

基于汉明距离与邻域搜索的多目标流水车间调度优化^①



杨兴坤, 杨 昊, 文 武, 徐 虹

(成都信息工程大学 计算机学院, 成都 610225)

通信作者: 杨 昊, E-mail: haoyang@cuit.edu.cn

摘 要: 针对多目标流水车间调度问题, 提出了一种邻域搜索 (NS) 算法, 旨在以加工时间和总延迟时间为优化目标构建调度模型. 该方法通过交换调度序列中的工件顺序生成新的解, 从而在解空间中高效探索. 算法从种群中的初始解开始, 不断通过工作顺序的交换产生候选解, 并利用适应值对这些解的优劣进行评估. 随着迭代优化的进行, 算法逐渐逼近全局最优解. 此外, 基于占优关系函数的方法用于评估解的相对优势, 通过参考点计算解在不同目标维度上的距离, 并以此衡量解的质量. 为了实现解空间的系统性探索, 本文采用基于汉明距离的解生成策略. 该方法通过构建解之间的最短汉明路径, 在保留解的优良特征的同时, 有效地探索未知解空间, 提升算法寻找全局最优解的能力. 通过 20 个不同规模算例的实验, 将 NS 算法与传统遗传算法 (GA) 及非支配排序遗传算法 (NSGA-II) 进行比较, 结果显示该方法在多目标优化上表现出色, 验证了其有效性和优势.

关键词: 多目标; 邻域搜索; 车间调度; 汉明距离

引用格式: 杨兴坤, 杨昊, 文武, 徐虹. 基于汉明距离与邻域搜索的多目标流水车间调度优化. 计算机系统应用, 2025, 34(7): 244–252. <http://www.c-s-a.org.cn/1003-3254/9936.html>

Multi-objective Flow Shop Scheduling Optimization Based on Hamming Distance and Neighborhood Search

YANG Xing-Kun, YANG Hao, WEN Wu, XU Hong

(School of Computer Science, Chengdu University of Information Technology, Chengdu 610225, China)

Abstract: A neighborhood search (NS) algorithm is proposed for the multi-objective flow shop scheduling problem, aiming to construct a scheduling model that optimizes processing time and total tardiness. This method efficiently explores the solution space by generating new solutions through the swapping of job sequences. Starting from an initial solution in the population, candidate solutions are iteratively generated by altering job sequences and are evaluated using fitness values. As the optimization proceeds, the algorithm gradually converges toward the global optimum. A dominance relationship function is employed to assess the relative superiority of solutions by calculating their distances to reference points across multiple objective dimensions, thus enabling effective quality evaluation. To ensure systematic exploration of the solution space, this study employs a solution generation strategy based on Hamming distance. By constructing the shortest Hamming paths between solutions, this approach preserves desirable characteristics. It also effectively probes unexplored regions of the solution space, thus enhancing the algorithm's capability to locate the global optimum. Experimental results from 20 test cases of varying scales show that the proposed NS algorithm outperforms the traditional genetic algorithm (GA) and the non-dominated sorting genetic algorithm (NSGA-II) in multi-objective optimization.

① 基金项目: 四川省科技计划 (2023JDZH0034); 四川省科技成果转化示范项目 (2024ZHCG0026)

收稿时间: 2024-11-14; 修改时间: 2025-02-12; 采用时间: 2025-03-06; csa 在线出版时间: 2025-05-16

CNKI 网络首发时间: 2025-05-19

demonstrating its effectiveness and advantages.

Key words: multi-object; neighborhood search (NS); flow shop scheduling; Hamming distance

复杂制造系统的优化调度理论与方法是当前国际学术界和工业界高度关注的跨学科前沿研究领域. 该研究方向具有广泛的工程应用背景, 涵盖工业工程与生产管理、机械工程、人工智能、控制科学与工程、计算机技术及管理工程等多个学科^[1]. 其中, 多目标流水车间调度问题 (multi-objective flow shop scheduling problem, MFSP) 的研究尤为重要. 该问题的核心在于通过优化调度策略, 综合考虑多种性能指标, 例如最大完工时间 (makespan)、总延迟时间 (total tardiness)、总能耗和机器利用率等, 从而实现制造系统的整体优化. 然而, 在实际应用中, 这些指标往往存在冲突. 例如, 最小化最大完工时间可能导致某些工件的延迟时间增加. 因此, 在这些目标之间找到适当的平衡点成为多目标优化的关键. 合理制定多目标车间调度方案不仅能够有效降低成本、优化资源利用率、降低能源消耗, 还能提升工艺稳定性, 增强企业的智能制造能力. 正因如此, 多目标流水车间调度问题的研究既具有重要的理论意义, 也具备很高的实际应用价值.

为了解决多目标调度问题, 研究者们提出了多种元启发式算法, 其中包括遗传算法 (genetic algorithm, GA)^[2]、模拟退火算法 (simulated annealing, SA)^[3]、粒子群优化算法 (particle swarm optimization, PSO)^[4]和蚁群优化算法 (ant colony optimization, ACO)^[5]等. 这些算法各有优劣, 遗传算法具有全局搜索能力强、适应性高的优点, 但容易陷入早熟收敛^[6]; 模拟退火算法在局部范围内的探索效率较高, 但整体收敛速度较为缓慢^[7]; 粒子群优化算法在处理连续优化问题时具有显著优势, 但在应对离散问题时需采用专门的调整策略^[8]; 蚁群优化算法在组合问题的求解中效果良好, 但其计算过程相对复杂^[9]. 为应对这些挑战, 研究者们不断探索新的算法改进. Jeen Robert 等^[10]设计了一种基于实数编码遗传算法 (RCGA) 和有效遗传算法 (EGA) 来应对流水车间调度问题的求解需求, EGA 通过修改传统遗传算法, 使用 NEH 启发式方法生成初始种群, 并在染色体水平上进行子染色体交叉和变异操作. Gao 等^[11]提出了一种改进的 GA-LS 算法, 基于遗传算法, 专门用于解决分布式置换流水车间调度问题. 该方法利用

部分作业序列集合来进行问题求解, 主要目标是缩短最大完工时间, 并通过邻域搜索策略来探索更优的解.

在这些改进的基础上, 研究者们逐步探索并总结出解决多目标流水车间调度问题的两大类主要方法^[12]: 一种是采用线性加权的方法, 将多目标中的每一个目标函数集成为一个复合函数, 还有种方式是基于帕累托 (Pareto) 支配, 将其他目标作为约束条件处理. 刘宇等^[13]设计了一种结合 Q-learning 的遗传强化学习算法, 用于求解该问题的 Pareto 最优解, 通过 Q 学习算法生成初始种群, 从而提高初始种群解的质量. Deb 等^[14]提出了 NSGA-II 算法, NSGA-II 是一种多目标优化算法, 用于处理复杂的多目标优化问题. 在处理 MFSP 问题, 提出了拥挤距离的方式, 使解空间均匀地分布在 Pareto 前沿上. Yang 等^[15]提出了一种名为 NPTSGA 的算法, 将遗传算法 (GA) 与禁忌搜索 (TS) 相结合. 该算法在保持 Pareto 最优解无支配性的基础上, 能够在折中曲线上实现对接近 Pareto 最优解的平衡处理. Zeitrög 等^[16]提出了一种结合参考点法和超启发式遗传编程的交互式多目标优化方法, 用于自动生成车间调度规则, 该调度规则能够在多个目标 (如完工时间、延迟时间和等待时间) 之间实现平衡, 同时考虑到决策者的偏好.

基于以上研究的启发, 本文提出了一种基于邻域搜索的算法, 专门针对多目标流水车间调度优化设计. 该算法通过比较两个目标上的双重指标来确定适应值, 并利用汉明距离避免种群陷入局部最优, 从而保证了解的多样性. 邻域搜索通过在当前解的邻域中生成新解, 并选择适应值更好的解来逐步改进调度方案. 同时, 汉明距离的应用确保了在高质量解集中进行充分搜索, 增强了算法的全局寻优能力和解的优越性.

1 问题描述及数学建模

1.1 问题描述

多目标流水车间调度问题 (MFSP) 主要研究在 m 台机器上对 n 个工件进行流水加工的过程. 在这个过程中, 每台机器负责不同的工序, 并且每个工件必须按照相同的顺序经过所有的机器进行加工. 本文的研究聚焦于两个关键目标的优化: 首先, 最小化所有工件

加工完成的总加工时间;其次,最小化所有工件的总延迟时间.在此基础上,所有工件的处理顺序需满足以下限制规则^[13].

- (1) 任何一台机器在同一时间只能加工一个零件.
- (2) 每个工件必须在所有机器上进行加工.
- (3) 每个工件的工序保持一致.
- (4) 机器在加工某个工件时不能终止、退出.
- (5) 不考虑工件的运输及机器设置的时间.

1.2 数学规划模型

符号定义:

J : 工件集合, $J = \{J_1, J_2, \dots, J_n\}$, 其中 n 代表工件数量.

M : 机器集合, $M = \{M_1, M_2, \dots, M_m\}$, m 代表机器数量.

D_i : 工件 J_i 能接受的最低加工时间.

$T_{i,j}$: 工件 J_i 在机器 M_j 上的完成时间.

C_i : 工件 J_i 的完成时间.

$process_i$: 工件 J_i 的总加工时间.

$delay_i$: 工件 J_i 的延迟时间.

$delay$: 所有工件的总延迟时间.

基于以上定义,多目标车间调度问题(MFSP)的数学模型如下.

$$\begin{cases} T_{1,1} = t_{1,1} \\ T_{i,1} = T_{i-1,1} + t_{i,1}, & i \in [2, n] \\ T_{1,j} = T_{1,j-1} + t_{1,j}, & j \in [2, n] \\ T_{i,j} = \max(T_{i-1,j}, T_{i,j-1}) + t_{i,j}, & i \in [2, n], j \in [2, n] \end{cases} \quad (1)$$

$$C_i = T_{i,n} \quad (2)$$

$$C_{\max} = \max_{1 \leq i \leq n} C_i \quad (3)$$

$$process_i = \sum_{j=1}^m t_{i,j} \quad (4)$$

$$delay_i = \begin{cases} D_i - C_i, & D_i > C_i \\ 0, & D_i \leq C_i \end{cases} \quad (5)$$

$$delay = \sum_{i=1}^n delay_i \quad (6)$$

$$F_1 = \min\{C_{\max}\} \quad (7)$$

$$F_2 = \min\{delay\} \quad (8)$$

其中,式(1)用于限制每个工件在各台机器上的加工时间;式(2)用于确定每个工件的最长加工时间;式(3)用于计算所有工件的最大完工时间;式(4)用于统计每个工件在所有机器上的加工总时长;式(5)用于评估每

个工件的延误时间;式(6)用于计算整体的延误时间总量;式(7)用于定义以最小化最大完工时间为目标;式(8)用于定义以最小化总延误时间为目标.

2 域搜索算法设计

2.1 初始化种群

在本算法中采用十进制编码作为解序列,根据工件的工序(机器号)生成一个随机的解.初始化种群大小为10,即随机生成10个不重复的解为原始种群.每个个体包含5个属性:(1) *array*: 解序列.(2) *process*: 该解序列的延迟时间.(3) *delay*: 该解序列的延迟时间.(4) *flag*: 标志位,为0时表示该解为Pareto前沿解,为1表示该解非Pareto前沿解.(5) *fit*: 适应值,适应值越大表示该解越优越.

2.2 适应值计算

在优化问题中,计算解的适应度是评估其优劣的重要步骤^[17].通过比较两个解之间的占优关系,确定解在种群中的相对优越性.

两个解之间的占优关系是通过占优关系函数:二元指标函数 $I(x, y)$ 的计算来确定的.该函数基于参考点计算两个解的区域面积.参考点是一个固定的点,该参考点在初始化种群时可确定具体的值为延迟时间和加工时间的最大值,该参考点用于度量解与目标的接近程度.

在计算两个解的占优关系时,需要考虑解在加工时间和延迟时间两个目标维度上的表现.通过比较这些维度,可以确定一个解是否在某个维度上优于另一个解.占优关系函数将这些比较结果转换为一个数值,表示两个解之间的相对优越性.

如果一个解在所有维度上都优于另一个解,称这个解为占优解.在这种情况下,占优关系函数返回两者的区域面积差值表示优劣程度.如果两个解在不同维度上有优劣,则需要计算它们的交集区域面积.这种情况下,占优关系函数会计算交集区域面积,并用此面积修正原始区域面积差值,从而更准确地反映两个解的相对优越性.

在图1中 $x(2, 3)$ 与 $y(3, 1)$ 无法确定明显的占优关系,即可通过占优关系函数来确定占优关系.计算方法是:

$$I(x, y) = Area(x, point) - Area(x, point) \cap Area(y, point) \quad (9)$$

其中, $Area(x, y)$ 为计算面积函数, 计算点 x 到点 y 所围成的矩形的面积. $point$ 为参考点. 式 (9) 描述为: $I(x, y)$ 为点 x 到参考点的面积减去 y 到参考点的面积和 x 到参考点的面积的交集.

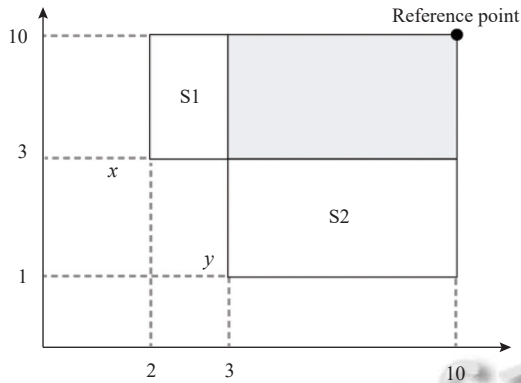


图1 非占优解的二元指标面积

在图2中 $x(2, 2)$ 与 $y(3, 3)$ 有明显的占优关系, 也可以通过占优关系函数来确定占优关系, 对于占优关系函数可以简化为 x 到参考点的面积减去 y 到参考点的面积.

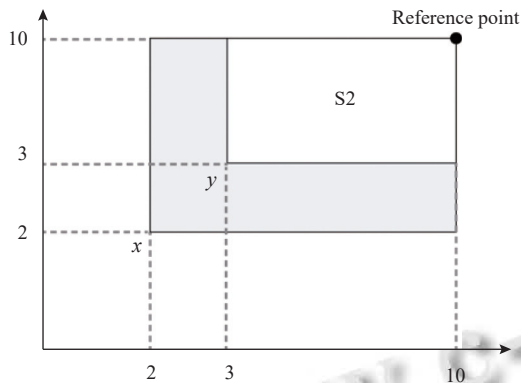


图2 占优解的二元指标面积

基于占优关系函数可以计算两个解之间的占优关系, 而每个解的适应度表示其在整个种群中的相对优劣程度. 通过遍历种群中的所有个体, 计算每个个体相对于其他个体的占优关系, 并累积这些结果, 我们可以得到每个解的适应度, 公式如下:

$$x_i \text{ fit} = \begin{cases} \sum_{j=1, j \neq i}^n I(x_i.array, x_j.array), & i \neq 1 \\ \sum_{j=2, j \neq i}^n I(x_i.array, x_j.array), & i = 1 \end{cases} \quad (10)$$

其中, x_i 表示待求解适应度的解; x_j 表示其他解; n 为种

群规模; $I(x, y)$ 为二元指标函数.

占优关系函数伪代码如算法1.

算法1. $I(x, y)$ 计算算法

输入: 两个解 x 和 y , 参考点 r .

输出: 占优关系值.

```

1. // 计算面积
2.  $area\_x \leftarrow (r.x - x.process) \times (r.y - x.delay)$ 
3.  $area\_y \leftarrow (r.x - y.process) \times (r.y - y.delay)$ 
4. // 判断支配关系
5. if  $(x.process \leq y.process \text{ AND } x.delay \leq y.delay)$  OR  $(y.process \leq x.process \text{ AND } y.delay \leq x.delay)$  then
6.   return  $area\_x - area\_y$ 
7. else
8.    $overlap\_width \leftarrow \min(r.x - x.process, r.x - y.process)$ 
9.    $overlap\_height \leftarrow \min(r.y - x.delay, r.y - y.delay)$ 
10.   $overlap\_area \leftarrow overlap\_width \times overlap\_height$ 
11.  return  $area\_x - overlap\_area$ 
12. end if
    
```

2.3 邻域搜索

在解决复杂优化问题时, 探索解空间的多样性是找到全局最优解的关键. 为此, 本文引入了邻域交换算法^[18], 通过交换解空间中的元素来生成新解, 从而增强解的多样性. 邻域交换包括两种变换策略: 交换法和插入法.

交换法: 产生2个范围为 $(0, \text{工件数}-1)$ 的不同的随机数, 交换对应位置的工件产生新解.

插入法: 产生2个范围为 $(0, \text{工件数}-1)$ 的不同的随机数, 将第1随机数位置的工件, 插入到第2个随机数位置工件前.

无论采用哪种方式产生新解, 都需要评估新解是否能加入原种群. 评估方法是判断该解是否在两个目标上都小于种群中的某一个解, 如果有则可以直接加入原种群. 如果没有, 则计算该解的适应值, 通过适应值来比较是否该解能够加入. 评估流程如图3所示.

在确定新解能够加入原种群后, 为了维持种群大小不变, 需要删除原种群中最差的一个解. 删除方法是优先删除标志位为1的解. 该标志位为1表示该解在种群中至少有一个解在两个目标上优于自己, 标志位为0表示种群中其他解不会同时在两个目标上优于自己, 即是 Pareto 前沿. 当种群中的解标志位都为0时, 删除一个适应值最低的解.

当一个解被删除时, 需要重新计算其他解的适应度. 对于每个剩余的解, 需要减去被删除解对其适应度

的影响. 确保种群中每个解的适应度始终准确反映其相对优劣程度.

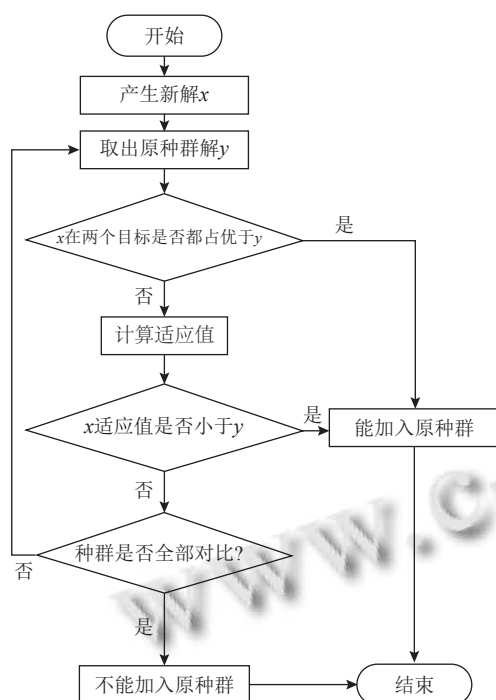


图3 新个体能否加入原种群流程图

式 (11) 可以描述删除时的种群适应值更新.

$$\begin{cases} x_i.fit \leftarrow x_j.fit - I(x_i.array, x.array) \\ \forall x_i \in \{x_1, x_2, \dots, x_n\} \setminus \{x\} \end{cases} \quad (11)$$

其中, x 为待删除的解, $I(x, y)$ 为二元指标函数.

向种群中添加新解时, 也需要计算新解的适应度. 包括计算新解相对于现有解的占优关系, 并更新现有解的适应度. 同时, 还需要计算现有解相对于新解的占优关系, 并累积这些结果得到新解的适应度.

式 (12) 可以描述添加新解时的种群适应值更新.

$$\begin{cases} x_i.fit \leftarrow x_j.fit - I(x_i.array, x.array) \\ \forall x_i \in \{x_1, x_2, \dots, x_n\} \setminus \{x\} \end{cases} \quad (12)$$

其中, x 为待添加的解, $I(x, y)$ 为二元指标函数.

2.4 汉明优化

2.4.1 基于汉明距离的解空间探索机制

为了防止陷入局部最优解, 本文引入汉明路径进行优化. 汉明距离 (Hamming distance) 是一种用于评估两个等长字符串之间差异程度的度量方法^[19], 该方法可以判断两个字符串的相似程度. 具体来说, 汉明距离表示在两个字符串中, 相同位置字符不匹配的次數. 它广泛应用于信息论、编码理论和计算机科学中, 尤其

是在错误检测和纠正领域. 给定两个等长字符串 s_1 和 s_2 , 汉明距离定义为:

$$d(s_1, s_2) = \sum_{i=1}^n \delta(s_1[i], s_2[i]) \quad (13)$$

其中, $\delta(s_1[i], s_2[i])$ 是一个指示函数, 当 $s_1[i] \neq s_2[i]$ 时值为 1, 否则为 0.

若存在 2 个等长的字符串, 其相应位置上不同字符的个数即汉明距离^[20]. 如图 4 所示, 两个长为 20 的解, 其中有 18 个位置不一致, 则汉明距离为 18.

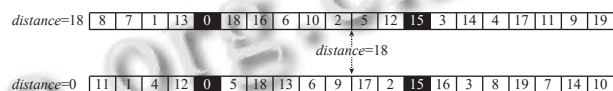


图4 汉明距离

目标解的第 1 个位置为 11, 则交换当前解的 8 与 11, 使汉明距离减小为 17, 如图 5 所示.

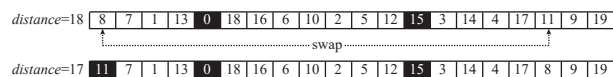


图5 交换一个工件顺序使得汉明距离缩小

汉明距离不断缩小为 0 的过程中所产生的解, 被称为汉明路径. 如图 6 所示, 两个解的汉明距离为 18, 则至多进行 18 次位置调整即可使得汉明距离为 0.

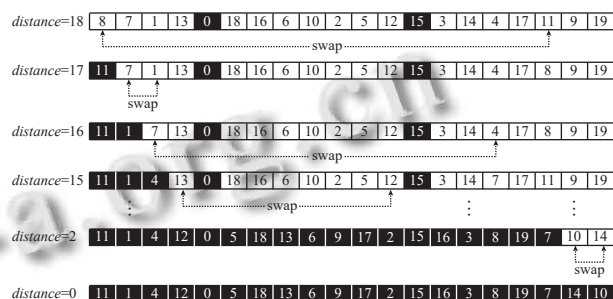


图6 汉明距离变为 0 的过程

2.4.2 汉明优化流程

基于汉明距离的优化策略 (Hamming-based optimization strategy, HOS) 的具体执行流程如下.

当种群在连续迭代中 (总迭代次数的 1%) 未发生更新时, 将触发汉明优化机制, 该机制通过以下步骤实现空间探索.

- 路径构建阶段: 从当前种群中随机选取两个个体, 计算它们之间的汉明路径. 该路径表示两个解之间的最短变换序列.

- 候选种群生成: 将汉明路径上的所有解作为候选种群, 这些解将作为深入探索的基础.

● 邻域搜索增强: 对候选种群中的每个解进行深度邻域搜索, 以发现局部空间中潜在的高质量解。

● 解评估与整合: 对邻域搜索获得的解进行评估, 评估标准包括:

(1) 支配关系判定: 新解的加工时间和延迟时间是优于原种群中的某个解。

(2) 适应度比较: 新解的适应度值是否高于原种群中的某个解。

● 种群更新机制: 满足上述任一标准的解将被视为优化成功, 可以被整合到原始种群中。

该优化策略通过汉明路径的系统性探索和邻域搜索的局部精化相结合, 有效提升了算法跳出局部最优并发现高质量解的能力。这种方法不仅保持了种群的多样性, 同时确保了解的质量不断提升。汉明优化的流程图如图7所示。

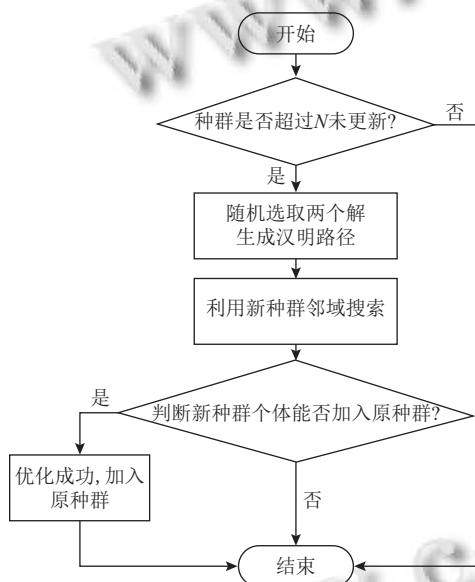


图7 汉明优化流程图

通过汉明路径计算和邻域搜索的结合, 有效增加了种群的多样性, 降低了陷入局部最优解的概率, 同时确保新种群的质量和有效性。这种方法在实际应用中表现出色, 有效提升了优化过程的效率和效果。

2.4.3 汉明优化对比实验

为了进一步验证汉明优化策略的有效性, 本研究选取了规模为 20×5 (20 个工件, 5 台机器) 的算例进行对比实验。

(1) 实验设置

- 算例规模: 20 个工件, 5 台机器。
- 迭代次数: 250 000。

● 种群大小: 10。

● 汉明优化次数: 10。

● 实验组: 加入汉明优化。

● 对照组: 未加入汉明优化。

(2) 性能指标

为了实时监控种群的多样性水平, 本研究采用汉明距离作为评估指标。在每次迭代过程中, 通过计算种群内个体间的平均汉明距离来度量种群的相似程度:

$$D = \frac{2}{N(N-1)} \sum_{i=1}^{N-1} \sum_{j=i+1}^N H(X_i, X_j) \quad (14)$$

其中, D 表示种群的平均汉明距离; N 为种群大小; i, j 为种群中的个体索引 ($i \neq j$); 较大的 D 值表示种群具有较高的多样性, 较小的 D 值表示种群趋于相似, D 值的变化趋势反映了种群多样性的动态变化过程。

(3) 实验结果

如图8所示, 未引入汉明优化, 种群相似度在迭代 5 000 次后算法趋于稳定, 且种群平均相似度约为 15.5, 算法进入局部最优。如图9所示, 引入汉明优化后, 每次触发汉明优化, 种群相似度出现显著波动, 解空间持续扩展, 避免了早熟收敛, 种群保持较高的多样性水平。

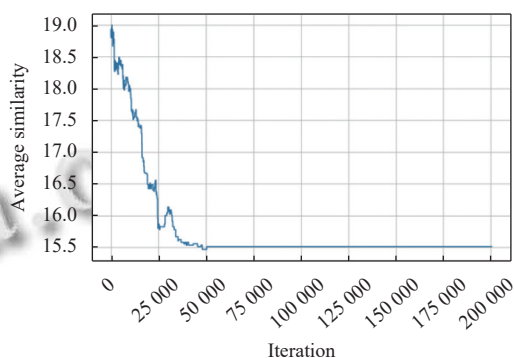


图8 未引入汉明优化

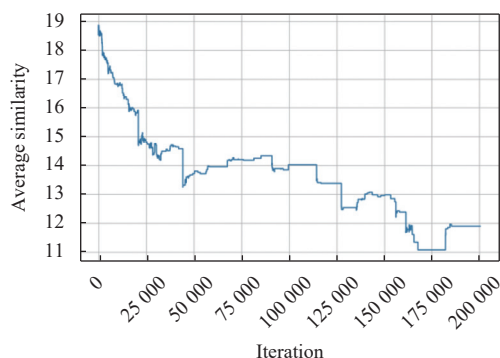


图9 引入汉明优化

实验数据清晰地表明,在相同规模的算例下,汉明优化能够有效防止算法陷入局部最优,通过定期的解空间扩展保持种群活力,从而提升算法的整体搜索性能。

3 实验结果与分析

3.1 实验设置

本实验旨在评估3种多目标优化算法(NS、NSGA-II和GA)在流水车间调度问题中的性能。实验使用了20个标准算例,每个算例的目标是最小化两个目标函数(如完工时间和总延迟)。每个算法在每个算例上运行多次,以确保结果的稳定性。实验的输出包括每个算法的帕累托前沿解集,用于后续的性能评估。

算例规模如表1所示。

表1 算例规模

规模 (工件数量×工序)	算例个数	规模 (工件数量×工序)	算例个数
20×5	2	80×30	2
20×10	2	100×30	2
20×15	2	150×30	2
20×20	2	200×30	2
50×20	2	200×40	2

算法参数如表2所示。

表2 算法参数

算法	迭代次数	种群大小	汉明优化次数
NS	30000	10	10
NSGA-II	30000	20	—
GA	30000	20	—

3.2 性能指标

为了全面评估算法的性能,使用以下两个指标。

(1) 超体积 (hypervolume): 超体积衡量帕累托前沿与参考点之间的体积,值越大表示解集的多样性和收敛性越好。

$$HV = \sum_{i=1}^n (x_{i+1} - x_i) \times (y_{i+1} - y_i) \quad (15)$$

(2) C 指标 (C-metric): C 指标衡量一个算法的帕累托前沿对另一个算法的支配比例,值越大表示前者对后者的支配性越强。

$$C(A, B) = \frac{| \{b \in B \mid \exists a \in A, a \leq b\} |}{|B|} \quad (16)$$

3.3 实验结果

我们对20个算例的实验结果进行了统计分析。每

个算例的超体积和C指标分别计算,并汇总为平均值。具体结果如表3,可以看出NS在超体积和C指标上均表现最好。

表3 超体积和C指标对比实验结果

算法	超体积	C指标		
		NS	NSGA-II	GA
NS	8711664.60	—	0.95	0.37
NSGA-II	811927.10	0	—	0
GA	4666880.55	0.01	0.52	—

由表4可见,通过对20组不同规模算例的分析,NS算法展现出了优异的性能和稳定性。在20个测试算例中,NS算法在17个算例中取得了完全优势,其余算例中也保持着极具竞争力的表现。这充分体现了NS算法在解决双目标调度优化问题时的卓越性能。

表4 算法对比实验结果

规模	NS	NSGA-II	GA
20_05_01	1297/2711	1335/5014	1324/2659
20_05_02	1396/2486	1368/5354	1416/3297
20_10_01	1656/2486	1656/3542	1747/2423
20_10_02	1812/4224	1768/6883	1817/4345
20_15_01	1950/5568	1985/6457	2017/5263
20_15_02	1951/4540	2006/6347	2021/4775
20_20_01	2434/6069	2380/8535	2463/6169
20_20_02	2269/6262	2155/7852	2296/6366
50_20_01	2854/36497	2831/42392	2883/36545
50_20_02	2943/35676	2932/45164	3000/35672
80_30_01	4668/98921	4672/123764	4726/105763
80_30_02	4633/95468	4645/118880	4706/93591
100_30_01	5483/161684	5512/192472	5535/161893
100_30_02	5556/170698	5609/194983	5631/168464
150_30_01	7458/383636	7565/461518	7555/378018
150_30_02	7426/385016	7459/449849	7461/379453
200_30_01	9410/703299	9486/796300	9496/692846
200_30_02	9525/706700	9620/824327	9620/691751
200_40_01	10044/703346	10217/832086	10163/688029
200_40_02	9951/698211	10105/808850	10048/694502

注: / 前后分别是总加工时间 (makespan) 和总延迟时间 (total delay)。

从图10实验结果可以直观地看出NS算法的加工时间 (process) 和等待延时 (delay) 更加靠近 (0, 0) 点,在两个目标整体都比较优于其他算法。且NS算法会平衡两个目标的关系,不会极端的在一个目标上尽可能小,因此NS算法比较其他算法更加稳定。

4 结论与展望

本文提出了一种基于邻域搜索算法的新方法,旨在优化多目标流水车间调度任务。该方法通过交换调度序列中的部分工件顺序来生成新的解,从而在解空

间中进行高效探索. 邻域搜索算法从一个初始解开始, 通过不断地交换工作顺序来产生新的候选解, 并通过

适应值评估这些候选解的优劣. 通过迭代优化, 算法逐渐趋向于全局最优解.

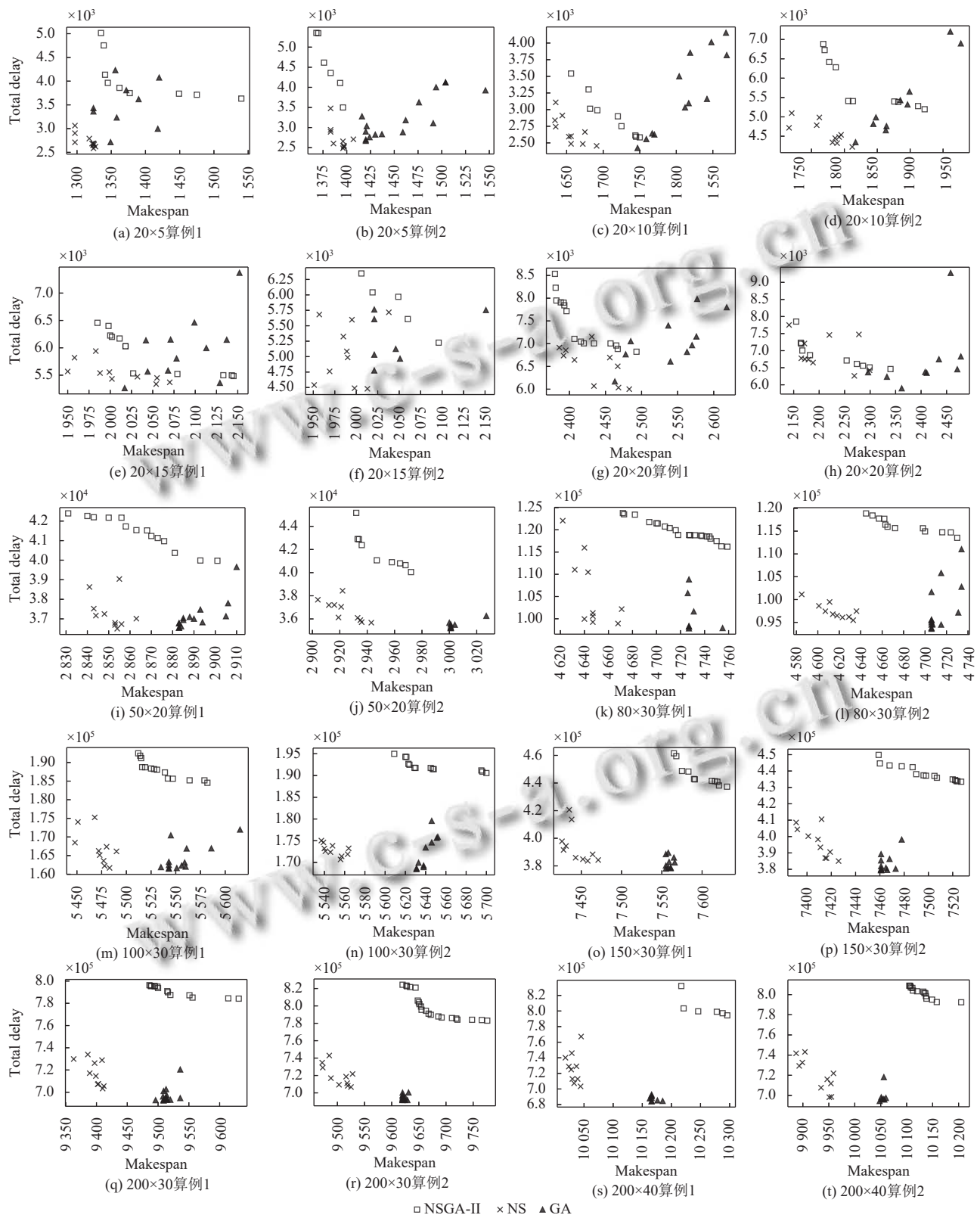


图 10 各算法实验对比

为了进一步提高算法的性能,本文还提出了一种基于占优关系函数的方法,通过比较两个解之间的占优关系来确定其相对优越性.占优关系函数基于参考点计算解在不同目标维度上的距离,并用这些距离衡量解的优劣.算法采用基于汉明距离的解生成策略来加强解空间的探索能力.该方法通过在解之间构建最短汉明路径,在保持原解优良特征的同时,能够系统地搜索路径上的潜在解,有效提升了算法发现全局最优解的能力.

本文通过进行包括20个算例的实验,将所提出的方法(NS)与传统遗传算法(GA)和非支配排序遗传算法II(NSGA-II)进行了比较.实验结果显示,NS算法在加工时间和总延迟时间两个目标维度上均表现出色,显著优于GA和NSGA-II算法.这不仅验证了NS算法的有效性和优越性,也为未来进一步改进多目标优化算法提供了有价值的参考.

参考文献

- 1 潘全科,高亮,李新宇.流水车间调度及其优化算法.武汉:华中科技大学出版社,2013.
- 2 Goldberg DE. Genetic Algorithms in Search, Optimization and Machine Learning. Boston: Addison-Wesley, 1989.
- 3 Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science*, 1983, 220(4598): 671–680. [doi: [10.1126/science.220.4598.671](https://doi.org/10.1126/science.220.4598.671)]
- 4 Kennedy J, Eberhart R. Particle swarm optimization. *Proceedings of the 1995 IEEE International Conference on Neural Networks*. Perth: IEEE, 1995. 1942–1948. [doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968)]
- 5 Dorigo M, Gambardella LM. Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 53–66. [doi: [10.1109/4235.585892](https://doi.org/10.1109/4235.585892)]
- 6 Mitchell M. An Introduction to Genetic Algorithms. Cambridge: MIT Press, 1998.
- 7 Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 1985, 45(1): 41–51. [doi: [10.1007/BF00940812](https://doi.org/10.1007/BF00940812)]
- 8 Shi Y, Eberhart R. A modified particle swarm optimizer. *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation*. Anchorage: IEEE, 1998. 69–73. [doi: [10.1109/ICEC.1998.699146](https://doi.org/10.1109/ICEC.1998.699146)]
- 9 Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 2006, 1(4): 28–39. [doi: [10.1109/MCI.2006.329691](https://doi.org/10.1109/MCI.2006.329691)]
- 10 Jeen Robert RB, Rajkumar R. An effective genetic algorithm for flow shop scheduling problems to minimize makespan. *Mechanics*, 2017, 23(4): 594–603. [doi: [10.5755/j01.mech.23.4.15053](https://doi.org/10.5755/j01.mech.23.4.15053)]
- 11 Gao J, Chen R. A hybrid genetic algorithm for the distributed permutation flowshop scheduling problem. *International Journal of Computational Intelligence Systems*, 2011, 4(4): 497–508. [doi: [10.1080/18756891.2011.9727808](https://doi.org/10.1080/18756891.2011.9727808)]
- 12 Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 2006, 91(9): 992–1007. [doi: [10.1016/j.ress.2005.11.018](https://doi.org/10.1016/j.ress.2005.11.018)]
- 13 刘宇,陈永灿,周艳平.求解多目标流水车间调度 Pareto 最优解的遗传强化算法. *计算机系统应用*, 2024, 33(2): 239–245. [doi: [10.15888/j.cnki.csa.009428](https://doi.org/10.15888/j.cnki.csa.009428)]
- 14 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182–197. [doi: [10.1109/4235.996017](https://doi.org/10.1109/4235.996017)]
- 15 Yang Y, Wu JF, Zhu XB, et al. A hybrid evolutionary algorithm for finding Pareto optimal set in multi-objective optimization. *Proceedings of the 7th International Conference on Natural Computation*. Shanghai: IEEE, 2011. 1233–1236. [doi: [10.1109/ICNC.2011.6022268](https://doi.org/10.1109/ICNC.2011.6022268)]
- 16 Zeiträg Y, Figueira JR. Automatically evolving preference-based dispatching rules for multi-objective job shop scheduling. *Journal of Scheduling*, 2023, 26(3): 289–314. [doi: [10.1007/s10951-023-00783-9](https://doi.org/10.1007/s10951-023-00783-9)]
- 17 王凌.智能优化算法及其应用.北京:清华大学出版社,2001.
- 18 Lin S, Kernighan BW. An effective heuristic algorithm for the traveling-salesman problem. *Operations Research*, 1973, 21(2): 498–516. [doi: [10.1287/opre.21.2.498](https://doi.org/10.1287/opre.21.2.498)]
- 19 Jaiswal MS, Cho M, Kang B. Hamming distance based robust output encoding for improved generalization: US, 11410043. <https://patents.google.com/patent/US11410043B2/en>. (2022-08-09).
- 20 丛培强,李梁,陈亚茹.基于汉明距离与免疫思想的粒子群算法. *重庆理工大学学报(自然科学)*, 2019, 33(4): 122–127. [doi: [10.3969/j.issn.1674-8425\(z\).2019.04.018](https://doi.org/10.3969/j.issn.1674-8425(z).2019.04.018)]

(校对责编:张重毅)