

基于 EDA 的绿色零等待作业车间调度问题求解

姚友杰^{1,2}, 钱 斌^{1,2}, 董钰明¹, 胡 蓉^{1,2}, 王 凌³

(1. 昆明理工大学信息工程与自动化学院, 云南昆明 650500; 2. 昆明理工大学云南省人工智能重点实验室, 云南昆明 650500;
3. 清华大学自动化系, 北京 100084)

摘 要: 针对绿色零等待作业车间调度问题, 本文提出一种基于分布估计算法 (Estimation of Distribution Algorithm, EDA) 的多目标优化算法, 用于最小化最大完工时间和总能量消耗. 采用双种群增加种群的多样性, 并加入启发式操作 NEH (Nawaz-Enscore-Ham, NEH) 来提高初始解的质量. 设计自适应学习速率来控制优良解信息在整个算法搜索过程中的引导作用. 结合零等待作业车间调度问题的结构性质与解码方式, 提出基于前端省略的快速评价方法, 同时利用 Insert 操作构建两种局部搜索策略, 进而加强算法的搜索效率和深度. 通过仿真实验和算法对比验证所提算法的有效性.

关键词: 零等待作业车间调度问题; 分布估计算法; 能量消耗; 绿色调度

中图分类号: TP302 **文献标识码:** A **文章编号:** 0372-2112 (2021)02-0225-08

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.12263/DZXB.20200091

EDA-Based for the Green No-wait Job Shop Scheduling Problem

YAO You-jie^{1,2}, QIAN Bin^{1,2}, DONG Yu-ming¹, HU Rong^{1,2}, WANG Ling³

(1. School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan 650500, China;
2. Yunnan Key Laboratory of Artificial Intelligence, Kunming University of Science and Technology, Kunming, Yunnan 650500, China;
3. Department of Automation, Tsinghua University, Beijing 100084, China)

Abstract: A multi-objective optimization algorithm based on estimation of distribution algorithm, namely EDA (Estimation of Distribution Algorithm) is presented to minimize the maximum completion time and the total energy consumption for the green no-wait job shop scheduling problem (GNWJSSP). Double populations are used to increase the diversity of the population, and a heuristic operation named NEH (Nawaz-Enscore-Ham) is added to improve the quality of the initial solution. An adaptive learning rate is designed to control the guidance of the information of excellent solutions during the entire algorithm search process. Combined with the structural properties and decoding method of the no-wait job shop scheduling problem, a fast evaluation method based on the front omission strategy is proposed, and two local search strategies are constructed by utilizing the Insert operation, thereby enhancing the search efficiency and depth of the algorithm. The effectiveness of the proposed algorithm for solving GNWJSSP is verified by simulation experiments and algorithm comparisons.

Key words: no-wait job shop scheduling problem; estimation of distribution algorithm; energy consumption; green scheduling

1 引言

作业车间调度问题 (Job Shop Scheduling Problem, JSSP) 作为一类重要的调度模型而被广泛研究. 但对各加工阶段操作连续性有特定要求的生产过程, 如冶金、炼钢、制药和化工企业等生产过程^[1,2], 需在 JSSP 中加入零等待约束形成零等待作业车间调度问题 (The No-

wait Job Shop Scheduling Problem, NWJSSP) 才符合生产实际. 同时, 随着绿色制造、节能减排的深入推进, 企业需要在生产过程中协同考虑经济指标 (时间成本) 和绿色指标 (能量消耗). 因此, 开展绿色零等待作业车间调度问题 (Green NWJSSP, GNWJSSP) 的研究具有重要的工程意义.

近年来, NWJSSP 的研究虽得到了一定的关注和发

展,但由于问题的强 NP-hard 属性^[3],相关文献还相对较少.部分研究者设计传统运筹学求解算法求解.譬如,Mascis 等^[4]提出一种 NWJSSP 的析取图模型,并设计分支定界算法进行求解. Pei 等^[5]提出列生成算法求解两台机器的柔性 NWJSSP.也有部分研究者将 NWJSSP 分为时间表子问题和排序子问题,并设计智能算法求解.譬如,Zhu 等^[6]设计移动时间表方法求解时间表子问题,进而提出带记忆机制的局部搜索算法进行求解,并在后续工作中对时间表子问题求解方法做出改进^[7,8].Sundar 等^[9]在评价解时采用工件直接插入时间表空隙的方法,可明显降低评价时间,进而提出带局部搜索的混合蜂群算法进行求解. Deng 等^[10]提出混合离散群搜索算法进行求解,之后又将时间表子问题求解方法^[6,8]与迭代贪心算法结合,可实现更高效的求解^[11].

绿色车间调度已开始受到重视.譬如,针对以最小化总能耗和总延误时间为优化目标的单机调度问题, Mouzon 等^[12]提出多目标自适应元启发式算法进行求解.针对以最小化生产效率和电力成本为优化目标的混合流水车间调度问题, Luo 等^[13]提出新型多目标蚁群优化算法进行求解.针对以最小化总能耗和总工期为优化目标的双机调度问题, Mansouri 等^[14]建立混合整数多目标优化模型,提出构造性启发式算法进行求解,并将该算法与商业求解器 Cplex 进行比较.针对以最小化碳排放和总加权延迟时间为优化目标的 JSSP, Zhang 等^[15]提出改进的多目标遗传算法进行求解.上述研究均采用调节机器转速的节能策略.然而,对于生产中广泛存在的 GNWJSSP,尚无相关研究,故亟需对其开展研究.

本文针对最小化最大完工时间和总能量消耗量的 GNWJSSP,提出一种基于分布式估计算法(Estimation of Distribution Algorithm, EDA)的多目标优化算法.该算法设计自适应学习速率来控制优良解信息在整个算法搜索过程中的引导作用,并结合 NWJSSP 的结构性质与解码方式提出快速评价方法,同时利用 Insert 操作构建两种局部搜索策略,从而可提升算法的性能.

2 问题描述

NWJSSP 可以描述为有 n 个工件需要在 m 台机器上加工,工件的加工路线和加工时间已经给定,且每个工件的各操作间存在零等待约束.此外, NWJSSP 也同样具有一般 JSSP 的普遍约束:任一工件的任一操作不允许被抢占加工,任一工件同一时刻只能进行一个操作,每个工件必须按照给定的加工顺序进行. NWJSSP 的零等待约束可以描述为:

$$S_{i,j+1} = S_{i,j} + T_{i,j} \quad (1)$$

式中 $S_{i,j}$ 代表第 i 个工件第 j 个操作的开始加工时间, $T_{i,j}$ 代表第 i 个工件第 j 个操作的加工时间.

本文所提 GNWJSSP,考虑在 NWJSSP 模型中增加机器的档位选择,通过机器加速改变各工件的实际操作时间,进而缩短企业生产的最大完工时间(C_{\max}).但机器转速的提升必定会影响机器的实际运行功率,从而改变整个生产过程的总能量消耗 E_{TEC} (the Total Energy Consumption, TEC).因此,本文考虑的两个优化的目标是最小化 C_{\max} 和 E_{TEC} ,使得企业在生产过程中兼顾经济效益和保护环境.两个优化目标分别可以描述为:

$$f_1 = \min C_{\max} \quad (2)$$

$$f_2 = \min E_{\text{TEC}} \quad (3)$$

通常将机器的状态分为加工状态 (processing) 和准备状态 (standby),处于加工状态的机器工作档位不同时,其单位时间内消耗的能量不同.假定 $V_{i,j}$ 代表机器 j 在加工工件 i 时机器的档位,且每台机器有 s 种不同的档位,即 $V_{i,j} \in S_v, S_v = \{V_1, V_2, \dots, V_s\}$.同时,档位的不同对当前工件的操作时间也有影响,由原来的 T 变为 T/V ,即机器档位的高低与工件的实际加工时间成反比,档位越高加工时间越短.

此外,机器档位的改变不仅改变工件的实际加工时间,对对应操作的机器能耗也会产生巨大影响.由于机器单位时间内的能耗与档位的高低并非成线性关系,所以于任意的机器档位 u, v 且 $u > v$,同一操作处于高档位 u 加工所造成的能耗必大于低档位 v 的能耗.这一性质已在文献[16]中被证明,其具体描述为式(4)~式(6).

$$\forall u > v \quad u, v \in S_v \quad (4)$$

$$T_{i,j}/V_u < T_{i,j}/V_v \quad (5)$$

$$P_{j,u} \cdot T_{i,j}/V_u > P_{j,v} \cdot T_{i,j}/V_v \quad (6)$$

假定 $x_{j,v}(t)$ 表示机器 j 在时刻 t 为加工状态且档位为 v , $y_j(t)$ 表示机器 j 在时刻 t 为准备状态. $x_{j,v}(t)$ 和 $y_j(t)$ 可以描述为:

$$x_{j,v}(t) = \begin{cases} 1, & \text{机器 } j \text{ 在 } t \text{ 时刻工作在 } v \text{ 档位} \\ 0, & \text{否则} \end{cases} \quad (7)$$

$$y_j(t) = \begin{cases} 1, & \text{机器 } j \text{ 在 } t \text{ 时刻为准备状态} \\ 0, & \text{否则} \end{cases} \quad (8)$$

综上所述,企业生产过程的 E_{TEC} 可以通过计算每一时刻下各个机器所消耗的能量进行求和(加工状态和准备状态),其具体计算公式如下:

$$E_{\text{TEC}} = \int_0^{C_{\max}} \left(\sum_{v=1}^s \sum_{j=1}^m P_{j,v}^p \cdot x_{j,v}(t) + \sum_{k=1}^m P_j^s \cdot y_j(t) \right) dt \quad (9)$$

其中, $P_{j,v}^p$ 表示机器 j 在档位 v 下进行加工时单位时间内消耗的能量值, P_j^s 表示机器 j 处于空闲状态时单位时间内消耗的能量值.为简化问题,假设各机器在同档位下具有相同的加工功率和空闲功率,借鉴文献[17]中的经验公式,设定 $P_{j,v}^p = 4 \cdot v^2$ 和 $P_j^s = 1$,本文中的机器速度档位 $S_v = \{1, 1.5, 2\}$.

GNWJSSP 同样具有零等待特性,其约束由式(1)变为:

$$S_{i,j+1} = S_{i,j} + T_{i,j}/V_{i,j} \quad (10)$$

由于零等待约束的存在,所以工件某个操作加速会改变其后续所有操作的开始加工时间,进而会改变所有工件在时刻表上的位置.因此,机器档位的加入使该问题的求解变得更加复杂.

从图 1 和图 2 的对比可以发现,加入机器档位后 GNWJSSP 的部分操作实际加工时间缩短,使整个过程的 C_{\max} 变小,但同时也造成某段时间内的能量消耗剧增,如 $t_0 \sim t_1$ 时间段(机器 1 和机器 2 均为加速状态),从而导致整个过程的 E_{TEC} 变大.由此看出,本文的两个优化目标最小化 C_{\max} (经济指标)和 E_{TEC} (绿色指标)之间存在着一定的冲突和制约.

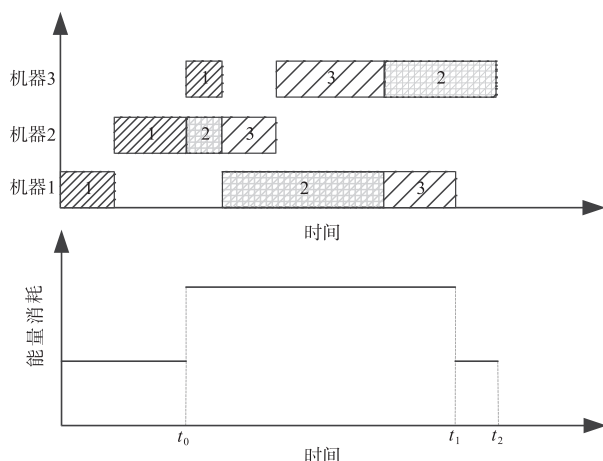


图1 一般的NWJSSP实时能量消耗图

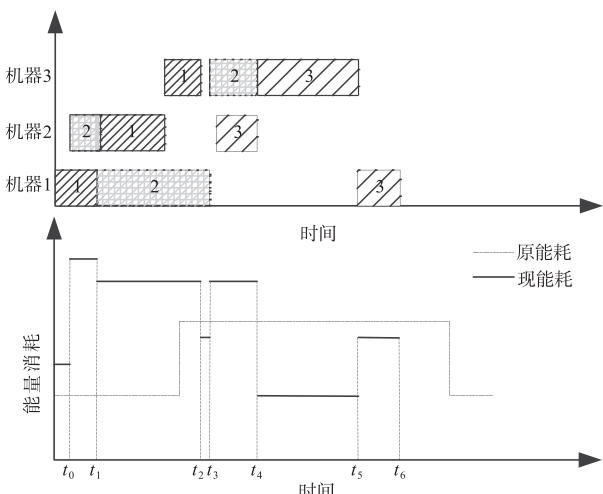


图2 带机器加速的NWJSSP实时能量消耗图

3 基于 EDA 的算法描述

3.1 编码与解码

本文采用整数编码的形式进行优化,每种排列 π

都代表问题的一个解.对于有 n 个工件的 GNWJSSP, $\pi = \{\pi_1, \pi_2, \dots, \pi_n\}$,该问题的解空间大小为 $n!$.

由于 GNWJSSP 的零等待约束,所以每个工件的所有操作可以视为一个整体.本文采用由 Brizuela^[18] 提出的直接插入时间表空隙的解码方式,该方式与其他较复杂的解码策略具有相当的性能^[9],但可明显缩短评价问题解的时间,其具体步骤如下:

步骤 1 将排列 π 的第一个工件直接放入时刻表;

步骤 2 计算每台机器上的空闲段的大小,最后一个空闲段的大小视为无穷大;

步骤 3 在不违反 GNWJSSP 任一约束的情况下,依次将排列 π 中的工件尽可能的往前放,若前面无空闲段放置该工件,则将工件往后移动,直到能够放置为止;

步骤 4 若排列 π 中的工件均放置完则停止该过程,否则跳转到步骤 2.

3.2 种群机制及初始化

本文采用启发式操作 NEH 构建初始种群,首先以每个工件总加工时间降序排列的结果作为 NEH 的初始排列,然后分别以最小化 C_{\max} 、最小化 E_{TEC} 为目标求得最优排列,进而提升初始种群的质量.此外,采用双种群的方式避免算法搜索后期新种群的同质化问题,即第一种群由 EDA 采样生成,另一种群均随机产生.

3.3 概率矩阵初始化和改进的更新机制

基于统计学原理的 EDA 是通过概率模型生成新个体,是算法进化寻优的基础.本文采用二维矩阵 P 来构建概率模型,如式(11)所示,矩阵中的数 $P_{ij}(g)$ 表示工件 j 出现在位置 i 的概率, g 表示进化代数.由于初始状态下工件出现在任一位置的概率是相同的,所以概率矩阵 P_{ij} 的初始值为 $1/n$.

$$P_{ij}(g) = \begin{pmatrix} P_{11} & \dots & P_{1n} \\ \vdots & \ddots & \vdots \\ P_{n1} & \dots & P_{nn} \end{pmatrix}_{n \times n} \quad (11)$$

对于 GNWJSSP 的求解,如何将非劣解中优良信息进行提取与保留是 EDA 的核心所在,本文对非劣解排列的采样包括工件的位置信息和块结构信息.如式(12)(13)所示:

$$\varphi_{i,j} = \begin{cases} 1, & \text{如果工件 } j \text{ 出现在位置 } i \text{ 及之前} \\ 0, & \text{其他} \end{cases} \quad (12)$$

$$\eta_{j,k} = \begin{cases} 1, & \text{如果工件 } j \text{ 紧邻工件 } k \text{ 出现} \\ 0, & \text{其他} \end{cases} \quad j \neq k \quad (13)$$

φ_{ij} 统计排列中工件位置信息, η_{jk} 统计排列中两个相邻工件的块结构信息,均为 $n \times n$ 的矩阵.同时,本文采用平均加权的形式的对工件的位置信息和块结构信息进行求和,以便生成当代的优良信息分布矩阵 E_{ij} 如式(14)所示.

$$E_{ij} = \frac{1}{2} \sum_{a=1}^b \left(\frac{\varphi_{ij}^a}{i \cdot n} + \eta_{jk}^a \right) \quad (14)$$

其中 b 表示当代采样优良解的个数, $a = 1, 2, \dots, b$; $i, j, k = 1, 2, \dots, n$. 举例: 假定采样的非劣解集中有 4 个非劣解, 分别是 $\pi_1 = [4, 1, 2, 3]$ 、 $\pi_2 = [2, 1, 4, 3]$ 、 $\pi_3 = [1, 2, 3, 4]$ 、 $\pi_4 = [4, 2, 3, 1]$, 则采样结果如式(15)~式(16)所示. 由于任何排列中的工件能且仅能出现一次, 因此 η_{jk} 中其对角线的值不存在.

$$\sum_{a=1}^4 \varphi_{ij}^a = \begin{pmatrix} 1 & 1 & 0 & 2 \\ 3 & 3 & 0 & 2 \\ 3 & 4 & 2 & 3 \\ 4 & 4 & 4 & 4 \end{pmatrix} \quad (15)$$

$$\sum_{a=1}^4 \eta_{jk}^a = \begin{pmatrix} \backslash & 2 & 0 & 1 \\ 1 & \backslash & 3 & 0 \\ 1 & 0 & \backslash & 1 \\ 1 & 1 & 1 & \backslash \end{pmatrix} \quad (16)$$

同时, 在采样得到当代的优良信息分布矩阵 E_{ij} 后, 需要对算法整体的概率矩阵 $P_{ij}(g)$ 进行更新. 为合理控制优良解信息在整个算法搜索过程中的引导作用, 设计一种自适应学习速率, 如式(18), 概率矩阵的学习速率 α 与算法的进化代数有关, 其取值范围为 $[\alpha_{\min}, \alpha_{\max}]$. 自适应学习速率可以强化优良解信息在算法运行初期的引导作用, 从而驱动算法全局搜索尽快到达解空间中的优质解区域; 同时保证在算法搜索的中后期, 适当弱化优良解信息的影响以避免过早收敛, 确保算法可在优质解区域内进行范围较广的搜索, 从而增加更优解发现的机率.

$$P_{ij}(g+1) = (1-\alpha)P_{ij}(g) + \alpha E_{ij} \quad (17)$$

$$\alpha = \alpha_{\max} - \frac{g}{g_{\max}}(\alpha_{\max} - \alpha_{\min}) \quad (18)$$

更新后的概率矩阵需要根据式(19)进行归一化处理. 此外, 算法通过概率模型生成新排列时, 利用轮盘赌形式选中工件 j 后要把对应列清零, 并对概率矩阵进行归一化再依次选择下一工件.

$$P_{ij} = \frac{P_{ij}}{\sum_{j=1}^n P_{ij}} \quad (19)$$

3.4 前端省略的快速评价方法

由于 NWJSSP 调度模型的复杂性, 算法的大部分时间都用于对解进行评价. 譬如, 对于 10×5 规模的 NWJSSP, 文献[9]所提混合蜂群算法(运行 25 代)的求解时间为 0.78s; 对于 30×10 规模的 NWJSSP, 所提混合蜂群算法(运行 500 代)的求解时间为 1797s(约 30 分钟). 由上可知, NWJSSP 的求解会随着规模的增加呈现非线性快速增长的趋势. 因此, 若能缩短解的评价时间, 算法可在相同时间内搜索解空间中更多的区域, 有利

于提高算法的搜索性能.

本文结合 NWJSSP 的问题特性及解码方式提出了一种基于前端省略的快速评价方法. 快速评价方法是指一个解的部分序列在没有被改变的情况下, 不需要重新计算. 如图 3 所示, 对解 $\pi = [5, 4, 2, 1, 3, 6]$ 局部搜索操作后, 解的部分排列 $[5, 4, 2]$ 却未发生改变, 并且该部分排列位于解的前端. 而本文所采用的解码方向是自前向后的, 因此对于新序列的评价只需重新计算 $[6, 1, 3]$ 即可完成对新解的快速评价. 解的前端中未发生改变序列的长度越长, 该方法节省的计算时间越多.

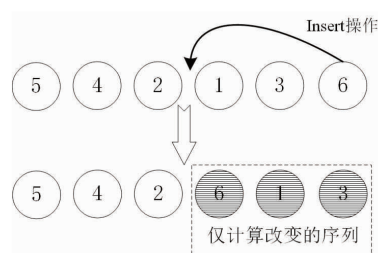


图3 前端省略评价方法示意图

3.5 基于快速评价的局部搜索

由于 GNWJSSP 不同于单目标问题的求解, 因此局部搜索策略的设计对算法性能有着至关重要的影响. 为了充分利用前端省略快速评价方法节约评价时间, 本文设计一种基于快速评价和 Insert 邻域操作的局部搜索策略 1; 同时, 也利用 Insert 和 Swap 邻域操作协同的方式设计局部搜索策略 2. 其中, Insert 操作是一种高效的局部搜索操作, 且适合应用前端省略的快速评价方法.

局部搜索策略 1 的具体步骤如算法 1.

算法 1 基于快速评价的局部搜索

- 1: 选中非劣解集中的解 π ;
- 2: 对解 π 进行评价, 并存储相应的时刻表;
- 3: $k \leftarrow n \div 2$ (n 为工件数);
- 4: For $i = k$ to n
- 5: For $j = 1$ to n
- 6: if $j \neq i$, 则 $\pi' = \text{Insert}(\pi, i, j)$, 并快速评价 π' ;
- 7: if π' 与 π 互不支配, 则将 π' 加入到非劣解集中;
- 8: End j
- 9: End i
- 10: 更新非劣解集, 剔除其中的受支配解.

局部搜索策略 2 分为扰动环节和探索环节, 搜索过程中的 p 和 q 均为随机生成, 具体步骤如算法 2.

算法 2 变邻域局部搜索

- 1: 选中非劣解集中的解 π .
- 2: 扰动环节:

```

3: 利用 Insert 对  $\pi$  进行扰动,得到  $\pi' = \text{Insert}(\pi, p, q)$ .
4: 探索环节:
5: loop ← 1
6: Repeat
7:    $c \leftarrow 0, c_{\max} \leftarrow 2$ ;
8:   Repeat
9:     if  $c = 0, \pi'' = \text{Swap}(\pi', p, q)$ ;
10:    if  $c = 1, \pi'' = \text{Insert}(\pi', p, q)$ ;
11:    if  $\pi'' > \pi'$  ( $\pi''$  支配  $\pi'$ ), then  $c \leftarrow 0, \pi' \leftarrow \pi''$ 
12:    else  $c \leftarrow c + 1$ ;
13:   Until  $c = c_{\max}$ 
14:   loop ← loop + 1;
15: Until loop =  $n \cdot (n - 1)$ 
16: if  $\pi' > \pi$ , then  $\pi \leftarrow \pi'$ 

```

局部搜索策略 1 能在较短时间内评价新解或邻域,可将节约的计算时间用于对邻域进行深入细致地搜索,从而有助于发现 Insert 邻域结构中更多的非劣解.局部搜索策略 2 能驱动算法跳出局部最优,可拓宽算法的搜索广度,避免非劣解的集中化;同时采用 Insert 和 Swap 两种邻域操作协同搜索也可一定程度上增加搜索的广度和深度,进而提高非劣解的质量.两种策略综合考虑了局部搜索的效率,以及局部搜索的深度与广度,故能有效地提高算法的整体性能.

3.6 EDA 求解 GNWJSSP

综上所述,基于 EDA 求解 GNWJSSP 的整体流程可以归纳为算法 3.

算法 3 EDA 求解 GNWJSSP:

```

步骤 1: 根据种群机制及初始化生成初始种群;
步骤 2: 选取初始种群的非劣解集;
步骤 3: 概率矩阵初始化;
步骤 4: 统计非劣解中的优良信息;
步骤 5: 根据公式对概率矩阵进行更新;
步骤 6: 通过概率矩阵生成第一个种群,另一种群均随机产生;
步骤 7: 选取新种群的非劣解并更新非劣解集;
步骤 8: 对非劣解集中的个体进行局部搜索,若随机数  $\leq P_{ls}$ , 执行局部搜索策略 1, 否则执行局部搜索策略 2;
步骤 9: 更新非劣解集;
步骤 10: 判断算法是否达到终止条件,若满足跳转至步骤 11, 否则转至步骤 4;
步骤 11: 输出非劣解集.

```

4 算法测试结果及分析

4.1 实验设置

本文的仿真实验采用的是国际标准测试数据 Benchmark, 共计 35 组数据进行仿真测试. 由于目前尚无其他直接求解 GNWJSSP 问题的算法, 故我们选用已有文献中求解生产调度问题的有效算法 IMMOGLS^[19]、

SGA^[20] 和 SFLA^[17] 进行对比. 其中, IMMOGLS 是一种利用随机权重确定搜索方向的有效多目标遗传算法, 其性能在求解生产调度问题上优于著名的多目标算法 SPEA 和 NSGAII; SGA 为一种多目标混合遗传算法, 采用模拟退火算法的逃逸机制以避免搜索陷入局部极小, 其性能优于一些当前较为有效的多目标优化算法; SFLA 采用随机分组的方式对蛙群进行分组优化, 并对每组蛙群进行多邻域的深度探索, 是求解绿色柔性作业车间调度问题的有效算法. 这些算法生成新个体和新邻域的核心操作均可作用于任何形式的排列或解, 同时也采用通用的非支配排序、非劣解集设置与更新等多目标处理技术, 故其搜索性能与具体问题的关联性较小. 因此, 将本文所提算法与这些算法在相同运行时间下进行对比, 可较客观反映本文算法的性能.

所有算法和测试均由 Delphi 2010 编程实现, 并在配置为 Win10、2.2GHz、8GB 的电脑上进行实验. 每种算法在求解同一测试问题时均在相同的时间下独立运行 20 次. 由于 GNWJSSP 的问题特性, 随着问题规模的增大, 算法评价解的时间也会相应延长, 但其关系呈非线性, 因此仿真实验结合文献[9]中实验数据对不同规模下算法的运行时间进行了调整, 时间设置为 La01 ~ La05 为 t_b , La06 ~ La10 为 $6t_b$, La11 ~ La15 为 $10t_b$, La16 ~ La20 为 $2t_b$, La21 ~ La25 为 $8t_b$, La26 ~ La30 为 $12t_b$, La31 ~ La35 为 $20t_b$, $t_b = n \times 100\text{ms}$.

4.2 性能评价指标

本文采用两个性能指标^[21]来评价算法的优劣性. 其中一个指标评价各算法所得到的非劣解集中非劣解的质量, 另一指标用来评价各算法所得到的非劣解集中非劣解的数量. 假定 S_{EDA} 、 S_{IMMOGLS} 、 S_{SGA} 、 S_{SFLA} 分别代表本文所提算法、IMMOGLS、SGA、SFLA 运行后得到的非劣解集. S_{sum} 代表所有算法求解 GNWJSSP 得到的非劣解集, 即 $S_{\text{sum}} = S_{\text{EDA}} + S_{\text{IMMOGLS}} + S_{\text{SGA}} + S_{\text{SFLA}}$. 由于各非劣解集之间存在互相支配的解, 所以计算两个性能指标时需要将被其他非劣解支配的解剔除.

$$R(S_j) = |S_j - \{\mathbf{x} \in S_j \mid \exists \mathbf{y} \in S_{\text{sum}}: \mathbf{y} < \mathbf{x}\}| / |S_j| \quad (20)$$

式中 S_j 表示某算法运行后的非劣解集, $\mathbf{y} < \mathbf{x}$ 表示解 \mathbf{x} 被解 \mathbf{y} 支配, 该部分解需要剔除. $R(S_j)$ 的值越大代表该算法所得非劣解集的质量越高.

$$N(S_j) = |S_j - \{\mathbf{x} \in S_j \mid \exists \mathbf{y} \in S_{\text{sum}}: \mathbf{y} < \mathbf{x}\}| \quad (21)$$

除了非劣解的质量是重要指标外, 非劣解集中解的数量也是一个很重要的指标, 该指标是指将算法所得非劣解集中被支配的解剔除后的剩余值. $N(S_j)$ 的值越大表示非劣解的数量越多, 则可以为提供的解决方案越多.

4.3 参数整定

本文所提算法中有三个关键参数对算法的性能有

影响,分别是种群的规模 P_{size} 、学习速率的范围 $[\alpha_{min}, \alpha_{max}]$ 、局部搜索策略的选择概率 P_{LS} 。针对这三个参数,采用实验设计方法(Design Of Experiment, DOE)进行实验分析,建立规模为 $L16(4^3)$ 的正交实验表,进而研究参数对算法性能的影响并最终确定算法的参数设置^[22]。经过实验分析^[23],最佳参数组合为 $P_{size} = 50$ 、 $[\alpha_{min}, \alpha_{max}] = [0.2, 0.5]$ 、 $P_{LS} = 0.6$ 。智能优化算法的优化效果与其参数选择密切相关,考虑到各对比算法被提出时求解的问题模型不同,其相应的算法最优参数

组合也会有所不同,为了显示三种算法对比的公平性,本文也对对比算法采用相同的实验方法进行参数整定。实验结果表明^[23],在求解 GNWJSSP 时,IMMOGLS 的最佳参数组合为种群的规模 $P_{Ca} = 60$ 、交叉概率 $P_c = 0.9$ 、变异概率 $P_m = 0.2$ 以及局部搜索概率 $P_{LS} = 0.1$,SAGA 的最佳参数组合为种群的规模 $P_N = 60$ 、交叉概率 $P_c = 0.85$ 、变异概率 $P_m = 0.05$ 以及模拟退火局部搜索迭代次数 $T_{LS} = 20$,SFLA 的最佳参数组合为种群的规模 $N = 60$ 、分组数 $s = 10$ 以及局部搜索次数 $\mu = 120$ 。

表 1 EDA 与 IMMOGLS、SGA、SFLA 的对比数据

问题	n_m	IMMOGLS		SGA		SFLA		EDA	
		R	N	R	N	R	N	R	N
La01	10_5	0.150	0.150	0.154	0.300	0.158	0.350	0.900	2.350
La02	10_5	0.200	0.300	0.125	0.200	0.317	0.500	0.950	2.650
La03	10_5	0.179	0.350	0.225	0.400	0.358	0.550	0.778	2.050
La04	10_5	0.217	0.350	0.342	0.550	0.379	0.750	0.798	2.250
La05	10_5	0.275	0.300	0.425	0.450	0.650	0.800	0.858	2.300
La06	15_5	0.275	0.350	0.075	0.100	0.233	0.450	0.700	0.750
La07	15_5	0.200	0.300	0.213	0.450	0.325	0.500	0.700	0.700
La08	15_5	0.150	0.350	0.279	0.550	0.325	0.500	0.800	0.900
La09	15_5	0.267	0.450	0.133	0.200	0.317	0.400	0.650	0.700
La10	15_5	0.325	0.650	0.230	0.450	0.299	0.650	0.700	0.750
La11	20_5	0.367	0.800	0.058	0.150	0.442	0.800	0.725	0.850
La12	20_5	0.288	0.550	0.130	0.300	0.350	0.700	0.850	0.900
La13	20_5	0.358	0.600	0.117	0.150	0.413	0.850	0.575	0.700
La14	20_5	0.235	0.550	0.237	0.350	0.317	0.550	0.850	0.900
La15	20_5	0.188	0.400	0.017	0.050	0.450	0.650	0.750	0.750
La16	10_10	0.000	0.000	0.000	0.000	0.050	0.050	0.950	1.100
La17	10_10	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.350
La18	10_10	0.050	0.100	0.025	0.050	0.050	0.050	0.950	1.200
La19	10_10	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.600
La20	10_10	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.550
La21	15_10	0.000	0.000	0.050	0.100	0.000	0.000	1.000	1.750
La22	15_10	0.000	0.000	0.050	0.050	0.150	0.250	0.932	2.100
La23	15_10	0.050	0.050	0.050	0.050	0.017	0.050	0.950	1.250
La24	15_10	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.800
La25	15_10	0.000	0.000	0.000	0.000	0.000	0.000	1.000	1.350
La26	20_10	0.050	0.050	0.000	0.000	0.150	0.200	0.850	1.100
La27	20_10	0.050	0.050	0.050	0.050	0.350	0.350	0.900	1.200
La28	20_10	0.150	0.200	0.050	0.050	0.325	0.450	0.828	1.850
La29	20_10	0.200	0.250	0.050	0.050	0.100	0.100	0.927	2.450
La30	20_10	0.200	0.200	0.075	0.100	0.125	0.150	0.800	1.250
La31	30_10	0.175	0.200	0.000	0.000	0.550	0.700	0.850	0.950
La32	30_10	0.125	0.200	0.050	0.050	0.450	0.600	0.600	0.650
La33	30_10	0.100	0.150	0.000	0.000	0.500	0.550	0.800	0.850
La34	30_10	0.100	0.150	0.000	0.000	0.300	0.450	0.900	1.150
La35	30_10	0.150	0.150	0.000	0.000	0.525	0.750	0.700	0.750

4.4 测试结果与对比分析

表 1 为所有算法仿真实验的运行结果, R 和 N 分别代表对应算法的 $R(S_j)$ 指标和 $N(S_j)$ 指标. 由表中数据可以看出, 在绝大多数情况下, 本文所提 EDA 算法的两个评价指标均优于对比算法 IMMOGLS、SGA 和 SFLA. 在部分问题类型的运行结果中, EDA 的 R 指标为 1 且其他算法 R 指标为 0, 此时代表 EDA 所得到的非劣解集完全支配其他算法的非劣解集; 当各算法 R 指标属于 $(0, 1)$ 之间时, 表示各算法之间均为部分非劣解被支配, 此时 R 指标的值越大代表该算法的占优情况越好, 对比结果表明 EDA 在所有问题求解中占据绝对优势; 同时, 从各算法对比数据来看, 无论是大规模还是小规模 EDA 的 R 指标均稳定在 0.6 以上且明显优于其他算法; 此外, EDA 的 N 指标在多数情况下均大于 1, 表明 EDA 所得非劣解集不仅具备优良的质量, 同时解的数量也优于对比算法.

综上所述, 在以上所有问题规模中, EDA 在求解 GNWJSSP 问题时, 其算法性能明显优于 IMMOGLS、SGA、SFLA 的结果, 能够找到数量更多、质量更好的非劣解. 因此, 对比结果证明了所提算法求解 GNWJSSP 的有效性.

5 结论

为综合考虑生产过程中的经济指标和环境指标影响, 本文提出一种基于 EDA 的多目标优化算法, 用于求解以最小化最大完工时间和总能量消耗量为优化目标的零等待作业车间调度问题. 首先, 采用双种群增加种群的多样性, 并加入启发式操作 NEH 以提高初始解的质量. 其次, 设计自适应学习速率以控制优良解信息在整个算法搜索过程中的引导作用. 然后, 结合零等待作业车间调度问题(NWJSSP)的结构性质与解码方式, 提出基于前端省略的快速评价方法, 并据此构建 Insert 邻域的快速局部搜索策略, 可提高搜索效率. 同时, 采用 Insert 和 Swap 两种邻域操作构建协同局部搜索策略, 有助于增强算法的搜索深度与广度. 最后, 通过不同规模经典测试问题的仿真实验和算法对比, 验证了所提算法求解 GNWJSSP 的有效性.

参考文献

- [1] Tang L, Liu J, Rong A, et al. A mathematical programming model for scheduling steelmaking-continuous casting production[J]. *European Journal of Operational Research*, 2000, 120(2): 423 - 435.
- [2] Grabowski J, Pempera J. Sequencing of jobs in some production system[J]. *European Journal of Operational Research*, 2000, 125(3): 535 - 550.
- [3] Cho S Y. Complexity of scheduling shops with no wait in process[J]. *Mathematics of Operations Research*, 1979, 4(4): 448 - 457.
- [4] Mascis A, Pacciarelli D. Job-shop scheduling with blocking and no-wait constraints[J]. *European Journal of Operational Research*, 2002, 143(3): 498 - 517.
- [5] Pei Z, Zhang X, Zheng L, et al. A column generation-based approach for proportionate flexible two-stage no-wait job shop scheduling[J]. *International Journal of Production Research*, 2019, 58(10): 1 - 22.
- [6] Zhu J, Li X, Wang Q. Complete local search with limited memory algorithm for no-wait job shops to minimize makespan[J]. *European Journal of Operational Research*, 2009, 198(2): 378 - 386.
- [7] Li X, Xu H, Li M. A memory-based complete local search method with variable neighborhood structures for no-wait job shops[J]. *International Journal of Advanced Manufacturing Technology*, 2013, 87(5 - 8): 1401 - 1408.
- [8] Li X, Xu H, Li M. A memory-based complete local search method with variable neighborhood structures for no-wait job shops[J]. *The International Journal of Advanced Manufacturing Technology*, 2016, 87(5 - 8): 1401 - 1408.
- [9] Sundar S, Suganthan P N, Jin C T, et al. A hybrid artificial bee colony algorithm for the job-shop scheduling problem with no-wait constraint[J]. *Soft Computing*, 2017, 21(5): 1193 - 1202.
- [10] Deng G L, Zhang Z W, Jiang H T, et al. Total flow time minimization in no-wait job shop using a hybrid discrete group search optimizer[J]. *Applied Soft Computing Journal*, 2019, 81(1): 1 - 15.
- [11] Deng G L, Su Q T, Zhang Z W, et al. A population-based iterated greedy algorithm for no-wait job shop scheduling with total flow time criterion[J]. *Engineering Applications of Artificial Intelligence*, 2020, 88(1): 1 - 12.
- [12] Mouzon G, Yildirim M B. A framework to minimise total energy consumption and total tardiness on a single machine[J]. *International Journal of Sustainable Engineering*, 2008, 1(2): 105 - 116.
- [13] Luo H, Du B, Huang G Q, et al. Hybrid flow shop scheduling considering machine electricity consumption cost[J]. *International Journal of Production Economics*, 2013, 146(2): 423 - 439.
- [14] Mansouri S A, Aktas E, Besikci U. Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption[J]. *European Journal of Operational Research*, 2016, 248(3): 772 - 788.
- [15] Zhang R, Chiong R. Solving the energy-efficient job shop scheduling problem: A multi-objective genetic algorithm with enhanced local search for minimizing the total weigh-

- ted tardiness and total energy consumption[J]. Journal of Cleaner Production, 2016, 112(4): 3361 – 3375.
- [16] Ding J Y, Song S, Wu C. Carbon-efficient scheduling of flow shops by multi-objective optimization[J]. European Journal of Operational Research, 2016, 248(3): 758 – 771
- [17] Lei D, Zheng Y, Guo X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption[J]. International Journal of Production Research, 2017, 55(11): 3126 – 3140.
- [18] Brizuela C A, Zhao Y, Sannomiya N. No-wait and blocking job-shops: Challenging problems for GA's[A]. IEEE International Conference on Systems [C]. Tucson, AZ, USA: IEEE, 2001. 2349 – 2354.
- [19] Ishibuchi H. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling[J]. IEEE Transactions on Evolutionary Computation, 2003, 7(2): 204 – 223.
- [20] Mokhtari H, Hasani A. An energy-efficient multi-objective optimization for flexible job-shop scheduling problem [J]. Computers & Chemical Engineering, 2017, 104(1): 339 – 352.
- [21] Qian Bin, LI Zihui, Hu Rong, et al. A hybrid differential evolution algorithm for the multi objective reentrant job-shop scheduling problem[A]. IEEE International Conference on Control and Automation (ICCA) [C]. Hangzhou: IEEE, 2013. 485 – 489.
- [22] 王春, 田娜, 纪志成, 等. 求解模糊柔性作业车间调度的多目标进化算法[J]. 电子学报, 2017, 45(12): 2909 – 2916.
WANG Chun, TIAN Na, JI Zhi-cheng, et al. Multi-objective evolutionary algorithm to solve fuzzy flexible job shop scheduling problem [J]. Acta Electronica Sinica, 2017, 45(12): 2909 – 2916. (in Chinese)
- [23] 姚友杰. 本文算法及对比例算法 SAGA, IMMOGLS 和 SF-LA 的参数实验及结果 [CP/DK]. <https://pan.baidu.com/s/1HVDMhWXHIQP2iRELJZ2ICA>, 2020-06-29.

作者简介



姚友杰 男, 1992 年 10 月出生, 河南洛阳人. 2019 年在昆明理工大学获工学硕士学位. 主要研究方向为智能优化与调度.
E-mail: youjie_yao@163.com



钱斌(通讯作者) 男, 1976 年 11 月出生, 云南曲靖人. 2009 年在清华大学获工学博士学位. 现为昆明理工大学教授, 博士生导师. 主要研究方向为调度与优化.
E-mail: bin.qian@vip.163.com



董钰明 女, 1996 年 12 月出生, 河南南阳人. 昆明理工大学信息工程与自动化学院硕士研究生. 主要研究方向为智能优化与调度.



胡蓉 女, 1974 年 11 月出生, 贵州安顺人. 2004 年在清华大学获工学硕士学位. 现为昆明理工大学副教授, 硕士生导师. 主要研究方向为优化方法和决策支持系统.



王凌 男, 1972 年 8 月出生, 江苏常州人. 1999 年在清华大学获工学博士学位. 现为清华大学教授, 博士生导师. 主要研究方向为智能优化调度理论与方法.