

基于网格搜索的船体不规则分段动态堆放方法

陆春霞*, 马少辉

(江苏科技大学 经济管理学院, 江苏 镇江 212003)

(*通信作者电子邮箱 chunxialu@hotmail.com)

摘要:针对船体分段建造后在堆场中的空间调度问题,提出基于网格搜索的分段动态空间调度算法。首先,利用改进的粒子群算法产生多个可行的分段堆放序列;然后,采用基于网格搜索的定位策略对堆放序列进行空间布局解码。在解码过程中,运用位图对场地及投影多边形进行信息描述,快速寻找多边形的最佳定位位置。考虑分段堆放问题的动态性的时空关联性,同时以场地平均利用率和需要挪动的场地内分段总数的综合加权作为评价函数,利用改进的粒子群算法对方案进行择优,得到近似最优解,实现了堆放方案的全局优化。通过对船厂实际生产数据的实证分析以及堆放算法间的对比分析,结果证明,所提算法在综合评价场地利用率、移动分段数和运算效率的条件下是最优的。

关键词:动态堆放;网格搜索;不规则分段;粒子群算法;位图

中图分类号: TP181 **文献标志码:** A

Approach on irregular block dynamic stacking in shipbuilding based on grid technology

LU Chunxia*, MA Shaohui

(School of Economics and Management, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003, China)

Abstract: In order to solve the problem of spatial scheduling of irregular blocks in stacking field after they were built, a dynamic stacking algorithm based on grid searching was proposed. An improved Particle Swarm Optimization (PSO) algorithm was used to determine the optimal processing sequence, and the locations of blocks were determined in spatial layout decoding by a dynamic location strategy based on grid technology. In the decoding process, bitmap was used to describe the information of yard and polygon and find the best location for every block. The space utilization and the quantity of blocks that need moving were used as the evaluation function, which fully considered the dynamic nature and the correlation between time and space in a stacking problem. Every particle in population was a stacking sequence, and the optimal solution could be found in the process of evolution by the improved PSO. Finally, the results of an actual data of a shipyard and the comparative analysis with other stacking algorithms show that the proposed algorithm is the best when comprehensively considering the space utilization, the number of movements of other blocks and efficiency of the algorithm.

Key words: dynamic stacking; grid search; irregular blocking; Particle Swarm Optimization (PSO); bitmap

0 引言

一艘船通常由上百个分段组成,且大多数为不规则分段。分段加工完成之后不一定直接进入船坞进行大合拢,需要在场地中进行堆置等待。不规则分段在场地堆放时,不仅要满足时空资源的限制,还需满足传统资源的限制,不能超出船厂中其他运输、加工等设备的负荷,称这类问题为场地内分段的堆放问题。

目前,大部分船厂中分段的堆放管理基本处于无序状态,单纯地根据人为经验和场地现状安排分段的堆放位置,最终使得分段堆放顺序混乱,在船体合拢时需要花费大量的人力和物力查找分段;且当需求时间较早的分段放置在场地中间时还需要挪动其外围的分段,这就大大增加了场地平板推车的负荷量,降低了场地的周转效率^[1]。因此,如何解决场地内分段的堆放问题,提高场地利用率且降低场地内运输设备的负荷量,成为船厂急需解决的重大问题之一。

针对分段建造过程中的时空资源约束限制问题,即空间调度问题,已有许多学者进行了相应研究。早期研究主要基于启发式规则,例如,文献[2]最早对船舶分段时空计划进行

研究,提出了基于船体不规则多边形分段的启发式规则,如最大剩余空间利用规则、最大可用矩形空间规则、初始定位规则、边规则及混合规则等;文献[3]提出了分批加工的一种启发式搜索空间调度方法,解决了生产过程中的空间调度问题,缩短了产品制造的工期;文献[4]论述了启发式算法的重要性,该算法是解决资源约束类型空间调度问题的有效方法之一;文献[5]以最小化最大工作量为目标,提出了一种资源适应度的启发式空间调度策略。启发式规则有简单、易于实现、计算复杂度低等优点,但往往对所得解的次优性无法进行评估。后期的研究主要是利用数学规划法和智能优化算法解决资源受限条件下的空间调度问题。例如:文献[6]利用线性规划将开工时间约束和顺序约束与空间调度结合;文献[7]以平均时空利用率为目标,构建了考虑加工优先顺序、交货期等因素的非线性规划模型。但数学规划方法只考虑模块间的相对时空位置,而且随着分段数量的增多,模型的方程数量呈指数增长,因此只能利用该方法求解小规模问题。在智能算法方面,文献[8]在对场地空间进行离散化、分段形状矩形化等简化处理后,用遗传算法进行了求解;文献[9]以空间利用率最大化为目标,应用遗传算法和最左最下原则,提出了一

收稿日期:2012-08-13;修回日期:2012-10-17。 基金项目:国家自然科学基金资助项目(70871057,71171100)。

作者简介:陆春霞(1987-),女,山东聊城人,硕士研究生,主要研究方向:决策支持系统; 马少辉(1972-),男,河北承德人,副教授,博士,主要研究方向:决策支持系统。

种船舶分段的空间调度方法;文献[10]按照 Bottom-Left-Fill 原则,提出了一种船舶分段自动布置算法,并应用元启发式方法提高了分段布置的准确性和效率;文献[11]针对矩形分段基于粒子群算法安排加工顺序、利用“右、前、上”启发策略安排空间位置。

但在现有的研究中,只考虑了以时间和空间为总体约束条件的空间调度问题,未考虑在各组立阶段找寻分段的容易性及从场地中移入、移出分段的复杂度,即忽略了分段加工完之后在场地中的堆放问题,增加了场地内平板推车的负荷,降低了生产效率。

本文针对船舶建造过程中分段堆场负荷加重、造船效率较低的瓶颈问题,将改进的粒子群算法与基于网格搜索的定位策略相结合,设计出一种船体分段的动态堆放系统,并做出了以下两点改进:1)利用位图和矩阵对场地及投影多边形进行描述,并得到了分段的网格搜索定位策略;2)在粒子群算法中加入了模拟退火算法的思想,同时引入遗传算法中的选择算子及变异算子,避免了局部最优情况的发生。

1 问题描述

船体不规则分段的堆放问题,即需在满足时间与空间双重约束条件下,通过时间规划和空间布局规划确定所有船舶分段的堆放信息,包括移入、移出场地的时间和分段的布局,实现时间和空间的综合优化指标。时间规划确定所有分段移入、移出场地的时间;空间布局规划确定所有分段在堆放场地上的具体放置位置及分段移入、移出场地时需要挪动的分段个数。

在本文中,只考虑单一场地内分段的堆放问题,提出的方法可以推广应用到多场地问题中。在优化系统运行时,输入信息主要包括工作场地信息:长(L)和宽(W);分段信息:分段数量 N 、最早进入场地时间(El)、最晚进入场地时间(Lt)、场地内堆放时间(DT)、第 i 个分段的投影多边形的顶点个数 g_i 及其顶点坐标向量 V_i 。

以最大化场地平均利用率(U)及最小化堆放过程中需要移动的分段数(YD)为综合优化目标,利用所设计的优化系统对所有堆放方案进行择优计算。最后,输出最优堆放方案的相关信息:分段的堆放序列($List$)、实际进入场地时间(ST)、第 i 个分段的投影多边形堆放过程中的顶点坐标 V_i' 、场地平均利用率(U)、堆放过程中需要移动的分段数(YD)、分段进出场地时选择的场地边界(BJ)。

2 动态堆放算法

2.1 改进的粒子群算法

粒子群算法(Particle Swarm Optimization, PSO)是在鸟群捕食的启发下提出的一种群智能优化算法^[12-13]。PSO的每一个“粒子”都是优化问题的一个解。PSO初始化为一群随机粒子,通过自身经验和社会经验来不断更新自己,最终找到最优解。

2.1.1 粒子群的编码

设种群的粒子数量为 M ,每个粒子均代表一个堆放顺序。在建立初始粒子群时,对 N 个分段以十进制数字依次进行编码 $\{1, 2, \dots, N\}$,以 El 作为参考产生初始调度序列,即:对于 El 相同的分段,先随机产生初始位移(x)及速度(v), x 与 v 均是一维的,且均在 $[0, 1]$ 范围内,然后对初始位移进行升序排序,则对应的分段会产生一个堆放序列;对于 El 不同的分段,则根据 El 先后次序排序。由于分段堆放序列决定了其进入空

间布局过程的先后,不同的堆放序列将直接影响分段在场地上的布局情况,形成不同大小和形状的空域,从而影响分段堆放的总体目标。

2.1.2 粒子的进化更新

粒子通过位移及速度的变化不断更新。粒子的速度位移更新公式如下:

$$v_k^{t+1} = \omega * v_k^t + c_1 * (Pb_k - x_k^t) + c_2 * (Gb - x_k^t) \quad (1)$$

$$x_k^{t+1} = x_k^t + v_k^{t+1} \quad (2)$$

其中:下标 k 代表第 k 个粒子,上标 t 代表迭代代数,如 x_k^t 和 v_k^t 分别是第 k 个粒子在第 t 次迭代中的位移和速度; ω 为惯性权重; c_1 和 c_2 是学习因子; Gb 是迭代过程中整个粒子群的最优位置; Pb_k 是粒子 k 在迭代过程中的最优位置。

在迭代过程中,为避免粒子飞出最优解的范围造成发散而得不到最优解,对基本粒子群算法做出以下几点改进:

1) 将粒子的速度限制在 $[-1, 1]$ 内,即第 k 个粒子在第 t 次迭代中速度满足

$$v_k^t = \begin{cases} 1, & v_k^t > 1 \\ -1, & v_k^t < -1 \end{cases} \quad (3)$$

2) 利用模拟退火算法的思想,令惯性权重 ω 的值随着迭代次数的增加线性减少。

3) 粒子更新时,引入遗传算法中选择算子及变异算子的思想。首先,直接将第 t 代粒子群 $G(t)$ 中评价函数值最高的10%个粒子复制到第 $t+1$ 代粒子群 $G(t+1)$ 中;然后,对于 $G(t)$ 中剩余的粒子,令变异的粒子个数 $Bg = pb * (M * 90\%)$,并利用式(1)、(2)更新随机选定的 Bg 个粒子位移及速度,其中, pb 也利用退火算法的思想,随着迭代次数的增加线性减少。对于选定的 Bg 个粒子, El 相同的分段,对其位移升序排列得到分段的堆放顺序;而 El 不同的分段,则按照 El 依次排序,最终得到 $G(t+1)$ 中的新粒子。

2.2 评价函数

根据生产实际要求,需要实现以下目标:

- 1) 在分段堆放过程中,场地平均利用率 U 越大越好;
 - 2) 分段移入(或移出)场地时,为了降低运输设备的负荷量,需尽量减少移动的分段数量,即 Y_{λ} 与 Y_{μ} 均越小越好。
- 将以上目标的综合加权作为判定粒子优劣的评价函数,即

$$f = k_1 \times \frac{1}{Y_{\lambda} + Y_{\mu}} + k_2 \times U + \sigma \quad (4)$$

其中: f 为评价函数; k_1 和 k_2 为权重因子; Y_{λ} 与 Y_{μ} 为分段移入、移出场地时需挪动的其他分段数量; σ 为调节系数,以避免一些个体过早淘汰。

2.3 空间定位算法与 PSO 解码

解码的主要目的是将可行分段堆放序列还原成场地内的分段布局图形及移动复杂度,即对所生成的可行分段堆放序列,按照改进的粒子群算法和基于网格搜索的定位策略依次将分段定位于堆放场地内部,且计算出在堆放布局过程中需要挪动的其他分段数量。利用基于网格搜索的空间定位策略对分段进行布局的具体过程将在第3章作详细说明。

对下一个待排分段 D ,判断其投影多边形 D' 的面积 $s_{D'}$ 是否小于矩形场地中剩余的空闲场地面积 $s_{\text{空}}$ 。若 $s_{D'} \leq s_{\text{空}}$,利用基于网格搜索的定位策略对该多边形 D' 进行定位,定位成功,记录其此时的顶点坐标;否则,将所有待排分段均推迟到次日继续定位。

主要流程包含以下步骤:

- 1) 在 T 天时,假设待解码的 n 个分段为 $H = \{h_1, h_2, \dots,$

$h_n\}$,其中 h_i 代表某一分段,投影凸多边形集合为 $H' = \{h_1', h_2', \dots, h_n'\}$,多边形面积为 $S = \{s_1, s_2, s_3, \dots, s_n\}$,设 $z = 1$ 。

2) 更新此时在场分段集合 $J = \{J_{T1}, J_{T2}, \dots, J_{Tm}\}$,投影凸多边形为 $J' = \{J_{T1}', J_{T2}', \dots, J_{Tm}'\}$,其中,面积为 $S' = \{s_1', s_2', \dots, s_m'\}$ 。

3) 计算空闲场地的面积 $s_{\text{空}} = L * W - \sum_{i=1}^m s_c'$,其中 $c \in [1, m]$ 。

4) 判断 s_z 是否小于 $s_{\text{空}}$,若 $s_z \leq s_{\text{空}}$,则利用基于网格搜索的定位策略对多边形 h_z' 进行定位;否则,将 h_z 及其以后的分段推迟, $T = T + 1$,转向2)。

5) 若多边形 h_z' 定位成功,则记录此时其顶点的坐标和实际的人场时间,并统计需要移动其他分段的数量, $z = z + 1$;若定位失败,则将 h_z 及其以后的分段推迟, $T = T + 1$ 。

6) 若 $z = n$,所有分段定位完成,结束;否则转向2) 继续定位。

3 基于网格搜索的空间定位策略

对于一个分段堆放序列,需对所有分段进行空间布局操作,即确定分段的空间位置(可简化为确定其投影多边形的顶点坐标)。在定位过程中,所有分段的投影多边形需要满足以下两个条件:

1) 第 T 天,场地内任意两个分段的投影多边形不能相互重叠;

2) 所有的投影多边形必须完全放置在场地内部。

为满足上述要求,本文对文献[14]提出的位图法进行改进,提出网格搜索定位策略对投影多边形进行布局。在对投影多边形 A 布局时,以多边形 A 的重心 G_A 作为参考点。由于不同角度下多边形的重心不尽相同,设定重心 G_A 为固定点,以

30° 为旋转间隔,对多边形 A 进行 $360/30 = 12$ 次场地定位,记录定位成功时 A 的旋转角度及多边形重心 G_A' ,可从中得到一个 G_A' 最低时的旋转角度,并记最低 G_A' 为多边形 A 的参考点的最终定位位置。若12次均定位失败,则表示分段的空间布局失败,需要推迟堆放。

3.1 场地及投影多边形的位图表示

位图(点阵图)是对平面的离散化描述,构成位图的最小单位是像素,像素包括位置和颜色等信息,位图即由像素阵列的排列来实现其显示效果。位图对图形的描述过程中会出现一定的不完整,但在误差允许的范围内,利用位图可以较快地描述图形的几何信息。

借助位图,将场地及投影多边形划分成较密集的点阵。在用位图描述的过程中,先选定像素的单位($x_{\text{unit}} \times y_{\text{unit}}$),如图1所示:以 1×1 为像素单位,对于长为 L 、宽为 W 的场地,可利用 $W * L$ 个像素来描述,记为 $W * L$ 的零矩阵 $O_{W * L}$,矩阵中对应的元素表示单元格的占用状态,当场地中的单元格未被占用的时候,标记为0,可放置多边形;否则标记为1,多边形需寻找其他位置进行放置^[14]。

对于投影多边形 A ,将其放置在场地的左下角位置,其顶点信息为 $\begin{pmatrix} 1 & 3 & 4 & 2 & 0 \\ 0 & 0 & 2 & 3 & 2 \end{pmatrix}$,其最大 x 坐标(X_{max})为4,最大 y 坐标(Y_{max})为3,此时场地(L 为10, W 为8)的位图信息如图1(a)(b)所示。

利用位图描述投影多边形,相当于对多边形进行凸多边形包络,因此,场地中的像素单位越小,即场地划分越紧密,多边形定位时占用的像素数量越接近其实际占用空间。当像素单位为 0.5×0.5 时,投影多边形 A 在场地地上占用的像素如图1(c)所示。



图1 场地及投影多边形的位图信息

当像素单位为 1×1 时,投影多边形 A 在场地中的初始位置可用 8×10 的矩阵

$$G_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}_{8 \times 10}$$

表示,可简化为 $3 \times \frac{\lceil Y_{\text{max}} \rceil}{y_{\text{unit}}}$ 的矩阵 $J_1 = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 1 \\ 4 & 4 & 4 \end{pmatrix}$,其中第 r

($r \in (1, 2, 3)$)列元素依次表示:占用像素的行标号、该行占用像素的起始列、总占用的像素数量。利用后者标记方式,可快速得出多边形的大致位置,在多边形的放置及移出过程中可大量缩短寻找时间,提高了实际生产的调运效率。

当像素单位为 0.5×0.5 时,投影多边形 A 在场地中的初

始位置可用 16×20 的矩阵

$$G_{0.5} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}_{16 \times 20}$$

表示,简化为 $3 \times \frac{3}{0.5}$ 的矩阵 $J_{0.5} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 2 & 1 & 1 & 1 & 3 \\ 6 & 6 & 8 & 8 & 8 & 4 \end{pmatrix}$ 。当

划分越细时,利用简化矩阵表示场地内的投影多边形的优势越明显,但寻找位置的复杂度也会随之提高。

3.2 基于网格搜索的空间定位策略

在利用基于网格搜索的空间布局策略对分段定位时,按照“靠左、靠下”规则寻找其最佳定位位置,即多边形重心的 x 坐标最小的位置作为最佳定位位置,若 x 坐标最小点的个数不少于2个,则这些点中 y 坐标较小的位置点作为最佳定位位置。

在寻找待排分段的空间位置时,先将其投影多边形放置

在场地的左下角位置。当场地中没有分段放置时,投影多边形不需要移动,左下角位置即为该多边形的最佳定位位置;否则,可利用位图及矩阵对场地及场地内的分段描述,并判断该投影多边形与场地中已放置分段的投影多边形之间是否发生重叠。场地内已有多边形情况及待排多边形 A 在场地上的位置,如图2(a)、(b)所示。将多边形 A 放置在场地上时,利用位图及矩阵表示场地的变化情况,如图2(c)、(d)所示。

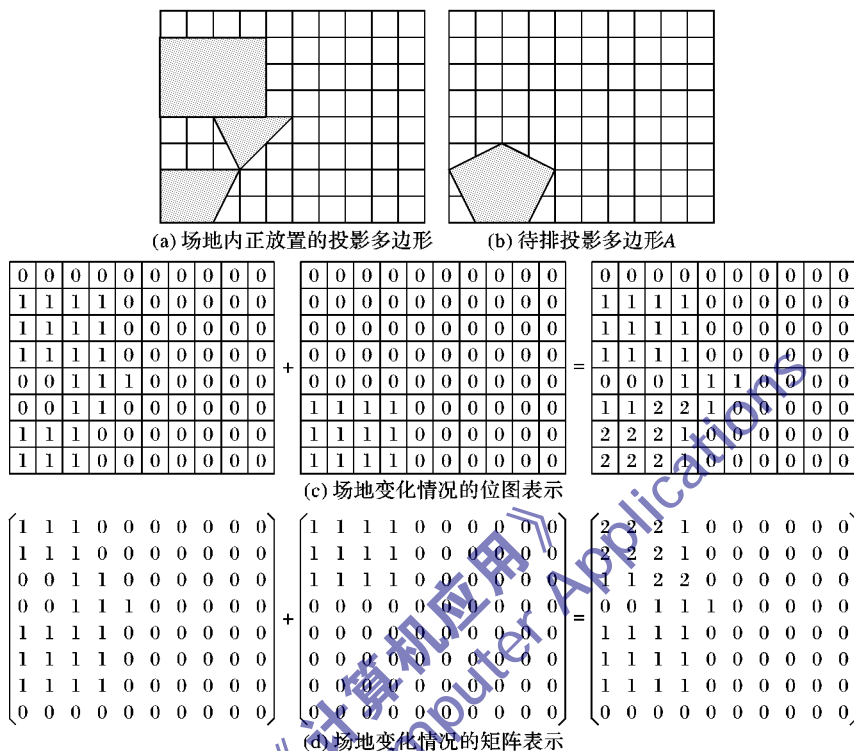


图2 待排投影多边形A的定位判断^[14]

此时场地的位图及矩阵中出现元素2,表示投影多边形 A 放置在场地上时, A 与场地内已有的多边形发生重叠,该位置不可行,需要将该多边形按照“靠左、靠下”规则移动,即多边形 A 从场地的左下角开始,先按照 y_{\min} 的步长逐渐向上移动并判断。若 A 未超出场地边界且场地的位图(或矩阵)中未出现元素2,则 A 定位成功,当前位置为多边形 A 的最佳定位位置,并更新当前的场地信息及分段信息;否则,需将多边形按照 x_{\min} 的步长右移至场地底边,再继续向上移动以寻找最佳定位位置。投影多边形 A 的移动过程如图3(a)所示。若 A 处于场地右上角位置时,场地的位图像素中仍出现元素2,则表示该分段定位失败,需将该分段及后续分段均延迟进行定位。投影多边形 A 定位成功后,场地的位图信息如图3(b)所示。

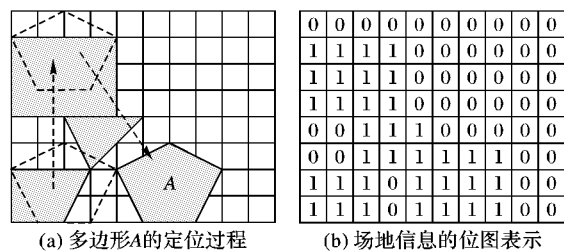


图3 多边形A的定位过程及场地的位图表示

4 验证与分析

4.1 船厂实际算例验证

以 Matlab R2010a 为编程环境实现本文的分段堆放算法。

以某船厂的50个不规则分段作为实验数据,部分分段信息如表1所示。场地长为20,宽为16。像素单位为 0.5×0.5 。PSO各参数分别设置为:最大进化代数50,种群规模100,惯性权重为 $0.4 \sim 0.05$,学习因子 c_1, c_2 在 $[0, 1]$ 区间随机获取,评价函数中的权重因子 $k_1 = k_2 = 1$,调节系数 $\sigma = 0$ 。

表1 不规则分段的投影多边形信息示例

分段编号	加工周期(d)	最早入场时间(Elit)	最晚入场时间(Lit)	投影凸多边形的原始坐标(逆时针)
1	4	2009-10-06	2009-10-08	$\begin{pmatrix} 0 & 5 & 5 & 0 \\ 0 & 0 & 4 & 4 \end{pmatrix}$
2	7	2009-10-07	2009-10-14	$\begin{pmatrix} 0 & 5 & 0 \\ 0 & 0 & 6 \end{pmatrix}$
3	6	2009-10-08	2009-10-18	$\begin{pmatrix} 0 & 4 & 4 & 2 & 0 \\ 0 & 0 & 3 & 4 & 3 \end{pmatrix}$
⋮	⋮	⋮	⋮	⋮
49	10	2009-10-15	2009-10-19	$\begin{pmatrix} 0 & 2 & 4 & 4 & 3 \\ 3 & 0 & 1 & 2 & 4 \end{pmatrix}$
50	7	2009-10-08	2009-10-17	$\begin{pmatrix} 0 & 1 & 5 & 5 & 2 & 0 \\ 3 & 1 & 3 & 5 & 6 & 5 \end{pmatrix}$

图4(a)给出了改进PSO的收敛过程。显然,解的集合是逐渐收敛的,并且在第18代之后个体的多样性开始消失,最优解开始收敛。

对于最优堆放方案,由于分段堆放时间及场地大小的双重约束,每天的场地利用率不尽相同,但总体围绕某值上下波动。如图4(b)所示,在最优堆放方案下,前15天的场地利

用率围绕 75% 上下波动。

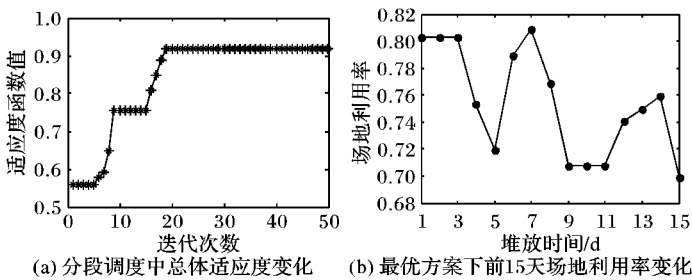


图4 改进的 PSO 收敛过程及最优方案中场地利用率变化

4.2 算法比较与分析

为判断算法的优劣,对比了三种算法的实验结果,如表 2 所示。其中算法 1 表示“PSO + 基于网格搜索的空间定位策略”,算法 2 表示“改进的 PSO + 基于临界多边形算法 (No-Fit Polygon, NFP) 的空间定位策略”,算法 3 表示“改进的 PSO + 基于网格搜索的空间定位策略”(即本文算法)。算法 1 与算法 3 的不同在于生成分段的堆放顺序的算法,前者利用基本的粒子群算法排序,后者则利用改进的粒子群算法排序;算法 2 与算法 3 的不同在于分段的定位策略,前者采用基于临界多边形的空间定位策略,后者则采用基于网格搜索的空间定位策略。

表 2 调度结果对比分析

分段的 动态堆 放算法	需移动的 分段数量 $Y_{\text{入}} + Y_{\text{出}}$	场地 平均利 用率/%	场地 最高利 用率/%	迭代一 次运算 时间/s	收敛 代数
算法 1	4 + 3	63.41	69.38	28.093	27
算法 2	9 + 7	73.96	82.66	79.016	20
算法 3	2 + 3	70.24	79.93	30.052	18

从表 2 可以看出:

1) 算法 1 与算法 3 比较而言:迭代一次的运行时间几乎相同,但算法 1 的收敛速度慢,最优方案下的场地利用率较低,有可能发生了局部最优的情况。这说明 PSO 算法中的改进点不占用太长时间,而且可以避免局部最优情况的发生。

2) 算法 2 与算法 3 比较而言:利用算法 2 对分段进行堆放时,场地利用率最高,但在场地内移动分段却最复杂,移动的分段数量超过了算法 3 中移动分段数量的三倍,运行时间也较长,在分段堆放问题中的实用性较差。

综上所述,将改进的粒子群算法与基于网格搜索的空间定位策略相结合的分段堆放算法可较好地满足船厂追求最大化的利用场地且降低运输设备负荷的要求,为最优的分段堆放算法。

5 结语

本文以提高场地平均利用率、减少分段移动数量为目标,综合运用改进的粒子群算法和基于网格搜索的空间定位策略,形成了两阶段混合时空堆放算法,解决了不规则分段的动态场地堆放问题。通过利用 Matlab 软件对船厂实际生产数据的仿真结果表明,该算法可以快速收敛到满意解,得到比较理想的分段堆放方案;且通过对比分析该算法与其他算法的结果表明该算法可更好地应用于船厂实际需求,有较高的实用性。

本文综合考虑船舶分段动态空间调度过程中时空结合的特征以及现阶段实现策略的不足与缺陷,针对凸多边形的不

规则性,对粒子群算法和分段定位算法都进行了相应改进:

1) 对原始的粒子群算法的改进:在粒子更新位移及速度时,引入了遗传算法中的选择算子和变异算子思想,且惯性权重 ω 及粒子更新的数量均利用了模拟退火算法的思想,令二者随着迭代次数的增加逐渐减小,以避免最优粒子被破坏,能较快得到最优方案;

2) 针对 Matlab 处理矩阵方面的优势,利用位图及矩阵表示场地及分段的投影多边形的相关信息,分段定位速度较快;

3) 在多边形定位过程中,为了寻找最佳位置,多边形的旋转角度由有限角度 (0° 、 90° 、 180° 、 270°) 改进到 $0^\circ \sim 360^\circ$ 的任一角度;

4) 在选择最优的空间布局策略时,综合考虑了场地平均利用率、分段移动复杂度,对船厂的分段堆场问题具有实际应用性。

参考文献:

- [1] 张志英, 江志斌, 陈强, 等. 曲面分段加工的虚拟流水线生产模式及关键技术探讨[J]. 中国造船, 2005, 46(3): 112 - 116.
- [2] LEE K J, LEE J K, CHOI S Y. A spatial scheduling system and its application to shipbuilding: DAS-CURVE [J]. Expert Systems with Applications, 1996, 10(3/4): 311 - 324.
- [3] SUNG C S, CHOUNG Y I. Minimizing makespan on a single burn-in oven in semiconductor manufacturing [J]. European Journal of Operational Research, 2000, 120(3): 559 - 574.
- [4] PAULUS J J, HURINK J. Adjacent-resource scheduling: why spatial resources are so hard to incorporate [J]. Electronic Notes in Discrete Mathematics, 2006, 25(1): 113 - 116.
- [5] JEFFREY D C, MICHAEL J M, GEORGE G P, et al. Scheduling parallel assembly workstations to minimize a shared pool of labor [J]. IIE Transactions, 2008, 40(8): 749 - 758.
- [6] RAJ P, SRIVASTAVA R K. Analytical and heuristic approaches for solving the spatial scheduling problem [C]// IEEE International Conference on Industrial Engineering and Engineering Management. Piscataway: IEEE, 2007: 1093 - 1097.
- [7] 张志英, 陈洁. 空间调度问题的非线性规划分析求解方法[J]. 计算机集成制造系统, 2010, 16(6): 1273 - 1279.
- [8] CHO K K, LEE D, OUM T H. An intelligent spatial planning and scheduling system [J]. International Journal of Industrial Engineering: Theory, Applications and Practice, 2002, 9(2): 133 - 140.
- [9] VARGHESE R, DUCK Y Y. Dynamic spatial block arrangement scheduling in shipbuilding industry using genetic algorithm [C]// INDIN '05: The 3rd IEEE International Conference on Industrial Informatics. Piscataway: IEEE, 2005: 444 - 449.
- [10] SHIN J C, KWON O H, RYU C. Heuristic and metaheuristic spatial planning of assembly blocks with process schedules in an assembly shop using differential evolution [J]. Production Planning and Control, 2008, 19(6): 605 - 615.
- [11] 张志英, 杨克开, 于瑾维, 等. 改进粒子群算法的动态空间调度方法[J]. 哈尔滨工程大学学报, 2009, 30(12): 1344 - 1350.
- [12] 满春涛, 孙明辉, 张礼勇. 粒子群优化算法在多峰函数寻优上的应用[J]. 哈尔滨理工大学学报, 2007, 12(2): 11 - 13.
- [13] EBERHART R, KENNEDY J. A new optimizer using particles swarm theory [C]// MHS '95: Proceedings of the 6th International Symposium on Micro Machine and Human Science. Piscataway: IEEE, 1995: 39 - 43.
- [14] 王蕾. 面向船体曲面分段建造的空间调度方法研究[D]. 上海: 上海交通大学, 2009.