C} y_{ks} \leq 1, \forall s \in S \quad (4)$$

$$\sum_{s \in S} y_{ks} = 1, \forall k \in C \quad (5)$$

$$\sum_{k \in C} y_{ks} \geq \sum_{k' \in C} y_{ks'}, \forall s \in S \quad (6)$$

$$\sum_{k \in C} y_{ks} \theta_k \geq \sum_{k' \in C} y_{ks'} \theta_{k'}, \forall s \in S \quad (7)$$

$$\sum_{s \in Q_i} \sum_{k \in C} y_{ks} w_k \leq SW_i, \forall i \in SN \quad (8)$$

$$\sum_{k \in C} y_{ks} (\|C\| - a_k) \geq \sum_{k' \in C} y_{ks'} (\|C\| - a_{k'}), \forall s \in S \quad (9)$$

$$0 \leq x_{kk'} + \frac{a_k - a_{k'}}{\|C\|} < 1, \forall k \in C; \forall k' \in \pi(k) \quad (10)$$

$$0 \leq z_{ks} + \frac{v_s}{\|D\|} < 1, \forall s \in S; v_s = \sum_{k \in C} y_{ks} d_k - \sum_{k' \in C} y_{ks'} d_{k'} \quad (11)$$

$$\frac{a_k}{\|C\|} \leq \sum_s y_{ks}, \forall k \in C \quad (12)$$

式中: $\|C\|$ 、 $\|D\|$ 为集合 C 、 D 的势,即元素个数.

式(1)~式(3)为目标函数:式(1)表示最小化堆场翻箱次数;式(2)和式(3)表示最小化船舶贝内翻倒箱数量及横倾力矩.式(4)~式(12)为约束条件:式(4)表示船舶贝内一个箱位至多被一个集装箱占用;式(5)表示一个集装箱必须占用船舶贝内一个箱位;式(6)表示船舶贝内的集装箱不悬空;式(7)表示船舶贝内集装箱“重不压轻”;式(8)表示船舶贝

内单列积载强度约束;式(9)表示船舶贝内先装船集装箱不可能位于后装船集装箱的下方;式(10)表示决策变量 $x_{kk'}$ 的取值限制,当 $a_k - a_{k'} \geq 0$ 时,表示堆场集装箱 k 的装船顺序晚于集装箱 k' ,此时集装箱 k' 不需要翻箱,强制 $x_{kk'}$ 为0,反之强制 $x_{kk'}$ 为1;式(11)表示决策变量 z_{ks} 的取值限制,当 $v_s \geq 0$ 时,表示占用箱位 s 的集装箱不是占用箱位 s 集装箱的阻塞箱,强制 z_{ks} 为0,反之强制 z_{ks} 为1;式(12)保证任意一个堆场集装箱 k 在装船时都必定能分配一个船舶贝内箱位.

2 两阶段算法设计

鉴于单一的SPP问题都是NP难题,设计SWO-HES两阶段算法求解SBPP.两阶段算法设计如图2所示.

2.1 SWO设计

SWO的构造器根据集装箱目的港及重量等级信息利用贪婪准则得到一个初始解,即让目的港远且更重的集装箱排在序列中更靠前位置.

(1) 个体编码.

采用基于实数的多维个体编码方法.第一维为集装箱编号,其先后顺序代表集装箱装船顺序;第二维为集装箱在堆场贝内所处位置;第三维为集

装箱的责罚值.

(2) 分析器.

分析器用来确定每个集装箱的责罚值,包括初始化和更新两部分.责罚值直接决定集装箱的装

船优先级,责罚值大的优先装船.现实作业中,为了避免后续港口的翻倒箱,让目的港远且更重的集装箱优先装船.假设集装箱 k 的初始责罚值 $h_k(0)$ 为

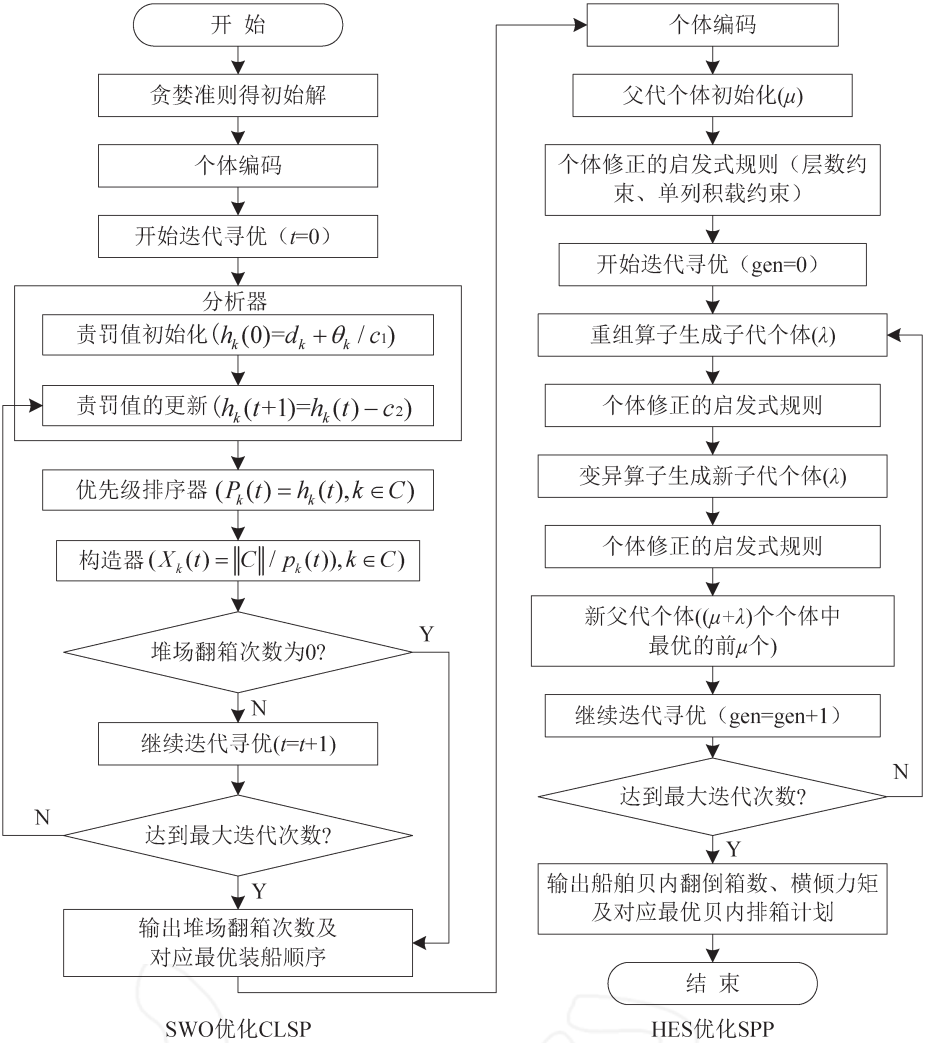


图2 两阶段算法设计
Fig. 2 Design of two-phase algorithm

$$h_k(0)=d_k+\theta_k/c_1,k\in C;c_1\text{为常数}\tag{13}$$

若某集装箱引起了堆场翻箱,则在下一次迭代时减小其责罚值.假设集装箱 k 在第 t 次迭代时引起堆场翻箱,此时的责罚值为 $h_k(t)$,则其在下一次迭代时,责罚值按式(14)更新.

$$h_k(t+1)=h_k(t)-c_2,k\in C;c_2\text{为常数}\tag{14}$$

(3) 优先级排序器.

优先级排序器依据责罚值来确定每个集装箱的装船优先级.假设集装箱 k 在第 t 次迭代时的优先级 $P_k(t)$ 为

$$P_k(t)=h_k(t),k\in C\tag{15}$$

(4) 构造器.

构造器根据每个集装箱的装船优先级得到新的装船顺序.假设集装箱 k 在第 t 次迭代时的装船顺序 $X_k(t)$ 为

$$X_k(t)=\|C\|/P_k(t),k\in C\tag{16}$$

2.2 HES设计

HES算法设计如图3所示.

(1) 个体编码.

采用基于实数的多维个体编码.第一、二维为

集装箱编号和责罚值,第三、四维为集装箱装船后所处列与层.其中第一、二维由SWO求解得到;第三维在船舶贝位列数约束范围内采用随机数生成;第四维根据第三维集装箱所在列出现的先后顺序依次确定其所在层数.

(2) 个体修正的启发式规则.

迭代寻优时,可能出现个体超出单列箱位数约束或积载强度约束变成非可行解.将超出单列箱位数(或积载强度)约束的集装箱调整到其他列,称作“落箱列”.“落箱列”需满足:

- ① 列未满,即该列还有空箱位;
- ② 加入“新”集装箱后,该列不会超出箱位层数(或积载强度)约束;
- ③ 加入的“新”集装箱不会成为阻塞箱.

若存在多个满足条件的列,则从船舶稳性的角度出发,选择剩余箱位最多的列,仍有多多个时,

任取一个.

个体修正主要对第三维操作,修正后需更新第四维编码,即集装箱所在层.这样可以保证依照既定装船顺序,自动将“新”集装箱调整到“落箱”列的合适位置,更好地避免其成为阻塞箱,提高算法的寻优效率.

(3) 重组算子.

采用基于三点交叉互换的重组算子,对个体的第三维进行操作.随机选择两个父代个体重组生成两个新的子代个体.重组后需对超出约束的子代个体进行修正.

(4) 变异算子.

采用个体内部基因两点互换的变异算子,对个体的第三维进行操作.设置一定的变异次数,对每个子代个体进行变异操作.变异后需对超出约束的子代个体进行修正.

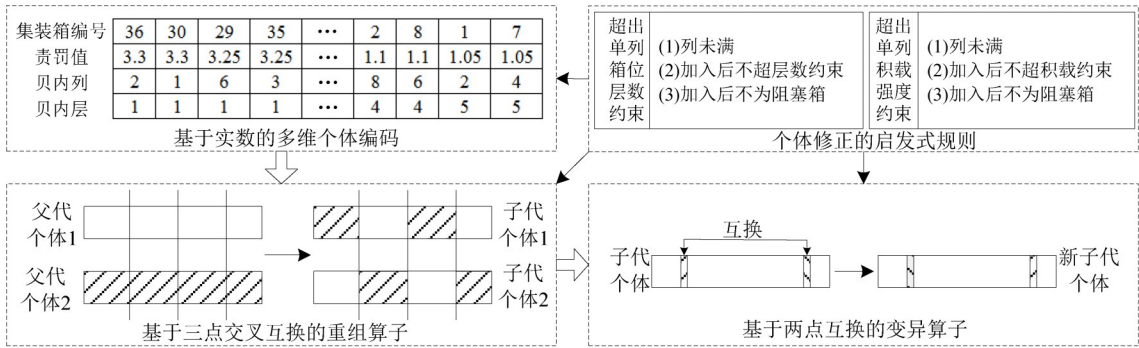


图3 HES算法设计

Fig. 3 Heuristic evolutionary strategy algorithm

3 算例研究

为了验证模型及算法的有效性,构建4组不同规模的16个算例进行分析,每个集装箱的重量在约束范围内随机生成,如表1所示.由于现实作业中,同一贝内出现混装时不同目的港的数目往往

不会太多.为了在考虑现实约束的同时,对算法的优化能力进行验证,算例中将目的港的最大数目设置为6个.为了保证船舶运输的经济性及算例求解的复杂性,所有算例中船舶贝内箱位利用率均在85%以上.

表1 算例数据

Table 1 Example data

| 算例编号 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---------|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|
| 堆场集装箱数 | 36 | 36 | 36 | 36 | 48 | 48 | 48 | 48 | 72 | 72 | 72 | 72 | 96 | 96 | 96 | 96 |
| 集装箱目的港数 | 2 | 3 | 4 | 6 | 2 | 3 | 4 | 6 | 2 | 3 | 4 | 6 | 2 | 3 | 4 | 6 |
| 船舶贝位箱位数 | 42 | 42 | 42 | 42 | 56 | 56 | 56 | 56 | 80 | 80 | 80 | 80 | 108 | 108 | 108 | 108 |

3.1 算法参数分析

(1) SWO责罚值分析.

责罚值设置是否合理将直接影响SWO性能,

对 c_1 、 c_2 的取值进行试验(表2).采用Matlab 2010b编程求解,SWO最大迭代次数设置为100,选取规模最大的算例16进行分析.每组设置下,连续运

行20次取平均值.

表2 c_1 、 c_2 设置分析
Table 2 c_1 、 c_2 set

| | | | |
|---------------|---------|---------|---------|
| c_1 、 c_2 | 10、0.05 | 10、0.10 | 10、0.15 |
| 翻箱数 | 0 | 0 | 0 |
| 求解时间/s | 0.006 | 0.006 | 0.006 |
| c_1 、 c_2 | 20、0.05 | 20、0.10 | 20、0.15 |
| 翻箱数 | 1 | 1 | 1 |
| 求解时间/s | 0.090 | 0.090 | 0.090 |
| c_1 、 c_2 | 30、0.05 | 30、0.10 | 30、0.15 |
| 翻箱数 | 0 | 0 | 0 |
| 求解时间/s | 0.008 | 0.009 | 0.009 |

从表2中可以看出,为了保证SWO算法的求解性能,将 c_1 、 c_2 设置为 $c_1=10$ 、 $c_2=0.05$, $c_1=10$ 、 $c_2=0.10$, $c_1=10$ 、 $c_2=0.15$ 均较为合理.选择将 c_1 、 c_2 设置为 $c_1=10$ 、 $c_2=0.05$.

(2) HES 参数设置.

HES 参数的设置将直接影响其优化性能,对父代及子代个体数(μ 、 λ)、变异算子的次数进行分析,通过不同设置(表3和表4)的对比试验来得出最优的设置.同样选取算例16进行分析,每组设置下,其他参数保持不变,连续运行20次取平均值.

表3 μ 、 λ 设置分析
Table 3 μ 、 λ set

| | | | | | | | | |
|------------------|---------|---------|---------|---------|---------|---------|---------|---------|
| (μ, λ) | (20,10) | (20,20) | (30,20) | (30,30) | (40,30) | (40,40) | (50,30) | (50,40) |
| 贝内翻倒箱 | 1.0 | 0.4 | 1.2 | 0.0 | 0.0 | 0.0 | 0.4 | 0.1 |
| 横倾力矩/(KN·m) | 137.6 | 260.1 | 5.5 | 8.2 | 8.2 | 5.5 | 24.6 | 2.7 |
| 求解时间/s | 39.5 | 81.4 | 70.7 | 109.7 | 123.7 | 164.2 | 121.1 | 165.2 |

表4 变异算子的次数设置分析
Table 4 Mutation operator times set

| | | | | | | | | |
|-------------|------|------|-------|-------|-------|-------|-------|-------|
| 变异算子的次数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 贝内翻倒箱 | 0.3 | 0.2 | 0.0 | 0.1 | 0.2 | 0.8 | 0.6 | 0.3 |
| 横倾力矩/(KN·m) | 0.0 | 2.7 | 8.2 | 5.5 | 2.7 | 0.0 | 0.0 | 0.0 |
| 求解时间/s | 63.9 | 91.9 | 109.7 | 132.0 | 156.0 | 166.8 | 207.5 | 251.1 |

从表3和表4中可看出,参数并非越大越好,合理的设置可以在保证算法稳定性的同时节约优化时间.综合来看,将父代及子代个体数均设为30,变异算子的次数设为3次较为合理.

3.2 算例分析

为了验证算法有效性,引入常见的智能算法及基于码头实际装船规则的启发式算法对比试验.采用惯性权重线性递减粒子群算法(PSO),种群大小设置为50,惯性权重由0.9线性递减至0.4,学习因子均为2.为了保证其求解性能,同样采用启发式规则进行改进.同时,对码头的实际装船规则进行改进,提出一种基于码头实际装船规则的启发式算法(Heuristic Algorithm Based on Actual Loading Rule, HA-ALR),该规则按照逐列装船作业.

为了在保证算法优化效果的同时提高求解效率,经多组设置多次试验求解后,HES和PSO最大迭代次数按算例规模不同分别设置,连续运

行20次取平均值.HA-ALR由于每次优化结果均一致,故每个算例优化一次.采用Matlab 2010b编程求解,结果如表5所示,其中 G 表示最大迭代次数, f_1 表示堆场翻箱次数, f_2 表示船舶贝内翻倒箱数量, f_3 表示船舶贝位横倾力矩(kN·m), T 表示求解时间(s).

不同算法对算例的求解结果对比如图4~图7所示.

从表5和图4~图7可以看出,由于堆场采用按目的港—箱型—重量等级等对集装箱进行分列集中堆放的方式,3种算法在优化堆场翻箱次数时均找到了最优解.但在其他方面却呈现出较大差异性:在算例规模较小时,3种算法均找到船舶贝内无翻倒箱的满意解,在优化横倾力矩方面,除算例2以外,SWO-HES均最优.随着算例规模增大,求解难度也随之加大,除SWO-HES外,其他两种算法均无法保证100%找到无翻倒箱的满意解.尤其

是PSO,无论在求解速度还是求解效果方面,都不太理想.而HA-ALR由于主要采用规则来实现,求解速度非常快,但求解效果却远不如SWO-HES.因此,SWO-HES在求解SBPP时更有效.

表 5 算法求解结果
Table 5 Results of algorithms

| 算例 编号 | SWO-HES | | | | | PSO | | | | | HA-ALR | | | |
|----------|---------|-------|-------|-------|--------|-------|-------|-------|-------|-------|--------|-------|-------|-------|
| | G | f_1 | f_2 | f_3 | T | G | f_1 | f_2 | f_3 | T | f_1 | f_2 | f_3 | T |
| 1 | 100 | 0 | 0 | 0.0 | 2.5* | 500 | 0 | 0.0 | 2.7 | 13.2 | 0 | 0 | 27.4 | 0.05 |
| 2 | 200 | 0 | 0 | 0.0 | 4.8 | 800 | 0 | 0.0 | 11.0 | 20.7 | 0 | 0 | 0.0 | 0.05* |
| 3 | 300 | 0 | 0 | 0.0 | 7.0* | 1 000 | 0 | 0.0 | 16.4 | 26.3 | 0 | 0 | 328.6 | 0.05 |
| 4 | 600 | 0 | 0 | 0.0 | 18.3* | 2 000 | 0 | 0.0 | 24.6 | 51.9 | 0 | 0 | 328.6 | 0.05 |
| 5 | 100 | 0 | 0 | 0.0 | 3.0* | 500 | 0 | 0.0 | 24.6 | 18.4 | 0 | 0 | 328.6 | 0.05 |
| 6 | 100 | 0 | 0 | 0.0 | 3.1* | 1 000 | 0 | 0.0 | 27.4 | 37.3 | 0 | 0 | 301.2 | 0.05 |
| 7 | 200 | 0 | 0 | 0.0 | 5.8* | 2 500 | 0 | 0.5 | 30.1 | 92.8 | 0 | 0 | 136.9 | 0.05 |
| 8 | 600 | 0 | 0 | 0.0 | 18.4* | 3 500 | 0 | 0.7 | 13.7 | 130.2 | 0 | 0 | 27.4 | 0.05 |
| 9 | 100 | 0 | 0 | 0.0 | 4.5* | 1 000 | 0 | 1.6 | 65.7 | 59.4 | 0 | 1 | 0.0 | 0.06 |
| 10 | 300 | 0 | 0 | 0.0 | 12.5* | 1 500 | 0 | 2.0 | 82.1 | 86.5 | 0 | 0 | 27.4 | 0.06 |
| 11 | 400 | 0 | 0 | 0.0 | 17.6* | 3 000 | 0 | 2.2 | 104.0 | 167.2 | 0 | 2 | 438.1 | 0.06 |
| 12 | 1 000 | 0 | 0 | 0.0 | 42.7* | 4 000 | 0 | 2.3 | 145.1 | 226.6 | 0 | 2 | 54.8 | 0.06 |
| 13 | 200 | 0 | 0 | 0.0 | 12.6* | 2 000 | 0 | 1.7 | 65.7 | 167.2 | 0 | 1 | 27.4 | 0.06 |
| 14 | 600 | 0 | 0 | 0.0 | 36.1* | 3 000 | 0 | 3.8 | 35.6 | 247.1 | 0 | 2 | 0.0 | 0.07 |
| 15 | 1 000 | 0 | 0 | 11.0 | 63.9* | 4 000 | 0 | 5.0 | 112.3 | 327.7 | 0 | 3 | 27.4 | 0.07 |
| 16 | 1 800 | 0 | 0 | 8.2 | 109.7* | 6 000 | 0 | 5.4 | 98.6 | 492.9 | 0 | 1 | 0.0 | 0.07 |

注:*表示该算法在求解算例时结果最优.

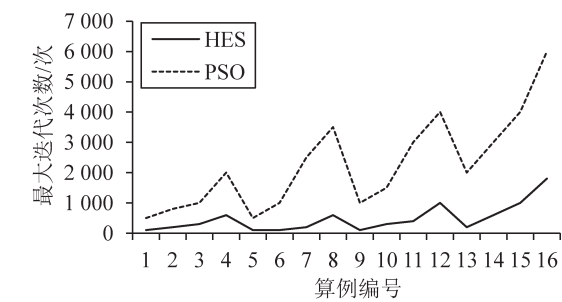


图 4 HES 与 PSO 最大迭代次数对比
Fig. 4 Iterations of HES and PSO

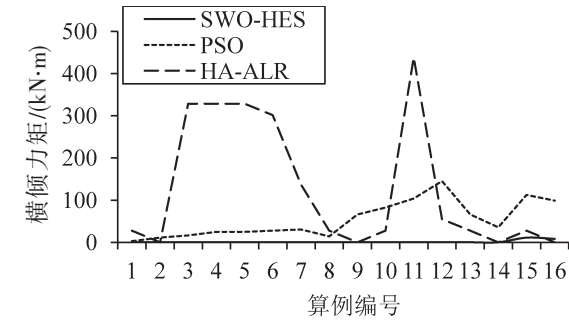


图 6 算法横倾力矩对比
Fig. 6 Heeling moments of algorithms

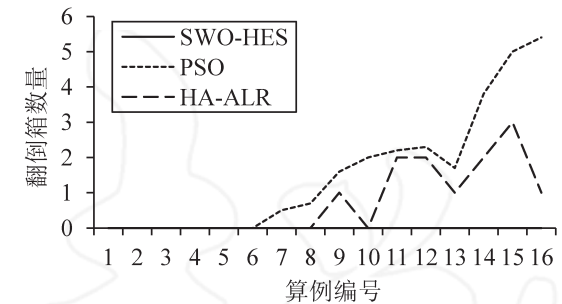


图 5 算法船舶贝内翻倒箱数量对比
Fig. 5 Turnover numbers of algorithms

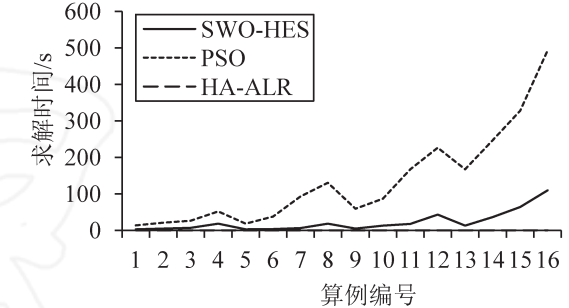


图 7 算法求解时间对比
Fig. 7 Solving time of algorithms

显然,对 CLSP 与 SPP 分别进行求解时, f_1 、 f_2 及 f_3 的理论下界值均为 0. SWO-HES 在求解 SBPP 时,算例 1~14 均达到理论下界,算例 15~16 中 f_1 、 f_2 达到理论下界, f_3 虽未达到理论下界,但也是满足约束的极小值,也是问题的满意解.因此,从求解结果与理论下界的对比中,也说明了 SWO-HES 求解 SBPP 的合理性.

4 结 论

本文从集装箱船舶动态配载的实际需求出发,提出 SBPP 问题,并构建了该问题的优化模型. SBPP 综合 CLSP 与 SPP,与单一问题相比更难求解,计算复杂度大幅上升,已有的求解方法难以直接对其进行优化.本文基于两阶段分层求解思想,设计 SWO-HES 两阶段算法优化 SBPP.通过不同规模算例研究,将 SWO-HES 与常见优化算法、基于实际装船规则的启发式算法等进行比对,验证 SWO-HES 两阶段算法在求解 SBPP 时的有效性.

参考文献:

- [1] 靳志宏,兰辉,边展,等. 基于现实约束的集装箱装船顺序优化[J]. 大连海事大学学报,2011,37(1):71-74. [JIN Z H, LAN H, BIAN Z, et al. Optimization on containership loading scheduling based on actual constraints[J]. Journal of Dalian Maritime University, 2011, 37(1): 71-74.]
- [2] JI M J, GUO W W, ZHU H L, et al. Optimization of loading sequence and rehandling strategy for multi-quay crane operations in container terminals[J]. Transportation Research Part E, 2015(80): 1-19.
- [3] 祝慧灵,计明军,郭文文,等. 基于配载计划的集装箱提箱顺序和倒箱策略优化[J]. 交通运输系统工程与信息,2016,16(2): 191-199. [ZHU H L, JI M J, GUO W W, et al. Optimization of the container retrieving sequence and rehandling strategy based on stowage plan[J]. Journal of Transportation Systems Engineering and Information Technology, 2016, 16(2): 191-199.]
- [4] 黎明,翟金刚. 集装箱装船顺序的多目标整数规划优化模型[J]. 计算机应用研究,2012,29(10): 3636-3639. [LI M, ZHAI J G. Multi-objective integer programming optimization model for loading sequence of container ship[J]. Application Research of Computers, 2012, 29(10): 3636-3639.]
- [5] DELGADO A, JENSEN R M, JANSTRUP K, et al. A constraint programming model for fast optimal stowage of container vessel bays[J]. European Journal of Operational Research, 2012(220): 251-261.
- [6] MONACO M F, SAMMARRA M, SORRENTINO G. The terminal-oriented ship stowage planning problems[J]. European Journal of Operational Research, 2014(239): 256-265.
- [7] DING D, CHOU M C. Stowage planning for container ships: A heuristic algorithm to reduce the number of shifts[J]. European Journal of Operational Research, 2015(246): 242-249.
- [8] 祝慧灵,计明军. 集装箱船舶全航线配载优化模型与改进遗传算法[J]. 交通运输工程学报,2014,14(5): 59-67. [ZHU H L, JI M J. Optimal model and improved genetic algorithm of containership stowage on full route[J]. Journal of Traffic and Transportation Engineering, 2014, 14(5): 59-67.]
- [9] ØVSTEBØ B O, HVATTUM L M, FAGERHOLT K. Optimization of stowage plans for RoRo ships[J]. Computers & Operations Research, 2011(38): 1425-1434.
- [10] UMANG N, BIERLAIRE M, VACCA I. Exact and heuristic methods to solve the berth allocation problem in bulk ports[J]. Transportation Research Part E, 2013(54): 14-31.