

变邻域保优遗传算法求解柔性车间调度问题

吴树景, 游有鹏, 罗福源

南京航空航天大学 机电学院, 南京 210016

摘 要:针对单目标柔性作业车间调度问题,以最大完工时间为优化目标,将遗传算法、变邻域搜索算法与精英保护策略相结合,提出一种运算效率和求解性能均较好的混合算法。首先建立数学模型,阐述算法的整体流程。对遗传算子进行改进,并添加改良的保优记忆库对精英个体进行保护。基于关键工序调整,设计了“同机器工序调整”“变机器工序调整”“双工序调整”三种邻域结构,增强了局部搜索能力,并给出一种高效的关键工序寻找法则。通过对基准算例及数值实验的测试,验证了该算法的有效性和可行性。

关键词:柔性作业车间调度(FJSP);遗传算法;变邻域搜索;精英保护策略

文献标志码:A **中图分类号:**TP18;TP301 **doi:**10.3778/j.issn.1002-8331.1908-0462

吴树景, 游有鹏, 罗福源. 变邻域保优遗传算法求解柔性车间调度问题. 计算机工程与应用, 2020, 56(22):236-243.

WU Shujing, YOU Youpeng, LUO Fuyuan. Genetic-variable neighborhood search algorithm with elite protection strategy for flexible job shop scheduling problem. Computer Engineering and Applications, 2020, 56(22):236-243.

Genetic-Variable Neighborhood Search Algorithm with Elite Protection Strategy for Flexible Job Shop Scheduling Problem

WU Shujing, YOU Youpeng, LUO Fuyuan

College of Mechatronics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract: Aiming at single-objective Flexible Job Shop Scheduling Problem(FJSP) with the objective of optimizing the maximum completion time, a hybrid algorithm incorporating genetic algorithm, variable neighborhood search and elite protection strategy is proposed. Firstly, the mathematical model is established and the overall flow of the algorithm is presented. Then, the genetic operator is improved and the improved memory base is added to protect the elite individuals. Secondly, based on the critical operations adjustment, three neighborhood structures are designed to enhance the local search ability, which are same-machine moving operation, cross-machine moving operation and double operations adjustment. Finally, the benchmark examples and numerical experiments are used to test and verify the effectiveness of the proposed method.

Key words: Flexible Job Shop Scheduling Problem(FJSP); genetic algorithm; variable neighborhood search; elite protection strategy

1 引言

柔性作业车间调度问题(Flexible Job Shop Scheduling Problem, FJSP)是当前智能制造及工业自动化领域研究的热点问题,在离散制造业、流程工业中应用广泛。FJSP是典型的NP-hard问题,工艺约束较多,每个工件的每道工序都可在可选择的有限台机器上加工,计

算复杂性高。目前求解FJSP常用的智能优化算法包括遗传算法、蚁群算法、粒子群算法、模拟退火算法、禁忌搜索算法等。

遗传算法(Genetic Algorithm, GA)是一种模拟自然进化机制的全局优化算法^[1],优化过程不受限制性条件的约束,全局搜索能力很强,但存在稳定性差、爬山能

基金项目:江苏省自然科学基金优秀青年基金(No.BK20160084);中央高校基本科研业务费专项资金(No.NS2016056)。

作者简介:吴树景(1994—),男,硕士研究生,研究方向为柔性作业车间调度与优化;游有鹏(1960—),通信作者,男,博士,教授,研究方向为智能制造与智能控制, E-mail: youypeng@163.com;罗福源(1977—),男,博士,副教授,研究方向为高速高精与智能数控技术。

收稿日期:2019-08-30 **修回日期:**2019-11-29 **文章编号:**1002-8331(2020)22-0236-08

CNKI网络出版:2019-12-06, <http://kns.cnki.net/kcms/detail/11.2127.TP.20191205.1621.016.html>

力弱、易陷入局部最优等缺点。周頔^[2]融合了遗传策略和蚁群策略,有效提升了择优能力和求解精度。宋存利^[3]采用贪婪策略对GA算子进行改进,增强了局部寻优能力。

变邻域搜索(Variable Neighborhood Search,VNS)算法是一种高效的局部优化算法^[4],它旨在通过不同邻域的递进式排查,在反复迭代中使当前的局部最优解向最优解进一步靠拢,局部搜索能力很强。VNS算法在深入搜索的过程中采用多个而非单个邻域结构,因此可获得比固定邻域局部搜索算法更强的寻优能力和搜索效率^[5]。崔琪等^[6]提出了改进混合变邻域搜索的遗传算法,设计了“两点交换”“反转逆序”“打乱互换”等五种邻域结构,提升了解集质量。王丹敬等^[7]设计了自适应变邻域搜索算法,利用“2-insertion”等邻域快速获得高质量的近优解。

目前大部分优化算法的盲目性和随机性较强,与FJSP的特点联系弱,稳定性差。由上述文献分析可知,GA与VNS算法构成极好互补,二者融合可有效增强FJSP求解性能。而国内此方面的研究较为匮乏,主要体现有:遗传算子跳坑能力不足,自适应性弱;缺乏优良个体保护机制;很多邻域本质为枚举,当问题规模变大时待试探的可行解集规模也会呈指数上升,寻优效率极大降低。故遗传算子优化及邻域设计方面仍有很大提升空间。

本文结合GA全局搜索能力强和VNS算法局部优化效果好的特点,将GA与VNS算法相结合以平衡寻优的广度与深度。同时,对GA算子进行改进,并引入保优记忆库策略对精英个体进行保护。其次,针对FJSP的特点设计了三种基于关键工序进行调整的邻域结构,给出了改进的关键工序寻找法则和甘特图向编码映射的工序调整方案。最后,通过算法测试与数值实验,采用基准算例与其他现有算法进行横向测评,证明所提算法的可行性与有效性。

2 FJSP数学模型

n 个工件(J_1, J_2, \dots, J_n)要在 m 台机器(M_1, M_2, \dots, M_m)上加工;每个工件包含一道至多道工序;每道工序可在多台不同机器上加工;不同机器加工各道工序的时间不同。由此,FJSP问题本质上由两个子问题组成:机器选择和工序排序。

本文选取“最大完工时间 C_{\max} 最小”作为优化目标,如式(1)所示:

$$C_{\max} = \min \left(\max_{1 \leq j \leq n} (C_j) \right) \quad (1)$$

式中, C_j 为第 j 个工件最后一道工序的完成时间, C_{\max} 为全部工件中最大的完工时间即最大完工时间(makespan)。优化目标的约束条件如下:

$$c_{ik} - t_{ik} + M(1 - a_{ikh}) \geq c_{ih} \quad (2)$$

$$c_{jk} - c_{ik} + M(1 - x_{ikh}) \geq t_{jk} \quad (3)$$

其中:

$$a_{ikh} = \begin{cases} 1, & \text{机器} h \text{ 先于} k \text{ 加工工件} i \\ 0, & \text{其他情况} \end{cases} \quad (4)$$

$$x_{ijk} = \begin{cases} 1, & \text{工件} i \text{ 先于工件} j \text{ 在机器} k \text{ 上加工} \\ 0, & \text{其他情况} \end{cases} \quad (5)$$

$$c_{ik} \geq 0 \quad (6)$$

$$i, j = 1, 2, \dots, n$$

$$h, k = 1, 2, \dots, m$$

式中, c_{ik} 和 c_{jk} 分别为工件 i 和 j 在机器 k 上的完成时间, c_{ih} 为工件 i 在机器 h 上的完成时间, t_{ik} 和 t_{jk} 分别为工件 i 和 j 在机器 k 上的加工时间; M 是一个足够大的正数; a_{ikh} 为决定工序先后顺序的决策变量, x_{ijk} 为工序分配机器的决策变量。

式(2)表示顺序约束条件,保证每个工件的加工顺序满足预先的要求。

式(3)表示资源约束条件,即机器加工各工件的先后顺序,限定每台机器一次只能加工一个工件。

3 变邻域保优遗传算法

本文综合遗传算法的全局搜索优势与变邻域搜索算法的局部搜索优势,加入改进的保优记忆库策略,提出了基于操作编码的变邻域保优遗传算法。

3.1 整体流程框架

整体算法流程框架如图1所示。

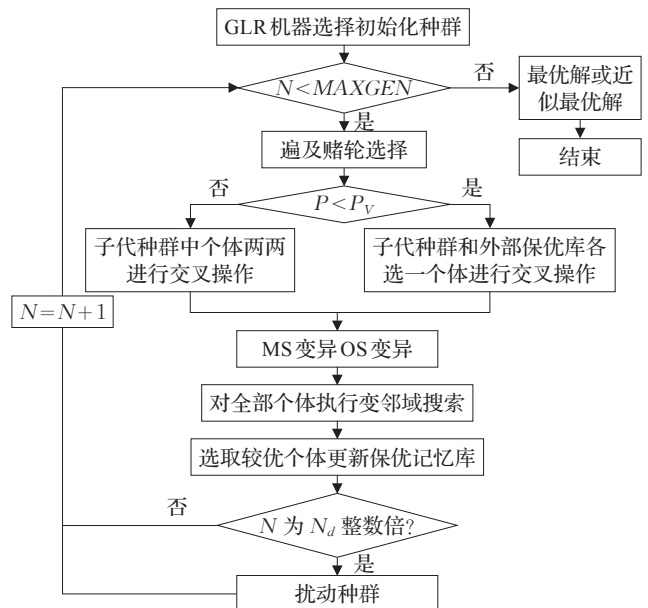


图1 整体算法流程框架图

算法具体步骤如下:

步骤1 使用GLR机器选择法^[8]对种群进行初始化,生成质量优秀的初始解集。

步骤2 判断当前迭代次数 N 是否已达最大迭代次数 $MAXGEN$ 。若是,则输出最优解或近似最优解;否则,继续执行步骤3。

步骤3 执行遍及赌轮选择操作,生成子代种群。

步骤4 根据交叉概率 P_c 选取进行交叉的个体,再根据交叉方式选择概率 P_v 采用以下两种方式中的一种执行交叉操作:子代种群中个体两两进行交叉,子代种群和外部保优记忆库中各选一枚个体进行交叉。

步骤5 根据变异概率 P_m 选取进行变异的个体,执行变异操作。

步骤6 对全部个体执行变邻域搜索,用优良解替代劣解。

步骤7 执行精英保留策略,选取当前代新种群中的较优个体更新保优记忆库,对精英个体进行保护。

步骤8 每隔 N_d 迭代次数对种群进行一次扰动,以随机选择的方式生成扰动比例 P_d 乘以种群规模数量的新个体,并替换当前代种群中同数目的较差个体,以此增强种群的活跃性和丰富度,防止陷入局部最优解。

3.2 染色体编码与解码

本文采用MSOS染色体编码方案^[9],如图2所示,编码由两部分组成:机器选择部分(Machines Selection, MS)和工序排序部分(Operations Sequencing, OS)。机器选择部分各基因位依次按照工件及工件工序的顺序排列,基因位的值表示该工序选择的加工机器在可选机器集中的序号;工序排序部分各基因位值为工件号,某一位置上工件号已出现的次数代表属于该工件的工序号。

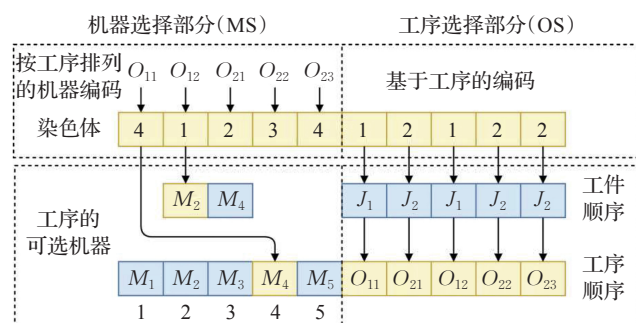


图2 FJSP染色体编码示例图

解码时采用文献[8]给出的前沿贪心方式解码成活动调度,目的是尽量将工序安排至对应机器的最早可行加工时刻,将整体调度的最大完工时间缩至最短。

3.3 遗传算法

3.3.1 种群初始化

本文采用GLR机器选择初始化的方式^[8],包含全局选择(Global Selection, GS)、局部选择(Local Selection, LS)和随机选择(Random Selection, RS)。GS旨在从全局的角度保证所有加工机器负载均衡;LS旨在从各工件独立的角度保证所有加工机器负载均衡,以提高机器的利用率;RS旨在使初始种群尽量地分布于整个解空间,提高种群多样性。三者按各自比例均匀作用于初始种群中的个体,可显著提高初始种群中的染色体在MS部分的质量。

间,提高种群多样性。三者按各自比例均匀作用于初始种群中的个体,可显著提高初始种群中的染色体在MS部分的质量。

3.3.2 选择算子

本文对传统的轮盘赌选择算子进行了优化,改进为遍及赌轮选择算子,如图3所示。给定参数GGAP为子代和父代种群大小差异的代沟比例,为赌盘均匀设置数目等同于子代所需个体数的多个取样指针,旋转一次即可根据适应度值筛选得出整个子代种群,在保证选择结果分布均匀的前提下有效提升选择效率。

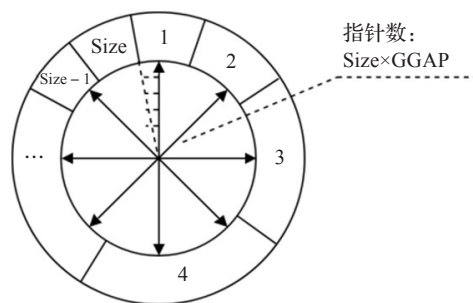


图3 遍及赌轮选择

3.3.3 交叉算子

本文在选择交叉操作执行的个体时,根据式(7)所示的交叉方式选择概率 P_v , N 为当前迭代次数, $MAXGEN$ 为总迭代次数, P 为随机生成概率。分别选用两种交叉方式: $P > P_v$ 时,子代种群中个体两两进行交叉; $P < P_v$ 时,子代种群和外部保优记忆库中各选一枚个体进行交叉。

$$P_v = \cos\left(\frac{\pi}{2 \times (MAXGEN - 1)} \times (N - 1)\right) \quad (7)$$

P_v 随 N 变化的曲线如图4所示,可看出:前期 P_v 较大,多采用第二种交叉方式,以此快速寻找到较优解,提升收敛速度;后期 P_v 较小,多采用第一种交叉方式,以此增加新个体产生的机会,避免陷入早熟的陷阱。

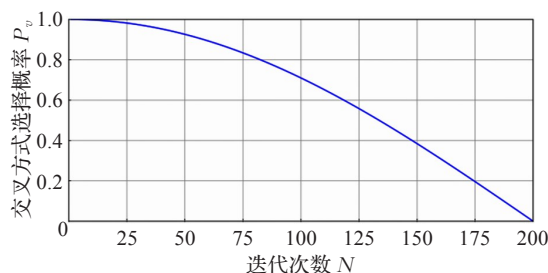


图4 交叉方式选择概率 P_v 图形

详细交叉策略如下:

机器选择(MS)部分采用均匀交叉操作,如图5所示,交换两父代染色体 n 个随机位置点处的基因片段。

工序排序(OS)部分采用改进的POX交叉方法^[10],如图6所示,划分工件集为两个非空集合,其中一集合中工件工序的基因在两父代染色体中保持不变,顺序交换两父代染色体另一集合中的基因。

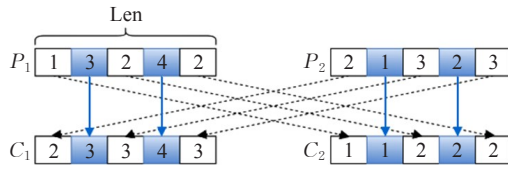


图5 均匀交叉

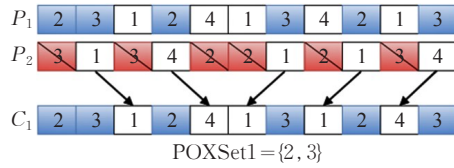


图6 改进 POX 交叉

3.3.4 变异算子

机器选择(MS)部分采用随机分配可选机器的操作^[11],如图7所示,随机选择 n 个基因位(n 不大于MS部分编码长度的 1/4),用其对应可选加工机器替换。

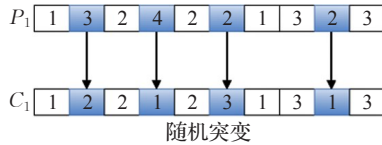


图7 MS部分变异

工序排序(OS)部分采用插入变异的方式执行变异操作,如图8所示,在工序编码中随机选取一个基因并将其插入至OS部分的一个随机位置上。

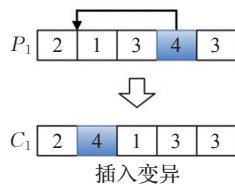


图8 OS部分变异

3.3.5 保优记忆库策略

传统遗传算法中交叉算子的作用个体全部来自于子代种群,此机制有两个明显的缺点:一是每轮迭代产生的优秀个体因得不到及时保护而易在下一轮迭代中被破坏;二是算法易过早收敛于局部最优解而导致早熟。本文针对该弊端提出一种改良的保优记忆库机制,将搜索过程中的精英个体加入库中进行保护,留待供下一轮迭代进行交叉操作。

为使保优记忆库中的精英个体尽可能分布均匀且种类丰富,引入“海明距离”的概念:两个染色体编码中相异的基因位的个数称作海明距离,简称 H 。

将精英个体更新至保优记忆库的算法流程如图9所示,其中 H_i 为待插入个体的海明距离, H_n 为保优库中第 n 个个体的海明距离。

3.4 变邻域搜索算法

本文以关键工序为调整对象,机器空闲时间为突破点缩减最大完工时间的思想构建邻域,在赵诗奎^[12]、王

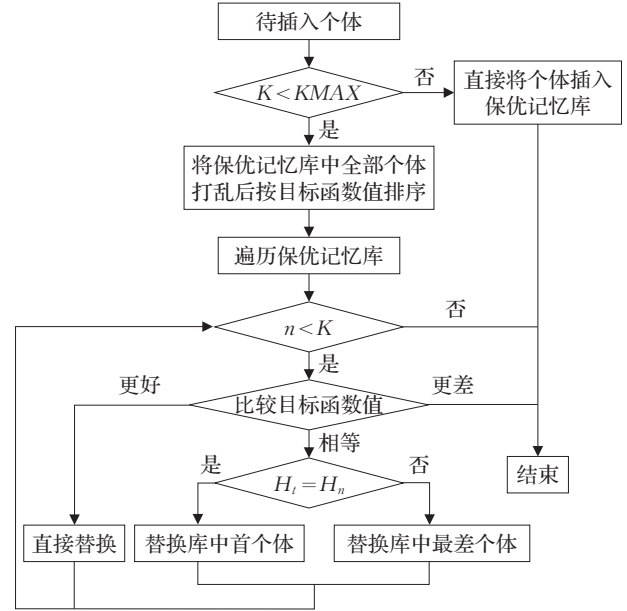


图9 保优记忆库更新流程图

磊^[13]等学者的研究基础上进一步改进:优化了关键工序的提取效率,拓展了空闲时间的搜索范围,提出了甘特图向编码映射的工序调整方案,融合了单工序与双工序的调整策略。由此提出三种邻域结构:同机器工序调整邻域、变机器工序调整邻域和双工序调整邻域。

3.4.1 变邻域搜索流程

变邻域搜索流程如图10所示,先进行单工序调整,随机选择“同机器工序调整”和“变机器工序调整”中的一个邻域先执行,再执行另一个;然后执行双工序调整。根据邻域搜索操作执行后个体的目标函数值进行取舍:若执行后的个体目标函数值优于原个体直接替换,差于原个体不替换,相等则以 50% 概率进行替换。

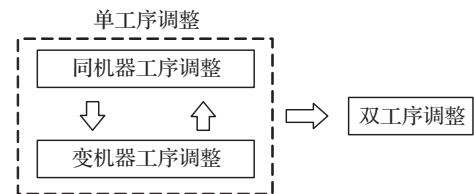


图10 变邻域搜索流程

3.4.2 关键工序提取改进方案

析取图中从起点到终点的最长路径为关键路径,对应于甘特图中工序间无时间间隔的最长路径,组成关键路径的工序为关键工序,其直接决定调度方案的最大完工时间。因此在邻域中对关键工序而非任意工序进行调整可减少搜索的盲目性,极大提高找到更优解的概率。由于每个个体进行变邻域搜索前都需对关键工序进行提取,故高效的提取方案对于整体算法快速求解尤为重要。

反向查找^[13]是较为常用的关键路径提取方案,如图11所示,其原理是:从最后完工的某一工序开始向前执行地毯式试探,同时遇到其工件前续工序和机器前续

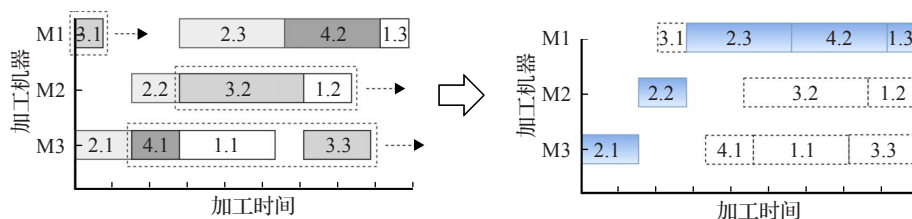


图12 关键工序锁定特性

工序时取其工件前续工序,直到完整拼凑出紧密连接的最长路径。此方法在规模较小的算例下效率较好,但随着问题规模增大,可试探的路径数量呈指数增加,搜寻效率和稳定性均会急剧下降。

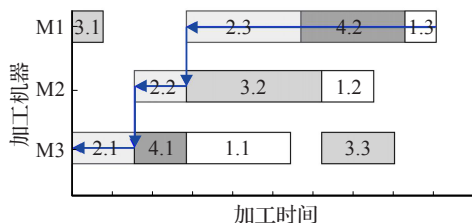


图11 关键路径反向查找法

如图12所示,在工序安排顺序和最大完工时间不变的前提下,受到属于同一工件/机器的前/后工序的最早/晚开工/完工时间约束,关键工序会被锁定,而非关键工序可在一定区间内自由移动。根据这一特性,本文提出一种较大规模算例下搜索速度和质量均较好的关键工序提取方案,相关符号变量说明如下: $s^E(h)$ 为工序 O_h 的最早开工时间, $s^L(h)$ 为工序 O_h 的最晚开工时间, $c^E(h)$ 为工序 O_h 的最早完工时间, $c^L(h)$ 为工序 O_h 的最晚完工时间; $PJ(h)$ 为工序 O_h 属于同一工件的前道工序, $SJ(h)$ 为工序 O_h 属于同一工件的后道工序, $PM(h)$ 为工序 O_h 属于同一机器的前道工序, $SM(h)$ 为工序 O_h 属于同一机器的后道工序; $p(h)$ 为工序 O_h 在当前机器的加工时间。提取过程如下:

(1) 采用文献[8]给出的前沿贪心方式将染色体解码成调度甘特图,工序 O_h 在图上的开始加工时间记为该道工序的 $s^E(h)$ 。

(2) 按式(8)计算得出工序 O_h 的 $c^L(h)$, $SJ(h)$ 或 $SM(h)$ 不存在则用 ∞ 代替, $c^L(h)$ 不可超过整体调度的最大完工时间。从后向前对各工序进行迭代计算,迭代过程中对已计算的变量进行记录,防止重复计算以提高效率。

$$c^L(h) = \min\{s^L[SJ(h)], s^L[SM(h)]\} \quad (8)$$

(3) 以式(9)为基准进行判断,若满足则判定该道工序为关键工序。

$$c^L(h) = s^E(h) + p(h) \quad (9)$$

3.4.3 同机器工序调整邻域

同机器工序调整邻域通过遍历调度方案中的每个关键工序,尝试将关键工序 O_h 调整至属于其同一加工

机器的其他工序间的空闲时间以最大限度地压缩最大完工时间。先按图13所示方法找出全部可缩减最大完工时间的调整方案,再按图14所示方法调整染色体编码,最终保留调整后进化程度最大的调整方案。

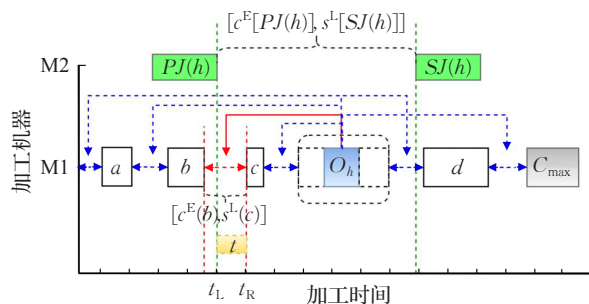


图13 同机器工序调整邻域结构示意图

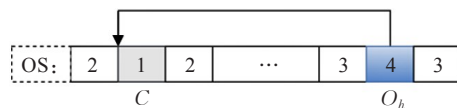


图14 甘特图向编码映射的工序调整方案

邻域结构示意图如图13所示。横向双箭头表示从0到 C_{\max} 区间内所有可供试探的空闲时间间隔。以“尝试将 O_h 移动至工序 b 和 c 间的空闲时间间隔”为例说明调整方式:对 O_h 的最大可调时间段 $[c^E[PJ(h)], s^L[SJ(h)]]$ 与工序 b 和 c 之间的最大可能空闲时间段 $[c^E(b), s^L(c)]$ 求交集,交集不为空时得到可利用区间 t ,其左边界 t_L 为 $\max\{c^E[PJ(h)], c^E(b)\}$,右边界 t_R 为 $\min\{s^L[SJ(h)], s^L(c)\}$,若 $t_R - t_L > 0$ 则代表该移动可减少最大完工时间。

染色体编码的调整方式如图14所示,同机器调整只需对编码的OS部分操作:将 O_h 对应的基因插入至工序 c 对应的基因之前,在前沿贪心式主动调度的影响下,此番调整后甘特图上 O_h 也将以最紧凑的方式插入至工序 c 之前。

3.4.4 变机器工序调整邻域

变机器工序调整邻域与同机器工序调整邻域的不同之处在于:前者尝试利用的是 O_h 当前加工机器之外的其他可选机器上工序间的空闲时间。甘特图搜寻方案和染色体编码调整方式分别如图15和图16所示。

邻域结构示意图如图15所示,遍历 O_h 当前加工机器之外的其他可选机器上0到 C_{\max} 区间内的空闲时间间隔进行试探,以“尝试将 O_h 移动至另一台加工机器上

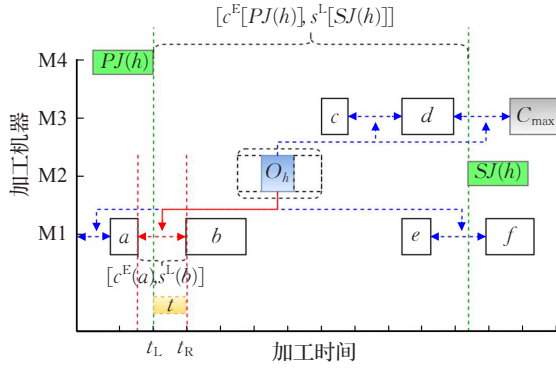


图15 变机器工序调整邻域结构示意图

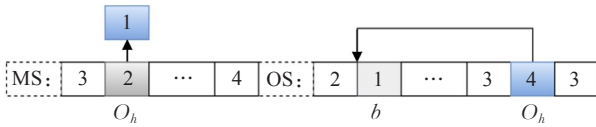


图16 甘特图向编码映射的工序调整方案

工序 a 和 b 间的空闲时间间隔”为例说明调整方式:对 O_h 的最大可调时间段 $[c^E[PJ(h)], s^L[SJ(h)]]$ 与工序 a 和 b 之间的最大可能空闲时间段 $[c^E(a), s^L(b)]$ 求交集,交集不为空时得到可利用区间 t ,其左边界 t_L 为 $\max\{c^E[PJ(h)], c^E(a)\}$,右边界 t_R 为 $\min\{s^L[SJ(h)], s^L(b)\}$,若 $t_R - t_L > p(h)$ 则代表该移动将 O_h 从原加工机器队列中抽出的同时,既不影响加入的机器队列中其他工序的原本状态,也不会产生新的关键路径,故可能减少最大完工时间。

染色体编码的调整方式如图16所示,由于是变机器调整,先将编码的MS部分 O_h 对应基因改为调整后的可选机器编号。再对编码的OS部分操作:若 O_h 基因在工序 b 基因之后,将 O_h 基因插入至工序 b 基因之前;否则无需调整OS部分编码。

3.4.5 双工序调整邻域

双工序调整邻域旨在对关键块(即关键路径上属于同一加工机器的相邻工序)内工序进行移动,且仅调整OS部分。邻域结构示意图如图17所示。该邻域基于 Van Laarhoven 等^[14]提出的邻域改进而来,删减了多处无效移动以缩小邻域规模,提升局部搜索效率的同时保证邻域结构的多样性。

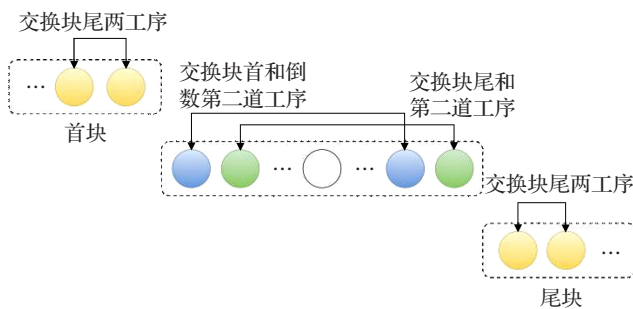


图17 双工序调整邻域结构示意图

移动过程如下:

(1)若首块包含两道及以上工序,交换块尾相连的

两道工序;若尾块包含两道及以上工序,交换块首相连的两道工序。

(2)除首块和尾块外的关键块,交换块首和倒数第二道工序,交换块尾和第二道工序。

4 算法测试与数值实验

4.1 参数设置

上述变邻域保优遗传算法采用 Visual Studio 下 C# 语言编程,程序在处理器为 Intel Core i7 四核CPU、主频为 2.60 GHz、内存为 8 GB、操作系统为 64 位 Windows10 的个人计算机上运行。

算法具体参数设置如下:

整体:种群规模 $Size = 100$,最大迭代次数 $MAXGEN = 200$,扰动间隔代数 $N_d = 20$,扰动比例 $P_d = 30\%$;种群初始化:GS、LS、RS 的比例分别为 0.6, 0.3, 0.1;选择算子:代沟比例 $GGAP = 0.9$;交叉和变异算子:交叉概率 $P_c = 0.8$,变异概率 $P_m = 0.05$;保优记忆库:库最大容量 $KMAX = 10$ 。

4.2 实验用例

(1)Kacem 8×8 、 10×10 、 15×10 三个实例^[15]。

(2)Brandimarte 十组实例(BRdata)^[16]。

4.3 变邻域搜索策略对比实验

为证明本文所设计的三个邻域拓展搜索深度的能力,使用 Brandimarte 十组实例(BRdata)中的 MK01 进行实验,分别采用添加变邻域搜索策略的遗传算法和去除变邻域搜索策略的遗传算法进行求解,重复 50 次,分别记录这两种算法求得解集中的最优个体目标函数值。

实验结果如图18所示,50次实验中,融合了变邻域搜索策略的遗传算法得到的最优个体 C_{max} 主要集中在最优解 40 上;而去除变邻域搜索策略的遗传算法得到的最优个体 C_{max} 大部分集中在次优解 42 和 41 上。经对比可证明,在求解稳定性和搜索深度上,融合了变邻域搜索策略的遗传算法明显优于去除了变邻域搜索策略的遗传算法,当问题规模增大、解空间拓扑结构复杂时,本文设计的变邻域搜索算法可在遗传算法全局优化的基础上快速稳定地深入寻找到更优解。

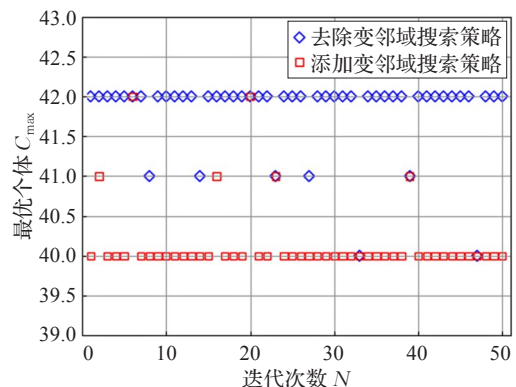


图18 算例MK01(10×6)50次求解结果

表1 各种算法对Kacem算例求解结果对比分析

问题	$n \times m$	GA+ACO		MAPSO1	Heuristic	BBO	TABC	IACO	本文算法	
		C_{\max}	$Aver$	C_{\max}	C_{\max}	C_{\max}	C_{\max}	C_{\max}	C_{\max}	$Aver$
Kacem	8×8	14	15.3	17	15	14	14	14	14	14.3
	10×10	7	7.6	8	7	7	7	7	7	7.0
	15×10	11	12.7	N/A	12	12	11	11	11	11.9

表2 各种算法对Brandimarte算例求解结果对比分析

问题	$n \times m$	LEGA		IGA		Heuristic	BBO	hDPSO	TABC	本文算法	
		C_{\max}	$Aver$	C_{\max}	$Aver$	C_{\max}	C_{\max}	C_{\max}	C_{\max}	C_{\max}	$Aver$
MK01	10×6	40	41.5	40	40.0	42	40	40	40	40	40.0
MK02	10×6	29	29.1	26	27.2	28	28	28	26	26	26.9
MK03	15×8	N/A	N/A	204	204.0	204	204	204	204	204	204.0
MK04	15×8	67	67.3	60	63.9	75	64	64	60	60	63.5
MK05	15×4	176	178.1	173	173.0	179	173	175	173	173	173.0
MK06	10×15	67	68.8	62	65.8	69	66	64	60	58	63.8
MK07	20×5	147	152.9	140	142.2	149	144	144	139	139	141.9
MK08	20×10	523	523.3	523	523.0	555	523	523	523	523	523.0
MK09	20×10	320	327.7	307	312.2	342	310	311	307	307	308.6
MK10	20×15	229	235.7	215	226.1	242	230	220	202	206	221.8

4.4 标准算例测试实验

为客观检验变邻域保优遗传算法求解FJSP问题的整体性能,进行基准算例测试,每个算例求解20次,记录最优值和平均值。

首先,对Kacem三个中小型实例进行测试,并与董蓉的GA+ACO^[17]、Nouiri的MAPSO1^[18]、Ziaee的Heuristic^[19]、Rhamati的BBO^[20]、Gao的TABC^[21]、Wang的IACO^[22]进行对比,结果如表1所示,其中 C_{\max} 为最优解, $Aver$ 为平均值。对比显示,在搜索深度上,本文算法所求得的最优值均已达到参考文献中的最好解,其余算法中仅GA+ACO、TABC和IACO算法可与本文算法持平;在搜索稳定性上,本文算法所求得的平均值优于GA+ACO算法。由此验证了本文算法应对中小型FJSP问题时深入且稳定的优势。

然后,对Brandimarte十个大中型实例进行测试,并与Ho的LEGA^[23]、刘琼的IGA^[24]、Ziaee的Heuristic^[19]、Rhamati的BBO^[20]、Gao的TABC^[21]、Shao的hDPSO算法^[25]进行对比,结果如表2和表3所示。对比显示,在求解深度上,除MK10算例外,本文算法的求解结果均取得了所列算法的最好解;在整体求解质量上,本文算法的优于解与差于解数目和TABC算法持平,而与其他五种算法相比优于解数目明显多于差于解数目。在搜索稳定性上,本文算法所求得的平均值均优于LEGA和IGA算法,且对于MK09大型算例的求解均值有明显改善。由此验证了本文算法应对大中型FJSP问题时,在保证搜索深度的前提下,进一步提升了寻优的稳定性和全面性。

表3 Brandimarte算例优于解和差于解数目对比分析

类别	LEGA	IGA	Heuristic	BBO	hDPSO	TABC
最优值优于解数目	7	3	9	6	7	1
最优值差于解数目	0	0	0	0	0	1

择取MK09实例进行完整记录, C_{\max} 平均值和最优值收敛曲线如图19所示。可见,在扰动策略的作用下,平均值收敛曲线每迭代20次会产生一次跃变,而最优值收敛曲线由次优解跳向更优解的行为也发生在跃变点附近,证明新元素的补充可适当增强种群的活跃度,配合变邻域搜索的局部优化能力可提高算法的跳坑能力。同时,在本文所设计保优记忆库策略作用下,平均值收敛曲线的跃变会快速回归平稳,且不会影响种群解的最优值。图20为该算例求得的最优调度甘特图。

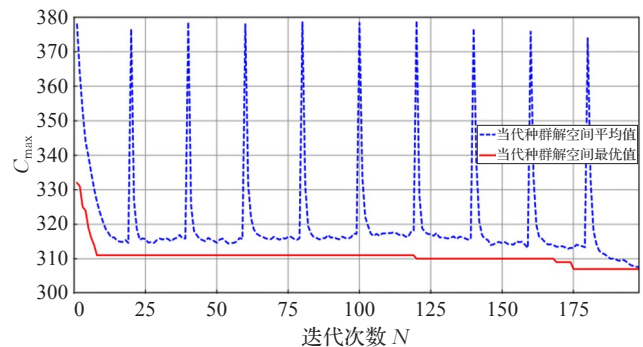


图19 算例MK09(20×10)收敛曲线

5 结束语

本文针对最小化最大完工时间的单目标FJSP,提出融合了遗传算法与变邻域搜索的混合算法,合理平衡了搜索过程的广度与深度,通过对Brandimarte等算例的测试验证了该算法的可行性与有效性。设计了一种保优记忆库策略,在有效保护精英个体的基础上进一步提升精英群体的丰富度。提出了高效稳定的关键工序搜寻方法,改进并设计了三种基于关键工序调整的邻域结构,并验证了其深化局部搜索能力、拓展寻优深度的良好效果。未来的研究中,将考虑与其他智能优化算法

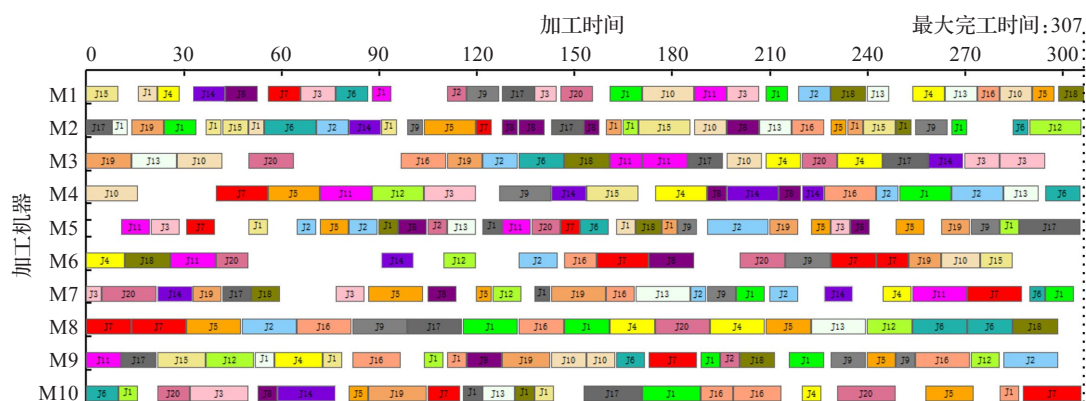


图20 算例MK09(20×10)最优调度甘特图

进行融合,并对多目标FJSP和动态调度问题进行深入探索,使求解作业车间调度问题的算法更加高效和完善,且具有更强的适应性。此外还考虑将本调度算法应用于制造车间的MES系统中,以应对实际加工过程中的任务订单调度需求。

参考文献:

- [1] Robilliard D, John H, Holland: signals and boundaries: building blocks for complex adaptive systems[J]. Genetic Programming and Evolvable Machines, 2013, 14(2): 279-280.
- [2] 周頔. 基于多种群多策略的混合遗传-蚁群算法及应用研究[J]. 计算机与数字工程, 2018, 46(12): 2390-2394.
- [3] 宋存利. 求解混合流水车间调度的改进贪婪遗传算法[J]. 系统工程与电子技术, 2019, 41(5): 1079-1086.
- [4] Hansen P, Mladenović N. Variable neighborhood search[J]. Computers & Operations Research, 1997, 24(11): 1097-1100.
- [5] 牟健慧, 潘全科, 牟建彩, 等. 基于遗传变邻域混合算法的带交货期的单机车间逆调度方法[J]. 机械工程学报, 2018, 54(3): 148-159.
- [6] 崔琪, 吴秀丽, 余建军. 变邻域改进遗传算法求解混合流水车间调度问题[J]. 计算机集成制造系统, 2017, 23(9): 1917-1927.
- [7] 王丹敬, 徐建有. 基于变邻域搜索的可重入流水车间调度[J]. 控制工程, 2018, 25(2): 362-366.
- [8] Zhang G, Gao L, Shi Y. An effective genetic algorithm for the flexible job-shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4): 3563-3573.
- [9] 张国辉. 柔性作业车间调度方法研究[D]. 武汉: 华中科技大学, 2009.
- [10] 彭运芳, 高雅, 夏蓓鑫. 不确定条件下基于遗传算法的作业车间调度问题[J]. 上海大学学报(自然科学版), 2016, 22(6): 793-803.
- [11] 济南大学. 基于加工时间等级的柔性作业车间调度机器选择方法: CN201610019013.2[P]. 2016-06-22.
- [12] 赵诗奎. 求解柔性作业车间调度问题的两级邻域搜索混合算法[J]. 机械工程学报, 2015, 51(14): 175-184.
- [13] 王磊, 黄文奇. 求解工件车间调度问题的一种新的邻域搜索算法[J]. 计算机学报, 2005, 28(5): 809-816.
- [14] Horsthuis G J B, Laarhoven J A H V, Rooij R C B M V, et al. Studies on upscaling parameters of the Gral high shear granulation process[J]. International Journal of Pharmaceutics, 1993, 92: 143-150.
- [15] Kacem I, Hammadi S, Borne P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic[J]. Mathematics and Computers in Simulation, 2002, 60: 245-276.
- [16] Brandimarte P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157-183.
- [17] 董蓉, 何卫平. 求解FJSP的混合遗传-蚁群算法[J]. 计算机集成制造系统, 2012, 18(11): 2492-2501.
- [18] Nouri M, Bekrar A, Jemai A, et al. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem[J]. Journal of Intelligent Manufacturing, 2018, 29(3): 603-615.
- [19] Ziaee M. A heuristic algorithm for solving flexible job shop scheduling problem[J]. The International Journal of Advanced Manufacturing Technology, 2014, 71: 519-528.
- [20] Rahmati S H A, Zandieh M. A new biogeography-based optimization (BBO) algorithm for the flexible job shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2012, 58: 1115-1129.
- [21] Gao K Z, Suganthan P N, Chua T J, et al. A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion[J]. Expert Systems with Applications, 2015, 42(21): 7652-7663.
- [22] Wang Lei, Cai Jingcao, Li Ming, et al. Flexible job shop scheduling problem using an improved ant colony optimization[J]. Scientific Programming, 2017(3): 1-11.
- [23] Ho N B, Tay J C. LEGA: an architecture for learning and evolving flexible job-shop schedules[C]//IEEE Congress on Evolutionary Computation, 2005.
- [24] 刘琼, 张超勇, 饶运清, 等. 改进遗传算法解决柔性作业车间调度问题[J]. 工业工程与管理, 2009, 14(2): 59-66.
- [25] Shao X, Liu W, Liu Q, et al. Hybrid discrete particle swarm optimization for multi-objective flexible job-shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2013, 67: 2885-2901.