DOI: 10.3901/JME.2018.03.148

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*

牟健慧¹ 潘全科² 牟建彩³ 徐汝峰¹ 于珊珊¹

- (1. 山东理工大学机械工程学院 淄博 255000;
2. 华中科技大学数字制造装备与技术国家重点实验室 武汉 430074;
3. 华立科技职业学院 广州 510000)

摘要: 针对带交货期的单机逆调度问题, 建立以最小化系统调整为目标函数的单机逆调度数学优化模型; 利用互补性能, 采用串行、并行和嵌入等结构, 将遗传算法与变邻域搜索算法相结合, 设计出遗传-变邻域搜索算法、遗传-变邻域搜索交替算法和遗传-变邻域搜索协同算法 3 种混合算法。为产生逆调度激发机制, 采用非最优调度法, 将随机初始化与局部初始化进行结合, 创造逆调度环境; 此外, 为提高算法的局部搜索能力, 基于交叉变异操作等思想来构建四种搜索邻域, 通过邻域结构的切换, 加强局部搜索能力; 最后, 将提出的混合算法用于求解不同规模的问题实例, 与其他算法的求解结果进行比较, 证明提出的混合算法是可行的和有效的。

关键词: 车间调度; 逆调度; 混合算法; 不确定加工参数

中图分类号: TG156

Research on Single-machine Inverse Scheduling Methods with Due-dates Based on Variable Neighborhood Search Hybrid Algorithm

MOU Jianhui¹ PAN Quanke² MOU Jiancai³ XU Rufeng¹ YU Shanshan¹

- (1. School of Mechanical Engineering, Shandong University of Technology, Zibo 255000;
2. State Key Lab of Digital Manufacturing Equipment and Technology,
Huazhong University of Science & Technology, Wuhan 430074;
3. Huali Science and Technology College, Guangzhou 510000)

Abstract: Aiming at the single-machine inverse scheduling problem with due-dates(SISPD), a mathematical model with the minimal adjustment of system as a target is constructed. Using complementary properties, three hybrid algorithms combining genetic algorithm with variable neighborhood search algorithm are proposed by using serial, parallel and embedded structure. In the algorithm, a double scheduling method which combines heuristic non-optimal scheduling method with random initial population and local initial population is designed to construct inverse scheduling mechanism. Then, based on the features of problem and encoding, four neighborhood structures are put forward to improve the local search ability by changing the neighborhood structure. Finally, the proposed algorithms are used to solve numerical instances. The computational results show that the proposed method could solve SISPD effectively.

Key words: shop scheduling; inverse scheduling; hybrid algorithm; uncertain processing parameter

0 前言

真实车间环境十分复杂, 趋于智能化发展, 调度系统是其中的关键环节, 以往调度系统过于理想

化, 假设工作环境固定, 工件的加工信息和状态都已知且不能更改^[1-2]。面对实际挑战, 亟需新型的调度理论与方法来解决智能车间的实际问题^[3-4]。

但是, 真实生产加工过程中, 车间实际状况具有不确定性^[5], 常存在工件随机到达、机器故障维修等突发事件, 使得原有调度方案, 由于车间状态的变化而失去最优性, 有时甚至变为不可行。以往解决思路是进行重调度或者动态调度, 这些方法通

* 国家自然科学基金(51605267)、山东省自然科学基金(ZR2016EEQ07)和山东省高等学校科技计划(J16LB04)资助项目。20161210 收到初稿, 20170818 收到修改稿

过不断调整调度方案来解决问题。但是，传统的调度方法难以有效应用于生产实践。因此，本文将逆优化原理与车间调度方法相结合，研究一种新型的逆调度方法^[6] (图 1)，解决传统车间调度系统动态多变的情况，具有重要的应用价值。

图 1 逆调度机理

“逆调度问题” (Inverse scheduling problem, ISP) 是指：通过最小限度地调整加工参数或相关资源，使得原来的可行调度在当前情况下变为最优调度，并且求得调整之后的真实值^[7]。目前，逆调度研究较少，仍属于初级阶段，国内外关于逆调度研究的前沿文章也不多。与传统调度问题相比，逆调度也已经被证明是 NP 问题^[8-9]。针对车间类型及问题研究层面看，国际上只有关于单机逆调度问题^[10-11]及两台机器流水车间逆调度^[8,12]等简单问题，主要以加工完成时间和最小为目标；从求解方法来看，目前逆调度的求解方法还仅限于精确算法，主要采用线性规划方法进行求解，提到的方法仅限于：线性规划、牛顿法等，无法应用于大规模或复杂车间调度问题。

本文主要针对带交货期的单机车间逆调度问题，以最小化交货期改变量为目标，提出了基于遗传、变邻域算法结合的混合算法进行求解，最后，为了验证其优越性和可行性，将所提算法用于求解典型的调度实例。

1 带交货期单机车间逆调度问题描述

实际生产中，某工厂接到一批特殊订单，生产加工前，客户和生产者依据经验或估计的交货期制定相应最优加工路线及最优调度顺序，然而在实际生产中往往有许多不确定因素，导致实际的交货期往往与估计值不同，同时，车间真实情况存在调度顺序约束。为了解决此问题，采用逆调度方法，通过调整某些工件的交货期来保证按期交货。基于以上实际问题，本文研究单机环境下带有交货期的车

间逆调度问题，通过调整交货期使得原先调度变得最优，并且考虑最小化最大拖期的问题为研究背景。该问题可以描述如下^[13]。

所谓单机线是指： n 个带有不同加工信息的工件在一台机器上逐一加工。每个工件具有不同的加工信息，包括产品交货期、加工时间和权重等参数。加工前交货期可以与客户沟通预先得到，根据此参数信息预先生成一个调度方案，并且可以在合理范围内调整，该调度方案往往由于车间状态的变化而失去最优性，通过调整相关参数既保证方案满足期望，又使得相关成本最低或方案改变最小。为了更清晰的描述该问题的数学模型，条件假设如下：一个工件在同一时刻只能被一台机器加工；机器可连续运转，在开始时，机器和工件都是可行的；工件加工顺序无约束；已知每个工件的加工时间和工件交货期；假设加工时间固定，交货期在给定随机均匀区间取值；工件的加工顺序是不变的，且此加工顺序非最优。

基于上面的条件，将问题模型中涉及到的相关符号做如下说明： n 为工件个数； J_i 为第 i 个工件； p_i 为第 i 个工件的加工时间； d_i 为第 i 个工件的交货期； σ 为一个可行排序，非最优排序； C_i 为工件 i 的完工时间； $\bar{d}_i$ 为工件 i 调整后的交货期； $L_{\max}$ 为最大拖期时间； $L_{\max}$ 为改变 d_i 后的最大拖期时间； $1||L_{\max}$ 为以最大拖期最小为目标的单机调度问题。

考虑带交货期的单机逆调度问题可以描述为：单台机器， n 个工件分别为 $J_1, J_2, \dots, J_n$ ，工件相应的加工时间为 $p_i (i=1, 2, \dots, n)$ ，交货期为 $d_i (i=1, 2, \dots, n)$ ，并给定可行排序 $\sigma=(1, 2, \dots, n)$ ，调整交货期 d_i 为 $\bar{d}_i$ ，使得可行排序 σ 在新的交货期下成为最优排序，并且同时使交货期的调整量 $\|d_i - \bar{d}_i\|$ 最小。

对于带有最大交货期的调度问题，满足最早交货期规则 (Earliest due date, EDD) 的情况下，可以得到最优解^[14]：即将带有不同交货期的加工工件按照升序进行排列。该规则基于交货期的调度规则，经常作为最小化拖期调度问题的比较基准。数学表达式为 $d_1 \leq d_2 \leq \dots \leq d_n$ 。在具体的模型中会使用该规则。

通过上述问题描述，则带交货期的单机逆调度问题数学模型如下

$$\begin{aligned} & \min \|d_i - \bar{d}_i\| \\ & \text{s.t. } \bar{d}_1 \leq \bar{d}_2 \leq \dots \leq \bar{d}_n \end{aligned} \quad (1)$$

$$L_{\max}^- \leq L_{\max} \quad (2)$$

$$d_i > 0 \quad i=1,2,\dots,n \quad (3)$$

$$p_i > 0 \quad i=1,2,\dots,n \quad (4)$$

$$\bar{d}_i > 0 \quad i=1,2,\dots,n \quad (5)$$

$$L_{\max} = \max\{C_1 - d_1, C_2 - d_2, \dots, C_n - d_n\}$$

$$\bar{L}_{\max} = \max\{C_1 - \bar{d}_1, C_2 - \bar{d}_2, \dots, C_n - \bar{d}_n\}$$

$\|d_i - \bar{d}_i\|$ 是在 L_1 距离下的值

$$\|d_i - \bar{d}_i\| = \sum_{i=1}^n |d_i - \bar{d}_i|$$

本文以最小交货期调整量 d_i 作为目标; 约束式(1)确保最早交货期规则; 约束式(2)保证调整后的最大交货期目标值小于调整前; 同时, 式(3)~(5)确保相关参数 d_i 、 p_i 、 $\bar{d}_i$ 均不小于 0。

此问题已经被证明是 NP-hard 问题^[12], 同时, 逆调度针对非最优调度问题进行优化, 如何获得非最优调度是研究难点, 而且逆调度主要针对加工参数进行调整, 因此, 如何在合理范围内进行调整, 使得系统改变量最小, 是研究的重点。算法设计方面, 如何设计高效的智能求解方法是本研究的难点。

2 遗传变邻域算法流程框架

2.1 变邻域搜索算法基本理论

变邻域搜索算法(Variable neighborhood search, VNS)由 Mladenović 和 Hansen^[15-16]于 1997 年被提出, 算法是通过在全部搜索期间, 为了增大搜索范围, 通过系统的改变邻域结构集 N_k , 获得此时局部最优解, 在上述过程中, 在当前局部最优附件不断调整邻域结构探索新的搜索空间来发现新的局部最优解。VNS 是围绕基本的局部搜索算法提出的, 该算法自提出后就表现出良好的性能, 简单易操作、时效性强、人性化、易于与其他全局搜索算法相融合。图 2 介绍了 VNS 的基本过程。

图2 变邻域搜索过程

VNS 算法主要取决于邻域结构、邻域结构之间的搜索顺序、搜索机制等方面。主循环由 3 部分组成: 随机扰动、局部搜索、邻域变换。当前解 s 在某一邻域结构内进行随机扰动产生一个新解 s' , 即为局

部搜索的初始解, 通过局部搜索方法找到新的局部最优解 s'' , 与当前解进行比较, 若 s'' 优于 s , 则用 s'' 替换 s , 重新开始新的迭代过程; 否则令 $k=k+1$, 采用新的邻域结构重新进行随机扰动。通过此方式, 防止陷入局部最优, 使其能够从局部最优解的“山谷”中脱离出来, 提高算法的局部搜索能力。

一般, 多数局部搜索算法最多只采用两种邻域结构, 而 VNS 算法中, 如果采用不同的邻域结构比仅使用一种邻域结果效果更好, 在搜索过程中更容易找到全局最优解。本文基于交叉、变异操作等思想来构建两种搜索邻域, 在本文后面将会详细介绍。变邻域搜索算法的一般流程图如图 3 所示。

- (1) 建立初始种群, 构建邻域结构集合 N_i , $i=1, \dots, i_{\max}$ 和设计终止准则, 产生初始解 s_0 。
- (2) 用下列操作对个体进行循环, 达到终止条件。假设 $m=1$, 当循环次数达到 $m=m_{\max}$, 采用下面操作: 对 s 个体的第 i 个领域结构进行晃动, 得到解 $s'(s' \in N_i(s))$; 针对解 s' , 通过执行不同局部搜索策略, 获得新的局部最优 s'' ; 变化领域结构, 如果解 s'' 好过经邻域变换后得到的解 s , 则令 $s=s''$, 并且采用此操作继续进行邻域搜索, 否则令 $m=m+1$ 。

图3 变邻域搜索流程图

遗传算法的全局搜索性能，能够迅速的搜寻到解空间的大部分解；并且遗传算法具有内在并行性的优点，具有高求解效率。然而，单一使用 GA 使得算法具有较差的整体局部搜索能力，以至于整个搜索过程比较费时，使得算法后期效率不高。相比，变邻域搜索算法弥补上述缺陷，具有较高的局部搜索能力。鉴于 GA 和 VNS 的特性，将二者有机结合^[17-18]，根据优势互补原则，在不清楚哪种算法组合对求解该问题效果更好的情况下，采用串行、并行和嵌入等混合结构，获得 3 种基于遗传变邻域的混合求解方法，通过比较这三种遗传变邻域混合算法的结果，获得较优求解方法^[19]。下面将对每一种算法的详细步骤进行描述。

2.2 遗传变邻域搜索算法

为了克服早熟收敛等问题，可以将 GA 和 VNS 两个方法相互结合，提出遗传变邻域搜索算法 (Genetic algorithm and variable neighborhood search, GAV)，GAV 将混合算法分为两个主要模块：模块一是遗传算法部分，为了获得较优的初始种群 A ，采用遗传算法对种群进行全局搜索；第二个模块是 VNS 部分，在种群 A 的基础上，抽取 M 的个体：部分来自 A 种群的最优个体，部分随机抽取。然后对 M 个个体进行 VNS 局部搜索，优化此部分个体。GAV 算法的流程如图 4 所示。

图 4 遗传-变邻域搜索算法流程图

2.3 遗传变邻域交替算法

遗传-变邻域搜索交替算法 (Genetic algorithm and variable neighborhood search in turn, GAVT) 与 GAV 略有不同，主要针对遗传算法和变邻域算法各

自的优缺点综合考虑，利用两种互补性能，对于各部分算法均有所改进，发挥各自优势，使其能够有效高地解决单机逆调度问题。GAVT 算法主要改进如下。

(1) 采用混合初始化方法，将非最优调度方法与混合的初始化方法进行结合，以此方法获得高质量的初始化种群，提高遗传变邻域交替算法的搜索效率，同时，采用外部档案进行个体保优，按照一定比例，把搜索过程中的优秀个体存入外部档案中。

(2) 为了避免种群陷入早熟，且更好地保持种群多样性，在交叉操作时，采用两种方式选择个体。一种是从外部档案集中选择一个父代，然后采用锦标赛选择方式从当前种群选择另一个父代，进行交叉操作，主要为了使算法初期加快收敛速度；另一种：使用采用锦标赛选择方法来挑选个体，在此过程中，不断涌现性能优良的个体，采用这种方法不仅能够防止算法过早成熟，而且能够增加种群多样性。

通过提出的初始化方法产生初始种群，将产生的优良个体按一定比例存入外部档案，分别采用多种方式进行交叉操作，然后对当前种群进行变异操作，最后采用邻域搜索算法求得每个个体的较优解。具体操作步骤如图 5 所示。

图 5 遗传-变邻域搜索交替算法流程图

2.4 遗传变邻域协同算法

遗传-变邻域搜索协同算法 (Genetic algorithm and variable neighborhood search cooperative algorithm, GAVC) 的基本思想如下：利用两个算法各

自良好特性, 对其分别进行并行协同处理, 首先按照提出的初始化方式进行种群初始化, 并将生成的群体分成两部分, 其中 M 个个体进行独立的 VNS 搜索, N 个个体进行遗传算法。通过该操作使遗传变邻域混合算法具有两者的各自特性, 在邻域搜索中, 遗传算法的全局搜索能力能够提高算法的搜索力度, 此方法提高了算法的质量和效率。其具体操作步骤如图 6 所示。

图 6 遗传-变邻域搜索协调算法流程图

3 遗传变邻域混合算法的关键操作

3.1 SISPD 问题编码方法

本文研究带交货期的单机车间逆调度问题, 同时考虑交货期不确定的情况, 在设计编码时需要针对逆调度问题特征考虑, 染色体必须能准确地表达出相应工件的交货期值, 并且加工参数在一定范围内是可调的。本文将采用基于交货期的编码方案, 针对单机调度问题特征, 工件仅在一台机器上加工, 具有任意置换染色体后总能得到可行调度的特点, 因此采用此编码和对应的解码方法简单易行, 易于在计算机上编程实现。

对于一个 n 个工件在单台机器上加工的单机车间调度问题, 假设第 i 个工件的加工时间和交货期分别为 p_i 和 d_i , 则染色体由 n 个基因组成, 每个基因分别代表工件交货期值, 从左到右扫描染色体。为了更好地描述染色体编码, 具体的实例如图 7 所示。以 7 个工件在单台机器上加工为例, 给定各自交货期的原始值以及可变范围。单机逆调度问题的解码相对其他问题比较简单, 由于本文采用基于交货期的编码方案, 因此解码对工件的染色体从左到右依次读取, 获得调整之后对应工件序号的交货期改变值。

图 7 基于交货期的编码方案

3.2 SISPD 种群初始化

种群初始化在进化算法中起着至关重要的作用, 初始解的质量直接影响着优化算法的求解效率^[17-18]。逆调度问题是针对给定的非最优调度进行调整。因此, 在整个算法执行之前, 需要得到一个初始近优调度方案。针对 $1||L_{\max}$ 调度问题。初始化第一步就是进行初始排序操作, 即根据原始加工信息或估计参数信息, 运用 EDD 规则, 获得逆调度的初始最优序列。为了创造逆调度环境, 随机交换某几个工件位置, 产生非最优调度序列进行逆调度。

随机初始化和局部初始化具体操作如下: 随机初始化是指不考虑原始参数信息, 通过随机的方式重新生成新调度参数。在此方法中, 所有工件加工参数都在可控范围内随机生成, 具体操作如图 8 所示。图中调度序列是一个非最优的调度顺序, 初始化前的染色体为各个工件的原始加工时间, 并且给出各个加工时间的调整范围。初始化后染色体代表在给定的加工时间调整范围内, 随机产生新的加工时间的染色体。

图 8 随机初始化方案

局部初始化是指保留某些原始加工参数不变, 而其他工件的加工参数在可控范围内随机生成, 具体程序如图 9 所示。图中调度序列是一个非最优的调度顺序, 对应染色体为每个工件的原始加工时间,

图 9 单机逆调度问题的局部初始化方案

并且给出各个加工时间的调整范围。初始化后染色体中, 工件 2、3、7 是原始加工时间, 工件 1、5、4、6 是在可控范围内随机生成的加工时间。局部初始化方案如图 6 所示。

本文提出的初始化方法的具体步骤如下。

(1) 根据原始加工参数, 按照 WSPT 规则生成此条件下的最优调度顺序。

(2) 为创造逆调度环境, 随机交换某几个工件的位置, 使得最优调度顺序变成非最优。

(3) 在此基础上, 依据一定比例 R_r , 选择随机初始化方式生成 M 个染色体。

(4) 选择局部初始化方式生成 $(N-M)$ 个染色体, 混合生成逆调度初始种群。

3.3 变邻域结构设计

针对逆调度问题特性, 邻域结构与传统邻域搜索算法中的略有不同, 本文借鉴逆调度中交叉、变异思想, 主要采用四种邻域结构进行局部搜索^[19], 所采用的邻域结构如下。

3.3.1 邻域结构 1

邻域结构 1 借鉴以往算法中交叉操作的思想, 从精英解中随机选择 k 个个体。在基因序列中随机选择两个基因位置, 则 k 个个体中位于两个位置点的相应基因组成当前解的邻域。这样得到不同邻域个体, 比较新个体的适应度值, 选择较优个体。这样所有邻居构成第 1 邻域。具体操作如图 10 所示。

图 10 邻域结构 1

3.3.2 邻域结构 2

邻域结构 2 基于交叉操作的思想, 主要以块交叉操作为基础, 选择 k 个较优个体。在基因序列中

随机选择两个基因位置, 两基因位置中间部分形成基因块。将精英解中基因块部分的基因依次取代当前解中相应基因。这样得到不同邻域个体, 比较新个体的适应度值, 选择较优个体。这样所有邻居构成第 2 邻域。具体操作如图 11 所示。

图 11 邻域结构 2

3.3.3 邻域结构 3

本文考虑交货期调整的情况, 调整范围为 $[d_i-4, d_i+4]$ 。因此, 将交货期区间分为 3 个步长, 分别为 $l_1=1, l_2=2, l_3=4$ 。在基因序列中随机选择选择一个位置, 按照各自步长调整。这样所有邻居构成第 3 邻域。具体操作如图 12 所示。

图 12 邻域结构 3

3.3.4 邻域结构 4

在基因序列 s_0 中, 随机调整交货期重新产生新的基因序列 s , 若 $f(s) < f(s_0)$, 则记下调调整的基因位置, 在基因序列 s 的基础上, 对该基因位置进行操作, 按照交货期在调整范围内交货期可取得的数值作为不同邻域, 产生不同的邻域解。这样所有邻居构成第 4 邻域。具体操作如图 13 所示。

4 试验结果与分析

本文提出的 GAIP 算法采用 C++ 语言编程, 其运行环境为 Intel Core 2 Duo CPU, 主频 2.29 GHz, 内存 2.86 GB。根据参数的不同组合, 设计多组不同规模的问题实例, 并将提出的算法在所设计的实例上进行试验验证。

图 13 邻域结构 4

4.1 试验设计

为了说明本文提出的 GAVT 算法求解带交货期单机逆调度问题的潜在性能, 将本文提出的 GAVT 算法在相同的加工条件下进行结果比较。VNS 方法的相关参数设置如下: $outloop=50$, $inloop=mn$, 根据前文给出的 VNS 方法的流程, 与本文提出的遗传算法相结合, 遗传算法参数设置如下: 种群规模(N)设为 100, 最大进化代数(T)设为 100, 交叉概率(P_c): 0.8, 复制概率(P_r): 0.01, 变异概率(P_m): 0.1。

4.1.1 问题实例生成方法

本文提出的 GAVT 方法将在一系列测试实例上

进行。不同规模的工件数量、加工时间等参数构成一系列问题实例^[20-21]。

针对不同问题实例, 其中 n 代表工件数量(个), 具有三个规模分别是: 20、50 及 100; 加工时间 p (时)分布如前章所述, 其取值区间是: $[5, 25]$ 、 $[30, 100]$ 及 $[80, 120]$; 同理, 工件的交货期 d 在离散的均匀分布区间内取值, 取值范围分别为 $[r + (P - r) \times 1 - D - R/2], r + (P - r) \times (1 - D + R/2)]$ 。其中, D 代表估计工件拖期的百分比, (紧张度因子), R (交货期范围因子)代表交货期的取值区间。两参数的设置分别为 0.1、0.5 或 0.9。 r 为工件的到达时间(时), 其取值范围符合离散的一致分布区间 $[0, k \times P]$, 其中 k 表示到达时间紧密度, 本文考虑密集到达和稀疏到达两种情况, 取值分别为 0.3、0.5、0.7, P 表示所有工件的加工时间之和。根据不同参数, 包括零件加工时间 p 和工件数量 n , 组合生成 $9(3 \times 3)$ 个问题实例, 记作 $TS_{np}1-9$ 。每个测试实例, 根据交货期参数的不同, 另外构成 $27(3 \times 3 \times 3)$ 类不同测试实例, 记作 $TS_{TRk}1-27$ 。这样, 一共生成了 36 个验证实例, $TS_{np}1-9$ 和 $TS_{TRk}1-27$ 。 $TS_{np}1-9$ 实例基础上, 根据工件交货期紧张度因子 D 、范围因子 R 和到达时间紧密度 k 等参数的不同进行测试; $TS_{TRk}1-27$ 实例基础上, 根据工件个数 n 和工件加工时间 p 的不同进行测试。不同实例的仿真参数如表 1 所示。

表 1 训练集合和验证集合包含的问题实例仿真参数值

TS_{np}	n	p	D	R	k	TS_{TRk}	D	R	k	n	p
1	10	$U[1,100]$				1	0.1	0.1	0.3		
2	10	$U[100,200]$				2	0.1	0.5	0.3		
3	10	$U[200,300]$				3	0.1	0.9	0.3		
4	50	$U[1,100]$				4	0.5	0.1	0.3		$U[1,100]$
5	50	$U[100,200]$	0.1, 0.5, 0.9	0.1, 0.5, 0.9	0.3, 0.5, 0.7	5	0.5	0.5	0.3	10, 50, 100	$U[100,200]$
6	50	$U[200,300]$				6	0.5	0.9	0.3		$U[200,300]$
7	100	$U[1,100]$				7	0.9	0.1	0.3		
8	100	$U[100,200]$				8	0.9	0.5	0.3		
9	100	$U[200,300]$				9	0.9	0.9	0.3		
10	0.1	0.1	0.5			19	0.1	0.1	0.7		
11	0.1	0.5	0.5			20	0.1	0.5	0.7		
12	0.1	0.9	0.5			21	0.1	0.9	0.7		
13	0.5	0.1	0.5		$U[1,100]$	22	0.5	0.1	0.7		$U[1,100]$
14	0.5	0.5	0.5	10, 50, 100	$U[100,200]$	23	0.5	0.5	0.7	10, 50, 100	$U[100,200]$
15	0.5	0.9	0.5		$U[200,300]$	24	0.5	0.9	0.7		$U[200,300]$
16	0.9	0.1	0.5			25	0.9	0.1	0.7		
17	0.9	0.5	0.5			26	0.9	0.5	0.7		
18	0.9	0.9	0.5			27	0.9	0.9	0.7		

4.1.2 遗传变邻域混合算法参数设置

遗传变邻域混合算法的相应参数设置如下: 外循环迭代次数 $O=50$, 内循环迭代次数 $I=mn$, 根据前文给出的 VNS 方法的流程, 与本文提出的遗传算

法相结合, 本文提出的新的初始化方法中, 涉及到两个参数的设置: 随机初始化比例 R_r 、局部初始化染色体中工件参数调整的个数 N_{lc} , 根据试验证明: 对于小规模问题实例(20 个工件), 参数设置 $R_r=0.5$,

$N_{lc}=30\%$ 时,可以取得最优结果: $RPD=0.12$, $Z=15$; 而对于中等规模问题实例(50 个工件), 参数设置 $R_r=0.7$, $N_{lc}=30\%$ 时,可以取得最优结果: $RPD=0.15$, $Z=41$; 同理大规模问题实例(100 个工件), 参数设置 $R_r=0.7$, $N_{lc}=70\%$ 时,可以取得最优结果, $RPD=0.21$, $Z=68$ ^[22-24]。不同规模的问题实例参数设置如表 2 中所示。

表 2 GAVT 算法控制参数设置

参数	数值
种群大小	100
终止条件	100 次迭代
外循环次数	O : 50
内循环次数	I : mn
工件个数	随机初始化方式+局部初始化方式
初始化	20 $R_r=0.5$, $N_{lc}=30\%$
方式	50 $R_r=0.7$, $N_{lc}=30\%$
	100 $R_r=0.7$, $N_{lc}=70\%$
变异	变异概率 0.1
交叉	交叉概率 0.8
复制	复制概率 0.01

4.1.3 比较对象

本文将两种算法进行混合,在此基础上,基于不同的混合结构,主要采用串行、并行和嵌入等结构。设计了遗传-变邻域搜索算法(GAV)、遗传-变邻域搜索交替算法(GA and VNS in turn, GAVT)和遗传-变邻域搜索协同算法(GA and VNS cooperative algorithm, GAVC) 3 种混合遗传算法。为了更进一步说明提出遗传变邻域混合算法彼此的性能,首先将遗传变邻域混合算法与未加入 VNS 局部搜索的

单一遗传算法相比较,验证结果表明了提出遗传变邻域混合算法的优越性;其次,为了验证 3 种不同混合算法的性能,将三种混合算法的结果进行对比。

4.2 结果分析与讨论

4.2.1 基于 GAVC 和 GA 的方法性能比较

为了验证本文所提出的遗传变邻域混合算法比单一算法优越, GAVC 和 GA 分别在测试实例 TS_{np}1-9 上进行测试。在相同的测试环境下,每个测试实例分别单独运行 10 次。工件交货期紧张度因子 T 、范围因子 R 随机选取,到达时间紧密度 k 分别取 3 种不同数值,分别代表不同紧密程度。表 3—表 5 分别显示了工件紧密程度 k 在 0.3、0.5 和 0.7 情况下的试验结果,参数 k 被定义为工件到达紧密程度。工件到达率,也就是说单位时间里的工件平均数量。当 $k=0.25$ 时,表示单位时间内会有 0.25 个工件到达。为了充分验证算法的性能指标,文中采用 3 种紧密程度不同的工件和 6 几种不同的工件数量。其中,按照工件到达程度分为两种:工件密集到达,工件稀疏到达;按照工件规模大小不同分为:大规模,中等规模和小规模。其中,计算结果

$$RPD_j = \frac{1}{n} \sum_{i=1}^n \frac{|S_{Gj} - B_j|}{B_j} \times 100\%,$$

计算每个实例的平均相对偏差,该公式中, S_{Gj} 表示第 j 个实例的最优结果; B_j 表示 j 组实例中所有算法求得结果的最优值,通过该指标能够反应解的波动情况。指标 Z 代表单机逆调度问题的目标函数,其公式为:

$$Z = \min \sum \|d_j - \bar{d}_j\|。$$

表 3 $k=0.3$ 时提出 GAVC 算法与单一 GA 算法结果比较

TS _{np} 1-9	1	2	3	4	5	6	7	8	9
$RPD(\%)$	67.5	72.8	69.6	56.1	55.3	49.9	32.7	46.1	39.4
GA									
Z	36	23	38	99	96	103	587	862	810
CPU	7.61	6.82	7.09	63.50	64.74	62.38	127.58	151.87	133.68
$RPD(\%)$	42.7	58.2	57.4	42.5	44.1	39.3	22.5	31.1	30.4
GAVC									
Z	23	17	26	78	69	72	473	612	658
CPU	15.61	12.78	14.78	134.16	145.32	118.65	381.93	378.70	431.28

表 4 $k=0.5$ 时提出 GAVC 算法与单一 GA 算法结果比较

TS _{np} 1-9	1	2	3	4	5	6	7	8	9
$RPD(\%)$	68.8	76.5	71.4	61.5	59.2	50.8	35.2	51.1	47.2
GA									
Z	38	33	38	96	106	112	592	894	836
CPU	8.61	8.82	7.59	68.55	69.94	69.08	140.08	153.47	136.62
$RPD(\%)$	43.7	58.9	59.1	45.5	47.8	41.6	32.5	37.1	35.3
GAVC									
Z	35	27	25	74	70	74	480	675	673
CPU	17.63	13.74	15.46	152.10	149.85	128.35	390.35	398.72	447.68

表 5 $k=0.7$ 时提出 GAVC 算法与单一 GA 算法结果比较

TS _{np} 1-9		1	2	3	4	5	6	7	8	9
GA	RPD(%)	67.5	72.8	69.6	56.1	55.3	49.9	32.7	46.1	39.4
	Z	37	24	39	103	104	115	593	872	832
	CPU	8.23	7.85	7.53	68.52	69.24	65.33	131.19	153.54	139.27
GAVC	RPD(%)	40.7	54.9	56.4	44.5	44.8	40.3	26.5	33.6	30.7
	Z	23	17	26	72	69	67	453	603	658
	CPU	19.61	15.78	18.18	157.26	155.36	129.65	397.94	402.75	473.36

表 3~5 给出了 k 取 0.3, 0.5 和 0.7 三种工件不同密集程度情况下, 混合遗传算法 GAVC 与单一遗传算法 GA 的比较结果。实例 TS_{np}1-9 根据工件数量分别为小、中、大三种规模, 其中 TS_{np}1-3 为小规模, TS_{np}3-6 为中规模, TS_{np}7-9 为大规模。从上述表中可以得到如下结果。

(1) 对于 9 组不同规模的问题实例, 每一实例中(包括不同的工件密集程度)GAVC 算法产生的平均相对偏差明显小于单一使用 GA 算法产生的平均相对偏差, 因此, 本文提出的遗传变邻域混合算法优于单一遗传算法。尤其是, 对于 3 组小规模实例(20 个工件), GAVC 求得的 RPD 结果优势更加明显。然而 Z 值代表取得的目标值(交货期调整量之和), 也就是说 Z 值越小越好。从上述 3 个表中可以看出, 本文提出的 GAVC 算法优化得到的目标值明显小于 GA 得到的 Z 值, 尤其是对于大规模的问题实例, 可以求得, 在不同工件密集程度下所有实例, 由 GAVC 和 GA 算法求得的平均 Z 值分别为 590.4 和 764.2。因此, 在此结果上本文所提出的混合遗传算法表现出较好性能。

(2) 从计算时间来看, 本文提出的混合遗传算法的运行时间比 GA 算法略长。这个主要是因为本文提出的算法加入了 VNS 局部搜索算法, 这些都影响到算法运行时间。但是 GAVC 的计算时间只是略高于 GA 算法。虽然本文提出算法所需总的运行时间略长, 但综合看来, 它们仍然是合理可接受的。

(3) 表 3~5 得到试验结果为工件紧密到达的情况, 参数设置分别为 0.3、0.5 和 0.7。本文提出的

混合遗传算法具有优良的性能。相比于工件稀疏到达的情况, 工件紧密到达的情况, 本文所提混合算法略微改善了调度系统, 同时也损失了调度效率。因此, 对于不同工件紧密程度情况下的试验结果略有差别。

4.2.2 三种混合遗传算法的性能比较

为了验证本文所提出的 3 种混合遗传算法性能, 本文将提出的 GAV 算法、GAVT 算法和 GAVC 算法分别在测试实例 TS_{Trk}1-27 上进行测试。每个遗传变邻域混合算法在相同的测试环境下进行, 每个测试实例分别单独运行 10 次。工件交货期紧张度因子 T 、范围因子 R 依据不同值选取, 到达时间紧密度 k 分别取 3 种不同紧密程度测试。针对不同的问题规模进行试验测试。表 6~8 分别显示了 TS_{Trk}1-9、TS_{Trk}10-18 和 TS_{Trk}19-27 的测试结果, 其中 TS_{Trk}1-9 为小规模, TS_{Trk}10-18 为中规模, TS_{Trk}19-27 为大规模。

为了评价提出方法的性能, 本文结果计算了 3 种指标进行比较。分别为结果测试的平均值、最优值和平均相对偏差 $ARPD$ 。

$$ARPD = \frac{1}{n} \sum_{i=1}^n \frac{|S_{HG} - S_B|}{S_B} \times 100\%$$

由于本文是首次采用智能优化算法求解逆调度问题, 因为目前没有任何结果可以拿来比较, 因此在 $ARPD$ 公式中, S_{HG} 代表所选用的算法求得结果, S_B 代表所有算法求得的最优解。通过 $ARPD$ 可以看出每种方法求的结果的波动情况。

表 6 提出的 3 种遗传变邻域混合算法试验结果比较(TSTRk1-9)

TS _{Trk} 1-9		1	2	3	4	5	6	7	8	9
GAV	平均值	39	27	41	27	28	33	35	31	28
	最优值	37	24	39	26	27	31	33	28	26
	ARPD(%)	38.7	35.9	36.4	33.7	39.1	36.2	39.7	36.2	36.7
GAVT	平均值	30	22	24	24	25	26	19	26	28
	最优值	17	20	19	21	22	21	18	21	23
	ARPD(%)	35.1	32.7	33.0	30.3	28.5	27.2	32.6	30.8	31.9
GAVC	平均值	32	22	26	21	22	21	19	19	20
	最优值	25	20	23	19	21	20	18	18	19
	ARPD(%)	34.3	30.5	32.6	30.1	29.5	26.4	30.7	30.4	30.6

表 7 提出的 3 种遗传变邻域混合算法试验结果比较(TSTRk10-18)

TS _{TRk} 10-18	10	11	12	13	14	15	16	17	18
平均值	96	97	93	99	93	96	90	90	88
GAV 最优值	88	83	84	81	77	81	83	82	76
ARPD(%)	56.1	56.3	47.9	53.7	59.1	56.2	49.7	46.2	47.7
平均值	90	86	83	99	93	89	80	94	88
GAVT 最优值	87	78	74	91	87	81	83	88	76
ARPD(%)	55.4	54.6	44.5	50.7	54.7	53.4	46.4	48.8	40.2
平均值	91	89	79	90	92	87	81	97	89
GAVC 最优值	88	89	72	84	83	85	85	90	79
ARPD(%)	56.4	57.9	43.5	53.7	52.8	51.7	44.9	47.7	45.0

表 8 提出的 3 种遗传变邻域混合算法试验结果比较(TSTRk19-27)

TS _{TRk} 19-27	19	20	21	22	23	24	25	26	27
平均值	601	578	537	627	573	533	635	551	528
GAV 最优值	587	562	510	604	527	531	603	528	526
ARPD(%)	38.7	45.9	47.4	38.2	39.1	46.2	40.6	36.2	39.0
平均值	593	571	520	596	565	501	604	548	524
GAVT 最优值	572	560	508	581	523	498	592	506	517
ARPD(%)	35.2	40.7	42.3	37.2	37.4	42.3	38.5	33.2	36.5
平均值	598	577	531	608	575	527	621	550	528
GAVC 最优值	579	565	510	597	523	518	599	528	522
ARPD(%)	38.3	43.5	47.3	37.9	39.5	45.8	39.1	34.8	37.9

表 6~8 给出了本文提出的三种混合遗传算法 GAV、GAVT 和 GAVC 的比较结果。实例 TS_{TRk}1-27 根据参数不同、工件数量不同分为不同测试实例。从上述表中可以得到如下结果。

(1) 3 种遗传变邻域混合算法求得的最优解和平均解都好于单一遗传算法，也就是说本文所提算法能够发挥混合优势，获得更优解。并且，从指标 ARPD 指标可以看出，所提三种算法都获得较小解，说明本文所提算法具有好的求解精度和较高的稳定性。

(2) 3 种遗传变邻域混合算法对 27 个问题实例，TS_{TRk}1-9 实例结果中，GAVT 和 GAVC 算法得到的平均值、最优值和 ARPD 均优于 GAV 算法，GAVT 和 GAVC 算法比较结果中，平均值和最优值差别不大，表明这两种混合方式对求解质量的影响不大，但 ARPD 略有不同，GAVC 优于 GAVT，相比而言，GAVC 的鲁棒性较优。

(3) TS_{TRk}10-18 实例结果中，GAVT 和 GAVC 算法得到的平均值、最优值和 ARPD 均优于 GAV 算法，GAVT 和 GAVC 算法比较结果中，平均值和最优值差别不大，表明这两种混合方式对求解质量的影响不大，并且 ARPD 结果差别不大，这表明 GAVC 和 GAVT 算法在解决该组实例中表现较优性能。

(4) TS_{TRk}19-27 实例结果中，GAVT 和 GAVC 算法得到的平均值、最优值和 ARPD 均优于 GAV 算法，

GAVT 和 GAVC 算法比较结果中，可以发现平均值、最优值和 ARPD 结果，GAVT 均优于 GAVC 算法，这表明在解决该组实例中 GAVT 表现较优性能。

三种混合算法对几个实例得到的结果基本差别不大，但 ARPD 的结果稍有不同，这表明所提算法表现了不同的鲁棒性。

4.2.3 遗传变邻域混合算法和单一算法的仿真过程比较

由图 14 可知，混合算法 GAVT 比单一算法 GA 以更快的速度收敛到更好的解。其中，图 14a、14b 表明遗传变邻域混合算法较单一算法，无论是目标值还是相对偏差，在不同问题上都表现明显优势。

5 结论

本文针对带交货期的单机车间逆调度问题，将逆优化机理引入调度问题，通过以最小化交货期改变量为目标进行调整；其次，将遗传算法的优良性能，结合具有较强局部搜索能力的变邻域搜索方法，利用互补性能，采用三种不同的混合结构，提出三种高效的基于变邻域搜索的混合遗传算法。为了验证算法性能，采用基准实例进行对比测试，先将遗传变邻域混合算法与单一算法进行比较，同时将几种算法之间也进行了比较，验证了研究车间逆调度对改善车间调度系统的必要性，证明提出的混合遗传算法的有效性和优越性。

图 14 遗传变邻域混合算法和单一算法的仿真过程图

参 考 文 献

- [1] NGUYEN S, ZHANG M, JOHNSTON M, et al. Automatic design of scheduling policies for dynamic multi-objective job shop scheduling via cooperative coevolution genetic programming[J]. *Evolutionary Computation*, IEEE Transactions on, 2014, 18(2): 193-208.
- [2] XU Y, WANG L, WANG S, et al. An effective teaching-learning-based optimization algorithm for the flexible job-shop scheduling problem with fuzzy processing time[J]. *Neurocomputing*, 2015, 148: 260-268.
- [3] HE T, LIU F, MA Y, et al. Study on shop-floor scheduling [J]. *Journal of Mechanical Engineering*, 2000, 36(5): 97-102.
- [4] TANG, D B. Energy-efficient approach to minimizing the energy consumption in an extended job-shop scheduling problem[J]. *Chinese Journal of Mechanical Engineering*, 2015, 28(5): 1048-1055.
- [5] GAO K Z, SUGANTHAN P N, TASGETIREN M F, et al. Effective ensembles of heuristics for scheduling flexible job shop problem with new job insertion[J]. *Computers & Industrial Engineering*, 2015, 90: 107-117.
- [6] BRUCKER P, SHAKHLEVICH N V. Inverse scheduling with maximum lateness objective. *Journal of Scheduling*, 2009, 12(5): 475-488.
- [7] BRUCKER P, SHAKHLEVICH N V. Inverse scheduling: two-machine flow-shop problem[J]. *J. Sched.*, 2011, 14(3): 239-256.
- [8] KOULAMAS C. Inverse scheduling with controllable job parameters[J]. *International Journal of Services and Operations Management*, 2005, 1(1): 35-43.
- [9] ZHANG FENG, NG T C, TANG G C. Inverse scheduling: Applications in shipping[J]. *International Journal of Shipping and Transport Logistics*, 2011, 3(3): 312-322.
- [10] 陈荣军, 陈峰, 唐国春. 单台机器总完工时间排序问题的反问题[J]. *上海第二工业大学学报*, 2005, 22(2): 1-7. CHEN Rongjun, CHEN Feng, TANG Guochun. Inverse problems of a single machine scheduling to minimize the total completion time[J]. *J. Shanghai Second Polytechnic University*, 2005, 22(2): 1-7.
- [11] 陈荣军. 单台机器总完工时间随机排序问题的反问题[J]. *常州工学院学报*, 2006, 19(6): 1-5. CHEN Rongjun. Inverse scheduling problem on a single machine stochastic scheduling to minimize the total completion time[J]. *Journal of Changzhou of technology*, 2006, 19(6): 1-5.
- [12] 陈荣军, 唐国春. 单机供应链排序及流水作业的反问题模型[J]. *运筹与管理*, 2009, 18(2): 80-84. CHEN Rongjun, TANG Guochun. Inverse problems of supply chain scheduling and flowshop scheduling[J]. *Operations research and management science*, 2009, 18(2): 80-84.
- [13] PHAM Hongtruong, 鲁习文. 平行机上单位加工时间加权总完工时间排序问题的反问题[J]. *华东理工大学学报*, 2012, 38(6): 757-761. PHAM Hongtruong, LU Xiwen. Inverse problem of total weighted completion time objective with unit processing time on identical parallel machines[J]. *Journal of East China University of Science and Technology*, 2012, 38(6): 757-761.
- [14] VERELA R, VELA C R, PUENTE J. GOMEZ A. A knowledge-based evolutionary strategy for scheduling problems with bottlenecks[J]. *European Journal of Operational Research*, 2003, 145: 57-71.
- [15] SHAPIRO J F. A survey of Lagrangian techniques for discrete optimization[J]. *Annals of Discrete Mathematics*, 1979, 5: 113-138.
- [16] HANSEN P, MLADENOVIC N. Variable neighborhood search[M]. New York: Springer, 2014.
- [17] XIA W, WU Z, ZHANG W, et al. Applying particle swarm optimization to job-shop scheduling problem[J]. *Chinese Journal of Mechanical Engineering*, 2004, 17(3): 437-441.
- [18] 潘全科, 王文宏, 朱剑英, 等. 基于粒子群优化和变邻

- 域搜索的混合调度算法[J]. 计算机集成制造系统, 2007, 13(2): 323-328.
- PAN Quanke, WANG Wenhong, ZHU Jianying, et al. Hybrid heuristics based on particle swarm optimization and variable neighborhood search for job shop scheduling. Computer integrated manufacturing system, 2007, 13(2): 323-328.
- [19] RANTALANKILA P, KANNALS J, RAHTU E. Generating object segmentation proposals using global and local search[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 2417-2424.
- [20] MOU J H, GAO L, LI X Y, et al. Optimization of the reverse scheduling problem by a modified genetic algorithm. International Journal of Production Research, 2015, 53(23): 6980 - 6993.
- [21] MOU J H, LI X Y, GAO L, et al. An improved genetic algorithm for single-machine inverse scheduling problem[J]. Mathematical Problems in Engineering, 2014, 2014(4): 1-14.
- [22] MOU J H, GAO L, LI X Y, et al. An effective L-MONG for solving multiobjective flow shop inverse scheduling problems[J]. Journal of Intelligent Manufacturing, 2015: 1-19.
- [23] 王松. 基于遗传算法的单机逆调度方法研究[D]. 武汉: 华中科技大学, 2014.
- WANG Song. Research on the solution method of single machine inverse scheduling based on genetic algorithm[D]. Wuhan: Huazhong University of Science and Technology, 2014
- [24] 牟健慧, 郭前建, 高亮, 等. 基于混合的多目标遗传算法的多目标流水车间逆调度问题求解方法[J]. 机械工程学报, 2016, 52(22): 186-197.
- MU Jianhui, GUO Qianjian, GAO Liang et al. Hybrid HMGA algorithm for solving multi-objective flow-shop inverse scheduling problems[J]. Journal of Mechanical Engineering, 2016, 52(22): 186-197.
-
- 作者简介: 牟健慧(通信作者), 女, 1983 年出生, 博士研究生。主要研究方向为车间调度与优化。
E-mail: mjhcr@163.com