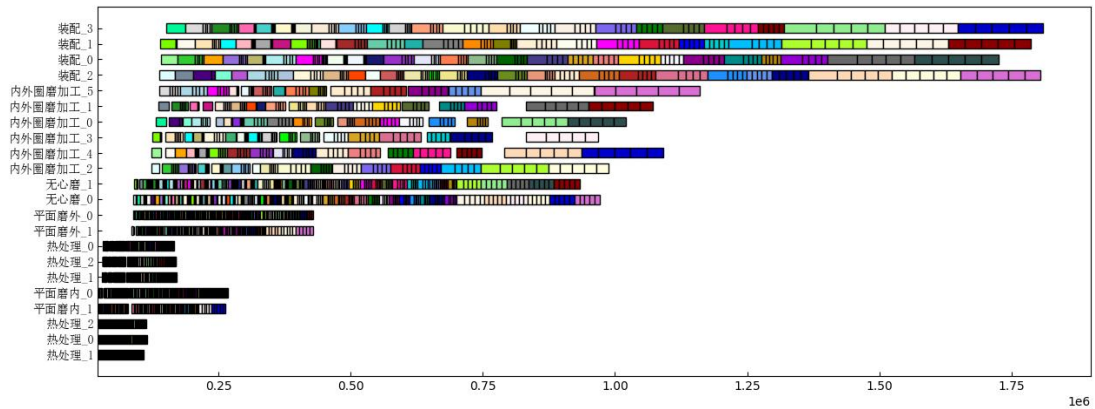


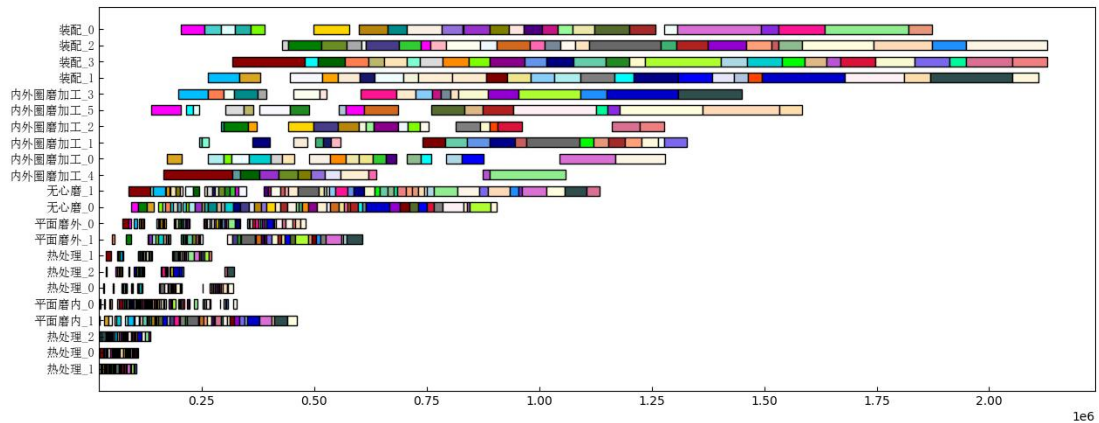
基于工艺路径柔性 and 批量流的柔性作业车间调度问题

1 整体思路简述

通过对给定数据集进行分析,可建模为一个含加工工艺柔性[1-2]、启动时间和序列依赖准备时间的柔性作业车间调度问题,优化目标为最小拖期和最大完工时间。对数据进行简单实验和分析,迭代次数和种群个数一致,且规模较小,得到结果如下图 1 和图 2 所示,其中图 1(a)为采用一致分批的排产结果,图 1(b)为整批的排产结果,图 2 为两种排产方式的最终解的情况,显而易见,考虑分批的最终解的质量远优于整批的解的质量。因此,将通过引入批量流技术,以保证求解速率的同时保证最终解的质量。



(a) 考虑分批



(b) 不考虑分批

图 1 两种调度方案的排产 Gantt 图

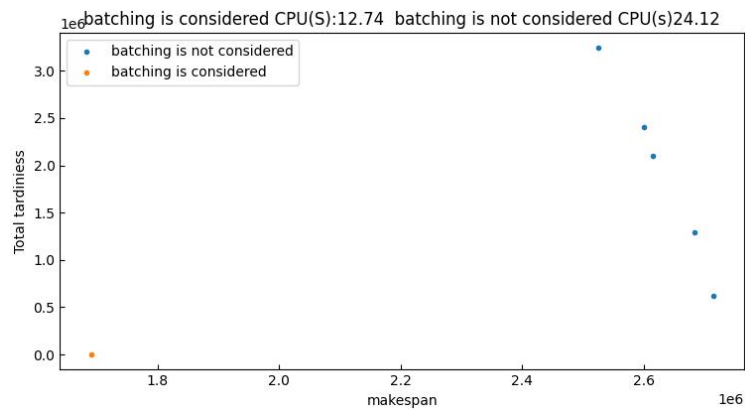


图 2 两种排产方案的最终 pareto 前沿解

批量流技术是将一个给定的批次分割成若干较小的子批，以实现在一个多阶段制造车间中同一批次在不同阶段中的并行加工。也就是说，批次的一个子批一旦在某一阶段完成加工，就可以立即运输到下游阶段。因此，它的好处是减少生产周期，从而更快地生产和交付产品。此外，它还具有减少在制品库存、线边仓和空间需求的优点。批量流的划分方法可分为三类：等量分批、一致分批和可变分批。在等量分批下，同一批次的不同子批规模是相同的，并且在不同阶段保持一致；在一致分批下，批次的不同子批规模可能不同，但在不同阶段保持一致；而在可变分批下，批次的不同子批规模可能不同，并且在不同阶段也保持变化。

由于所给算例规模较大且各工件批的差异较大，将采用一致分批的方式。相对等量分批而言，一致分批包含等量分批但解空间优于等量分批；可变分批虽包含一致分批，更有可能得到较佳的目标值，但因其对于每道工序需要重新分批的复杂性，现场难以管控。因而折中方案的一致分批是我跟倾向的一种批量流技术。

因为问题的 NP-hard 特性，为了解决所给数据的排产调度问题，将采用多目标进化算法进行求解，针对调度问题的耦合问题，如批次排序、机器分配、工艺路径选择和批量分割，将在后文给出合理的编码方式对解空间进行表达，并结合启发式规则对编码进行解码操作。

3 问题描述和建模

本文将所给问题建模为考虑柔性工艺路径的多目标变分批柔性作业车间调度问题(Multi-objective Hybrid Flexible job shop Scheduling Problem with Consistent Sub-lots and processing flexibility , MOFJSP_CS&PF)。在 MOFJSP_CS&PF 中，一系列工件批次要经过多道工序进行加工，其中，工件批的某些局部加工工序顺序不定，每道工序可由若干的不同的机器进行加工，并且每个工件批次包含若干待加工工件，待加工工件的数量称为批次规模。在变分批策略下，每个工件批被分割成若干子批，子批的数量可根据现场需求设定最大限定值。每个子批包含不同数量的待加工工件，子批待加工工件的数量称为子批规模。批次的子批数量和子批规模在不同阶段是发生变化的。来自同一批次的不同子批在各道工序可由不同机器上进行加工，子批内的待加工工件是连续性地加工。也就是说，一个子批的加工时间是子批规模和单个工件加工时间（一个工件地加工时间）的乘积。当工件完成当前工序加工时，需要静置一段时间才能进行下一道工序；此外，当工件放入机器进行加工前，若机器的上一加工工件与当前工件不一样，则需要换模，不同的工件具有不同的静置时间和换模时间。

3.1 数学模型

MOFJSP_CS&PF 需要确批次分割、批次排序、机器分配以及各批次子批的工艺路径。为了表达工艺路径约束，引入了变量 $Z_{j,e,r,r'}$ ，以此来解决工艺路径指派问题，并使用基于 MTZ 约束的方法，消除工艺路径的子环。定义的相关参数符号和决策变量如下：

参数和索引：

n	批次数目
m	阶段数目
I	阶段集合， $I = \{1, \dots, i, \dots, m\}$
J	批次集合， $J = \{1, \dots, j, \dots, n\}$
σ_i	在阶段 i 的机器数
M_i	机器集合， $M_i = \{1, \dots, k, \dots, \sigma_i\}$
T_j	批次 j 中加工单元的数目
L	每个批次的最大子批数
e	子批索引， $e \in \{1, 2, \dots, L\}$
r	工艺路线索引， $r \in \{1, \dots, R\}$
$p_{i,j,k}$	批次 j 在阶段 i 的机器 k 上的单位加工时间
$t_{i,j}$	批次 j 在阶段 i 的静置时间
$f_{j,e,k,j',e'}$	响铃工件批 j 的 e 子批和 j' 的 e' 子批在机器 k 上的准备时间

H_i 阶段 i 的启动时间

d_j 批次 j 的交货期

Q 一个充分大的正数

决策变量:

$B_{i,j,e}$ 批次 j 的子批 e 在工序 i 上的开始加工时间

$E_{i,j,e}$ 批次 j 的子批 e 在工序 i 上的完成加工时间

$S_{j,e}$ 批次 j 的子批 e 在工序 i 上的规模

$W_{i,j,e}$ 二进制变量, 批次 j 的子批 e 上的大规模大于 0 时取 1, 否则取 0.

$D_{i,j,k,e}$ 二进制变量, 批次 j 的子批 e 在工序 i 上分配给机器 k 上时取 1, 否则取 0.

$X_{j,e,j',e}$ 二进制变量, 批次 j 和批次 j' 为同种类型工件时取 1, 否则取 0.

$Y_{i,j,e,j',e',k}$ 二进制变量, 批次 j 的子批 e 优先批次 j' 的子批 e' 在工序 i 上的机器 k 上时取 1, 否则取 0.

$Z_{j,e,r,r'}$ 二进制变量, 批次 j 的子批 e 的工艺路径从工序 r 到工序 r' 时取 1, 否则取 0.

模型的目标定义为最小化最大完工时间和最小化总拖延时间

目标函数:

$$\text{Min } \max(E_{m,j,L})$$

$$\text{Min } \sum_{j=1}^n (\max(0, E_{m,j,L} - d_j))$$

约束函数:

$$S_{i,j,e} \geq 0 \quad \forall (i, j, e) \quad (1)$$

$$\sum_{e=1}^{I_j} S_{i,j,e} = T_j \quad \forall i \in I, j \in J \quad (2)$$

$$\sum_{k=1}^{\sigma_j} D_{i,j,k,e} = 1 \quad \forall (i, j, e) \quad (3)$$

$$B_{i,j,e} \geq H_i \quad \forall (j, e) \quad (4)$$

$$\sum_{r=1}^R Z_{j,e,r,r'} = 1 \quad \forall r', r \neq r' \quad (5)$$

$$\sum_{r'=1}^R Z_{j,e,r,r'} = 1 \quad \forall r, r \neq r' \quad (6)$$

$$\mu_i - \mu_j + R \times Z_{j,e,r,r'} \leq R - 1 \quad \forall r, r', r, r' \neq 0, r \neq r' \quad (7)$$

$$\mu_i \geq 0, \mu_i \in R, \forall (i) \quad (8)$$

$$Y_{i,j,e,j,e',k} + Y_{i,j,e,j',e',k} \leq 1 \quad \forall (j, e, j', e', k) \quad (9)$$

$$Y_{i,j,e,j,e',k} + Y_{i,j,e,j',e',k} \leq D_{i,j,k,e} + D_{i,j',k,e} \quad \forall (i, j, j', k, e, e') \quad (10)$$

$$D_{i,j,k,e} + D_{i,j',k,e'} - 1 \leq Y_{i,j,e,j,e',k} + Y_{i,j',e,j,e',k} \quad \forall (i, j, j', k, e, e') \quad (11)$$

$$E_{i,j,e} - B_{i,j,e} = p_{i,j,k} \times S_{j,e} \quad \forall (i, j, e) \quad (12)$$

$$B_{i',j,e} - E_{i,j,e} - t_{i,j} - (1 - Z_{j,e,i,i'}) \times Q \geq 0 \quad \forall (i, i', j, e), i \neq i' \quad (13)$$

$$B_{i,j,e} - E_{i,j',e'} - f_{i,j} \times X_{j,e,j',e'} + Q \times (3 - Y_{i,j',e',j,e,k} - D_{i,j,k,e} - D_{i,j',k,e'}) \geq 0 \quad \forall (i, j, k, j', e, e') \quad (14)$$

其中, 式(1)保证在各工序每个子批的规模要大于等于 0; 式(2)保证在每一工序对于给定的批次, 子批规模之和等于批次的总规模; 式(3)保证了每个批次都要经过所有工序的加工, 并且在每道只能分配到一台机器上; 式(4)要求在各工序, 每个批次的各子批的开始加工时间要大于它的启动时间; 式(5)-(8)表示任意一个工件只能选择一条工艺路线; 式(9)-(11)共同定义了变量 $D_{i,j,k,e}$ 和 $Y_{i,j,e,j',e',k}$ 的关系, 两个变量的取值决定了批次的机器分配和调度次序; 式(12)保证了每个子批的加工不能被打断; 式(13)保证了每个子批要按照给定工艺路径进行加工; 式(14)保证了机器上的加工时间约束, 当子批类型不一样时则需要换型时间。

4 算法设计

4.1 编码设计

根据问题描述, 求解 MOHFSP_VS&PF 需要同时解决四个相互耦合的问题, 即工艺路径选择、批次分割、批次排序和机器分配。利用元启发式算法求解优化问题时, 解的编码需要承载必要的信息, 能够利用解码规则翻译成详细的调度方案。在有限的计算成本限制下, 批次序列编码在研究中得到了广泛应用并且表现优异。批次序列表示在第一阶段的批次调度顺序, 在后续阶段中, 批次的调度顺序由相应的启发式规则得到, 机器分配同样如何。由于所给数据考虑了序列依赖准备时间, 机器上相同物料编码的工件连续加工可省略准备时间(换型时间), 基于批次序列编码的表示方法难以体现这一方面的优化。因而, 本文的解的编码分为三个部分两种形式, 包括固定编码形式和可变编码形式。

固定编码形式包含两个部分, 第一个部分为工艺路径选择矩阵的集合, 即 $I = \{\xi_1, \dots, \xi_j, \dots, \xi_n\}$, 其中 $\xi_j = \{r_1, \dots, r_e, \dots, r_L\}$, r_e 表示可选工艺路径索引, n 表示批次的数量, L 表示最大分批数; 第二个部分为批次分割矩阵集合, 即 $\Delta_n = \{\psi_1, \dots, \psi_j, \dots, \psi_n\}$, 其中 $\psi_j = \{S_{j,1}, \dots, S_{j,e}, \dots, S_{j,L}\}$, 其中 $S_{j,e}$ 表示第 j 个批次的第 e 个子批的批量。

可变编码主要针对批次排序编码, 第一种形式为 n 维向量 $\pi_n^1 = \{\pi_1, \dots, \pi_j, \dots, \pi_n\}$, 其中 π_j 表示批次索引, n 表示批次的数量; 第二种形式为长度为 $n \times m$ 的一维的向量 $\pi_n^2 = \{\tau_1, \dots, \tau_i, \dots, \tau_m\}$, 其中 $\tau_i = \{\pi_1, \dots, \pi_j, \dots, \pi_n\}$ 。

4.2 解码方法

解码过程是将解的编码翻译成可行调度方案的过程。针对前文提出的编码方式, 求解 MOHFSP_VS&PF, 解码有两种方式。

第一种解码方式主要针对可变编码的第一种形式, 解码过程简述如下:

- 步骤 1** 在第一阶段, 即 $i=1$, 依次从序列 $\pi_n^1 = \{\pi_1, \dots, \pi_j, \dots, \pi_n\}$ 中取出 π_j , $I = \{\xi_1, \dots, \xi_j, \dots, \xi_n\}$ 中取出 ξ_j , $\Delta_n = \{\psi_1, \dots, \psi_j, \dots, \psi_n\}$ 中取出 ψ_j 。执行以下流程:
- 步骤 2** 通过 ψ_j 和 ξ_j 确定了批次 π_j 的子批分批方案以及各子批的工艺路径, 从第一个子批开始, 依次从 $\psi_j = \{S_{j,1}, \dots, S_{j,e}, \dots, S_{j,L}\}$ 和 $\xi_j = \{r_1, \dots, r_e, \dots, r_L\}$ 中取出 r_e 和 $S_{j,e}$ 进行解码, 根据“先到先得”规则, 确定当前阶段各子批的分配机器和机器空闲时间, 并基于加工约束, 依次调度各子批。
- 步骤 3** 在其他阶段, 即 $i=2, \dots, m$, 根据“批次优先”规则或“交期早者优先”规则, 得到新的批次排序序列 $\pi_n^i = \{\pi_1', \dots, \pi_j', \dots, \pi_n'\}$, 依次从序列 π_n^i 中取出 π_j' , 重复步骤 3。

步骤 4 重复上述步骤, 直到所有阶段解码完毕。

第二种解码方式主要针对可变编码的第二种形式, 解码过程简述如下:

- 步骤 1** 依次从序列 $\pi_n^2 = \{\tau_1, \dots, \tau_m\}$ 中取出 τ_i , τ_i 表示第 i 阶段各批次的加工顺序, 获取当前阶段的可加工机器集 σ_i 。
- 步骤 2** 依次从并 $\tau_i = \{\pi_1, \dots, \pi_j, \dots, \pi_n\}$ 读取 π_j , 并从 $I = \{\xi_1, \dots, \xi_j, \dots, \xi_n\}$ 和 $\Delta_n = \{\psi_1, \dots, \psi_j, \dots, \psi_n\}$ 中取出对应批次的 ξ_j 、 ψ_j 。
- 步骤 3** 通过 ψ_j 和 ξ_j 确定了批次 π_j 的子批分批方案以及各子批的工艺路径, 从第一个子批开始, 依次从 $\psi_j = \{S_{j,1}, \dots, S_{j,e}, \dots, S_{j,L}\}$ 和 $\xi_j = \{r_1, \dots, r_e, \dots, r_L\}$ 中取出 r_e 和 $S_{j,e}$ 进行解码, 根据“先到先得”规则, 确定当前阶段各子批的分配机器和机器空闲时间, 并基于加工约束, 依次调度各子批。
- 步骤 4** 重复上述步骤, 直到序列 $\pi_n^2 = \{\tau_1, \dots, \tau_m\}$ 读取完毕。

4.3 初始化方法

好的初始化生产方法有利于提高元启发式算法搜索效率。在本小节中, 基于所给编码, 我们设计了初始解生产方法。初始化方式主要针对两种固定编码和第一种可变编码, 针对工艺路径选择矩阵 I , 设计了两种初始化方式, 即随机初始化和均匀初始化方式, 由于混合流水车间调度问题的调度, 最后一个阶段决定最终完工时间, 因此带工艺路径柔性的 HFSP 问题则可通过优化前几阶段的批次完工时间, 来缩短所有工

件加工批次的加工时长，采用均匀初始化的方式可以尽可能让具有柔性的几个阶段负荷更加均匀来达到缩短整体加工时长，采用随机初始化的方式可用于丰富种群多样性。

工艺路径选择矩阵的均匀初始化包含基于整批的均匀初始化方法和基于子批的均匀初始化方法，其中基于整批的均匀初始化是将各可选工艺路径均匀且随机的分配给各整批，子批的工艺路径与整批一致，而基于子批的均匀初始化则将各可选工艺路径均匀且随机的分配所有已分割的子批。工艺路径选择矩阵的随机初始化则也从整批和子批两个角度进行初始化，基于整批的随机初始化则对每个整批随机分配工艺路径，子批工艺路径与整批一致，而基于子批的随机初始化则对各子批随机分配工艺路径。

针对批次序列和第一种可变编码方式则采用文献[4]中所给的三种方式初始化方式进行初始化。

4.4 算法框架

算法采用 NSGA-II+变领域搜索的框架，与传统的变领域搜索不同的是，本文的变领域搜索是在种群每个个体的基础上进行搜索，此处我们将 NSGA-II 的进化过程作为主过程，变领域搜索过程作为子过程。主过程采用含第一种批次排序的编码方式进行进化操作，子过程则通过主过程个体解码方案转换为第二种批次排序的编码形式，子过程对主过程转换得到的第二种批次排序编码进行领域搜索，若领域搜索得到的解的质量优于主过程的解的质量，则将子过程的目标函数值作为主过程的目标函数值，即为当前个体的目标函数值，算法的大概框架如下图 5 所示。

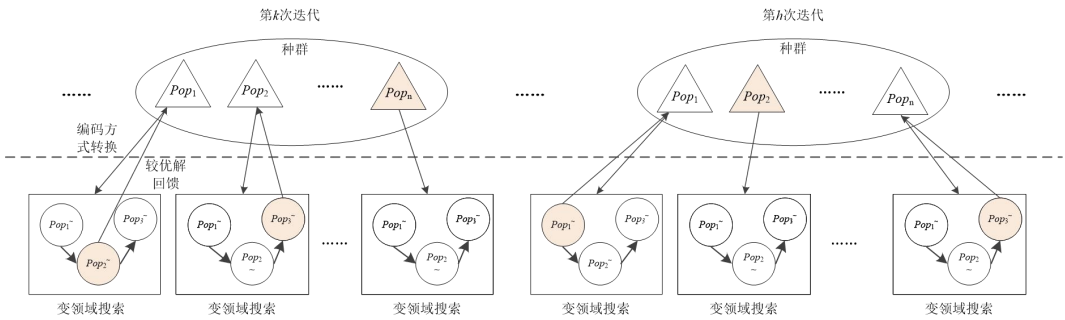


图 5

4.5 进化操作

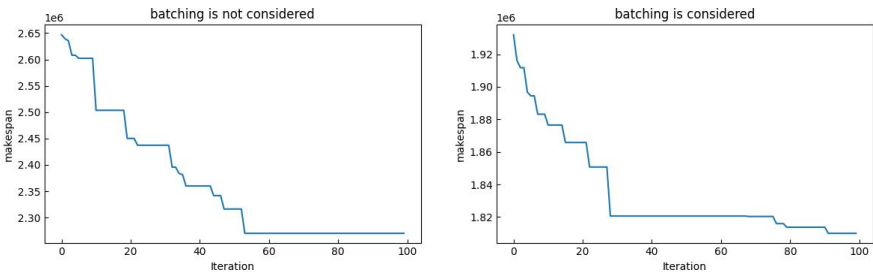
时间问题，略

5 排产结果

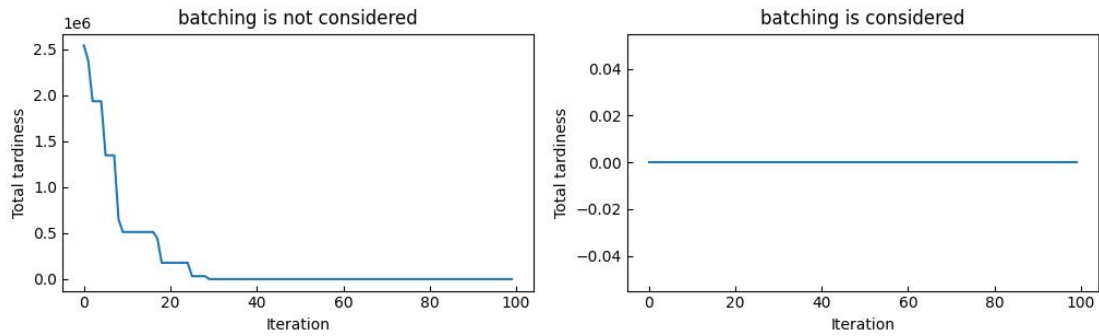
相关参数，设置最大分批数：10，最小分批规模：2000，分批方式：一致分批，迭代次数和规模：100*100。

输出表：result.xlsx 文件,中包含 scheduling_result 和 Batching_result 两张表，其中 scheduling_result 表达了排产结果，表头为['Machine','Order_Id'],Order_Id 可解释为：工件批的订单编号_批次索引_子批规模；Batching_result 表达了各工件的分批结果，表头为['Order_Id','batching'],batching 以列表的形式反应分批方案和索引。

最终实验结果迭代曲线：时间单位为（秒：s）



(1) 完工时间



(2) 总拖期

相关文献:

- [1] Zhang, M., Tan, Y. T., Zhu, J. H., Chen, Y. N., & Chen, Z. T. (2020). A competitive and cooperative Migrating Birds Optimization algorithm for vary-sized batch splitting scheduling problem of flexible Job-Shop with setup time. *Simulation Modelling Practice and Theory*, 100, 102065.
- [2] Zhang, B., Pan, Q. K., Meng, L. L., Zhang, X. L., Ren, Y. P., Li, J. Q., & Jiang, X. C. (2021). A collaborative variable neighborhood descent algorithm for the hybrid flowshop scheduling problem with consistent sublots. *Applied Soft Computing*, 106, 107305.
- [3] Yan, J., Liu, Z. F., Zhang, C. X., Zhang, T., Zhang, Y. Z., & Yang C. B. (2021). Research on flexible job shop scheduling under finite transportation conditions for digital twin workshop. *Robotics and Computer-Integrated Manufacturing*, 72, 102198.
- [4] Zhang, Biao, Pan, Quan-ke, Meng, Lei-lei, Lu, Chao, Mou, Jian-hui, Li, Jun-qing. An automatic multi-objective evolutionary algorithm for the hybrid flowshop scheduling problem with consistent sublots[J]. *KNOWLEDGE-BASED SYSTEMS*, 2022, 238.
- [5] Pan, Y. X., Gao, K. Z., Li, Z. W., & Wu, N. Q. (2022). Solving Biobjective Distributed Flow-Shop Scheduling Problems With Lot-Streaming Using an Improved Jaya Algorithm. *IEEE transactions on cybernetics*.
- [6] Daneshamooz, F., Fattahi, P., & Hosseini, S. M. H. (2022). Scheduling in a flexible job shop followed by some parallel assembly stations considering lot streaming. *Engineering Optimization*, 54(4), 614-633.