

不确定环境下预制构件生产鲁棒性调度研究

于森,兰宁,许音

(沈阳建筑大学管理学院,辽宁 沈阳 110168)

摘要:在确定了预制构件生产工艺和资源约束情况的基础上,分析了预制构件生产调度计划对鲁棒值的影响;基于关键链技术制定了考虑二次资源冲突的集中缓冲进度计划,构建了以预制构件生产完工时间最短和鲁棒值最大为目标函数的双层规划模型,并通过嵌套式遗传算法对该模型进行求解。实证结果表明:基于该模型得出的调度计划鲁棒值最大且在面对不确定环境时更为稳定。

关键词:预制构件;生产调度;双层规划;遗传算法;鲁棒性

中图分类号:TU756

文献标志码:A

引用格式:于森,兰宁,许音.不确定环境下预制构件生产鲁棒性调度研究[J].沈阳建筑大学学报(社会科学版),2024,26(1):43-49.

装配式建筑是中国近年来大力发展的建筑形式,预制构件作为装配式建筑最重要的组成部分,其生产问题已受到广泛关注。与传统现浇建筑相比,装配式预制构件的生产过程往往由于工序延迟、资源短缺等问题导致其控制难度较大、项目调度情况更为复杂、环境不确定性更强,这些困难导致其对生产管理的专业化程度要求较高^[1],预制构件在生产过程中更容易受到环境影响,仅依靠生产经验或调度规则很难保证预制构件的有序生产,一旦发生生产调度混乱问题将会导致工期的延误与成本的增加。因此,在面对构件生产过程的不确定环境时,可以借助鲁棒性(Robustness)概念使预制构件生产过程更加稳定,即在调度计划的某些参数扰动下依然能够维持其原有性能,抵抗不确定性因素的干扰,保证预制构件生产按时完成。因此开展不确定环境下预制构件生产鲁棒性调度方面的研究意义重大。

目前,预制构件生产车间的研究、资源受限项目调度问题的研究已经被学者们高度关注。一些学者从生产过程入手并结合现实生产条件对生产模型进行了研究与创新,Chan W T等^[2]在构建预制构件生产调度模型时考虑到养护窖的数量约束,由此得到优化后的预制构件生产调度计划;Ko C H等^[3]以Chan W T等的研究为基础在约束条件方面增加了缓冲区约束,得出了更符合实际情况的生产调度计划。资源受限工程调度问题(Resource-Constrained Project Scheduling Problem, RCPSP)是一类经典优化问题,在研究预制构件生产过程中能够发挥重要作用,但由于其并不能很好地应对不确定因素,往往导致了实际结果与计划的偏离。Goldratt E M^[4]提出了对基准调度计划进行调整的关键链法(Critical Chain Method, CCM),即在关键链与非关键链中间插入相应的缓冲区以确保调度计划的稳定,其在项目调度上起到

了重要作用,从某种角度上说,CCM 也可看作是鲁棒性调度。近年来,有学者在传统关键链技术基础上从鲁棒性角度进行了更深入的研究。Al - Fawzan M 是对鲁棒性进行研究的首位学者,他将实际资源约束情况纳入考虑范畴,计算得出了工序间自由时差,并以自由时差之和衡量调度计划的鲁棒性^[5];张静文等^[6]在研究关键链技术时考虑到了二次资源冲突问题,以最大化鲁棒性为目标提出了冲突消除策略,由此得到了稳定性更高的调度计划。

上述研究大多集中在理想确定环境下的预制构件生产调度问题,也有部分学者以项目生产装配全阶段作为一个整体进行鲁棒性研究,总体来说,对于预制构件生产阶段的鲁棒性研究较少,然而预制构件的生产往往处于复杂多变的环境,其对稳定性的要求更高。基于此,笔者基于预制构件的生产工序及资源约束情况对预制构件生产的鲁棒性调度问题进行研究,运用遗传算法生成调度方案后借助鲁棒性评价指标进行分析与评价,以期获得更加稳定的生产调度计划。

一、问题描述

预制混凝土构件生产工序如图 1 所示,共包含 6 道工序,可大致分为并行工序和串行工序,并行工序是在满足资源约束条件的情况下可同时处理多个工件的工序,若不可同时处理则为串行工序^[7]。在加工构件时,需要生产 j 个工件,所有工件都需要依次完成 N_k 道工序且这些工件的加工顺序相同。预制构件的生产需满足以下要求:①工件必须按照工艺要求进行加工;②混凝土浇筑、混凝土养护及储存 3 道工序必须按照相应步骤进行加工且不可抢占;③除混凝土养护和储存外其余工序允许等待。

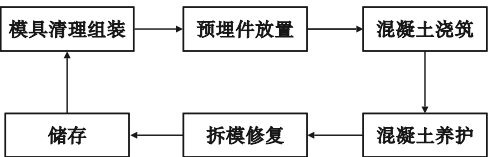


图 1 预制混凝土构件生产工序

二、预制构件生产调度模型

基于以上描述建立预制构件生产调度模型,相关参数及定义如表 1 所示。

表 1 相关参数含义

名称	含义
j	构件编号, $j = (1, 2, \dots, 8)$
N_k	工序编号, $k = (1, 2, \dots, 6)$
(j, N_k)	预制构件 j 的第 N_k 道工序
$P(j, N_k)$	预制构件 j 的第 N_k 道工序加工持续时间
$F(j, N_k)$	预制构件 j 的第 N_k 道工序加工完成时间
C	生产完工时间
D	工作天数
H_w	日常正常工作时间
H_A	加班时间
H_L	日常非正常工作时间
T	累计完工时间
C_{ij}	养护前构件 i 在第 j 道工序的完工时间
S_B	构件浇筑前的等待时间

1. 优化目标

在对预制构件生产调度计划进行评价时需计算生产完工时间,即最后一个构件的最后一道工序的加工完成时间。笔者将预制构件生产最短完工时间 ($\min C$) 设定为上层目标,其计算式为

$$\min C = \min F(j, N_6)$$

(1)

2. 约束条件

(1) 工序约束

工序分为可中断工序和不可中断工序。首先计算可中断工序的完工时间。模板清理组装、预埋件放置、拆模修复这 3 道工序在作业时均可中断,若在正常作业时段内构件不能完成加工,则可将加工完成时间延长至第二天继续加工直至完成,可中断工序的完工时间为

$$F(j, N_k) = \begin{cases} T & T \leq 24D + H_w \\ T + H_L & T > 24D + H_w \end{cases}$$

(2)

式中: $k = 1, 2, 5$ 。

不可中断工序包括混凝土浇筑和混凝土养护,其完工时间需分别进行计算。

①混凝土浇筑。该工序无法间歇作业,如果在当天浇筑作业不能完成,那么必须加

班进行,或将此作业推迟至第二天进行,预制构件混凝土浇筑工序完工时间为

$$F(j,N_k) \begin{cases} T & T \leq 24D + H_w + H_A \\ 24(D+1) + P_{j,k} & T > 24D + H_w \end{cases} \quad (3)$$

式中: $k=3, H_A \leq H_w$ 。

②混凝土养护。该工序必须在浇筑作业结束后进行且不可抢占加工,养护窖可同时养护多种混凝土构件,因此混凝土浇筑工序结束后需立即进行该工序的作业,其完成时间为

$$F(j,N_k) = \begin{cases} T & T \leq 24D + H_w \text{ or } T \geq 24(D+1) \\ 24(D+1) & 24D + H_w < T < 24(D+1) \end{cases} \quad (4)$$

式中: $k=4, T=F(j,N_4)$ 。

③储存。该工序作为一道并行工序不需要占用任何劳动力资源,因此在上一道工序(拆模修复)结束以后即可进行,该工序的完成时间为

$$F(j,N_k) = \begin{cases} T^* & T \leq 24D + H_w \text{ or } T \geq 24(D+1) \\ 24(D+1) & 24D + H_w < T < 24(D+1) \end{cases} \quad (5)$$

式中: $k=6, T^* = F(j,N_4) + P(j,N_5)$ 。

$$S_{\beta} = \begin{cases} \max(C_{j2}, C_{(j-1)3}) & i < M \\ \max(C_{j2}, C_{(j-1)3}, \min(C_{\leq j-M \leq N \leq j+1})) - P(j,N_3) & i \geq M \end{cases} \quad (7)$$

式中: M 为养护窖内可容纳的构件数量; C_{j2} 和 C_{j3} 分别为预制构件 j 的第 2 道和第 3 道工序的生产完工时间。

三、预制构件生产进度计划的制定

1. 平均时间 t_{ij} 的确定

构件加工时间长短存在不确定性,因此对于构件加工平均时间 t_{ij} 的确定选用模糊时间估计法^[8],即在计算求得构件加工的最快完成时间(t_{aij})、最可能完成时间(t_{mij})及最慢完成时间(t_{bij})后进行去模糊化处理,得到构

(2)资源约束

资源约束分为模板约束和养护窖数量约束两种情况,其工序等待时间需分别进行计算。

①模板约束。在生产预制构件时,模板是极其重要的资源,其几乎贯穿构件生产的整个过程^[9]。由于当前建筑市场对构件的生产要求较高,所需的模板样式也较多,这使得构件的生产成本增高。构件厂商为降低生产经营成本对一些类型不同但结构相似的构件使用同一个模板,即“一模多用”,这对生产调度提出了更高要求。若在生产某个预制构件时发现并无空闲模板,则必须等到所有采用该模板的构件都拆模完成后进行此构件的模板清理工作,假设第 j 个预制构件等待 A 模板的时间为

$$F(j,N_0) = \min X_A \{ \forall y \{ F(j,N_5) \} \} \quad (6)$$

式中: N_0 为等待模板工序; X_A 为 A 模板的数量; y 为其余使用 A 模板的构件。

②养护窖数量约束。高温蒸汽养护在混凝土浇筑完成之后即可进行,养护窖有空位时可以同时养护多种不同类型的构件^[9]。此外,由于构件生产工艺的约束,混凝土浇筑与混凝土养护两道工序之间不能存在时间间隙,否则将会影响混凝土的强度。因此,在进行混凝土浇筑作业之前需要注意养护窖内有无空位,若存在空位可以进行浇筑;反之则需等待,构件浇筑前的等待时间为

$$t_{ij} = \frac{t_{aij} + 4t_{mij} + t_{bij}}{6} \quad (8)$$

2. 集中缓冲进度计划

在基准进度计划基础上结合关键链技术制定预制构件生产集中缓冲进度计划,并运用启发式协调策略解决二次资源冲突问题^[10]。

对于关键工序与非关键工序的判断依据为工序之间时差的大小,按照工艺要求与资源约束情况计算各工序的开始时间和结束时

间,构件 j 在满足紧后工序开始时间的前提下可以后移的最大时间量为该工序的时差,若时差存在即为非关键工序;若时差不存在则为关键工序^[11]。

在制定进度计划时,对于缓冲尺寸的计算选用剪切粘贴法,由于项目计划制定者在估计工序持续时间时加入了大量的安全时间,但是往往由于人为因素导致这些安全时间在项目实施过程中被浪费掉,因此要将这些安全时间进行削减,所以在计算缓冲尺寸时需要将安全时间剔除^[12]。非关键工序上的缓冲尺寸为剩余时间的 50%,用 FB 表示;关键工序上缓冲尺寸为剩余时间的 30%,用 PB 表示,由此得到项目缓冲尺寸及输入缓冲尺寸为

$$PB = 30\% \sum_{i=1}^n \Delta D_i \tag{9}$$

$$FB = 50\% \sum_{i=1}^n \Delta D_i \tag{10}$$

式中: PB 为项目缓冲尺寸; FB 为输入缓冲尺寸; ΔD_i 为安全时间。

3. 考虑二次资源冲突的启发式解决策略

在基准调度计划的基础上插入缓冲尺寸会导致部分工序后移,这将引发养护窖或模板的资源冲突,为了解决此问题,有学者提出了一种启发式解决策略^[13],具体步骤如下。

①以工序的最早开始时间为起点,按照不同工序开始时间的先后插入 FB ,若与当前 FB 发生资源冲突,则需要解决当前与紧后 FB 之间的冲突。

②若因插入 FB 引起的冲突涉及关键工序和非关键工序,为保证生产能够按时完成且关键工序按照计划进行生产,可后移后续非关键工序的开始时间,且后续工序的 FB 要吸收相应的时间量。

③若因插入 FB 引起的冲突只涉及关键工序,则需要将后续关键工序的开始时间后移,此时 PB 发生作用,后移的时间量将被 PB 所吸收,若后移时间量大于 PB ,将会导致构件生产的完工时间延长。

4. 鲁棒性指标

通过上述启发式策略解决二次资源冲突

问题,能保证调度计划在面对不确定环境时更加稳定。这是因为在基准进度计划的基础上制定集中缓冲进度计划,并根据启发式解决策略消除二次资源冲突的影响后得到新的集中缓冲进度计划,能够在最大程度上缩减基准进度计划与新集中缓冲进度计划的偏离值,且能使 FB 都紧跟在非关键工序之后,最终能使非关键工序调度情况更加稳定。

基于上述两次调度计划的鲁棒性指标进行结果评价,即评价这两种缓冲形式对调度计划稳定性的贡献度。计算 FB 在鲁棒性评价指标中的贡献度,即进行第二次调度后的 FB 与构件生产完工时间的比值,且当 FB 小于非关键工序本身的自由时差时,虽不会出现资源冲突的情况,但由于其属于时间空隙,因此直接选用自由时差作为 FB ; PB 的贡献在于推迟 FB 后导致关键工序的开始时间向后移动,此时后移量被 PB 吸收。根据 FB 与 PB 的贡献度构建最大化预制构件生产调度计划鲁棒性指标,并将此作为下层目标。

$$\max R = \frac{\sum_{m=1}^M \max(FB_{ij}^{m'}, ff_{ij})}{f_j'} + \frac{PB}{\sum_{j \in CC} d_j} \tag{11}$$

式中: R 为考虑资源冲突后的生产进度计划鲁棒性; M 为输入缓冲总数; m 为输入缓冲序号且 $m = 1, 2, \dots, M$; f_j' 为经过二次调度后的生产完工时间; $FB_{ij}^{m'}$ 为考虑二次资源冲突后的输入缓冲尺寸; ff_{ij} 为非关键链工序的时差大小; d_j 为工序 j 的完工时间; CC 为关键链。

四、嵌套式遗传算法设计

为有效求解优化模型,笔者将该模型看作双层规划问题,运用嵌套式遗传算法进行求解,该算法核心内容主要包括染色体编码、生成初始种群、交叉变异等要素,其按照优胜劣汰的方式将优秀个体保留下来,其他个体将被淘汰,直到搜索出问题的最优解^[14]。嵌套式遗传算法的具体步骤如下。

1. 染色体编码

采用整数编码的方式将各组按顺序进行编码,从左到右的基因位置表示加工顺序。例如,编码序列[1,2,3,4,5,6,7,8,9]表示同一工序所有构件被加工的顺序依次为第 1 位至第 9 位。

2. 初始化种群

随机生成一个种群规模,通过计算适应度值判断其是否满足上层目标及约束条件,若满足则继续初始化下层种群,若不满足则进行步骤 3^[15]。

3. 交叉、变异

寻找两个符合交叉条件的个体,假定为 P_1 和 P_2 ,随机交换两个个体并进行修补以此形成两个新个体,随后寻找符合变异条件的个体,通过不断改变基因片段来保证多样性^[16]。

4. 更新种群,判断收敛

将交叉、变异后产生的新解作为当前解保留至下一代。记录当前最优解,直至达到最大迭代次数后输出最优解^[17]。

完整的嵌套式遗传算法流程如图 2 所示。

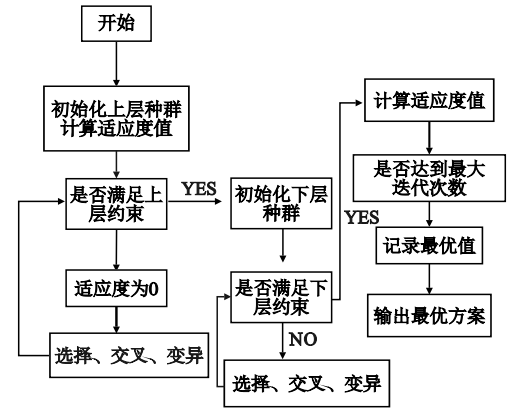


图 2 算法流程

五、案例分析

以沈阳某预制构件厂的外墙板生产线为研究对象进行构件的生产信息收集,工序加工预估时间由生产工人提供(见表 2)。其中, N_1 为模板清理组装; N_2 为预埋件放置; N_3 为混凝土浇筑; N_4 为混凝土养护; N_5 为拆模修复; N_6 为储存。在构件生产过程中,使用 $A、B、C$ 这 3 种模板,其中, A 模板有 3 个、 B 模板有 3 个、 C 模板有 2 个,养护窖数量为 2 个,日常正常工作时间(H_w)为 8 h,加班时间(H_A)不能超过 4 h。

表 2 工序预估时间

构件 编号	各工序预估时间/h ($t_{aij}, t_{mij}, t_{bij}$)					
	N_1	N_2	N_3	N_4	N_5	N_6
1	2.2,2.6,3.0	1.5,1.9,2.3	3.0,3.5,4.0	6.0,8.0,10.0	1.5,1.9,2.3	2.0,3.0,4.0
2	2.2,2.6,3.0	1.5,2.0,2.5	3.0,3.6,4.2	6.0,8.0,10.0	2.0,2.3,2.6	2.0,3.0,4.0
3	2.0,2.5,3.0	2.0,2.3,2.6	3.0,3.6,4.2	6.0,8.0,10.0	1.8,2.2,2.6	2.0,3.0,4.0
4	2.3,2.7,3.1	1.8,2.2,2.6	3.0,3.5,4.0	6.0,8.0,10.0	2.0,2.3,2.6	2.0,3.0,4.0
5	2.0,3.0,4.0	2.3,2.7,3.1	3.3,3.9,4.5	6.0,8.0,10.0	2.0,2.3,2.6	2.0,3.0,4.0
6	2.2,3.1,4.0	2.3,2.8,3.3	2.8,3.8,4.8	6.0,8.0,10.0	1.8,2.2,2.6	2.0,3.0,4.0
7	2.0,2.5,3.0	2.0,2.3,2.6	3.0,3.7,4.4	6.0,8.0,10.0	1.8,2.1,2.4	2.0,3.0,4.0
8	2.2,2.6,3.0	1.8,2.2,2.6	2.8,3.8,4.8	6.0,8.0,10.0	1.8,2.2,2.6	2.0,3.0,4.0

1. 基准调度方案求解

对表 2 中的数据分析,运用式(8)进行去模糊化处理,得到预制构件生产平均时间(见表 3)。

用 matlab r2018b 软件进行编程,设置初始种群规模 $NP=100$,交叉概率 $CR=0.9$,最大迭代次数 $G=300$,对模型进行求解得到构件加工的初始完工时间(见表 4)。

表 3 预制构件生产信息

模板 类型	构件 编号	构件各工序生产平均时间/h					
		N_1	N_2	N_3	N_4	N_5	N_6
A	1	2.6	1.9	3.5	8.0	1.9	3.0
B	2	2.6	1.8	3.6	8.0	2.3	3.0
A	3	2.5	2.3	3.6	8.0	2.2	3.0
B	4	2.7	2.2	3.5	8.0	2.3	3.0
B	5	3.0	2.7	3.9	8.0	2.3	3.0
A	6	3.1	2.8	3.8	8.0	2.2	3.0
C	7	2.5	1.6	3.7	8.0	2.1	3.0
C	8	2.6	1.7	3.8	8.0	2.2	3.0

表4 构件完工时间

构件 编号	完工时间/h					
	N_1	N_2	N_3	N_4	N_5	N_6
7	2.5	4.8	56.8	80.0	243.1	246.1
8	5.1	7.3	60.6	104.0	245.3	266.1
6	8.2	28.0	75.8	128.0	247.5	269.1
4	27.9	30.2	79.3	152.0	266.8	272.1
2	30.5	32.5	82.9	176.0	269.1	292.1
1	50.1	52.0	99.5	200.0	271.0	295.1
5	53.1	55.8	103.4	224.0	290.3	315.1
3	55.6	75.1	107.0	248.0	292.5	318.1

表5 最优调度计划

构件 编号	加工结束时间/h					
	N_1	N_2	N_3	N_4	N_5	N_6
7	2.5	5.1	56.8	80.0	243.1	246.1
8	5.1	8.2	60.6	104.0	245.3	266.1
6	8.2	28.0	75.8	128.0	247.5	269.1
4	27.9	30.5	79.3	152.0	266.8	272.1
2	30.5	50.1	82.9	176.0	269.1	292.1
1	50.1	53.1	99.5	200.0	271.0	295.1
5	53.1	55.8	103.4	224.0	290.3	315.1
3	55.8	80.0	224.0	266.8	292.5	318.1

2. 集中缓冲进度计划

在上述基准进度计划的基础上寻找关键链与非关键链,得到关键链 $CC = (j, N_k)$, 即 $CC = [(1,1)(1,3)(1,4)(1,5)(1,6)(3,1)(3,4)(3,6)(5,1)(5,4)(5,6)(2,1)(2,4)(2,6)(8,1)(8,4)(8,6)(6,1)(6,6)(6,4)(7,1)(7,4)(7,6)(4,6)]$; 非关键链 $NC = (j, N_k)$, 即 $NC_1 = (3, 1)$, $NC_2 = (7, 2)$, $NC_3 = (8, 2)$, $NC_4 = [(6, 2)(4, 2)]$, $NC_5 = (2, 2)$, $NC_6 = (1, 2)$, $NC_7 = [(5, 2)(3, 2)]$, $NC_8 = [(8, 3)(6, 3)(4, 3)(2, 3)(1, 3)(5, 3)(3, 3)]$, $NC_9 = (3, 4)$ 。

得到关键链与非关键链后,根据式(9)、式(10)分别计算 FB 、 PB ,将缓冲尺寸插入相应位置,由于调度计划中存在多条非关键链,需要插入多个输入缓冲,此时需要考虑插入缓冲尺寸后引起的二次资源冲突即模板数量和养护数量的冲突,根据消除二次冲突的策略不断进行调整,最终得到 $FB_1 = 5.40$, $FB_2 = 0.58$, $FB_3 = 0.90$, $FB_4 = 1.25$, $FB_5 = 17.60$, $FB_6 = 1.10$, $FB_7 = 4.90$, $FB_8 = 117.00$, $FB_9 = 18.80$, $PB = 15.74$ 。

3. 鲁棒指标分析

根据式(11)可以得到考虑二次冲突后的集中缓冲进度计划的鲁棒性指标值为0.5761。鲁棒指标最大的调度计划即为最优的调度计划,其在预制构件生产调度过程中最稳定,更能保证生产按时完成,最优调度计划结果如表5所示。

使用嵌套式遗传算法得到的收敛效果如图3所示,其中横坐标表示迭代次数,纵坐标表示每代最优目标函数值,根据迭代曲线可

以发现使用该算法求解得到的收敛性较好,故可以用其来解决此类问题。

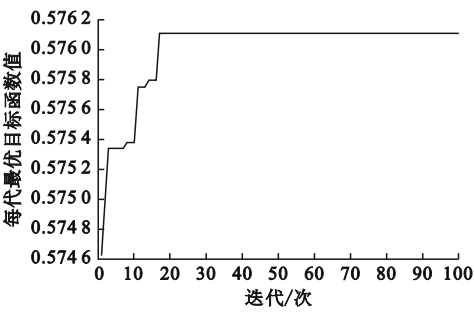


图3 收敛曲线

六、结 语

根据预制构件生产工艺要求,制定了生产调度计划,选取嵌套式遗传算法对模型进行求解,得到了完工时间最短的调度计划。在此基础上制定了集中缓冲进度计划,运用启发式策略解决了二次资源冲突问题,进而找出了鲁棒性指标并对其进行评价,最终得到了鲁棒性最大且完工时间最短的调度计划。实证结果表明:在预制构件生产阶段对生产调度计划进行鲁棒性评价,不仅能达到提高预制构件生产计划稳定性的目的,还能提高生产订单的管理效率。

参考文献:

[1] 陈伟,容思思. 装配式住宅项目多空间鲁棒性调度研究[J]. 建筑经济,2017,38(1):69-73.

[2] CHAN W T,HAO H. Production scheduling for precast plants using a flow shop sequencing model[J]. Journal of computing in civil engineering,2002,16(3):165-174.

[3] KO C H, WANG S F. Ga-based decision support systems for precast production planning [J]. Automation in construction, 2010, 19 (7) : 907 – 916.

[4] GOLDRANT E M. Critical chain [M]. Great Barrington: North river press, 1997.

[5] AL-FAWZAN M, HAOUARI M. A bi-objective model for robust resource-constrained project scheduling [J]. International journal of production economics, 2005, 96 (2) : 175 – 187.

[6] 张静文, 乔传卓, 刘耕涛. 基于鲁棒性的关键链二次资源冲突消除策略 [J]. 管理科学学报, 2017, 20 (3) : 106 – 119.

[7] 于森, 赵洁, 焦红超. 基于多目标差分进化算法的预制构件生产重调度研究 [J]. 沈阳建筑大学学报 (社会科学版), 2021, 23 (6) : 589 – 595.

[8] 李聪波, 刘飞, 易茜, 等. 基于关键链的再制造系统不确定性生产调度方法 [J]. 机械工程学报, 2011, 47 (15) : 121 – 126.

[9] 王茹, 班丹梅, 王月, 等. 考虑资源约束的预制混凝土构件生产调度问题研究 [J]. 制造业自动化, 2020, 42 (12) : 63 – 67.

[10] 于森, 徐宁, 马健. 基于关键链技术的装配式鲁棒性项目调度方法 [J]. 沈阳建筑大学学报 (社会科学版), 2022, 24 (4) : 392 – 398.

[11] 吴超. 关键链管理法在资源受限多项目调度中的应用 [J]. 中国管理信息化, 2016, 19 (17) : 87 – 89.

[12] 林晶晶. 考虑资源可替代性的关键链识别与缓冲设置方法研究 [D]. 成都: 西南交通大学, 2011.

[13] 张静文, 刘金波, 李若楠. 基于关键链技术的项目鲁棒性优化调度方法研究 [J]. 科技管理研究, 2013, 33 (6) : 217 – 220.

[14] 喻寿益, 邝溯琼. 嵌套式模糊自适应遗传算法 [J]. 控制工程, 2010, 17 (1) : 75 – 79.

[15] 王中原, 陈小波. 基于遗传算法的板类预制构件生产中的模板排布优化研究 [J]. 工程管理学报, 2019, 33 (1) : 45 – 49.

[16] 赵建峰, 朱晓春, 汪木兰, 等. 基于遗传算法柔性制造系统生产调度的优化与仿真 [J]. 制造业自动化, 2010, 32 (5) : 156 – 159.

[17] 张大斌, 王婧, 刘桂琴, 等. 模糊自适应遗传算法 [J]. 计算机工程与设计, 2008 (18) : 4783 – 4785.

Research on Robustness Scheduling of Prefabricated Components under Uncertain Environment

YU Miao, LAN Ning, XU Yin

(School of Management, Shenyang Jianzhu University, Shenyang 110168, China)

Abstract: Based on determining manufacturing technique of prefabricated components and resource constraint, the influence of production scheduling plan of prefabricated components on the robustness value was studied. On the basis of the key chain technology, a centralized buffer schedule considering secondary resource conflicts was constructed, and a two-layer programming model with the shortest production completion time and maximum robustness value as the objective function was established. A nested genetic algorithm is designed to solve the model. Empirical results show that the obtained scheduling plan on the basis of this model has the largest robustness value and is more stable in the face of uncertain environment.

Key words: prefabricated component; production scheduling; bi-level programming; genetic algorithm; robustness

(责任编辑:徐聿聪 英文审校:林 昊)