

DOI:10.3876/j.issn.1000-1980.2021.02.013

基于遗传-剩余矩形算法的装配式预制构件堆场优化

金 睿¹,丁宏亮²,沈西华²,刘东海¹

(1. 天津大学水利工程仿真与安全国家重点实验室,天津 300350; 2. 浙江省建工集团有限责任公司,浙江 杭州 310012)

摘要:为了解决装配式建筑施工现场预制构件堆放缺乏统筹规划和信息化管理不足的问题,对预制构件的现场堆放方案和吊装距离进行分析,建立了基于遗传算法和剩余矩形算法的堆场优化数学模型并提出了求解方法,开发了堆场优化系统。利用该系统可以对可堆叠和不可堆叠类型的构件堆放进行优化,给出含有构件堆放位置和堆放顺序的优化排布方案,使得构件吊装距离最小;且该系统中构件状态管理流程明确,管理人员可根据优化方案高效安排现场构件堆放。验证结果表明,堆场优化系统可提高预制构件堆放的管理水平和吊装效率,降低对现场管理人员经验的依赖,实现预制构件堆放的信息化管理。

关键词:装配式建筑;预制构件;堆场优化;遗传算法;剩余矩形算法;优化系统

中图分类号:TU756 **文献标志码:**A **文章编号:**1000-1980(2021)02-0182-08

Packing optimization of precast concrete components for prefabricated construction based on genetic-surplus rectangle algorithm

JIN Rui¹, DING Hongliang², SHEN Xihua², LIU Donghai¹

(1. State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin 300350, China; 2. Zhejiang Construction Engineering Group Co., Ltd., Hangzhou 310012, China)

Abstract: To solve the lack problems of overall planning and information management of precast concrete components stacking in prefabricated construction site, a mathematical model for the packing optimization was established based on the analysis of the site stacking scheme and the hoisting distance of precast concrete components. The solving method for the model was proposed based on the genetic algorithm and the surplus rectangle algorithm, and a packing optimization system was developed. The packing optimization system could be used for both the stackable condition and the non-stackable condition by providing the optimum scheme including the stacking position and the stacking order with the minimum hoisting distance. Moreover, the management process of precast components is clear, and managers can arrange the precast packing according to the optimum scheme. A case study demonstrates that the optimization system improves the management level and hoists the efficiency of precast. Meanwhile, it can reduce the reliance on the site management experience and realize the information management of precast.

Key words: prefabricated construction; precast concrete component; packing optimization; genetic algorithm; surplus rectangle algorithm; optimization system

随着装配式建筑的发展,预制构件在施工现场堆场的堆放优化问题随之产生^[1-2]。现场管理人员凭借经验对预制构件堆放方案进行人工规划,但经验堆放方案往往优化程度低,导致构件的堆放缺乏整体性,影响吊装效率;同时目前还存在对预制构件的信息化管理不足、构件信息不全和信息不可追溯等问题^[3-4]。

预制构件在堆场的合理堆放问题,其实是在有限的空间内实现构件的优化排布,可通过优化排样算法进行计算求解。常用的优化算法有遗传算法、粒子群算法、贪心算法、模拟退火算法、蚁群算法等^[5-7]。遗传算法^[8]是一种借鉴生物界自然选择和自然遗传机制的随机搜索算法,其特点是具有良好的全局寻优性能^[9]。遗传算法结合剩余矩形算法^[10]、BL 算法^[11]、下台阶算法^[12]、最低水平线算法^[13-14]等排样算法,可以有效求解矩形件排样问题。刘海明等^[13]通过改进最低水平线方法,并与基于分阶段遗传算子的遗传算法相结合,共同求解矩形件排样问题,其提出的优化算法能够有效改善排样效果,提高材料利用率。Charalambous

基金项目: 国家重点研发计划(2016YFC0702005)。
作者简介: 金睿(1975—),男,正高级工程师,硕士,主要从事建设工程技术研究。E-mail:hzrui@126.com
引用本文: 金睿,丁宏亮,沈西华,等. 基于遗传-剩余矩形算法的装配式预制构件堆场优化[J]. 河海大学学报(自然科学版),2021,49(2):182-189.
JIN Rui, DING Hongliang, SHEN Xihua, et al. Packing optimization of precast concrete components for prefabricated construction based on genetic-surplus rectangle algorithm [J]. Journal of Hohai University(Natural Sciences), 2021, 49(2):182-189.

等^[10]提出一种剩余矩形算法,将板材可用区域不断划分为矩形块并逐个排入矩形零件,以尽可能减少板材浪费。韩喜君等^[15]利用遗传算法并结合剩余矩形排样算法求解矩形件正交排样问题,实例验证表明混合遗传算法具有较高的准确性、有效性和可行性。然而,上述研究成果主要面向矩形件的排样领域,用于提高排样后材料的利用率,而预制构件的现场堆放还需考虑后续的吊装成本,所以现有算法难以直接应用于预制构件在堆场的堆放排布问题。

本文采用遗传算法和剩余矩形算法对预制构件堆场优化问题进行研究,建立优化模型并提出求解方法,并开发了堆场优化系统,为预制构件的现场堆放提供优化排布方案,以提高堆场管理水平,减少吊装成本。

1 堆场优化数学模型

1.1 问题描述

预制构件在施工现场的堆放优化问题,本质上是在满足构件堆放位置和堆放顺序要求的情况下进行构件堆放,达到尽可能减少构件吊装成本的目的。对于构件吊装,塔吊单次往返时间包括挂钩、就位、小车平移提升、大臂转动和摘钩返回下一吊的时间,塔吊工作距离越短则塔吊单次往返时间越少。而减少构件吊装距离,可以减少塔吊小车平移提升和摘钩返回下一吊的工作距离,从而减少塔吊单次往返的时间。当进行多次构件吊装作业时可以节省大量时间,从而提高吊装效率,降低构件的吊装成本。因此,本文将堆场优化问题转化成以构件吊装距离最小为优化目标的数学问题,建立堆场优化数学模型。

按照构件是否可以堆叠,可分为:(a)不可堆叠的构件,如预制柱、预制外墙等;(b)可堆叠的构件,如叠合板、叠合梁、预制楼梯等。对于不可堆叠的构件,其在二维空间的优化排布问题其实就是求解在已知宽度的堆场上,使得吊装距离最小、同时占用堆场长度尽可能小的堆放所有构件的方案。对于可堆叠的构件,可利用最小包络矩形法先将堆叠的构件看成一个整体,形成最小包络矩形集,再将问题转化为不可堆叠的构件优化排布问题。

1.2 建模前准备

设堆场优化数学模型的堆场宽为 W ,长为 L ,堆场的面积足以排下所有的预制构件。要排的预制构件共有 N 个,构件 i 的高度为 h_i ,宽度为 w_i 。以堆场左下角为原点建立平面直角坐标系(图 1),构件在堆场上的位置可以由该构件左下角和右上角的坐标完全确定,即第 i 个构件位置关系可以通过式(1)或(2)确定。

横放的情况：
$$\begin{cases} x_{r,i} = x_{l,i} + h_i \\ y_{r,i} = y_{l,i} + w_i \end{cases} \quad (1)$$

竖放的情况：
$$\begin{cases} x_{r,i} = x_{l,i} + w_i \\ y_{r,i} = y_{l,i} + h_i \end{cases} \quad (2)$$

式中: $x_{l,i}$ ——第 i 个构件左下角横坐标; $y_{l,i}$ ——第 i 个构件左下角纵坐标; $x_{r,i}$ ——第 i 个构件右上角横坐标; $y_{r,i}$ ——第 i 个构件右上角纵坐标。

1.3 模型目标函数

求解在已知长和宽的堆场上堆放 N 个预制构件,在满足构件堆放位置和堆放顺序要求的情况下,使得构件吊装距离最小,同时占用堆场长度尽可能小的方案。在堆场上堆放构件后,构件中心距离塔吊中心的距离之和为

$$S = \sum_{i=1}^N \sqrt{\left(\frac{x_{l,i} + x_{r,i}}{2} - x_0\right)^2 + \left(\frac{y_{l,i} + y_{r,i}}{2} - y_0\right)^2} \quad (3)$$

式中: S ——所有预制构件中心与塔吊中心距离之和,m; x_0 ——塔吊相对于堆场的横坐标; y_0 ——塔吊相对于堆场的纵坐标。

1.4 约束条件

a. 预制构件在堆场堆放时不出界:

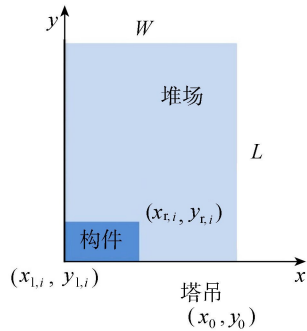


图 1 预制构件与堆场坐标示意图
Fig.1 Schematic diagram of precast components and yard coordinates

$$\begin{cases} x_{r,i} \leq W \\ y_{r,i} \leq L \end{cases} \tag{4}$$

在构件堆放时,从左下角向右上角依次堆放,任一构件要满足不出界的约束。

b. 预制构件坐标关系:

$$\begin{cases} x_{r,i} = x_{1,i} + (1 - r_i) w_i + r_i h_i \\ y_{r,i} = y_{1,i} + (1 - r_i) h_i + r_i w_i \end{cases} \tag{5}$$

式中: r_i ——变量,取值0或1。当 $r_i=1$ 时表示构件*i*横放, $r_i=0$ 时表示构件*i*竖放。任一构件的位置完全由其左下角坐标和其摆放方式(横放或竖放)决定,因此该构件坐标满足上述约束关系。

c. 邻接预制构件之间高度和宽度的限制条件:

$$\begin{cases} y_{r,i} \leq y_{1,j} + (1 - t_{ij}) L \\ x_{r,i} \leq x_{1,j} + (1 - s_{ij}) W \end{cases} \quad (i \neq j) \tag{6}$$

式中: t_{ij} 、 s_{ij} ——变量,取值0或1。构件*i*和*j*为任意2个构件,当 $s_{ij}=1$ 时说明构件*i*在*j*的左侧;当 $s_{ij}=0$ 时说明构件*i*在*j*的右侧;当 $t_{ij}=1$ 时说明构件*i*在*j*的下方,当 $t_{ij}=0$ 时说明构件*i*在*j*的上方。构件堆放时相邻构件之间高度和宽度需满足上述限制条件。

d. 预制构件位置不出现重叠:

$$s_{ij} + s_{ji} + t_{ij} + t_{ji} \geq 1 \quad (i < j) \tag{7}$$

为防止构件位置出现重叠,各构件之间必须满足上或者下,左或者右的关系。

e. 各个变量的约束:

$$W, L, x_{r,i}, y_{r,i}, x_{1,i}, y_{1,i} \geq 0 \tag{8}$$

1.5 可堆叠构件情况分析

对于可堆叠构件的堆放情况,每一堆构件就是一个堆区,采用最小包络矩形法求得各堆区在堆场中的宽度和长度,组成新的矩形集,然后采用不可堆叠的构件优化数学模型进行求解。因此,在解决可堆叠构件优化排布问题前,需要对构件的堆叠顺序和堆叠后形成的最小包络矩形进行预处理。

首先将*N*个构件分堆,每堆构件可堆叠*m*层($m \leq 5$),则可分成 $n = \lceil N/m \rceil$ 堆($\lceil \cdot \rceil$ 表示向上取整)。将每堆构件按照吊装优先级进行排序,优先级较高的构件排在上面,得到构件从上到下的堆叠顺序。

采用最小包络矩形法,取各堆构件中高和宽的最大值生成最小包络矩形,并形成最小包络矩形集,矩形数为 $\lceil N/m \rceil$,该矩形集中各矩形大小即代表各堆区大小。然后采用不可堆叠构件优化数学模型进行求解,得到优化方案。因此,对于可堆叠构件,构件吊装距离最小的优化模型如下:

$$\min S = m \sum_{i=1}^{n-1} \sqrt{\left(\frac{x_{1,i} + x_{r,i}}{2} - x_0\right)^2 + \left(\frac{y_{1,i} + y_{r,i}}{2} - y_0\right)^2} + (N - nm) \sqrt{\left(\frac{x_{1,n} + x_{r,n}}{2} - x_0\right)^2 + \left(\frac{y_{1,n} + y_{r,n}}{2} - y_0\right)^2} \tag{9}$$

$$\text{s. t. } \begin{cases} n = \left\lceil \frac{N}{m} \right\rceil & (m \leq 5) \\ x_{r,i} \leq W \\ y_{r,i} \leq L \\ x_{r,i} = x_{1,i} + (1 - r_i) w_i + r_i h_i \\ y_{r,i} = y_{1,i} + (1 - r_i) h_i + r_i w_i \\ y_{r,i} \leq y_{1,j} + (1 - t_{ij}) L_{\min} & (i \neq j) \\ x_{r,i} \leq x_{1,j} + (1 - s_{ij}) W & (i \neq j) \\ s_{ij} + s_{ji} + t_{ij} + t_{ji} \geq 1 & (i < j) \\ W, L, x_{r,i}, y_{r,i}, x_{1,i}, y_{1,i} \geq 0 \\ r_i, s_{ij}, t_{ij} \in \{0, 1\} \end{cases}$$

2 优化算法及堆放策略

采用遗传算法结合剩余矩形算法,对优化模型进行 Matlab 编程^[16],求解在已知长和宽的堆场上堆放完

N 个预制构件,使得构件吊装距离最小的方案。优化算法以吊装距离的倒数为适应度函数,基本步骤见图 2。

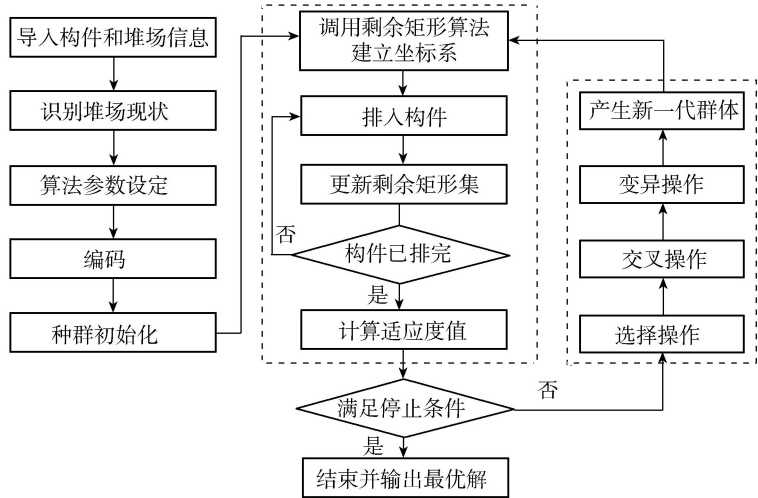


图 2 优化算法流程

Fig.2 Flow chart of optimization algorithm

- 第 1 步:导入所有待堆放构件的信息、堆场的尺寸(宽 W ,长 L)、塔吊相对于堆场的位置坐标 (x_0,y_0) 。
- 第 2 步:识别现有堆场的堆放情况,确定待堆放构件在堆场中的可堆放空间(最低可堆放位置)。
- 第 3 步:设置遗传算法参数,种群大小 pop_size ,每个染色体或个体的长度为 chromo_size ,最大代数 G ,交叉率 P_c ,变异率 P_m 。
- 第 4 步:采用十进制编码,以排样顺序中的 N 个构件编号 R_i 对应 w_i 的大小,产生 $w_1,w_2,\cdots,w_N$ 序列编码组成染色体,当 $w_i>0$ 时,判断为竖放,否则为横放。
- 第 5 步:种群初始化,随机产生初始种群,获得 pop_size 个染色体。
- 第 6 步:优化目标为吊装距离最小,适应度函数取目标函数的倒数,计算种群适应度值 $f_i=1/S$ 。对个体适应度值升序排序,找出最优个体并保留。
- 为求解适应度值,需知道构件在堆场上的排布情况。使用剩余矩形算法,算法如下:(a)建立以 W 为横轴, L 为纵轴的直角坐标系,定义堆场的左下角为原点 $(0,0)$,右上角坐标为 (W,L) ,得到初始矩形 $R_1=[(0,0),(W,L)]$ 。(b)排入构件 $[(x_{l,i},y_{l,i}),(x_{r,i},y_{r,i})]$,将剩余矩形集合中的每一个矩形都减去此矩形所占的位置,并记录剩余矩形。(c)去掉面积为 0 或已无法排下所剩的任何一个构件的剩余矩形;把具有完全包含关系的剩余矩形中面积小的矩形去除,有相交关系的矩形全部保留,得到新的剩余矩形集。(d)向剩余矩形集中继续排入构件,排放时选择宽、高均大于等于待排放构件的剩余矩形。该剩余矩形是剩余矩形集中最低且最靠左的剩余矩形,以使待排放构件与剩余矩形的左下角重叠。排放完成后跳转到 c,直到 N 个构件全部排完。
- 第 7 步:依次进行选择、交叉和变异操作。(a)选择操作。使用“轮盘赌”选出一批适应度高的父代,并遗传到下一代,个体 i 被选中的概率为 $P_i=f_i/\sum_{i=1}^M f_i$ 。同时,保留这一代的最佳个体,遗传至下一代,替代上一代中的最差个体。(b)交叉操作则是在父代染色体中随机设定两个交叉点并按交叉率交叉互换,产生新个体。(c)变异操作。最后按照变异率 P_m 随机选取染色体某一点的值进行单点变异。
- 第 8 步:重复验算,达到预定的进化代数时结束,保存最佳个体。
- 编程求解该遗传-剩余矩形算法后可以给出堆场优化数学模型的相对最优解,获得构件堆放的优化方案。使用该算法求解堆场优化问题,具有以下特点:(a)由于构件在堆场的堆放是一个动态过程,因此在算法开始时设置堆场现状识别功能以迅速确定可堆放空间;(b)始终保留最佳个体遗传至下一代,避免最佳个体在计算过程中丢失;(c)快速获得人工在短时间内难以得到的优化方案。

3 堆场优化系统

为实现遗传-剩余矩形算法的转化应用,并服务于项目现场构件的管理,在现有“浙江省建工集团工业化项目管理平台”中进行二次开发,建立堆场优化系统。如图 3 所示,该系统由系统输入模块、系统优化模块和系统输出

模块三部分组成。项目管理人员根据项目信息,采用该系统获取堆场信息和所有构件编码信息,从数据库调取构件类型和构件尺寸等参数信息。系统通过编码识别、数据提取、堆场分配过程,结合堆场信息对构件的堆放位置和堆放方式进行优化,然后由优化算法产生优化排布方案,管理人员将产生的优化排布方案进行评价并最终确认。最后这些数据由系统输出模块从前端展示,指导现场构件堆放。

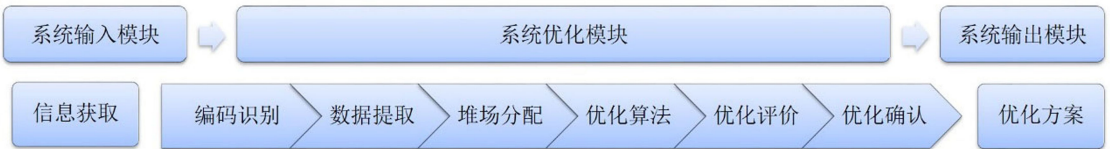


图3 预制构件堆场优化系统构成示意图

Fig.3 Schematic diagram of packing optimization system of precast concrete components

堆场优化系统前端界面如图4所示,该界面展示了构件堆放的位置信息,选中某一构件后还可展示完整的构件属性信息,如构件编号、构件类型、所属状态等。由此,现场和企业管理人员可以快速了解堆场和构件现状,并使用该信息化的管理手段辅助项目管理。

堆场优化系统移动端如图5所示,现场管理人员可对构件的进场和堆放进行管理。当预制构件到达现场通过质量验收并点击“确认”后(图5(a)),系统会自动给出优化推荐的堆场(图5(b))。选择优化推荐的堆场后,则界面显示该堆场的现状(图5(c)),同时待入库构件的具体位置也可从移动端获得,操作人员据此可将预制构件放入指定位置。在移动端查看堆场详情时,根据构件编号可以直观看到每个构件所在的堆场、堆区和层(图5(d)),实现对构件的精确定位和管理。



图4 堆场优化系统界面

Fig.4 User interface of packing optimization system



(a) 入库 (b) 选择堆场位置 (c) 堆场现状 (d) 构件所在堆场情况

图5 堆场优化系统移动端

Fig.5 Mobile terminal of packing optimization system

本文开发的堆场优化系统,将堆场优化数学模型和遗传-剩余矩形算法成果应用于工程实际,经验只是起辅助作用,复杂的计算和堆放方案由系统提供,人的作用只是判断系统的优化方案是否合理与可行。利用该系统可提高现场工作效率,减少不必要的时间浪费,降低对经验的依赖程度,实现降本增效的目的。现场工作人员能快速学会并使用该堆场优化系统,使建筑施工由粗放向精细化方向转变。

4 系统应用分析

某房建工程楼板和楼梯采用预制装配式混凝土建造。预制叠合板和预制楼梯与结构同步施工,垂直运输机械采用塔吊吊运。通过吊运分析,各栋楼的预制叠合板均在塔吊辐射半径以内,且吊运能力满足预制叠合板的重量要求,因此全部采用塔吊进行吊装。现需要在施工现场的堆场堆放预制构件,堆场为宽15 m、长30 m的

矩形,堆场初始状态为无构件堆放状态,塔吊相对于堆场的坐标位置为(7.5,-9),待堆放构件的尺寸信息见表1。

表1 构件及其吊装优先级

Table 1 List of precast components and hoisting priority

序号	宽/m	高/m	依次吊装顺序	序号	宽/m	高/m	依次吊装顺序
1	4.76	2.38	23号构件	17	2.78	1.98	27号构件
2	4.48	2.58	14号构件	18	2.78	2.08	29号构件
3	4.18	2.78	15号构件	19	2.78	1.18	30号构件
4	3.98	2.78	8号构件	20	2.50	2.30	18号构件
5	3.50	2.04	28号构件	21	2.06	1.88	25号构件
6	3.38	1.18	12号构件	22	2.04	0.98	11号构件
7	3.38	1.84	13号构件	23	1.98	1.18	26号构件
8	3.38	2.78	21号构件	24	1.88	1.58	16号构件
9	3.28	1.18	2号构件	25	1.88	1.10	24号构件
10	3.28	1.84	4号构件	26	1.76	0.80	22号构件
11	3.28	2.78	5号构件	27	1.18	0.48	1号构件
12	3.28	1.68	10号构件	28	0.98	0.48	6号构件
13	2.89	1.58	19号构件	29	0.96	0.48	17号构件
14	2.78	0.98	7号构件	30	0.86	0.48	3号构件
15	2.78	2.09	9号构件	31	0.86	0.48	31号构件
16	2.78	1.28	20号构件	32	0.86	0.48	32号构件

目前,构件的现场堆放方案由管理人员人工规划,在有限时间内所得的经验方案优化程度低。根据经验,一般需要对堆场情况和构件大小信息进行初步判断,然后按照卸车顺序对构件逐个堆放,因此管理人员需要对构件进行多次判断并堆放,消耗时间多,吊装效率低,堆放方案不能达到减少吊装距离、节约吊装成本的目的。

现采用堆场优化系统来指导构件的现场堆放。先将堆场、构件等信息导入系统。

a. 在考虑不可堆叠的情况下进行优化求解,可以快速得到优化后所有构件的堆放顺序和堆放位置坐标。系统自动调用 Matlab 优化算法程序,并集成在工业化项目管理平台中使用。图6(a)为构件在堆场的堆放位置排样。从图6(a)可以看到每个构件的堆放方式和具体位置。在最优堆放方案下,塔吊最小工作距离为438.3 m。图6(b)为经验方案的模拟堆放图,按照构件卸车顺序在堆场堆放。从图6(b)可以看到,采用经验方案进行构件堆放,占用空间大,且该方案下的吊装距离为486.2 m,增加的吊装距离增加了吊装成本。采用堆场优化系统得到的优化方案,使构件在堆场中可以合理排布,降低了对现场管理人员经验的依赖性,同时减少了吊装距离,提高了吊装效率,降低了综合成本。

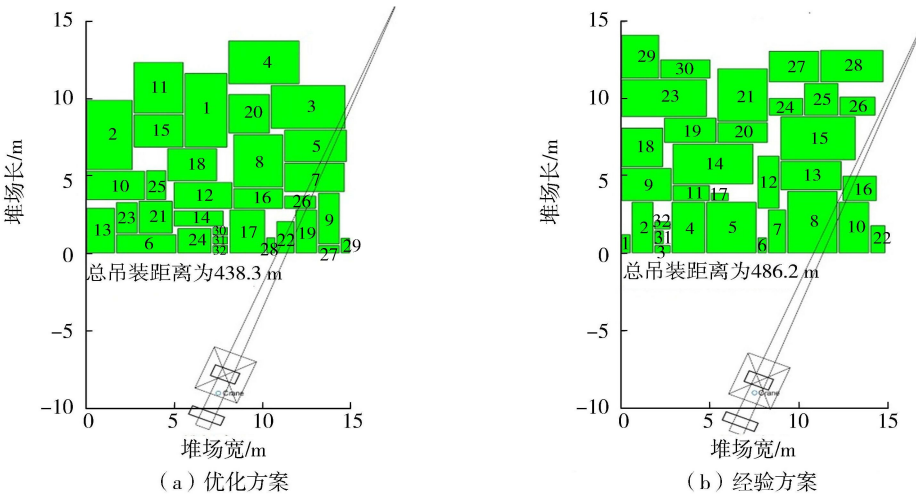


图6 不可堆叠情况下构件堆放位置排样

Fig. 6 Layout of packing scheme under non-stackable condition

b. 在考虑可堆叠的情况下进行优化求解,得到优化后所有的最小包络矩形在堆场中的排列顺序和堆放位置坐标,自动绘制成预制构件堆叠图,见图7。从图7(a)可以看出最优堆放方案下的构件排布方式,对于各最小包络矩形,在最优堆放方案下塔吊最小工作距离为342.5 m。图7(b)是经验方案的模拟堆放图,该方案是较优的经验方案,在考虑各堆区中构件的吊装顺序后进行堆叠。从图7(b)可以看出,构件在堆场所

占用的堆场长度大于优化方案,同时,该方案下的吊装距离为387.0 m,相比于优化方案,较多的吊装距离增加了吊装成本。

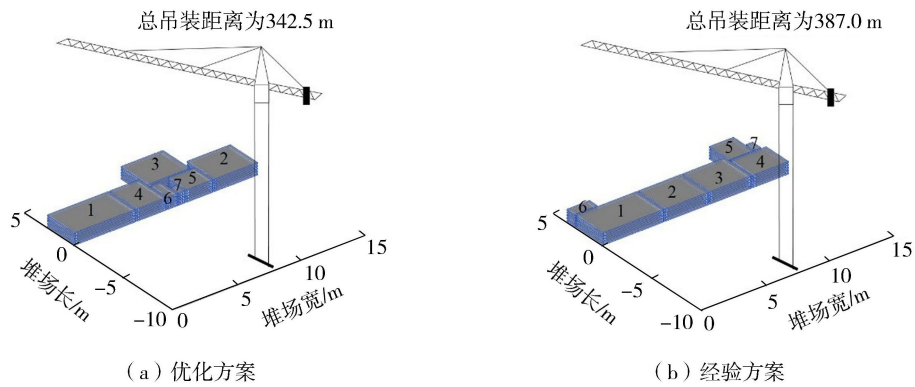


图7 可堆叠情况下构件堆放位置排样

Fig. 7 Layout of precast packing scheme under stackable condition

表2为构件在各堆区中从上到下的堆放顺序,如堆区1代表第一堆构件,该堆区最小包络矩形的宽和长分别为4.76 m和2.78 m,构件从上到下的堆叠顺序按照吊装优先级依次为8、14、15、23、28,这种堆叠顺序确保了先吊装构件在上层。在可堆叠情况下采用堆场优化系统得到的优化方案,考虑了构件的吊装顺序后进行合理堆叠,避免了吊装过程中需要吊装堆放在下层位置构件的情况,减少了吊装距离,提高了吊装效率,降低了管理成本,从而降低了综合成本。

图8为在不可堆叠和可堆叠工况下遗传-剩余矩形算法迭代过程中各代最小吊装距离和平均吊装距离随着迭代次数增加的收敛图。从图8(a)可以看出,在不可堆叠工况下,随着迭代次数的增加,算法逐渐收敛。同时,算法在第50代和第68代时都跳出局部最优,找到更优的解并保持稳定,可以看出该算法在一定迭代次数后得到的结果是稳定准确的。图8(b)显示,由于在可堆叠情况下最小包络矩形整体数量较少,算法迅速收敛并在迭代30代后最小吊装距离和平均吊装距离都基本保持稳定,证明该算法得到的结果也是稳定准确的。根据上述收敛情况,遗传-剩余矩形算法不仅合理、可靠,而且收敛速度快,可求解得到满足要求的相对最优解。

表2 构件的堆放顺序

Table 2 Stacking order of precast components

堆区	堆区尺寸/m		依次吊装顺序	堆区	堆区尺寸/m		依次吊装顺序
	宽	长			宽	长	
1	4.76	2.78	8号构件	4	2.78	2.30	18号构件
			14号构件				20号构件
			15号构件				27号构件
			23号构件				29号构件
			28号构件				30号构件
2	2.78	3.38	2号构件	5	2.06	1.88	11号构件
			4号构件				16号构件
			12号构件				24号构件
			13号构件				25号构件
3	2.78	3.28	21号构件	6	2.78	3.28	26号构件
			5号构件				1号构件
			7号构件				3号构件
			9号构件				6号构件
			10号构件				17号构件
			19号构件	7	0.48	0.86	22号构件
							31号构件
							32号构件

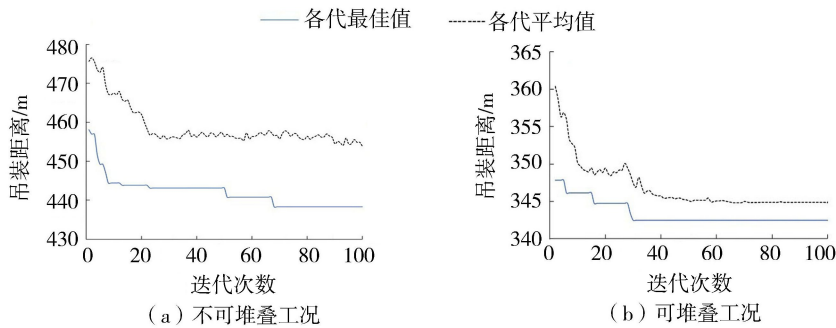


图8 优化算法收敛曲线

Fig. 8 Iterative convergence curves of optimization algorithm

5 结 语

对于预制构件的现场堆放问题,本文提出了基于遗传-剩余矩形算法的堆场优化数学模型及其求解方法,利用 Matlab 编程求得相对最优解,获得构件堆放的优化方案。在工业化项目管理平台上二次开发后形成堆场优化系统,应用于项目堆场管理后可以看出,该系统可以针对可堆叠和不可堆叠类型的构件进行优化,给出的优化方案使得吊装距离最小,降低了对现场管理人员经验的依赖程度,提高了现场工作效率,降低了堆放和吊装综合成本。该系统解决了对构件管理不足、构件在现场堆放无序等问题,实现了构件的管理信息化,助推装配式建筑施工向智能化方向发展。

参考文献:

[1] MA Zhiliang,YANG Zhitian,LIU Shilong,et al. Optimized rescheduling of multiple production lines for flowshop production of reinforced precast concrete components[J]. Automation in Construction,2018,95:86-97.

[2] 周冲,薛守斌,王伟. 我国建筑结构装配式施工技术的应用与展望[J]. 建筑技术,2018,49(6):605-608. (ZHOU Chong,XUE Shoubin,WANG Wei. Application and outlook for domestic construction technique of prefabricated building structure in China[J]. Architecture Technology,2018,49(6):605-608. (in Chinese))

[3] 董娜,张宁,赵雪媛,等. 基于 BIM 与 RFID 的工业化建筑构件进度管理研究[J]. 施工技术,2018,47(10):5-10. (DONG Na,ZHANG Ning,ZHAO Xueyuan,et al. Research on progress management of industrial building components based on BIM and RFID[J]. Construction Technology,2018,47(10):5-10. (in Chinese))

[4] 于振欢,韦永斌. 基于物联网的预制构件信息管理系统设计与应用[J]. 工程建设与设计,2017(15):222-227. (YU Zhenhuan,WEI Yongbin. The design of information management system for precast concretes based on IOT[J]. Construction & Design for Project,2017(15):222-227. (in Chinese))

[5] 李明. 智能优化排样技术研究[D]. 杭州:浙江大学,2006.

[6] 赵晓东. 矩形件优化排样算法的研究与实现[D]. 大连:大连交通大学,2008.

[7] WU Taihsi,CHEN Jengfung,LOW Chinyao,et al. Nesting of two-dimensional parts in multiple plates using hybrid algorithm[J]. International Journal of Production Research,2003,41(16):3883-3900.

[8] 韩玉兵. 遗传算法数学机理分析[J]. 河海大学学报(自然科学版),2001,29(3):92-94. (HAN Yubing. Analysis on mathematic mechanism of genetic algorithm[J]. Journal of Hohai University (Natural Sciences),2001,29(3):92-94. (in Chinese))

[9] 杨海霞,谢轶鹏. 一种改进的遗传算法及其在结构优化设计中的应用[J]. 河海大学学报(自然科学版),2007,35(6):659-663. (YANG Haixia,XIE Yipeng. An improved genetic algorithm and its application to structural design optimization[J]. Journal of Hohai University (Natural Sciences),2007,35(6):659-663. (in Chinese))

[10] CHARALAMBOUS C,FLESZAR K. A constructive bin-oriented heuristic for the two-dimensional bin packing problem with guillotine cuts[J]. Computers & Operations Research,2011,38(10):1443-1451.

[11] JAKOBS S. On genetic algorithms for the packing of polygons[J]. European Journal of Operational Research,1996,88(1):165-181.

[12] LIU Dequan,TENG Hongfei. An improved BL-algorithm for genetic algorithm of the orthogonal packing of rectangles[J]. European Journal of Operational Research,1999,112(2):413-420.

[13] 刘海明,周炯,吴忻生,等. 基于改进最低水平线方法与遗传算法的矩形件排样优化算法[J]. 图学学报,2015,36(4):526-531. (LIU Haiming,ZHOU Jiong,WU Xinsheng,et al. Optimization algorithm for rectangle packing based on improved lowest horizontal method and genetic algorithm[J]. Journal of Graphics,2015,36(4):526-531. (in Chinese))

[14] WEI Lijun,OON W C,ZHU Wenbin,et al. A skyline heuristic for the 2D rectangular packing and strip packing problems[J]. European Journal of Operational Research,2011,215(2):337-346.

[15] 韩喜君,丁根宏. 矩形件优化排样问题的混合遗传算法求解[J]. 计算机技术与发展,2006,16(6):219-221. (HAN Xijun,DING Genhong. Solving the packing problem of rectangles with hybrid genetic algorithm[J]. Computer Technology and Development,2006,16(6):219-221. (in Chinese))

[16] 姜启源,谢金星,叶俊. 数学模型[M]. 5 版. 北京:高等教育出版社,2018.