

# 柔性作业车间调度的精确邻域结构混合进化算法

王家海, 李营力, 刘铮玮, 刘江山

(同济大学 机械与能源工程学院, 上海 201804)

**摘要:** 为解决现有基于关键路径的邻域搜索存在无效移动多、盲目性大以及仅优化单一目标的问题, 设计了更加明确精准有效的邻域结构, 包括同机器移动和跨机器移动两步操作; 在此基础上, 给出相应的关键工序精确移动条件, 并将其从优化最大完工时间推广到多目标优化; 为兼顾算法局部搜索和全局搜索, 将其与进化算法进行混合, 实现局部与全局的优势互补, 并给出相应的混合算法框架; 最后, 通过两个国际通用的案例集进行测试, 并将测试结果与成熟的算法进行对比, 验证了所设计算法的有效性和高效性。

**关键词:** 多目标优化; 车间调度; 邻域搜索; 进化算法

**中图分类号:** TP301. 6

**文献标志码:** A

## Evolutionary Algorithm with Precise Neighborhood Structure for Flexible Workshop Scheduling

WANG Jiahai, LI Yingli, LIU Zhengwei, LIU Jiangshan  
(School of Mechanical Engineering, Tongji University, Shanghai 201804, China)

**Abstract:** In order to solve the problems of the existing neighborhood search based on critical path, such as too many invalid moves, too much blindness and optimization one objective, a more precise and effective neighborhood structure is designed, including the two-step operation of the same machine movement and the cross-machine movement. Based on which, the corresponding operation movement conditions are given and extended from the optimization of the maximum completion time to multi-objective optimization. Besides, to realize the complementary advantages of local and global search, the algorithm is mixed with the evolutionary algorithm, and the corresponding hybrid algorithm framework is given. Moreover, two internationally used case sets are tested, and the test results are compared with those of other algorithms to

verify the effectiveness and efficiency of the proposed algorithm.

**Key words:** multi-objective optimization; workshop scheduling; neighborhood search; evolutionary algorithm

生产调度优化是制造执行系统的核心功能模块,也是公认的Np-hard难题<sup>[1]</sup>。求解生产调度优化问题对于企业提高设备利用率、节约生产成本以及缩短产品制造周期具有十分重要的意义。

柔性作业车间调度问题(flexible job-shop scheduling problem, FJSP)是经典作业车间调度问题的扩展,由于其打破了工序单一设备资源的约束,使生产过程具备一定的柔性,因此,FJSP更加符合生产实际,同时问题求解也变得更加复杂,尤其是多目标FJSP(multi-objective FJSP, MOFJSP)。

近年来,以智能算法为代表的MOFJSP求解方法由于其优越的全局搜索能力和易于实现的特点得到了广泛的研究和应用。然而,仅仅使用智能算法求解问题的最优解仍有一定的局限性。混合算法已被多位学者证实往往能够得到更优越的解,尤其是混合智能算法和邻域搜索。例如:Gao等<sup>[2]</sup>混合遗传算法和变邻域搜索求解MOFJSP,结果表明,该混合算法性能优于非混合智能算法。Li等<sup>[3]</sup>混合人工蜂群算法和邻域搜索,通过设计工序移动和机器置换两类邻域搜索方法有效提升了算法的求解能力。Gong等<sup>[4]</sup>混合模因算法和邻域搜索,通过设计N1和N2两种工序邻域结构求解问题的最优解,对比试验表明,该算法性能优越。Gao等<sup>[5]</sup>混合离散和声搜索算法和邻域搜索,运用公共关键块理论,针对不同优化目标采取不同邻域策略,试验结果表明,该算法明显优于非混合算法。

尽管混合邻域搜索的智能算法已有大量研究成

收稿日期: 2020-06-08

基金项目: 国家重点研发计划(2017YFE0101400)

第一作者: 王家海(1964—),男,副教授,博士生导师,工学博士,主要研究方向为数字化制造及相关技术。

E-mail: jhwang@tongji.edu.cn



论文  
拓展  
介绍

果,然而在进行邻域搜索时盲目、随机、过多无效移动以及仅能优化单一目标的问题仍然存在<sup>[6-7]</sup>。因此,如何设计有效邻域结构、如何准确选择邻域工序并将其扩展到多目标优化,仍然是当前研究的重点和难点。

为进一步提高邻域搜索的准确性和有效性,本文对基于关键路径的邻域结构进行深入研究,对已有的关键工序同机器移动和跨机器移动操作进行改进和扩展,使其更加精准有效的同时,适用于多目标邻域搜索。在此基础上,将其与进化算法进行混合求解MOFJSP,通过国际通用的基准案例进行测试,验证所提方法的有效性。

## 1 问题描述

MOFJSP问题描述为<sup>[8]</sup>:有 $n$ 个工件在 $m$ 台机器上加工;每个工件有多道工序;同一工件的不同工序之间有顺序约束;每道工序的可选加工设备有多个;同一时刻,一道工序只能在一个选定的机器上进行加工且不能中断;同一时刻,每台机器也只能加工一道工序;不同工件之间以及不同工件的工序之间无优先级约束;加工准备时间包含在工序加工时间内。

本文采用多目标车间调度研究文献中最常用的三个优化目标<sup>[9]</sup>:最小化最大完工时间 $F_1$ ,最小化关键机器负载 $F_2$ 以及最小化机器总负载 $F_3$ 。文献<sup>[10]</sup>表明这三个目标之间在一定程度上相互冲突。目标的数学描述如式(1)–(3)所示,其工序和设备约束的数学描述如式(4)–(6)所示:

$$F_1 = \min \left( \max_{i=1}^n C_i \right) \quad (1)$$

$$F_2 = \min \left[ \max \left( \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ijk} x_{ijk} \right) \right] \quad (2)$$

$$F_3 = \min \left( \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m p_{ijk} x_{ijk} \right) \quad (3)$$

s. t.

$$C_{ij} - C_{i(j-1)} \geq p_{ijk} x_{ijk} \quad (4)$$

$$\left[ (C_{hg} - C_{ij} - t_{hgk}) x_{hgk} x_{ijk} \geq 0 \right] \vee \left[ (C_{ij} - C_{hg} - t_{ijk}) x_{hgk} x_{ijk} \geq 0 \right] \quad (5)$$

$$\sum_{k \in M_{ij}} x_{ijk} = 1, \forall i, j \quad (6)$$

式中: $C_i$ 为工件 $i$ 的完工时间; $p_{ijk}$ 为工序 $O_{ij}$ 在机器 $k$ 上的加工时间; $x_{ijk}$ 为决策变量,如果工序 $O_{ij}$ 在机器 $k$ 上加工,其值为1,否则为0。

## 2 邻域搜索

基于析取图理论产生的关键路径邻域搜索是一种高效局部最优解求解方法,通常包括两步:同机器移动关键工序和跨机器移动关键工序。其核心思想是最大化利用机器上相邻工序间空闲时间,难点在于如何实现精准邻域搜索以减少无效移动。

文献<sup>[11]</sup>认为邻域搜索时,机器选择是工序排序优化的前提和基础,因此,应先进行跨机器移动关键工序,再进行同机器移动关键工序。与其观点不同,本文认为跨机器移动关键工序是寻找解空间中的点,而同机器移动关键工序是在每一个点周围寻找局部最优解。只有当前机器分配下的工序排序达到最优,才能对比机器分配方案的优劣,才能有针对性地进行跨机器移动,避免盲目跨机器移动操作。因此,应先进行同机器移动关键工序,再跨机器移动关键工序,以防止跨机器移动产生的最优解与当前非最优解比较造成当前局部最优解的遗失。

### (1) 同机器移动关键工序

长期试验发现,有效同机器关键工序移动往往存在于关键工序前移中,为此,该部分对工序前移进行深入研究,给出工序前移邻域结构和关键工序精确前移条件,与现有同机器关键工序移动操作相比,本文所设计方法更加精确,避免大量试探性无效移动,且只需对一种邻域结构进行精确移动,便可达到预期优化效果。

为介绍同机器精确移动关键工序操作,首先对一些符号进行解释说明:假设 $u$ 为一道工序,则其工件前道工序表示为JP[ $u$ ],其工件后续工序表示为JS[ $u$ ],其机器前道工序表示为MP[ $u$ ],其机器后续工序表示为MS[ $u$ ]; $S(u)$ 为工序 $u$ 的开始加工时间(也是最早开始加工时间), $S^l(u)$ 为工序 $u$ 的最晚开始加工时间, $C(u)$ 为工序 $u$ 加工结束时间(也是最早加工结束时间), $C^l(u)$ 为工序 $u$ 的最晚加工结束时间, $p_k(u)$ 为工序 $u$ 在机器 $k$ 上的加工时间。同机器移动关键工序示意图如图1所示。

由于同机器移动关键工序并不会改变工序在机器上的分配结果,因此,同机器移动关键工序可有效降低目标值 $F_1$ ,但对于目标值 $F_2$ 和 $F_3$ 并无影响。首先识别关键路径上的关键块以及关键块的后续工序。如果要减小 $F_1$ ,则关键块的后续工序的开始加工时间会因为关键块中关键工序的移动而提前,因此,工序块的后续工序之前必须要有可提前的空间,如式(7)所示:

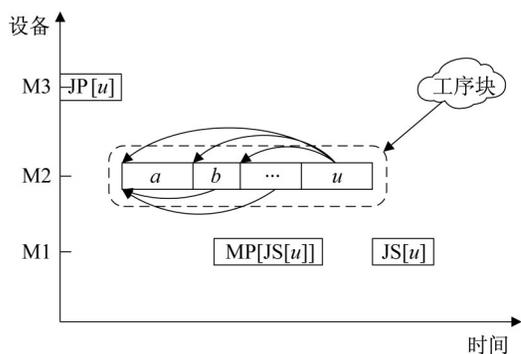


图1 同机器移动关键工序

Fig. 1 Critical operation on the same machine

$$S(JS[u]) - C(MP[JS[u]]) > 0 \quad (7)$$

式(7)称之为邻域结构条件1。同时,工序块后续工序的工件前道工序 $u$ 要包含在工序块中。此时,得到一个推论:如果该条件成立,那么 $u$ 一定为工序块的块尾工序。此条件称之为邻域结构条件2。

证明:反证法,假设 $u$ 不是工序块的块尾工序, $v$ 是工序块的块尾工序, $u$ 在 $v$ 之前,由于 $JS[u]$ 之前存在时间间隔,那么 $S(JS[u])$ 不再是工序块块尾工序的完成时间,而是 $\max\{C(u), C(MP[JS[u]])\}$ ,显然,这与关键路径的工序先后顺序约束关系矛盾,问题得证。

仅同时满足邻域结构条件1和2的工序块才会被考虑进行同机器移动关键工序操作,显然,该邻域结构可有效去除无效工序块,降低工序块选择的盲目性。如果工序块后续工序的工件前道工序 $u$ 不包括在此时的关键块中,那么其必定属于其他的关键路径并与其他工序块或关键工序构成上述关系。

在满足上述邻域结构的情况下,要使 $S(JS[u])$ 提前,有两种实现方案,第一, $u$ 位于块尾的位置不变,通过移动块内其他关键工序使得工序块整体开始加工时间提前;第二,移动块尾工序 $u$ 到工序块的其他位置,使得 $u$ 的完工时间 $C(u)$ 提前。

针对第一种情况,采取将关键块中位于块首和块尾之间的关键工序分别移动到块首之前的移动操作。假设 $a$ 为块首工序, $b$ 为块首和块尾之间的工序, $e$ 为 $b$ 移动到块首位置的跨越工序,那么只有在 $b$ 移动到块首位置之后,其开始加工时间小于原来块首 $a$ 的开始加工时间,并且其跨越工序的移动后的开始加工时间要小于其移动前最晚开始加工时间才能降低 $F_1$ 的值。为此,该类移动需要满足式(8)~(10)的约束,该约束称之为精确前移条件1。

$$S^{af}(b) < S^{bc}(a) \quad (8)$$

$$S^{af}(e) < S^{bc}(e) \quad (9)$$

$$S^{af}(e) = S^{af}(b) + p_k(a) + p_k(b) \quad (10)$$

式中:上标 $af, bc$ 分别表示工序移动之后和工序移动之前。

针对第二种情况,采取将块尾工序分别移动到块中非块尾工序前方的操作,假设 $u$ 为块尾工序, $g$ 为块中任意一个非块尾工序,那么只有块尾工序的跨越工序移动后的开始加工时间小于其移动前的最晚开始加工时间才能降低 $F_1$ 的值,为此,该类移动需要满足式(11),该约束称之为精确前移条件2。

$$S^{af}(g) < S^{bc}(g) \quad (11)$$

仅当精确前移条件1和2至少满足其一时才进行同机器关键工序移动,该方法显然比传统试探性移动判别方法更加清楚、简洁以及运算量小,避免大量无效移动。

## (2) 跨机器移动关键工序

文献[12]介绍了一种基于同机器相邻工序最早结束时间和最晚开工时间的跨机器移动关键工序方法来减少最大完工时间,与其不同,本文对其进行扩展,使其适用于求解MOFJSP。

跨机器移动关键工序示意图如图2所示,文献[12]中设定,仅当被移动的工序(此处假设为 $u$ )在重分配的机器上有足够的时间间隔时,才能减小目标值 $F_1$ ,即:

$$\{C(JP[u]), S^L(JS[u])\} \cap \{C(x), S^L(y)\} > p_1(u) \quad (12)$$

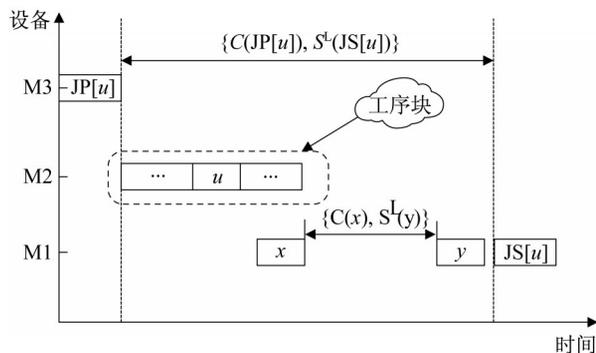


图2 跨机器移动关键工序

Fig. 2 Critical operation across machines

然而,公式(12)在求解MOFJSP时并不适用,因为该方法虽然能够降低目标 $F_1$ 的值,但跨机器移动可能会导致目标值 $F_2$ 或 $F_3$ 的增加,新解与原解构成非支配关系,因此,解并未得到优化。此外,当公式(12)取等号时,虽然不能减小目标 $F_1$ 的值,仅仅是产生了新的关键路径,然而,如果目标值 $F_2$ 和 $F_3$ 同

时被降低或者其中一个降低另外一个不增加,则新解支配原来的解,解得到了优化。因此,修正后的跨机器移动条件如式(13)~(15)所示,通过该约束,将跨机器移动关键工序由优化单目标扩展为优化多目标。

$$\{C(JP[u]), S^L(JS[u])\} \cap \{C(x), S^L(y)\} \geq p_1(u) \quad (13)$$

$$\left( \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ij1} x_{ij1} \right)^{af} \leq \max^{bc} \left( \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ijk} x_{ijk} \right) \quad (14)$$

$$\left( \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m p_{ijk} x_{ijk} \right)^{af} \leq \left( \sum_{i=1}^n \sum_{j=1}^{n_i} \sum_{k=1}^m p_{ijk} x_{ijk} \right)^{bc} \quad (15)$$

进行跨机器移动关键工序时,候选机器选择顺序也是要解决的难题,在现有文献报道中,都只提到要进行跨机器移动关键工序,但是对于候选机器选择顺序问题却很少提及,造成跨机器移动存在盲目性和大量无用试探性计算,为此,本文首次提出跨机器移动关键工序的机器选择方法:

在满足式(14)、式(15)的前提下,进行空载计算。机器可用区间空载时间小于被移动工序在该设备上加工时间的机器将被移除,即:

$$\begin{cases} \max_{i=1}^n C_i - C(JP[u]) - \sum_{i=1}^n \sum_{j=1}^{n_i} p_{ijk} x_{ijk} < p_k(u) \\ C_{ijk} > C(JP[u]) \end{cases} \quad (16)$$

为保持设备间的负载平衡,提出最小机器负载优先原则,对剩余候选设备进行负载计算,并按负载大小进行排序,负载小的设备优先进行跨机器移动;如果负载相同,则进行信息度测量计算,信息度小的将被优先选择,即:

$$\begin{cases} O_{mf} = \frac{2 \sum_i \sum_j \text{count}(O_{ijk}) \sum_i \sum_j p_{ijk} x_{ijk}}{\left[ \max_{i,j} (p_{ijk} x_{ijk}) + \min_{i,j} (p_{ijk} x_{ijk}) \right] \sum_{i=1}^n \sum_{j=1}^{n_i} \text{count}(O_{ij})} \\ C_{ijk} > C(JP[u]) \end{cases} \quad (17)$$

式中: $O_{mf}$ 代表信息度。

### 3 进化算法

与经典进化算法相比,由于本文将使用更为精确有效的邻域搜索,因此,在进化算法设计时重点在于扩大种群多样性,在解空间中获得更分散的点。同时为了便于复现算法以及保持论文结构完整性,该部分将对改进的进化算法框架进行简要介绍。

#### (1) 编码与解码

MOFJSP 包括机器选择和工序排序两个子问题,即:首先为工序选择合适的加工机器,然后对每台机器上分配的工序进行排序以满足优化目标的要求。为便于编程实现,本文采用最常用的简洁的A-B链编码方式,A链为机器选择链,B链为工序链。如图3所示,A链中每个基因代表机器号;B链中每个数字代表一个工件号,工件号出现的次数代表该工件的第几道工序。比如:A链中第一个数字2,代表工序 $O_{11}$ 由机器 $M_2$ 加工;B链中数字3第一次出现时,代表工序 $O_{31}$ ,数字3第二次出现时,代表工序 $O_{32}$ ,以此类推。

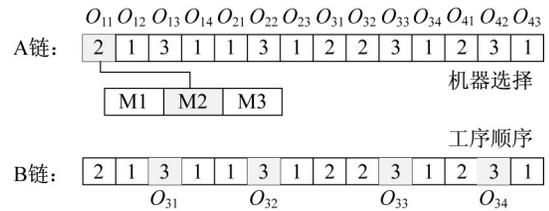


图3 染色体编码方式

Fig. 3 Chromosome coding

解码采用基于主动调度理论的左移插空法解码方式。按照B链中的工序顺序在A链对应的机器上依次进行解码,解码过程中,总是在机器上寻找可用的时间间隔,从而使得生产过程更加紧凑,缩短加工完成时间。

#### (2) 种群初始化

为确保种群多样性,种群初始化采用混合策略生成法。机器选择混合随机、最小处理时间、局部最小处理时间、全局最小处理时间4种规则;工序排序混合随机、最多负荷剩余、最多工序剩余以及最短处理时间4种规则,详细解释可参考文献[6]。通过不断随机组合不同的规则生成所需数量初始种群。

#### (3) 交叉操作

与经典遗传算法不同,为保持物种多样性,本文不再采用单一的交叉方式,混合交叉方法在本文中被设计并使用,A链交叉操作包括:MPX<sup>[1]</sup>和Uniform Crossover<sup>[10]</sup>;B链交叉操作包括:POX<sup>[13]</sup>和IPOX<sup>[1]</sup>。每次交叉,都随机选择一种交叉方法完成交叉操作。

#### (4) 变异操作

为进一步增强种群多样性,在进化算法中设计并使用了较大扰动变异方式。A链变异采用双点替换法,从A链中随机选择两个基因位,分别从各自候选设备集中随机选择与当前机器不同的机器替换当

前的机器;B链变异采用三点重置法,随机选择三个基因位,重新打乱三个基因位的位置关系。

#### (5) 选择操作

MOFJSP中染色体选择通常有两种方式:基于加权加的染色体适应度选择方法和基于支配关系的染色体选择方法。由于加权和解方式中加权数的确定较为困难,而基于支配关系的选择方式不需要考虑加权数,因此,本文采用基于支配关系的染色体选择方法。基于支配关系的染色体选择具体操作流程可参考文献[3]。

## 4 混合算法设计

进化算法具有较强全局搜索能力,然而其局部求解能力较弱,邻域搜索具有较强局部寻优能力,然而其缺点是容易陷入局部最优解。因此,结合进化算法全局搜索能力以及邻域搜索局部寻优优势可以有效克服单一算法的弊端,提高算法的整体求解性能。

混合算法框架如图4所示。先执行改进的进化算法进行全局寻优,对寻优结果执行邻域搜索以求解局部最优解。为避免重复计算,在邻域搜索之前加入去重操作,从种群列表中依次读取每个染色体,将其与其后的染色体依次进行对比,如果编码相同,后者将从种群中去除。为了维持种群数量不变,种群初始化方法再次被调用,生成所需数量的染色体。为了保存每次迭代的最优解,引入临时档案集,档案集在每次迭代中执行更新策略,非支配解被加入,支配解被删除。

## 5 案例测试

采用Python编程语言实现所述混合算法。测试所用计算机为: Intel i5-6400 CPU,主频 2.70 GHz,内存 8.00 GB。算法参数设置如下:种群规模为  $10n$ ,交叉概率为 0.8,变异概率为 0.3,最大迭代次数为 150代。为了测试所提算法的性能,本文将使用两个国际通用的标准案例集进行测试并与其他成熟的算法进行对比。第一个案例集为 Kacem<sup>[14]</sup>,其包括 5 个问题实例,分别为:  $4 \times 5, 8 \times 8, 10 \times 7, 10 \times 10, 15 \times 10$ 。第二个案例集为 BRdata<sup>[15]</sup>,其包括 10 个问题实例: MK01-MK10。

首先测试 Kacem 案例集,并将测试结果与 4 个成熟的算法进行对比,包括: HTABC<sup>[16]</sup>、P-

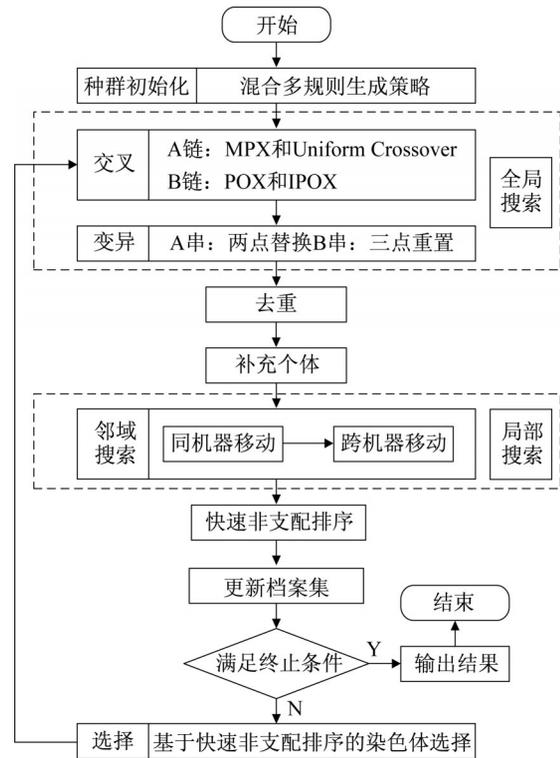


图4 混合算法框架图

Fig. 4 Framework of hybrid algorithm

DABC<sup>[3]</sup>、PSO+TS<sup>[8]</sup>以及STA<sup>[17]</sup>,本文所设计的混合算法用NGA表示。各算法求解结果如表1所示,表中加粗的解代表该解被对比算法中的相应解支配。

为了更好地描述本文所提算法的优越性,对表1中的结果进行统计综合分析,统计结果如表2所示,其中  $h_1$  为被对比算法支配的解个数,  $h_2$  为支配对比算法中解的个数,  $H$  为解的数量。从表2中可以看出,与 HTABC 相比,在 Kacem 五个问题实例上,NGA 共求得 12 个最优解,而 HTABC 仅求得 8 个最优解,在 NGA 的 12 个最优解中,仅有 1 个解被 HTABC 支配,然而,HTABC 中有 2 个解被 NGA 支配,因此,NGA 优于 HTABC;与 P-DABC 相比,尽管 P-DABC 共求得 14 个解,数量多于 NGA,然而,其有 4 个解被 NGA 支配,NGA 中不存在解被其支配,因此,NGA 要优于 P-DABC;与 PSO+TS 相比,其仅获得 5 个最优解,数量远低于 NGA,且 5 个最优解中有 2 个被 NGA 支配,NGA 中无解被其支配,因此,NGA 明显优于 PSO+TS;与 STA 相比,其仅获得 9 个最优解,且有 6 个最优解被 NGA 支配,因此,NGA 同样明显优于 STA。

为了进一步说明所提算法的优越性,采用综合评价指标反向世代距离 IGD (inverted generational

表1 Kacem 案例集求解结果  
Tab. 1 Solution results of Kacem data

算例	目标	算法											
		HTABC		P-DABC			PSO+TS		STA		NGA		
4 x 5	F <sub>1</sub>	11	11	11	12	13	11		12		11	11	12
	F <sub>2</sub>	9	10	10	8	7	10		8		9	10	8
	F <sub>3</sub>	34	32	32	32	33	32		32		34	32	32
8 x 8	F <sub>1</sub>	14	15	14	15	16	14	15	16	16	14	15	16
	F <sub>2</sub>	12	12	12	12	13	12	12	12	13	12	12	11
	F <sub>3</sub>	77	75	77	75	73	77	75	80	73	77	75	77
10 x 7	F <sub>1</sub>	11		11	12	12			11	11	11	11	
	F <sub>2</sub>	11		11	11	12			11	10	10	11	
	F <sub>3</sub>	61		63	61	60			62	63	62	61	
10 x 10	F <sub>1</sub>	7		7	8	8	7		7	8	7	8	8
	F <sub>2</sub>	6		5	5	7	6		5	6	5	5	7
	F <sub>3</sub>	41		43	42	41	43		45	42	43	42	41
15 x 10	F <sub>1</sub>	11	12	11	12		11		12	13	11		
	F <sub>2</sub>	11	11	11	11		11		10	12	11		
	F <sub>3</sub>	93	91	93	91		93		95	93	91		

表2 Kacem 案例集统计结果  
Tab. 2 Statistics of Kacem data

算例	HTABC	P-DABC	PSO+TS	STA	NGA
	<i>h</i> <sub>1</sub> , <i>h</i> <sub>2</sub> , <i>H</i>	<i>h</i> <sub>1</sub> , <i>H</i>			
4 x 5	0, 0, 2	0, 0, 3	0, 0, 1	0, 0, 1	0, 3
8 x 8	0, 0, 2	0, 0, 3	0, 0, 2	1, 0, 2	0, 3
10 x 7	0, 0, 1	2, 0, 3		2, 0, 2	0, 2
10 x 10	0, 1, 1	0, 0, 3	1, 0, 1	2, 0, 2	1, 3
15 x 10	2, 0, 2	2, 0, 2	1, 0, 1	1, 0, 2	0, 1
合计	2, 1, 8	4, 0, 14	2, 0, 5	6, 0, 9	1, 12

distance, IGD)进行分析,IGD 计算方法如公式(18)所示,其中, $P^*$ 表示帕累托前沿,其由最优解构成; $|P^*|$ 代表帕累托前沿解的个数; $d(x, y)$ 为欧式距离。计算时,所有染色体目标值均做归一化处理,处理方法如公式(19)。IGD 值越小,说明越靠近帕累托前沿,解越好。表3为IGD 计算值(以  $I_{GD}$  表示),从表中可以看出,NGA 相比其他算法更接近帕累托前沿面,因此,得出结论NGA 更优越。

$$I_{GD}(A, P^*) = \frac{1}{|P^*|} \sum_{x \in P^*} \min_{y \in A} d(x, y) \quad (18)$$

$$\tilde{f}(x) = \frac{f_i(x) - f_i^{\min}}{f_i^{\max} - f_i^{\min}}, i = 1, 2, 3 \quad (19)$$

表3 Kacem 案例的IGD 测度值  
Tab. 3 IGD value of Kacem data

案例	IGD				
	HTABC	P-DABC	PSO+TS	STA	NGA
4 x 5	0.533 760	0.263 523	0.846 856	0.695 434	0.195 434
8 x 8	0.381 342	0.190 662	0.381 324	0.627 289	0.190 662
10 x 7	0.589 197	0.422 531		0.656 125	0.388 889
10 x 10	0.617 584	0.235 702	0.715 215	0.676 925	0.235 702
15 x 10	0.683 013	0.683 013	0.683 013	0.612 372	0.612 372

此外,给出了NGA在10×7问题实例上最优解[11, 11, 61]的甘特图,如图5所示。

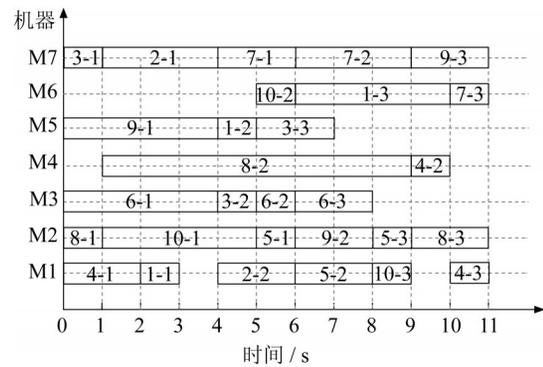


图5 10×7问题实例解[11, 11, 61]的甘特图  
Fig. 5 Gantt chart of the solution [11, 11, 61] of the 10×7 problem

对于BRdata 案例集,得到的最优解不止一个,限于篇幅,无法通过列表的形式全部给出,因此采用两种方法来验证所设计算法的性能。第一,对比单个解的质量,从所设计算法的所有解中选择一个最优的解与其他成熟的多目标算法同规则选择的最优解进行对比,最优解选择规则为: $F_1$ 最小的解优先被选中,如果 $F_1$ 值相同,则选择 $F_2$ 最小的解;第二,对比所有解的质量,基于支配关系对比解的总体特性。

对比单个解的质量,对比算法包括:AESM<sup>[18]</sup>和AIA<sup>[19]</sup>。表4给出了对比结果,其中粗体解表示该解被对比算法中的相应解支配。从表4中可以看出,与AESM 相比,其有4个解被NGA 支配,而NGA 中仅有2个解被其支配,因此,NGA 在多目标求解能力上优于AESM;与AIA 相比,其有8个解被

NGA支配,然而NGA中没有解被AIA支配,因此,得出结论,NGA的多目标求解能力要明显优于AIA。

表4 对比单个解的质量

Tab. 4 Comparison of quality of single solution

案例	AESM			AIA			NGA		
	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>	F <sub>1</sub>	F <sub>2</sub>	F <sub>3</sub>
MK01	42	42	162	40	36	171	40	36	167
MK02	28	28	155	26	26	154	26	26	154
MK03	204	204	852	204	204	1 207	204	204	1 092
MK04	68	67	352	60	60	403	60	60	396
MK05	177	177	702	173	173	686	172	172	687
MK06	75	67	431	63	56	470	58	56	447
MK07	150	150	717	140	140	695	139	139	693
MK08	523	523	2 524	523	523	2 723	523	523	2 629
MK09	311	299	2 374	312	306	2 591	307	301	2 560
MK10	227	221	1 989	214	206	2 121	200	198	1 857

对比所有解的质量,对比算法包括:MGA<sup>[1]</sup>、PSO<sup>[20]</sup>以及BEG<sup>[21]</sup>,对比结果如表5和表6所示。为了表明所提算法的优越性,C度量指标被引入,如

式(20)所示,其代表B中的解被A中解所支配的比例,C(A,B)值越小,表明B中解的质量越好。

$$C(A, B) = \frac{|\{b \in B | \exists a \in A: a > b\}|}{|B|} \quad (20)$$

由表5可知,与MGA相比,NGA在7个问题实例上优于MGA,仅在1个问题实例上MGA优于NGA;与PSO相比,NGA在所有问题实例上都优于PSO;与BEG相比,NGA在9个问题实例上优于BEG,而BEG没有问题实例优于NGA。因此,得出结论,本文所提算法求得的解总体特性更好。

由表6可知,NGA与MGA、PSO、BEG相比IGD值更小,解更贴近帕累托前沿面,因此,同样得出结论,NGA求得的解的总体特性更加优越。

为了表明解在三维空间中的分布特性,给出了问题实例MK01、MK03、MK07和MK10的帕累托前沿图,如图6所示。图中,总机器负载计算的是机器在一定时间内有多少时间处于工作状态。

表5 C测度对比结果

Tab. 5 Comparison of results of metric C

案例	MGA		PSO		BEG	
	C(NGA, MGA)	C(MGA, NGA)	C(NGA, PSO)	C(PSO, NGA)	C(NGA, BEG)	C(BEG, NGA)
MK01	0.750 000	0	1.000 000	0	0.875 000	0
MK02	0.333 333	0.125 000	0.600 000	0.250 000	0.333 333	0.125 000
MK03	0.100 000	0.076 923	0.166 667	0.076 923	0.100 000	0.076 923
MK04	0.400 000	0.200 000	0.333 333	0	0.500 000	0.133 333
MK05	0	0	0.250 000	0.090 909	0.285 714	0
MK06	0.100 000	0	0.333 333	0	0.100 000	0
MK07	0	0	0.200 000	0.066 667	0	0
MK08	0	0.111 111	0.250 000	0.222 222	0.375 000	0.111 111
MK09	0.666 666	0	1.000 000	0	0.666 666	0
MK10	1.000 000	0	1.000 000	0	1.000 000	0

表6 BRdata案例的IGD测度值

Tab. 6 IGD value of BRdata

案例	IGD			
	MGA	PSO	BEG	NGA
MK01	0.234 773	0.407 700	0.234 773	0
MK02	0.124 628	0.221 093	0.124 628	0.040 816
MK03	0.341 530	0.402 746	0.341 530	0.086 043
MK04	0.240 007	0.396 883	0.259 973	0.040 089
MK05	0.238 565	0.317 146	0.098 543	0.006 949
MK06	0.140 546	0.286 652	0.140 546	0.054 867
MK07	0.155 072	0.273 034	0.155 072	0.005 325
MK08	0.128 835	0.501 509	0.054 861	0.058 706
MK09	0.349 444	0.404 490	0.349 444	0.019 005
MK10	0.209 871	0.474 554	0.277 925	0

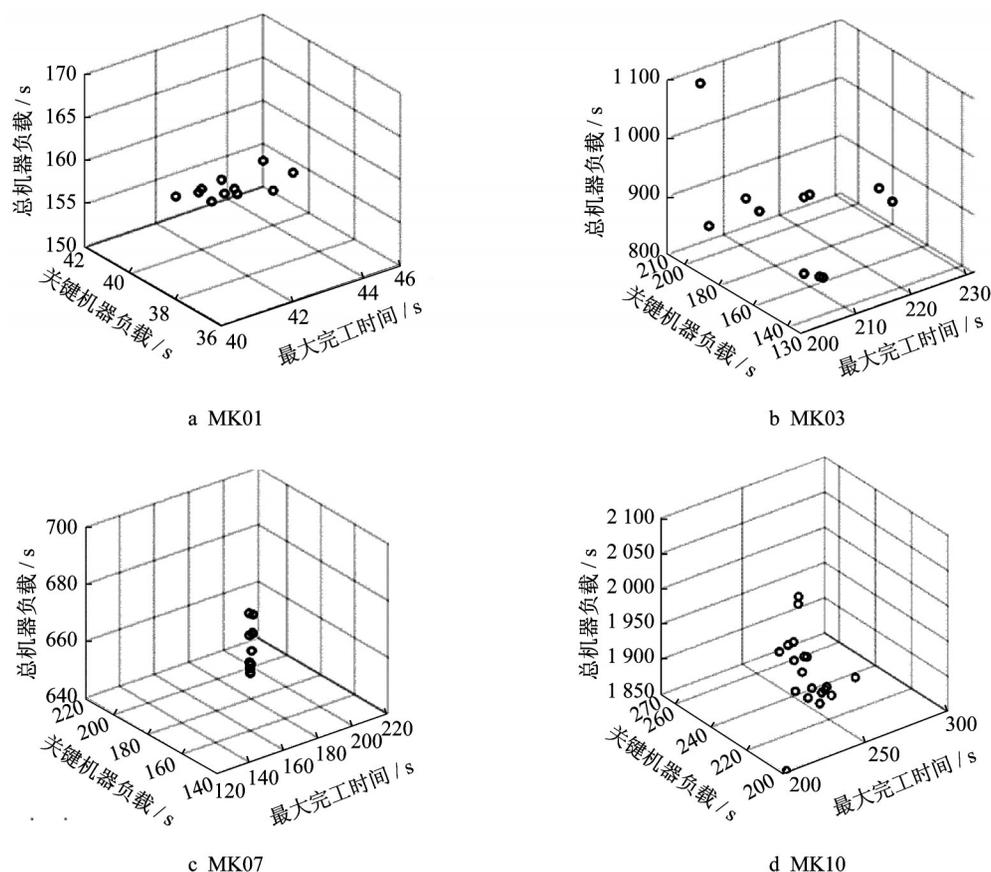


图6 帕累托前沿面

Fig. 6 Pareto front

## 作者贡献申明:

王家海:提出研究方向,给予建设性建议;

李营力:完成实验并撰写文章;

刘铮玮:协助实验,分析数据;

刘江山:给出论文修改建议和格式检查。

## 6 结论

本文针对MOFJSP问题,提出了一种混合精确邻域搜索的多目标进化算法。首先,针对同机器移动关键工序设计了更为明确精准有效的邻域搜索结构,为了适应该邻域结构,给出了目标优化的两种关键工序精确移动条件,该设计有效避免了同机器移动关键工序中工序块选择和关键工序移动的盲目性;针对跨机器移动关键工序,将现有的单目标邻域搜索推广到多目标,并首次提出了精确的候选机器选择方法,该方法可有效降低跨机器移动关键工序中候选机器选择的盲目性,避免大量无效移动。其次,将精确邻域搜索与改进的进化算法混合,实现了局部搜索与全局搜索的优势互补。最后,通过国际基准案例测试以及与其他成熟算法的对比分析验证了所设计算法的有效性和高效性,结果表明,该算法性能优越。

## 参考文献:

- [1] WANG X, GAO L, ZHANG C, *et al.* A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem [J]. *The International Journal of Advanced Manufacturing Technology*, 2010, 51(5/8): 757.
- [2] GAO J, SUN L, GEN M. A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems [J]. *Computers & Operations Research*, 2008, 35(9): 2892.
- [3] LI J Q, PAN Q K, GAO K Z. Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems [J]. *International Journal of Advanced Manufacturing Technology*, 2011, 55(9/12): 1159.
- [4] GONG G, DENG Q, CHIONG R, *et al.* An effective

- memetic algorithm for multi-objective job-shop scheduling[J]. Knowledge Based Systems, 2019, 182: 104840.
- [5] GAO K Z, SUGANTHAN P N, PAN Q K, *et al.* Pareto-based grouping discrete harmony search algorithm for multi-objective flexible job shop scheduling[J]. Information Sciences, 2014, 289: 76.
- [6] LI J Q, PAN Q K, SUGANTHAN P N, *et al.* A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem[J]. International Journal of Advanced Manufacturing Technology, 2011, 52(5/8): 683.
- [7] VITAL-SOTO A, AZAB A, BAKI M F. Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility[J]. Journal of Manufacturing Systems, 2020, 54: 74.
- [8] ZHANG G, SHAO X, LI P, *et al.* An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem[J]. Computers & Industrial Engineering, 2009, 56(4): 1309.
- [9] CHIANG T C, LIN H J. A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling[J]. International Journal of Production Economics, 2013, 141(1): 87.
- [10] WANG L, ZHOU G, XU Y, *et al.* An enhanced Pareto-based artificial bee colony algorithm for the multi-objective flexible job-shop scheduling[J]. International Journal of Advanced Manufacturing Technology, 2012, 60(9/12): 1111.
- [11] 赵诗奎. 柔性作业车间调度的改进邻域结构混合算法[J]. 计算机集成制造系统, 2018, 24(12): 144.  
ZHAO Shikui. Hybrid algorithm based on improved neighborhood structure for flexible job shop scheduling[J]. Computer Integrated Manufacturing Systems, 2018, 24(12): 144.
- [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(14): 181.  
ZHAO Shikui. Bilevel neighborhood search hybrid algorithm for the flexible job shop scheduling problem[J]. Journal of Mechanical Engineering, 2015(14): 181.
- [13] ZHANG C Y, LI P G, RAO Y Q, *et al.* A new hybrid GA/SA algorithm for the job shop scheduling problem[C]// European Conference on Evolutionary Computation in Combinatorial Optimization. Berlin: Springer, 2005:246-259.
- [14] KACEM I, HAMMADI S, BORNE P. Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic[J]. Mathematics & Computers in Simulation, 2002, 60(3/5): 245.
- [15] BRANDIMARTE P. Routing and scheduling in a flexible job shop by tabu search[J]. Annals of Operations Research, 1993, 41(3): 157.
- [16] 孟冠军, 杨大春, 陶细佩. 基于混合人工蜂群算法的多目标柔性作业车间调度问题研究[J]. 计算机应用研究, 2019, 36(4): 972.  
MENG Guanjun, YANG Dachun, TAO Xipei. Study on multi-objective flexible job-shop scheduling problem based on hybrid artificial bee colony algorithm[J]. Application Research of Computers, 2019, 36(4): 972.
- [17] 吴贝贝, 张宏立, 王聪, 等. 基于正态云模型的状态转移算法求解多目标柔性作业车间调度问题[J/OL]. [2020-06-05]. <https://doi.org/10.13195/j.kzyjc.2019.1233>.  
WU Beibei, ZHANG Hongli, WANG Chong, *et al.* State transition algorithm based on normal cloud model for solving multi-objective flexible job shop scheduling problem. [2020-06-05]. <https://doi.org/10.13195/j.kzyjc.2019.1233>.
- [18] XING L N, CHEN Y W, YANG K W. An efficient search method for multi-objective flexible job shop scheduling problems[J]. Journal of Intelligent Manufacturing, 2009, 20(3): 283.
- [19] BAGHERI A, ZANDIEH M, MAHDAVI I, *et al.* An artificial immune algorithm for the flexible job-shop scheduling problem[J]. Future Generation Computer Systems, 2010, 26(4): 533.
- [20] 裴小兵, 李依臻. 基于三方博弈的改进遗传算法求解多目标柔性作业车间调度[J]. 工业工程与管理, 2020, 25(4): 59.  
PEI Xiaobing, LI Yizhen. Improved genetic algorithm based on three-party game for multiobjective flexible job shop scheduling[J]. Industrial Engineering and Management, 2020, 25(4): 59.
- [21] DENG Q W, GONG G L, GONG X R, *et al.* A bee evolutionary guiding nondominated sorting genetic algorithm II for multiobjective flexible job-shop scheduling[J]. Computational Intelligence and Neuroscience, 2017, 2017:1.