doi: 10.12194/j.ntu.20230710001
引文格式: 刘慧霞, 张铭心. 基于改进粒子群算法的柔性制造系统无死锁优化调度[J]. 南通大学学报(自然科学版), 2024, 23(1):38-48.

基于改进粒子群算法的柔性制造系统无死锁优化调度

刘慧霞, 张铭心
(南通大学 电气工程学院, 江苏 南通 226019)

摘要: 柔性制造系统的优化调度问题是一个复杂的组合优化和 NP-hard 问题。以赋时 Petri 网为模型、最小化最大完工时间为优化目标,利用改进粒子群算法对一类柔性制造类系统建立了一种新的无死锁优化调度方法。该方法首先采用 2 层编码方式对路径和工序进行编码,建立工序与粒子位置之间的一一映射关系;其次,基于实时在线的死锁避免策略对粒子进行死锁检测与修复,保证所搜索的粒子均能解码为无死锁的可行调度序列;然后,设计了 2 种改进策略:粒子工序定向调整策略和局部搜索策略,以提高算法的寻优效率和局部搜索能力,保证快速得到最优或次优的可行序列;最后,利用 2 个仿真实验验证所提算法的有效性。实验结果表明:与其他已有算法相比,改进粒子群算法在求解柔性制造系统无死锁优化调度问题上具有较好的寻优能力。

关键词: 柔性制造系统;死锁避免策略;粒子群算法;定向调整;局部搜索
中图分类号: TP301 **文献标志码:** A **文章编号:** 1673-2340(2024)01-0038-11

Deadlock-free scheduling based on improved particle swarm algorithm for flexible manufacturing systems

LIU Huixia, ZHANG Mingxin
(School of Electrical Engineering, Nantong University, Nantong 226019, China)

Abstract: The optimization scheduling problem of flexible manufacturing systems is a complex combinatorial optimization and NP-hard issue. Using timed Petri nets as the model and aiming to minimize the maximum completion time, a novel deadlock-free optimization scheduling method for a class of flexible manufacturing systems has been established through an improved particle swarm optimization algorithm. This method first adopts a two-layer coding strategy for paths and processes, establishing a one-to-one mapping relationship between processes and particle positions. Secondly, it employs a real-time online deadlock avoidance strategy to check and repair the feasibility of particles, ensuring that the searched particles can be decoded into a deadlock-free feasible scheduling sequence. Then, two improvement strategies are designed: a particle process directional adjustment strategy and a local search strategy, to enhance the algorithm′s optimization efficiency and local search capability, ensuring the rapid acquisition of optimal or sub-optimal feasible sequences. Finally, the effectiveness of the proposed algorithm is verified through two simulation experiments. Experimental results demonstrate that, compared to other existing algorithms, the improved particle swarm optimization algorithm exhibits superior optimization capability in solving the deadlock-free optimization scheduling problem of FMSs.

Key words: flexible manufacturing systems (FMSs); deadlock avoidance policy; particle swarm optimization; directional adjustment; local search

收稿日期: 2023-07-10 **接受日期:** 2023-07-28
基金项目: 山东省自然科学基金面上项目(ZR2018MF024);江苏省“双创博士”项目(JSSCBS20211103);南通市基础科学研究项目(JC2021203);烟台市科技创新发展计划(2022XDRH005)
第一作者简介: 刘慧霞(1979—), 女, 教授, 博士, 主要研究方向为制造系统死锁控制和优化调度。E-mail:liuhx@ntu.edu.cn

近年来，柔性制造系统(flexible manufacturing systems)在工业生产领域得到了广泛应用，柔性制造系统调度问题的研究也变得愈发重要[1-2]。调度问题指在当前系统的生产约束下，寻找最佳的加工顺序以实现各个工件在系统中的最佳加工效率。由于柔性制造系统具有资源共享和多路径选择的特点，在加工过程中容易出现死锁状态，即资源循环等待。这种状态会导致整体或部分生产停滞，给生产带来巨大损失。针对死锁问题，国内外的许多研究者已做了大量工作，并提出了多种死锁控制策略[3-6]。例如，文献[3]利用 Petri 网中的图形结构——资源变迁回路，对一类柔性制造系统设计了具有多项式时间复杂性的最优死锁避免策略。文献[5-6]则对具有不可靠资源的柔性制造系统，基于资源变迁回路设计了两种结构不同的鲁棒死锁控制器。

上述这些控制策略可以从逻辑上保证系统在整个生产过程中无死锁发生，但无法对系统生产性能进行直接优化。目前，针对柔性制造系统调度问题的无死锁优化方法趋向于将死锁避免策略与智能算法相结合[7-23]。例如，文献[7]在搜索过程中嵌入了死锁避免策略，基于可达图和最小加工时间矩阵，设计了一种无死锁的混合启发式搜索算法。该算法是基于状态空间搜索的，因此，存在空间爆炸问题。文献[8]提出一种混合粒子群算法，采用一种变邻域局部搜索方法在各种邻域空间中进行搜索，提升算法的局部搜索能力；但该算法复杂性较高、参数选择较为困难。文献[9]提出一种高容错性死锁控制器的任意分支定界算法，利用 2 种剪枝规则提高算法的搜索速度；但该算法在处理大规模问题时，计算时间成本相对较高。

本文采用赋时 Petri 网对一类柔性制造系统建模，利用文献[3]提出的死锁避免算法对所搜索调度序列进行可行性检测与修复，采用粒子群算法求解该系统调度问题。基于粒子群算法对调度序列进行搜索时，由于多个不同粒子可能同时共享同一变迁序列，破坏了种群多样性，影响了算法的寻优效率，故本文设计一种最小工件号最先开始调度的定向规则来限制粒子的搜索空间。另外，在寻优过程中，粒子群算法具有其性能受限于问题本身的复杂性和易于陷入局部最优的缺点，为克服该缺点本文又设计了一种分组优化的局部搜索策略，以提高搜索能力。最后通过 2 个仿真实验验证了所提改进策略的有效性。上述 2 个“设计”分别称为粒子工序定向调整策略和局部搜索策略。与已有参考文献[7-9,20]相比，本文的主要优点如下：1)设计了粒子工序定向调整策略缩小粒子的搜索空间；2)设计了一种局部搜索策略，加强算法的局部搜索能力，提高种群质量。

1 柔性制造系统的 Petri 网模型

1.1 Petri网的基本概念

赋时 Petri 网[8-9]是一个四元组 $N=(P,T,F,D)$，其中：$P=\{p_1,p_2,\cdots,p_n\}$为有限库所集合；$T=\{t_1,t_2,\cdots,t_m\}$为有限变迁集合；$F=(P\times T)\cup(T\times P)$为有向弧集，表示库所集到变迁集或者变迁集到库所集的流关系；$D=\{d_1,d_2,\cdots,d_n\}$为库所的时延集，其中 d_i 表示工件在库所 p_i 上的加工时间。Petri 网 N 的标识是一个映射 $M:P\to Z$，$\forall p\in P$，$M(p)$表示在 M 下 p 中的 token 个数，这里 $Z=\{0,1,2,\cdots\}$。M_0 是 N 的初始标识，即初始状态下库所 P 中 token 的分布情况，(N,M_0)称为标识赋时 Petri 网。$\forall x\in P\cup T$，$\cdot x=\{y\in P\cup T\mid(y,x)\in F\}$表示 x 的输入节点集，$x\cdot=\{y\in P\cup T\mid(x,y)\in F\}$表示 x 的输出节点集。

给定标识 M 和变迁 t，如果 $\forall p\in\cdot t$，$M(p)\geqslant 1$，则称 t 在 M 下是使能的。在 M 下引发使能变迁 t 使得系统状态从 M 转移到新的标识 M'，记作 $M[t>M'$，其中 $\forall p\in\cdot t\backslash t\cdot$，$M'(p)=M(p)-1$，$\forall p\in t\cdot\backslash\cdot t$，则 $M'(p)=M(p)+1$；否则 $M'(p)=M(p)$。对于任意一个变迁序列 $\alpha=t_0t_1t_2\cdots t_n$，如果 $M_i[t_i>M_{i+1}$ $(i=0,1,2,3,\cdots,n)$，则 α 为 M_0 下的可行序列，M_i 为 M_0 下的可达标识。用 $R(N,M_0)$表示 Petri 网从初始状态 M_0 出发得到的所有可达状态的集合。

1.2 Petri 网模型

对任一柔性制造系统，我们记第 i 类工件的加工过程为 $p_{is}t_{i1}p_{i1}t_{i2}p_{i2},\cdots,t_{ij}p_{ij}t_{i(j+1)}p_{if}$，其中 p_{is} 和 p_{if} 是闲置库所，表示第 i 类工件上传和卸载缓冲区；p_{ij} 是操作库所，表示第 i 类工件的第 j 个加工过程。对具有多类型待加工工件、每类工件有多个加工路径的柔性制造系统，其 Petri 网模型建立如下：

南通大学学报（自然科学版）

用 $P_{sf} = \{p_{is}, p_{if}, i = 1,2,\cdots,m\}$表示闲置库所集;用 $P = \{p_{ij}, i = 1,2,\cdots,m, j = 1,2,\cdots,n\}$表示操作库所集;用 $P_R = \{r_i, i = 1,2,\cdots,k\}$表示资源库所集;用 $T = \{t_{ij}, i = 1,2,\cdots,m, j = 1,2,\cdots,n+1\}$表示变迁集, 其中 t_{ij}表示第 i 类工件的第 j 个操作的开始和第 i 类工件的第 $j-1$ 个操作的结束。机器的请求和释放通过有向弧 F 表示,当 p_{ij}请求机器资源 r_m 时,从 r_m 引出一条弧线指向输入变迁 t_{ij};当释放机器资源 r_m时,从输出变迁 $t_{i(j+1)}$引出一条弧线指向 r_m。M_0表示系统的初始状态,p_{is} 的 token 数表示代加工的毛坯品个数,r_i 中的 token 数表示 r_i 的最大容量,其余库所的初始状态均为 0。上述 Petri 网模型可以表示为$(N, M_0) = (P_{sf} \cup P \cup P_R, T, F, D, M_0)$。给定一个标识 $M \in R(N, M_0)$,如果 $M(^{(p)}t) > 0$,则变迁 t 在 M 标识下是操作使能的;如果 $M(^{(r)}t) > 0$,则变迁 t 在 M 标识下是资源使能的。

考虑一个经典的柔性制造系统,该系统可以同时加工 3 种类型的工件 $J = \{J_1, J_2, J_3\}$,3 类待加工工件的个数分别为 8,12,8,其中第二类待加工工件有 2 条加工路径。该系统中含有个资源 $R = \{m_1, m_2, m_3, m_4, r_1, r_2, r_3\}$,每个资源的容量分别为 $C(m_1) = 2, C(m_2) = 2, C(m_3) = 2, C(m_4) = 2, C(r_1) = 1, C(r_2) = 2$ 和 $C(r_3) = 1$。每个库所对应的时延时间列在该库所旁边。该系统的 Petri 网模型如图 1 所示。

1.3 死锁描述

给定 Petri 网(N, M_0), $\forall M \in R(N, M_0)$和 $t \in T$,如果 $\forall M' \in (N, M_0)$,$M'[t >$ 都不成立,则称变迁 t在 M 下是死的,M 称为死锁标识;如果存在 M 下的可行变迁序列 α,$M[\alpha > M'$且 $M'[t >$ 都成立,则称变迁 t 在 M 下是活的。考虑图 2 所示的 Petri 网模型,该系统加工 2 类工件,每类待加工的工件数为 5,系统初始标识 $M_0 = 5p_{10} + 5p_{20} + 2r_1 + 2r_2 + 2r_3$。

这里只讨论死锁,故为使网图形简单,在该网中将 p_{is}与 p_{if}合并,使每个加工路径形成一个闭环。因为只考虑死锁问题,所以省略库所时延。

在 M_0下引发变迁序列 $t_{11}t_{21}t_{11}t_{21}t_{22}t_{22}$得到新标识 $M_1 = 3p_{10} + 3p_{20} + 2p_{11} + 2p_{22} + 2r_3$。在标识 M_1 下,资源 r_1 和 r_2 均为空,变迁 t_{12} 和 t_{23} 是操作使能但非资源使能的,从而 t_{12} 和 t_{23} 都是死的。如果当前死锁不及时处理,继续引发变迁序列 $t_{21}t_{21}$得到另一标识 $M_2 = 3p_{10} + 3p_{20} + 2p_{11} + 2p_{21} + 2p_{22}$,如图 2 所示。在 M_2下,系统所有变迁均是死的,导致系统生产全面停滞。

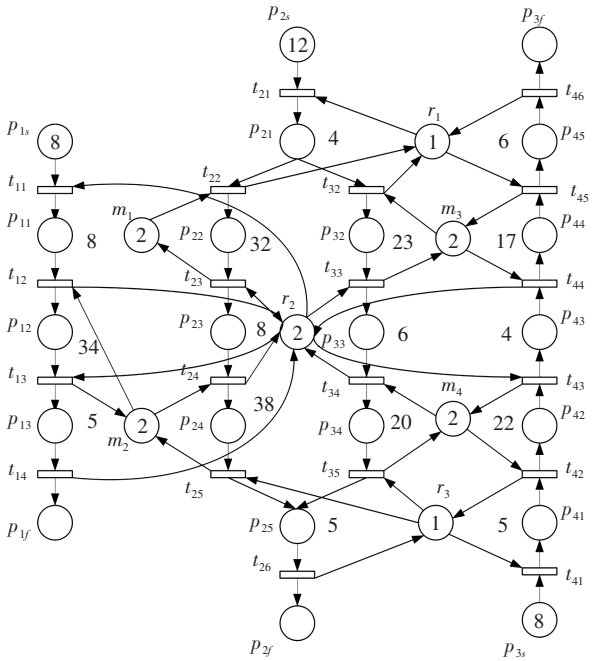

图 1 一个柔性制造系统的 Petri 网模型

Fig. 1 Petri net of a flexible manufacturing system

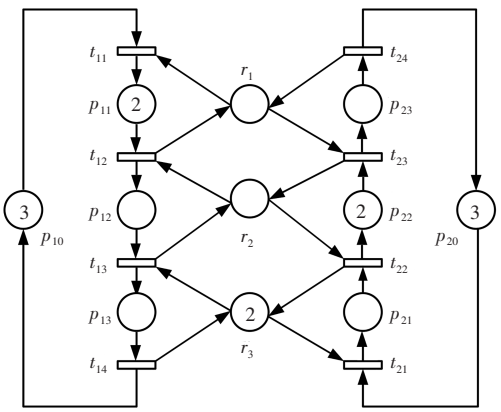

图 2 Petri 网的一个死锁标识

Fig. 2 A deadlock marking of Petri net model

2 基于改进粒子群求解制造系统无死锁调度问题

2.1 粒子群算法的基本概念

粒子群算法(particle swarm optimization)中多个

南通大学学报(自然科学版)

粒子构成一个种群[11-12]。假设问题的解是 D 维的，种群大小为 N，第 i 个粒子的位置和速度可以表示为 $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$，$V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$。粒子的位置和速度更新为

$$\begin{cases} v_{id}(t+1) = \omega(t) \times v_{id}(t) + c_1 \times r_1 \times \\ \quad (P_{i\text{best}}(t) - x_{id}(t)) + c_2 \times r_2 (G_{\text{best}}(t) - x_{id}(t)), \\ x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \end{cases} \tag{1}$$

其中：$i = 1, 2, \cdots, N$；$d = 1, 2, \cdots, D$；r_1 和 r_2 为 0 和 1 之间的随机数；ω 为惯性权重；c_1 和 c_2 为学习因子。

根据粒子自身的适应度值对惯性权重和学习因子进行自适应动态更新[13-16]，更新表示为

$$\omega(t) = \begin{cases} \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \dfrac{Fit_X - Fit_{\text{best}}}{Fit_{\text{avg}} - Fit_{\text{best}}}, Fit_X \leqslant Fit_{\text{avg}} \\ \omega_{\max}, Fit_X < Fit_{\text{avg}} \end{cases}, \tag{2}$$

$$c_1 = 2 + \frac{Fit_X - Fit_{\text{avg}}}{Fit_{\text{avg}} - Fit_{\text{best}}}, c_2 = 2 - \frac{Fit_X - Fit_{\text{avg}}}{Fit_{\text{avg}} - Fit_{\text{best}}}, \tag{3}$$

其中：$\omega_{\max}$ 和 $\omega_{\min}$ 分别为惯性权重的最大、最小值；Fit_X 为当前粒子适应度值；Fit_{best} 为 G_{best} 的适应度值；Fit_{avg} 为当前迭代中所有粒子的平均适应度值。

在每一次迭代的过程中，粒子通过跟踪个体极值 $P_{i\text{best}}$ 和全局极值 G_{best} 来更新自身的位置和速度。个体极值和全局极值的位置更新表示为

$$P_{i\text{best}}(t+1) = \begin{cases} X_i(t+1), Fit(X_i(t+1)) < Fit(P_{i\text{best}}(t)) \\ P_{i\text{best}}(t), \text{其他} \end{cases}, \tag{4}$$

$$G_{\text{best}} = \min\{P_{i\text{best}}\}, i = 1, 2, \cdots, N,$$

其中：$P_{i\text{best}}$ 表示粒子 i 所经历的最优位置；G_{best} 表示粒子在迭代过程中经历的最优位置。

2.2 编码与解码

由于编码反应了工件的工序、粒子的位置和速度，所以根据柔性制造系统和粒子群算法的特性，本文采用的编码方式如下：设工件个数为 K，工件集 $J = \{j_1, j_2, j_3, \cdots, j_K\}$，每个粒子表示为长度为 D 的三维列向量组，第一维是工序编码，第二维是位置向量 $\boldsymbol{X}$，第三维是速度向量 $\boldsymbol{V}$，粒子的长度 D 等于每个工件工序数之和。由于柔性制造系统具有柔性路径，粒子工序编码采用两层编码方式。第一层编码为工件的路径选择编码 $w = \{w_1, w_2, w_3, \cdots, w_k\}$，每个工件必须在指定的加工路径上进行加工；第二层编码为工件的工序编码，该编码由工件集 J 的工件编号组成，工件编号 $i(i \in K)$ 重复出现的次数表示工件 i 对应的第几道工序。通过采用两层编码，可以使调度问题更加简化、减少搜索空间，并且在 2 个层次上进行优化，可以有效提高算法的收敛速度。加工系统模型如图 3 所示，其粒子编码如表 1 所示。

表 1 粒子编码的向量表示

Tab. 1 Vector representation of particle encoding

路径	工序	位置	速度
1,2,1,1	2	19.87	−2.71
	2	−18.87	0.62
	4	−3.44	8.84
	3	14.42	9.90
	1	−1.01	8.40
	1	−14.02	−7.14
	2	6.69	9.98
	3	10.47	9.06
	4	0.80	−6.48
	1	11.37	3.73
	4	−3.57	−4.26
	3	17.93	−7.55

为了建立工序与粒子位置之间的映射关系，根据位置实数从小到大进行排序。随着粒子位置的更新，工序和速度的排序也会发生改变，即间接的产生一个新的离散工序解。排序后的新粒子和新粒子的加工顺序如表 2 所示。

2.3 粒子死锁检测与修复方法

根据 2.2 解码得到的变迁序列不一定是可行加工序列，需要对其进行死锁检测与修复。本文利用文献[3]提出的死锁避免算法，对每一个粒子的变迁序列进行检测与修复，保证变迁序列中的每个变迁都能依次引发。在对粒子修复的过程中，粒子的位置和速度保持不变。

例 1 如图 3 所示 Petri 网，系统初始标识 $M_0 = 2p_{1s} + 2p_{2s} + r_1 + r_2 + 2r_3 + 2r_4$。给定一个变迁序列

南通大学学报（自然科学版）

表 2 新粒子编码的向量表示

Tab. 2 Vector representation of new particle encoding

路径	工序	位置	速度	加工顺序
1,2,1,1	2	−18.87	0.62	t_{21}
	1	−14.02	−7.14	t_{11}
	4	−3.57	−4.26	t_{31}
	4	−3.44	8.84	t_{32}
	1	−1.01	8.40	t_{12}
	4	0.80	−6.48	t_{33}
	2	6.69	9.98	t_{22}
	3	10.47	9.06	t_{31}
	1	11.37	3.73	t_{13}
	3	14.42	9.90	t_{32}
	3	17.93	−7.55	t_{33}
	2	19.87	−2.71	t_{23}

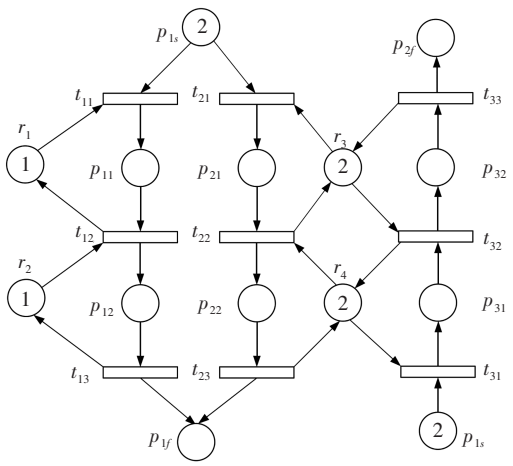

图 3 Petri 网系统模型

Fig. 3 Petri net model

表 3 粒子死锁检测算法

Tab. 3 Particle deadlock detection algorithm

算法 1 粒子死锁检测算法

输入:可行标识 M 和使能变迁 t

输出:$\chi(M,t)$ = 可行或不可行

1. 初始化变迁集 $T_1 = \varnothing, T_2 = \varnothing, T_3 = \varnothing$,

$\chi(M,t)$ = 可行;

2. 令 $r_1 = {}^{(r)}t$;$M[t > M_s$;

if $(M_s(r_1) > 0)$ {

output $\chi(M,t)$;

exit;

} else {

$T_1 = \{t \in T | R({}^{(p)}t) = r_1, M_s({}^{(p)}t) > 0\}$;

$R_1 = \{r_1\}$;

}end if−else

3. while$(T_1 \backslash T_2 = \varnothing)$, do {

chose $t \in T_1 \backslash T_2$, let $r = {}^{(r)}t$;

if $(M_s(r) > 0)$ {

output $\chi(M,t)$;

exit;

} else {

$T_3 := \{t \in T | R({}^{(p)}t) = r, M_s({}^{(p)}t) > 0\}$;

$T_1 := T_1 \cup T_3$;

$T_2 := T_2 \cup \{t\}$;

} end if−else

} end while

4. $\chi(M,t)$ = 不可行;

$\alpha = t_{21}t_{31}t_{21}t_{31}t_{32}t_{22}t_{33}t_{23}t_{32}t_{22}t_{33}t_{23}$,在 M_0 下引发 $t_{21}t_{31}t_{21}$ 得到新标识 $M_1 = 2p_{21} + p_{31} + p_{2s} + r_1 + r_2 + r_4$。在 M_1 下,变迁 t_{31} 可以申请资源 r_4,故 t_{31} 是使能变。但引发 t_{31} 使得资源 r_3 和 r_4 均被占满,这样系统生产就出现了资源循环等待现象,即死锁的发生。从而 t_{31} 不能引发,变迁序列 α 是不可行的。借助文献[3],本文给出如表 3 所示的算法对上述死锁现象进行检测。

我们对上述变迁序列 $\alpha = t_{21}t_{31}t_{21}t_{31}t_{32}t_{22}t_{33}t_{23}t_{32}t_{22}t_{33}t_{23}$ 进行如下修复:利用算法 1,$\chi(M_1, t_{31})$ 是不可行的,即 t_{31} 在 M_1 下是不能引发的,则在后续变迁序列 $t_{32}t_{22}t_{33}t_{23}t_{32}t_{22}t_{33}t_{23}$ 中,找到 M_1 下可行变迁 t_{22} 移至变迁 t_{31} 之前,即原变迁序列 α 变为新的变迁序列 $t_{21}t_{31}t_{21}t_{22}t_{31}t_{32}t_{33}t_{23}t_{32}t_{22}t_{33}t_{23}$;然后在 M_1 下引发 t_{22},得到新标识 $M_2 = p_{21} + p_{22} + p_{2s} + p_{31} + r_3 + r_1 + r_2$,利用算法 1,得到 $\chi(M_2, t_{31})$ 是不可行的,即 t_{31} 在 M_2 下也是不能引发的,则继续执行上述步骤,在后续变迁序列中找到可行变迁 t_{32} 移至变迁 t_{31} 之前;依次进行下去,最终得到一个可行的无死锁变迁序列 $\alpha = t_{21}t_{31}t_{21}t_{22}t_{32}t_{31}t_{33}t_{23}t_{32}t_{22}t_{33}t_{23}$。

南通大学学报(自然科学版)

2.4 适应度值计算

本文以粒子的最大完工时间作为适应度函数。给定一个可行变迁序列 $\alpha = t_1 t_2 t_3 \cdots t_k$，每个资源中至少含有一个加工单元，即工件可以选择加工单元进行加工。粒子 P 的适应度函数可以表示为 $Fit(P) = S(t_k)$，其中 t_k 表示最后一道加工工序。

约束条件 1 如下：

$$\begin{cases} E(t_i) = \min(T(^{(r)}t_i)) + D(t_i), i = 1,2,\cdots,k-1 \\ T(^{(r)}t_i) = E(t_i), i = 1,2,\cdots,k-1 \\ S(t_i) = E(t_{i-1}), i = 1,2,\cdots,k \end{cases}, \tag{5}$$

式中：t_{i-1} 表示变迁 t_i 的前一道工序，t_{i+1} 表示变迁 t_i 的下一道工序；$S(t_i)$ 表示变迁 t_i 的开始时间；$E(t_i)$ 表示变迁 t_i 的结束时间；$D(t_i)$ 表示工件通过变迁 t_i 在库所 p_i 上的加工时间；$T(^{(r)}t_i)$ 表示变迁 t_i 申请资源的加工时间。

约束条件 2 如下：若变迁 t_i 申请的资源中含多个加工单元，则选择加工时间最短的加工单元；若变迁 t_i 申请资源的加工时间大于 t_{i-1} 的结束时间，则更新变迁 t_i 的开始时间、t_{i-1} 的结束时间和变迁 t_{i-1} 申请资源的加工时间。时间更新为

$$\begin{cases} S(t_i) = \mathrm{Max}(\min(T(^{(r)}t_i)), E(t_{i-1})) \\ T(^{(r)}(t_{i-1})) = E(t_{i-1}) = S(t_i) \end{cases}。 \tag{6}$$

2.5 定向调整策略

在使用粒子群算法对柔性制造系统调度时，多个不同粒子可能同时共享同一变迁序列，且粒子在寻优过程中可能会追随多个不同的优秀粒子，会大大影响粒子群算法的寻优效率。比如在同一类型下存在 2 组加工路径相同的工件 1 和 2、工件 3 和 4，交换每组工件中的工序顺序不会影响解码后的变迁序列。出现以上这种情况，工件 1 和 2、工件 3 和 4 为可交换工件组。

种群中存在可交换组会使得种群多样性遭到破坏，导致算法搜索效率变差。为解决该问题，本文设计一种最小工件号最先开始调度的规则进行定向调整。通过规定在可交换工件组中，工件加工的先后顺序必须按照工件号的大小进行调整，即工件号小的工件总是在工件号大的工件前进入系统加工。调整方法为：比较可交换工件组中各工件工序段中的最小位置值，依次将位置值小的工件的工序段与可交换工件组中工件号小的工件的工序段进行交换。通过这种方式，可以限制粒子的搜索空间，保持种群多样性，防止算法陷入局部最优解，提高寻优效率。

2.6 局部搜索策略

粒子群算法虽然收敛速度快，但易于陷入局部最优；又因为柔性制造系统的结构复杂，因此有必要加强算法的局部搜索能力，更好地寻找最优解。

由于种群中粒子优劣程度不同，本文将种群以平均适应度值为界限划分成精锐组和普通组，采用不同的更新方法对粒子进行优化。对于适应度较优的粒子，通过与种群中的随机粒子进行领域搜索，以扩大最优解的范围；对于适应度较差的粒子，通过追随当前全局最优粒子和个体最优粒子来平衡种群质量。对于粒子优化，本文采用随机变异方法对路径层进行更新，采用优先级运算交叉（precedence operation crossover，POX）方法[17-19]对工序层进行更新。

1）针对路径层，结合图 4 的步骤如下：

步骤 1 选择粒子 i 的路径层编码[1,1,2,2,1,3,2]；

步骤 2 随机产生一个变异点 $C = 3$ 和变异长度 $D = 2$，得到变异段 $l_{CD} = [2,2]$；

步骤 3 将变异段上的路径编号在其路径集中随机选择，得到新的路径层编码[1,1,1,3,1,3,2]。

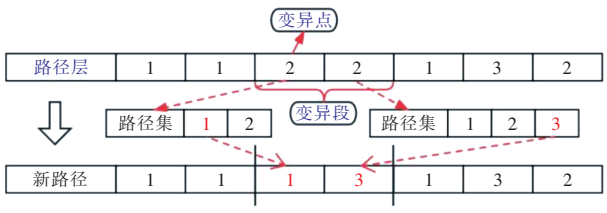

图 4 路径层的变异方法

Fig. 4 Mutation method of path layer

2）针对工序层，结合图 5 的步骤如下：

步骤 1 将粒子 i 的工序层作为父代 P_1，将随机粒子或 G_{best} 或 P_{ibest} 的工序层作为父代 P_2；

步骤 2 将所有工件随机分成 2 个互补集合

南通大学学报（自然科学版）

G_1 和 G_2,集合大小为工件总数量的1/3 至 1/2;

步骤 3　将 P_1 包含在 G_1 的工序编号复制到 C_1 且位置对应在 P_1 中的位置,将 P_2 包含在 G_2 的工序编号依次复制到 C_1 中的空缺位置;

步骤 4　将 P_2 包含在 G_2 的工序编号复制到 C_2 且位置对应在 P_2 中的位置,将 P_1 包含在 G_1 的工序编号依次复制到 C_2 中的空缺位置。

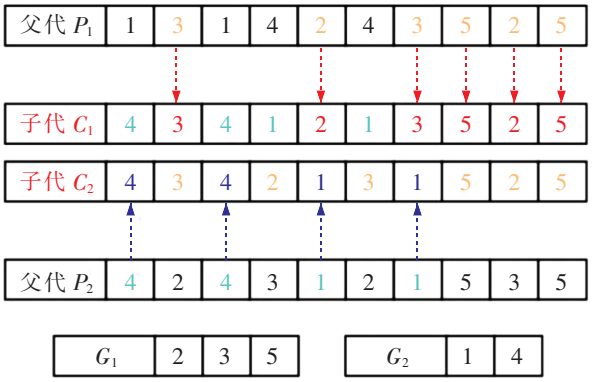

图 5　工序层的交叉方法

Fig. 5　Intersection method of process layer

3)为了扩大精锐组粒子的寻优范围,避免算法陷入局部最优,在 POX 交叉的基础上进行如下步骤:

步骤 1　将粒子 i 与随机粒子进行 POX 交叉,生成子代 C_1 和 C_2,保留适应度值优的子代并复制到 C_3;

步骤 2　将 C_3 倒序得到 C_4,比较 C_3 和 C_4 的适应度值,将适应度值最优的粒子作为优化后的新粒子;

步骤 3　更新粒子的 $P_{i\text{best}}$ 和 G_{best}。

4)为了提高普通组粒子的种群质量,使得粒子更快接近最优解,在 POX 交叉方法的基础上进行如下步骤:

步骤 1　将粒子 i 与 G_{best} 进行 POX 交叉,保留适应度值优的子代并复制到 C_3;将 $P_{i\text{best}}$ 与 G_{best} 进行POX 交叉,保留适应度值优的子代并复制到 C_4;

步骤 2　比较 C_3 和 C_4 的适应度值,将适应度值最优的粒子作为优化后的新粒子;

步骤 3　更新粒子的 $P_{i\text{best}}$ 和 G_{best}。

2.7　改进粒子群算法的求解流程

根据上述策略,对柔性制造系统调度问题的求解流程图如图 6 所示。

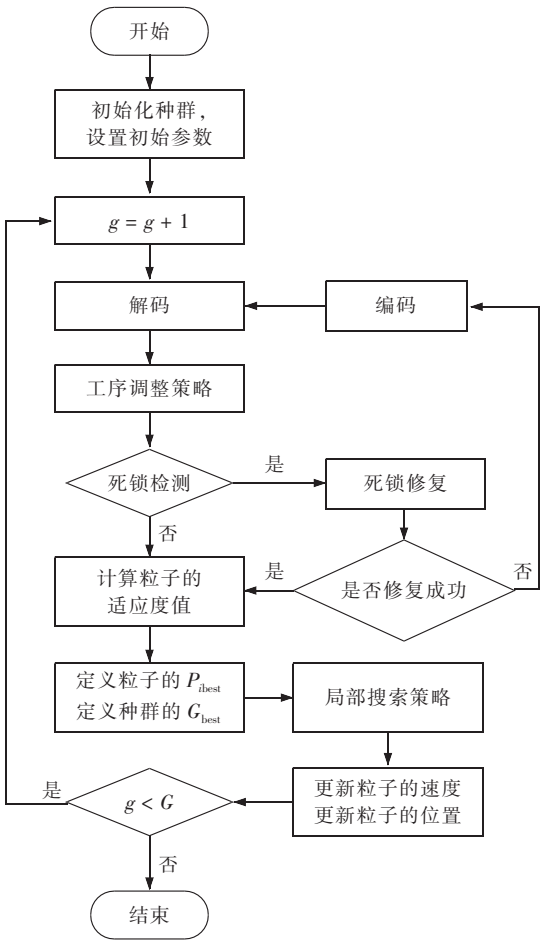

图 6　改进粒子群算法流程图

Fig. 6　Improved particle swarm optimization algorithm flow chart

3　实验结果与分析

为了验证本文所提算法在求解无死锁优化调度时的有效性,以图 1 和图 7 所示的柔性制造系统为例进行仿真测试,具体加工时间数据如表 4 和表 7 所示。算法采用 java 语言编程,基于主频 3.3 GHz CPU、内存 16 G 的 PC 机上运行。每个算法在各算例中独立运行 20 次。

3.1　实验 1

针对图 7 模型,考虑含不同批量 $\varphi(J)$ 和不同资源容量 $C(R)$ 的 12 个算例 In01 至 In12,如表 5 所示。将本文提出的两个改进策略与基本粒子群算法(PSO)进行比较,验证定向调整策略和局部搜索策略的有效性。含定向调整策略的粒子群算法,记作 DA-PSO;含局部搜索策略的粒子群算法,记作 LA-

南通大学学报(自然科学版)

PSO。加粗数据为同一算例下的最优结果，不同算法在不同算例下的运行结果如表 6 所示。可以看出，DA-PSO 和 LA-PSO 在所有算例中的性能均优于 PSO。其中 DA-PSO 与 PSO 相比，性能上均有一定的提升，这说明了定向调整策略是有效的；而在大批量的 In03、In06、In09、In12 算例中，LA-PSO 的结果远优于 PSO，这也说明了本文提出的局部搜索策略具有较好的寻优能力，能避免过早陷入局部最优。

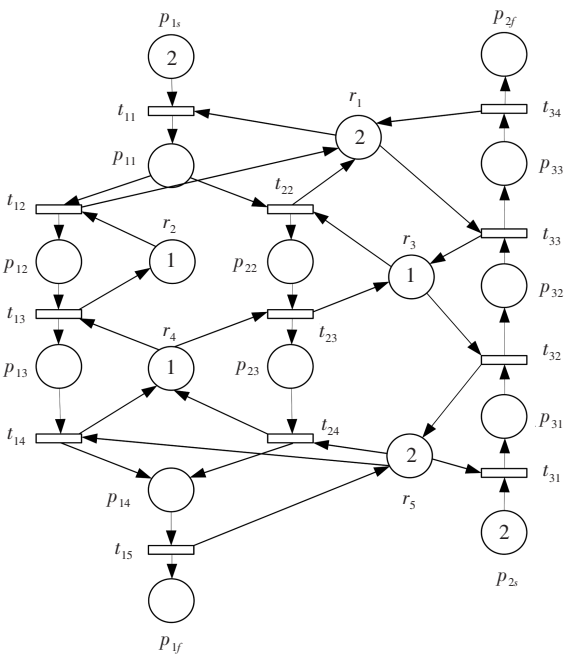

图 7　算例模型

Fig. 7　An example

表 4　图 7 的加工时间

Tab. 4　Processing time of figure 7

J_1		J_2
w_1	w_2	w_3
O_{11}:4	O_{21}:4	O_{31}:5
O_{12}:21	O_{22}:12	O_{32}:23
O_{13}:7	O_{23}:4	O_{33}:6
O_{14}:5	O_{24}:5	

3.2　实验 2

针对图 1 模型，考虑含不同批量和不同资源容量的 10 个算例 Case01 至 Case10，如表 8 所示。将改进的粒子群算法（改进-PSO）与文献[20]中提出的定向启发式算法（DHS）、文献[7]中提出的无死锁动态窗口搜索（D2WS）算法和 PSO 进行对比，加粗数据为同一算例下的最优结果，如表 9 所示。可以看出，改进-PSO 除了在 Case10 中的效果稍差于 DHS

表 5　基于图 7 的 12 个算例

Tab. 5　12 examples based on figure 7

算例	$\varphi(J_1)$	$\varphi(J_2)$	$C(r_1)$	$C(r_2)$	$C(r_3)$	$C(r_4)$	$C(r_5)$
In01	10	10	1	1	1	1	1
In02	20	20	1	1	1	1	1
In03	30	30	1	1	1	1	1
In04	10	10	2	2	2	2	2
In05	20	20	2	2	2	2	2
In06	30	30	2	2	2	2	2
In07	10	10	3	3	3	3	3
In08	20	20	3	3	3	3	3
In09	30	30	3	3	3	3	3
In10	10	10	4	4	4	4	4
In11	20	20	4	4	4	4	4
In12	30	30	4	4	4	4	4

表 6　实验 1 结果

Tab. 6　Results of experiment 1

算例	DA-PSO		LA-PSO		PSO	
	最优值	平均值	最优值	平均值	最优值	平均值
In01	**241**	241.5	**241**	241.0	**241**	243.3
In02	493	510.9	**481**	501.2	508	516.6
In03	791	806.3	**727**	746.3	798	813.0
In04	137	140.4	**126**	126.7	140	141.3
In05	284	292.5	**247**	255.6	288	292.7
In06	456	478.3	**427**	443.5	468	480.1
In07	**103**	103.2	**103**	103.0	**103**	103.3
In08	210	217.5	**176**	181.3	216	218.3
In09	318	327.6	**271**	282.5	325	330.3
In10	**80**	80.0	**80**	80.0	**80**	80.3
In11	159	163.6	**137**	139.3	161	164.1
In12	252	262.1	**207**	217.7	257	263.3

表 7　图 1 的加工时间

Tab. 7　Processing time of figure 1

J_1	J_2		J_3
w_1	w_2	w_3	w_4
O_{11}:8	O_{21}:4	O_{31}:4	O_{41}:5
O_{12}:34	O_{22}:32	O_{32}:23	O_{42}:22
O_{13}:5	O_{23}:8	O_{33}:6	O_{43}:4
	O_{24}:38	O_{34}:20	O_{44}:17
	O_{25}:5	O_{35}:5	O_{45}:6

南通大学学报（自然科学版）

外,在其余算例中的最优值和平均值均优于其他算法,因此改进-PSO 具有较好的寻优能力。图 8 为基本 PSO 与改进 PSO 在 10 个算例中的标准差对比结果。从图 8 可以看出,改进-PSO 的标准差普遍较小,因此,求得的最优解更集中,算法性能更稳定。

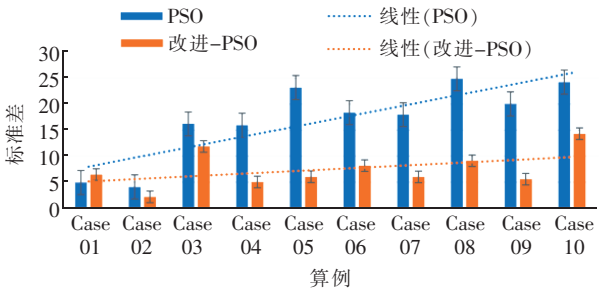

图 8　标准差

Fig. 8　Standard deviation

表 8　基于图 1 的 10 个算例

Tab. 8　10 examples based on figure 1

算例	$\varphi(J_1)$	$\varphi(J_2)$	$\varphi(J_3)$	$C(m_i)$	$C(r_1)$	$C(r_2)$	$C(r_3)$
Case01	5	5	5	2	1	1	1
Case02	5	5	5	2	1	2	1
Case03	10	10	10	2	1	1	1
Case04	10	10	10	2	1	2	1
Case05	8	12	8	2	1	1	1
Case06	8	12	8	2	1	2	1
Case07	8	12	8	2	2	2	2
Case08	10	20	10	2	1	1	1
Case09	10	20	10	2	1	2	1
Case10	10	20	10	2	2	2	2

表 9　实验 2 结果

Tab. 9　Results of experiment 2

算例	DHS	D2WS		PSO			改进-PSO		
	最优值	最优值	平均值	最优值	平均值	标准差	最优值	平均值	标准差
Case01	—	—	—	190	203.3	4.92	**159**	168.5	6.46
Case02	—	—	—	159	165.9	4.09	**133**	136.9	2.17
Case03	—	—	—	416	444.6	16.21	**320**	337.2	11.87
Case04	—	—	—	313	352.0	15.95	**254**	263.9	5.03
Case05	312	303	334.2	323	358.6	23.21	**297**	309.1	6.04
Case06	—	255	280.5	292	349.1	18.38	**237**	257.9	8.18
Case07	273	252	272.6	288	315.9	18.00	**237**	248.8	6.01
Case08	506	426	454.9	513	552.2	24.89	**420**	435.2	9.13
Case09	—	—	—	456	492.7	20.06	**369**	376.7	5.58
Case10	**316**	360	388.6	399	445.2	24.25	333	355.8	14.32

4　结论

本文为一类柔性制造系统设计了一种基于改进粒子群算法的无死锁调度策略。其中,对粒子进行编码并建立一种合适的映射关系,利用死锁避免策略保证调度序列的可行性, 同时设计定向调整策略和局部搜索策略以加强算法的搜索效率和搜索能力。最后,通过 2 个仿真实验对比验证了 2 种改进策略的有效性,且算法在性能上有明显提升。

基于本文的设计思想,下一步的研究方向是为含不可靠资源的柔性制造系统建立性能优良的重调度优化策略。

参考文献:

[1] 公艳庆. 柔性制造系统在工程机械产品制造中的应用[J]. 中国设备工程, 2022(18):87-89.
GONG Y Q. Application of flexible manufacturing system in the manufacture of construction machinery products[J]. China Plant Engineering, 2022(18):87-89. (in Chinese)

[2] 洪良, 王艺翔, 南恺恺, 等. 柔性制造系统时延 Petri 网模型排产优化研究[J]. 机械设计与制造, 2022(4):262-265.
HONG L, WANG Y X, NAN K K, et al. Scheduling optimization study of timed Petri net modelsfor flexible manufacturing systems[J]. Machinery Design & Manufacture, 2022(4):262-265. (in Chinese)

[3] XING K Y, ZHOU M C, LIU H X, et al. Optimal petri-

南通大学学报（自然科学版）

net－based polynomial－complexity deadlock－avoidance policies for automated manufacturing systems[J]. IEEE Transactions on Systems, Man, and Cybernetics－Part A: Systems and Humans, 2009, 39(1):188–199.

[4] LUO J C, LIU Z Q, ZHOU M C. A Petri net based deadlock avoidance policy for flexible manufac turing systems with assembly operations and multiple resource acquisition [J]. IEEE Transactions on Industrial Informatics, 2019, 15(6):3379–3387.

[5] 刘慧霞，李俊红，王红梅，等. 具有不可靠资源柔性制造系统的鲁棒控制器设计[J]. 控制与决策，2022，37(8):2040–2048.
LIU H X, LI J H, WANG H M, et al. Design of robust controllers for flexible manufacturing systems with multiple unreliable resources[J]. Control and Decision, 2022, 37(8):2040–2048. (in Chinese)

[6] LIU H X, FENG Y X, LI J H, et al. Robust Petri net controllers for flexible manufacturing systems with multitype and multiunit unreliable resources[J]. IEEE Transactions on Systems, Man, and Cybernetics:Systems, 2023, 53(3):1431–1444.

[7] LUO J C, XING K Y, ZHOU M C, et al. Deadlock–free scheduling of automated manufacturing systems using Petri nets and hybrid heuristic search[J]. IEEE Transactions on Systems, Man, and Cybernetics:Systems, 2015, 45(3):530–541.

[8] LI X L, XING K Y, LU Q C. Hybrid particle swarm optimization algorithm for scheduling flexible assembly systems with blocking and deadlock constraints[J]. Engineering Applications of Artificial Intelligence, 2021, 105:104411.

[9] LUO J C, ZHOU M C, WANG J Q. AB&B:an anytime branch and bound algorithm for scheduling of deadlock－prone flexible manufacturing systems[J]. IEEE Transactions on Automation Science and Engineering, 2021, 18(4):2011–2021.

[10] 李大成，罗继亮，孙莎莎，等. 基于平行 Petri 网的制造系统调度与控制一体化方法[J]. 自动化学报，2023，49(4):845–856.
LI D C, LUO J L, SUN S S, et al. The integrated method of scheduling and control for manufacturing systems based on parallel Petri nets[J]. Acta Automatica Sinica, 2023, 49(4):845–856. (in Chinese)

[11] 杨恒. 基于改进粒子群算法的作业车间调度优化[J]. 机械设计与制造工程，2019，48(2):73–76.
YANG H. An improved partical swarm optimization algorithm for solving job shop scheduling[J]. Machine Design and Manufacturing Engineering, 2019, 48(2):73–76. (in Chinese)

[12] 刘洪铭，曾鸿雁，周伟，等. 基于改进粒子群算法作业车间调度问题的优化[J]. 山东大学学报(工学版)，2019，49(1):75–82.
LIU H M, ZENG H Y, ZHOU W, et al. Optimization of job shop scheduling based on improved particle swarm optimization algorithm[J]. Journal of Shandong University (Engineering Science), 2019, 49(1):75–82. (in Chinese)

[13] 张渊博，邹德旋，张春韵，等. 自适应惯性权重的粒子群优化算法[J]. 计算机仿真，2023，40(4):350–357.
ZHANG Y B, ZOU D X, ZHANG C Y, et al. Particle swarm optimization algorithm with adaptive inertial weight [J]. Computer Simulation, 2023, 40(4):350–357. (in Chinese)

[14] 王翠，姜学军. 基于动态变化自适应惯性权重混沌粒子群算法[J]. 沈阳理工大学学报，2022，41(6):13–18.
WANG C, JIANG X J. Chaotic particle swarm optimization based on dynamic change adaptive inertial weight[J]. Journal of Shenyang Ligong University, 2022, 41(6):13–18. (in Chinese)

[15] 仝秋娟，赵岂，李萌. 基于自适应动态改变的粒子群优化算法[J]. 微电子学与计算机，2019，36(2):6–10.
TONG Q J, ZHAO Q, LI M. Particle swarm optimization algorithm based on adaptive dynamic change[J]. Microelectronics & Computer, 2019, 36(2):6–10. (in Chinese)

[16] 喻明让，陈云，张志刚. 离散粒子群优化算法求解多目标柔性作业车间调度问题[J]. 制造技术与机床，2019(1):159–165.
YU M R, CHEN Y, ZHANG Z G. Adiscrete version of particle swarm optimization for multi－objective flexible job–shop scheduling problems[J]. Manufacturing Technology & Machine Tool, 2019(1):159–165. (in Chinese)

[17] 吴晓雯，郑巧仙. 基于改进粒子群的柔性作业车间调度问题优化研究[J]. 湖北大学学报(自然科学版)，2022，44(5):501–507.
WU X W, ZHENG Q X. Optimization of flexible job shop scheduling problem based on improved particle swarm[J]. Journal of Hubei University (Natural Science), 2022, 44(5):501–507. (in Chinese)

[18] 李稚，周双牛. 面向绿色智能制造的高维多目标动态作业车间调度优化[J]. 运筹与管理，2023，32(1):47–53.

LI Z, ZHOU S N. High dimensional multi-objective dynamic job shop scheduling optimization for green intelligent manufacturing[J]. Operations Research and Management Science, 2023, 32(1):47-53. (in Chinese)

[19] 张桐瑞, 吴定会. 混合竞争群优化算法求解柔性车间调度问题[J]. 控制工程, 2021, 28(9):1820-1828.
ZHANG T R, WU D H. Hybrid competitive group optimizer algorithm for flexible job-shop scheduling problem[J]. Control Engineering of China, 2021, 28(9): 1820-1828. (in Chinese)

[20] LEI H, XING K Y, HAN L B, et al. Hybrid heuristic search approach for deadlock-free scheduling of flexible manufacturing systems using Petri nets[J]. Applied Soft Computing, 2017, 55:413-423.

[21] 周鹏鹏, 翟志波, 戴玉森. 基于改进遗传算法的柔性作业车间调度问题研究[J]. 组合机床与自动化加工技术, 2023(3):183-186.
ZHOU P P, ZHAI Z B, DAI Y S. Research on flexible job shop scheduling problem based on improved genetic algorithm[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2023(3):183-186. (in Chinese)

[22] 管赛, 熊禾根. 混合禁忌搜索的车间调度遗传算法研究[J]. 智能计算机与应用, 2023, 13(5):171-174.
GUAN S, XIONG H G. A hybrid taboo search genetic algorithm for shop floor scheduling[J]. Intelligent Computer and Applications, 2023, 13(5):171-174. (in Chinese)

[23] 晏晓凡. 改进灰狼优化算法求解机器人作业车间调度问题[J]. 计算技术与自动化, 2023, 42(2):158-163.
YAN X F. An improved grey wolf optimization algorithm for robotic job shop scheduling problem[J]. Computing Technology and Automation, 2023, 42(2):158-163. (in Chinese)

(责任编辑:仇慧)

(上接第 37 页)

[24] FU J, LIU J, TIAN H J, et al. Dual attention network for scene segmentation[C]//Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York:IEEE Xplore, 2019:3141-3149.

[25] MEI H, ZHANG H, JIANG Z. Self-attention fusion module for single remote sensing image super-resolution[C]// Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, July 11-16, Brussels, Belgium. New York:IEEE Xplore, 2021:2883-2886.

[26] STAAL J, ABRÀMOFF M D, NIEMEIJER M, et al. Ridge-based vessel segmentation in color images of the retina[J]. IEEE Transactions on Medical Imaging, 2004, 23(4):501-509.

[27] ALOM M Z, HASAN M, YAKOPCIC C, et al. Recurrent residual convolutional neural network based on U-Net (R2U-Net) for medical image segmentation[EB/OL]. (2018-05-29)[2023-02-21]. https://arxiv.org/abs/1802.06955.

[28] AZAD R, ASADI-AGHBOLAGHI M, FATHY M, et al. Bi-directional ConvLSTM U-net with densley connected convolutions[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), October 27-28, 2019, Seoul, Korea (South). New York: IEEE Xplore, 2019:406-415.

[29] 孙颖, 丁卫平, 黄嘉爽, 等. RCAR-UNet:基于粗糙通道注意力机制的视网膜血管分割网络[J]. 计算机研究与发展, 2023, 60(4):947-961.
SUN Y, DING W P, HUANG J S, et al. RCAR-U Net: retinal vessels segmentation network based on rough channel attention mechanism[J]. Journal of Computer Research and Development, 2023, 64(4):947-961. (in Chinese)

(责任编辑:仇慧)

南通大学学报(自然科学版)