

12-16-2020

## Automatic Discovery Method of Dynamic Job Shop Dispatching Rules Based on Hyper-Heuristic Genetic Programming

Suyu Zhang

*Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;*

Wang Yan

*Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;*

Zhicheng Ji

*Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

# Automatic Discovery Method of Dynamic Job Shop Dispatching Rules Based on Hyper-Heuristic Genetic Programming

## Abstract

**Abstract:** The dynamic job shop has the uncertainty of resource state and the randomness of tasks, so it is difficult to find the common dispatching rules applicable to a variety of complex production scenarios. A method for automatic discovery of dynamic shop dispatching rules based on Hyper-Heuristic genetic programming is proposed, with makespan and average weighted tardiness as the optimization goals, is improved by using the automatic discovery of machine sequencing rules and the dynamic adaptability of workshop scheduling under different production scenarios. Through the semantic analysis of dispatching rules, the function of terminators on different optimization objectives is analyzed. The experiment result shows that the proposed algorithm can effectively generate appropriate dispatching rules which is obviously better than the manual designed benchmark rules for different production scenarios.

## Keywords

genetic programming, dynamic job shop, dispatching rules, automatic discover

## Recommended Citation

Zhang Suyu, Wang Yan, Ji Zhicheng. Automatic Discovery Method of Dynamic Job Shop Dispatching Rules Based on Hyper-Heuristic Genetic Programming[J]. Journal of System Simulation, 2020, 32(12): 2494-2506.

# 基于超启发式遗传规划的动态车间调度方法

张苏雨, 王艳, 纪志成

(江南大学教育部物联网技术应用工程中心, 江苏 无锡 214122)

**摘要:** 动态作业车间存在资源状态的不确定性和任务的随机性, 难以寻找适用于多种复杂生产情景的通用调度规则。提出一种基于超启发式遗传规划的动态车间调度规则自动化发现方法, 以最大完工时间和平均加权迟到时间为优化目标, 利用机器排序规则的自动化发现, 来提高不同生产情景下车间调度的动态适应性。通过对演化调度规则的语义分析, 分析了 GP 树终端属性对不同优化目标的作用。实验结果表明, 所提算法能够针对不同生产场景, 生成适合的调度规则, 且性能优于人工设计的基准调度规则。

**关键词:** 遗传规划算法; 动态作业车间; 调度规则; 自动化发现

中图分类号: TP391.9

文献标识码: A

文章编号: 1004-731X (2020) 12-2494-13

DOI: 10.16182/j.issn1004731x.joss.20-FZ0452

## Automatic Discovery Method of Dynamic Job Shop Dispatching Rules Based on Hyper-Heuristic Genetic Programming

Zhang Suyu, Wang Yan, Ji Zhicheng

(Engineering Research Center of Internet of Things Technology Applications Ministry of Education, Jiangnan University, Wuxi 214122, China)

**Abstract:** The dynamic job shop has the uncertainty of resource state and the randomness of tasks, so it is difficult to find the common dispatching rules applicable to a variety of complex production scenarios. A method for automatic discovery of dynamic shop dispatching rules based on Hyper-Heuristic genetic programming is proposed, with makespan and average weighted tardiness as the optimization goals, is improved by using the automatic discovery of machine sequencing rules and the dynamic adaptability of workshop scheduling under different production scenarios. Through the semantic analysis of dispatching rules, the function of terminators on different optimization objectives is analyzed. The experiment result shows that the proposed algorithm can effectively generate appropriate dispatching rules which is obviously better than the manual designed benchmark rules for different production scenarios.

**Keywords:** genetic programming; dynamic job shop; dispatching rules; automatic discover

## 引言

调度是指以优化一个或多个性能指标为目标, 将有限资源分配给各种生产活动, 并将其视为一个

NP-hard 的组合优化问题。作业车间调度问题(Job Shop Scheduling Problem, JSSP)就是确定每道工序的排序问题, 即确定了车间里机器上排队等待加工工序的次序<sup>[1-3]</sup>。在大部分的作业车间调度问题的研究中, 都是使用启发式算法。使用启发式算法生成的调度方案, 只是给出了在特定问题下的一个解, 一旦初始条件改变, 调度方案也随之改变, 又需要生成另一个新的调度方案。但在实际生产过程中, 不同的性能指标和客户需求会导致作业车间环



收稿日期: 2020-04-29 修回日期: 2020-07-08;  
基金项目: 国家自然科学基金(61973138), 国家重点研发计划(2018YFB1701903);  
作者简介: 张苏雨(1996-), 女, 江苏苏州, 博士生, 研究方向为动态离散作业车间调度规则发现;  
王艳(1978-), 女, 江苏无锡, 博士, 教授, 博导, 研究方向为制造系统能效优化。

<http://www.china-simulation.com>

• 2494 •

境变得更复杂, 拥有许多动态变化, 具有不确定性, 这类问题即动态作业车间调度问题<sup>[4]</sup>(Dynamic Job Shop Scheduling Problem, DJSSP)。复杂工况下的动态车间的调度策略(Scheduling Strategy)研究是当前的研究热点和难点之一, 可以通过使用机器学习来解决这一困难, 尤其是遗传规划, 来创建特定于问题的调度算法。与以遗传算法(Genetic Algorithm, GA)为代表的启发式算法相比, 遗传规划<sup>[5]</sup>(Genetic Programming, GP)给出的是用于求解调度问题的规则, 而不是特定问题的特定调度方案。超启发式算法, 实际就是产生调度方案的算法, 可以通过调度规则得到不同调度问题的方案。

近年来, 利用调度规则(Dispatching Rule, DR)的自动启发式求解动态车间调度问题取得了一些研究进展。其算法的思路是<sup>[5]</sup>, 当一台机器可用时, 基于优先级的调度规则将检查等待的作业, 并自动选择接下来要处理的具有最高优先级的作业。调度规则主要可以分成以下 3 类<sup>[6]</sup>: (1) 单一调度规则(Simple Dispatching Rule, SDR), 主要基于与工件本身相关的信息; (2) 加权调度规则(Weighted Dispatching Rule, WDR), 主要基于多个工件的相关信息; (3) 复合调度规则(Composited Dispatching Rule, CDR)。调度规则无法保证为问题提供最优解<sup>[7]</sup>, 但其对动态环境的适应能力, 工程适用性强。在研究和实践中得到了广泛的应用。复合调度规则主要是单一调度规则的启发式组合, 旨在继承单个调度规则的优势。复合调度规则的研究成果较为丰富, 相关研究结果表明<sup>[8-9]</sup>, 合理的复合调度规则在调度性能上确实优于单一调度规则, 但大多数的规则都是面向静态作业车间, 未考虑车间的动态不确定性, 因此提出了针对不同生产场景的调度规则的演化生成。

Burke 等<sup>[10]</sup>团队提出了超启发式遗传规划算法(Genetic Programming based on Hyper-heuristic, GPHH)的框架, 为后续的调度和组合优化问题的发展打下了基础。Tay 等<sup>[6]</sup>利用遗传规划生成多目标

柔性作业车间的调度规则, 考虑了最大完工时间、平均迟到时间以及平均流动时间 3 个冲突目标, 利用权重线性组合。张国辉等<sup>[11]</sup>通过遗传规划对作业车间调度问题标准测试集的求解, 验证了该算法求解作业车间调度问题的有效性。范华丽等<sup>[12]</sup>针对平均加权迟到时间为调度目标, 研究分析了遗传规划算法自动发现调度规则的鲁棒性。

本文主要任务就是将最小化完工时间和最小化平均加权迟到时间作为调度优化目标, 研究基于超启发式遗传规划的动态作业车间调度规则的自动化发现。本文提出了在 GP 中融入 NSGA-II 的带精英策略的非支配排序方法来研究非支配调度规则的帕累托前沿。实验中, 将演化调度规则与已有的基准调度规则进行了仿真比较, 表明了调度规则是如何有效地解决动态作业车间调度问题。

## 1 动态作业车间调度问题

### 1.1 问题描述

在动态离散作业车间调度问题中<sup>[13]</sup>, 设有  $m$  台加工机器  $M=\{M_1, M_2, \dots, M_m\}$  和  $n$  个待加工工件  $J=\{J_1, J_2, \dots, J_n\}$ , 以服从某种分布形式随机到达加工车间, 各个工件都有其到达时间  $at_i$  (Arrival Time) 和交货期  $dd_i$  (Due Date)。工件  $J_i$  被随机指定一道或多道工序  $J_i=\{J_{i1}, J_{i2}, \dots, J_{in}\}$ , 其中  $n_i$  为工件  $J_i$  的工序数, 且服从某种离散分布形式。每个工件都按照一定的工艺路线在指定加工机器上进行加工, 第  $i$  个工件的第  $j$  道工序在机器  $M_k$  的加工时间为  $pt_{ijk}$ 。

根据实际生产条件和约束, 动态车间调度问题具有满足以下假设:

假设 1. 在任何给定的时间内, 每台机器最多只能处理一个工件的一个工序;

假设 2. 在任何给定的时间内, 每个工件的每道工序只能在一台机器上加工;

假设 3. 一旦一道工序在机器上开始处理, 机器将继续处理它, 直到完成。在此期间, 机器不能被其他操作中断;

假设 4. 作业车间中的所有机器都是持续可用的,可以在工序与工序之间空闲时使用它们来处理新工序;

假设 5. 工件数是确定的,但工件在到达作业车间之前的加工信息是完全未知的;

假设 6. 每个工件的交货期在作业执行期间预先确定且固定不变。

## 1.2 优化目标

调度就是满足问题假设约束条件的前提下,找到一个切实可行的计划,优化特定的性能指标。本文使用两个目标函数:最大完工时间、平均加权迟到时间。

最大完工时间<sup>[14]</sup>是调度问题中最常使用的优化目标,也就是最后一个工件的完成时间,提高最大完工时间,可以提高客户的满意度。

$$C_{\max} = \max \{C_i | i = 1, 2, \dots, n\} \quad (1)$$

式中:  $C_i$  为工件  $J_i$  的完成时间。

平均加权迟到时间<sup>[15]</sup>是目前动态作业车间调度问题中被广泛使用的优化目标,考虑了工件的重要程度,并满足了工件的交货期的约束。

$$T = \frac{\sum_{i=1}^n \omega_i \max \{C_i - dd_i, 0\}}{n} \quad (2)$$

式中:  $n$  为工件数;  $\omega_i$  为工件  $J_i$  的权重;  $C_i$  为工件  $J_i$  的完成时间;  $dd_i$  为工件  $J_i$  的交货期。

## 2 超启发遗传规划算法

遗传规划<sup>[16]</sup>同遗传算法一样,也是模仿生物进化的思想,随机产生初始种群,种群中的每个个体以树的形式表现,计算每个个体的适应度值,并进行比较,再经过复制、变异和交叉等遗传操作,对问题进行多次的迭代后,生成最优解或近似最优解。与启发式优化算法的不同在于,遗传规划的个体不是具体的解,而是启发式规则,即产生具体解的算法或规则。

在动态作业车间调度问题中,通过遗传算法和调度规则来产生调度方案,然后根据调度方案计算个体的适应度值。基于适应度值选择合适的个体作

为父代,再通过遗传操作算子繁殖子代。图 1 所示为求解动态作业车间排序调度规则的算法流程图。

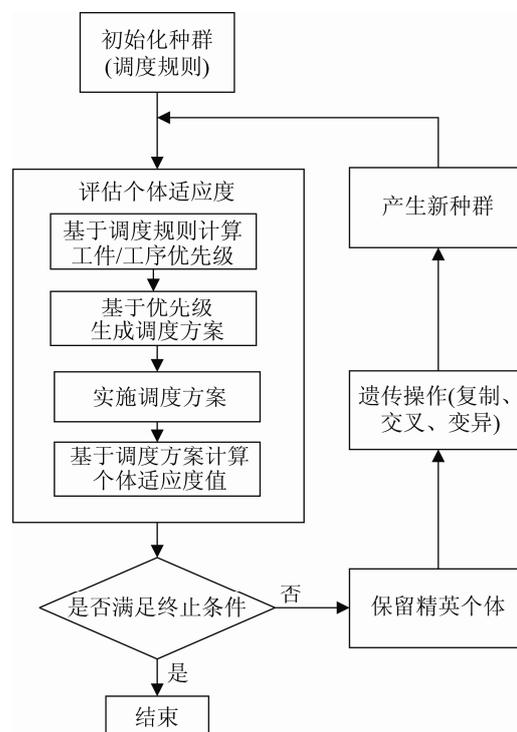


图 1 遗传规划求解动态作业车间调度规则算法流程图  
Fig. 1 Overall framework of Genetic Programming to solve Dispatching Rules of Dynamic Job Shop

### 2.1 调度规则的表达方式

基于遗传规划超启发式调度可以根据车间的状态和将要调度的机器的状态,学习新的调度规则,自适应地对队列中的操作进行排序。调度规则通常被描述为一个优先级函数,因此,优化调度规则等同于优化优先级函数。在遗传规划中,优先级函数可以用 GP 树来表示<sup>[17]</sup>,GP 树是由函数节点和终端叶节点组成。函数和终端的选择是整个优化过程中的关键步骤,因为它们包含了算法运行过程中所使用的相关信息并生成有效的解决方案。因此,在遗传规划算法解决具体问题时,合理的函数节点和终端叶节点可以缩小算法的搜索空间,提高了调度算法的效率。

在车间调度问题中,各种操作参数都会影响到调度结果质量,这些参数都可以成为一个调度规则。通常终端集是用来考虑作业车间属性的,在评

估了许多与 JSSP 相关的参数后, 本文 GP 算法中使用的终端如表 1 所示, 将车间全局和局部信息有效整合在一起, 从而做出排序决策。在动态作业车间调度问题中, 当一台机器处于空闲状态时, 工件到达该机器时, 立马进行加工处理。如果机器完成当前的加工工序, 而队列中有多个待加工的工序, 则根据基于调度规则的优先级, 选择优先级高的工件进行先加工。

表 1 遗传规划算法所使用的终止符集  
Tab. 1 Terminal set of Genetic Programming Algorithm

终端	含义
W (weight)	工件权重
DD (due-date)	工件的交货期
PT (processing-time)	工序的加工时间
NPT (next processing time)	下个工序加工时间
ORT (operation's ready time)	工序的准备时间
WKR (work_remaining)	工件剩余工作量
NOR (the number of operations remaining)	工件剩余工序数
OWT (the waiting time of the operation)	工序等待时间
rDD (the relative due date)	工件交货剩余时间
SL(slack)	工件宽裕时间

函数节点使用 {+, -, \*, /, max, min, if}, 其中“+”, “-”, “\*”, “/”为基本的算术运算符, “min”, “max”为接受两个参数并返回较小或较大的值, “if”表示为在 {a,b,c} 中, 若 a>0, 则输出值为 b, 若 a<0, 则输出 c。

图 2 所示为“2PT+WINQ+NPT”调度规则的 GP 树, 其 GP 树的表达形式为(+ (\* 2 PT) (+ WINQ NPT))。

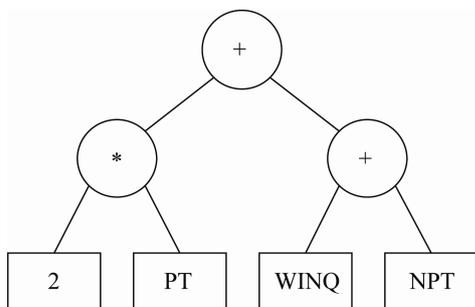


图 2 “2PT+WINQ+NPT”的 GP 树  
Fig. 2 GP Tree of 2PT+WINQ+NPT

## 2.2 种群的初始化

在遗传规划中, 我们通常事先不知道(或不希望指定)解决方案的大小和形状。因此, 第一代, 种群随机初始化, 采用了 Ramped-half-and-half 的方法<sup>[14]</sup>, 就是实行完全生法(full)和成长生成法(grow)相结合的混合生成方式。full 方法, 是给定深度来创建 GP 树, 从树的根到端点的所有路径都是相同的长度, 因此具有相同的形状。相反, grow 方法为给定深度值来创建的 GP 树, 从树的根到端点的路径深度都不会大于给定的深度值。因此, 这些 GP 树在形状上有很大的差异。假设给定 GP 树最大深度为 3, 从函数集 {+, -, ×, /} 和表 1 所示的终端集中, 随机生成初始化种群。图 3 和图 4 所示分别为 full 方法和 grow 方法。Full 方法生成的种群, 个体深度为定值, 都为 3; 而 Grow 方法随机生成深度为 1~3 之间的 GP 树。这样 grow 和 full 混合的方法, 使得 GP 树有各种各样的大小和形状, 可以保持了种群的多样性。

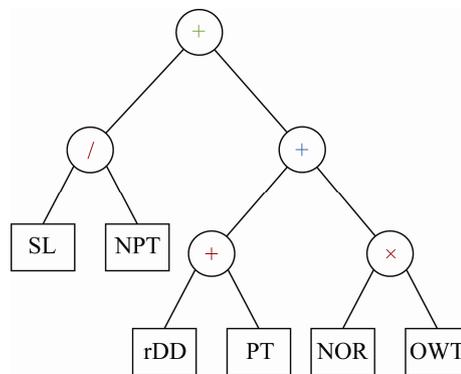
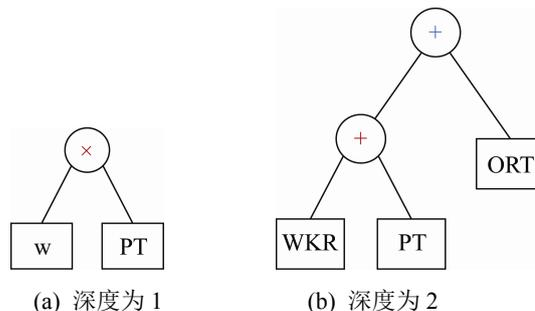
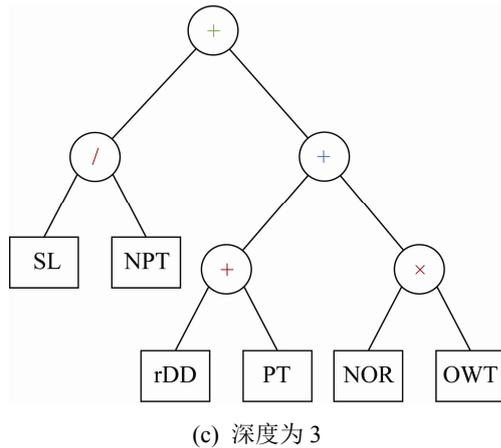


图 3 Full 法生成 GP 树  
Fig. 3 Generating GP tree by full method





(c) 深度为 3

图 4 Grow 法生成 GP 树

Fig. 4 Generating GP tree by growing method

种群初始化个体包括 2 个部分, 分别为根和子节点。初始化种群的过程为: 从函数集随机选择 GP 树的根和延伸的子节点; 若子节点深度等于最大给定深度, 使用完全生长法从终端集中为 GP 树选择终端; 若子节点深度小于最大给定深度, 使用成长生长法随机选择继续延伸子节点还是从终端集中选择终端, 终止个体生长。

### 2.3 适应度评价

无论是遗传规划还是遗传算法, 适应度函数始终是算法的驱动力, 以个体适应度的大小来判断个体的优劣程度, 决定其是否可以遗传操作。本文为了能够简单明了的观察种群个体的性能, 直接将最大完工时间和平均加权迟到时间这两个优化目标作为适应度函数, 适应度值越小, 个体的性能就越好, 有更大的可能性进行遗传操作, 传给下一代。GP 进化后, 遍历个体的 GP 树调度规则, 机器选择待加工工件队列中优先级最高的工件进行加工, 产生合适的调度方案, 基于调度方案计算个体的适应度值<sup>[18]</sup>。

对于动态车间来说, 不能通过单一的优化目标来评价车间的调度性能, 为了权衡两个调度目标, 本文在 GP 算法中采用了 NSGA-II 的带精英策略的非支配排序的方法<sup>[19]</sup>, 求得两个优化目标的非支配解集。精英策略, 将上一代中适应度值最高的个体不经过遗传操作保留到下一代中, 来替代下一代

中经过一系列遗传操作适应度值最低的个体。其中, 非支配可以解释为, 对于  $m$  个目标分量  $f_i(i=1,2,\dots,m)$ , 任意给定两个决策变量  $x_1, x_2$ , 若满足以下两个条件, 则  $x_1$  支配  $x_2$ 。

(1) 当且仅当, 对于  $\forall i \in \{1,2,\dots,m\}$ , 都  $f(x_1) < f(x_2)$ , 则  $x_1$  支配  $x_2$ ;

(2) 当且仅当, 对于  $\forall i \in \{1,2,\dots,m\}$ , 有  $f(x_1) \leq f(x_2)$ , 且至少存在一个  $j \in \{1,2,\dots,m\}$ , 使  $f(x_1) < f(x_2)$ , 则  $x_1$  弱支配  $x_2$ 。

本文使用非支配排序方法计算种群中各个 GP 树个体的非支配等级, 再通过拥挤度计算非支配等级相同的个体的优先级关系, 来得到每代的非支配解集。

### 2.4 遗传操作算子

在 GP 中, 遗传操作主要分为 3 步: 复制、交叉和变异。在动态车间中, 仿真的工件数量十分庞大, 因此种群规模也非常巨大。通过采用锦标赛选择法, 在庞大的种群中选择合适的个体作为父代, 进入下一步的遗传操作, 适应度值并不突出的个体也有机会被选中, 保持了种群的多样性<sup>[20]</sup>。

交叉操作如图 5 所示, 随机选择两个父代 GP 树, 随机选择其中的子节点, 进行交叉操作形成子代。

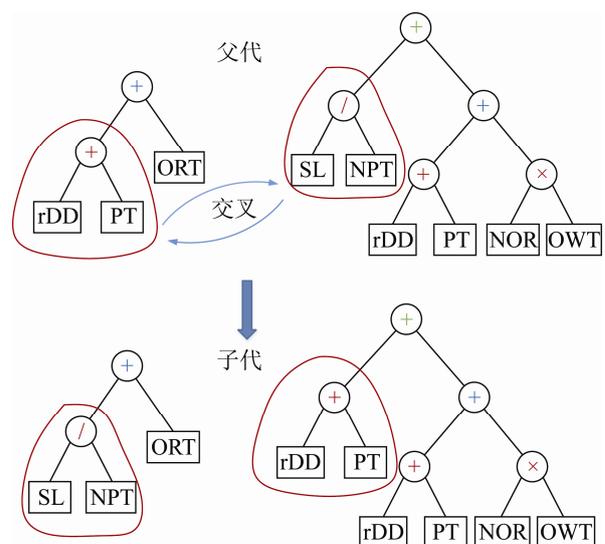


图 5 GP 树的交叉操作

Fig. 5 Crossing of GP tree

变异操作如图 6 所示, 在父代中随机选择一个子节点作为突变点, 随机生成一个子树去替换父代中以突变点作为根节点的子树。此外, 精英保留策略<sup>[15]</sup>可以保证种群中最优的个体可以被保留到下一代中。在 GP 中, 给遗传操作算子分配一定的复制率、交叉率以及变异率, 来控制对父代的遗传操作。

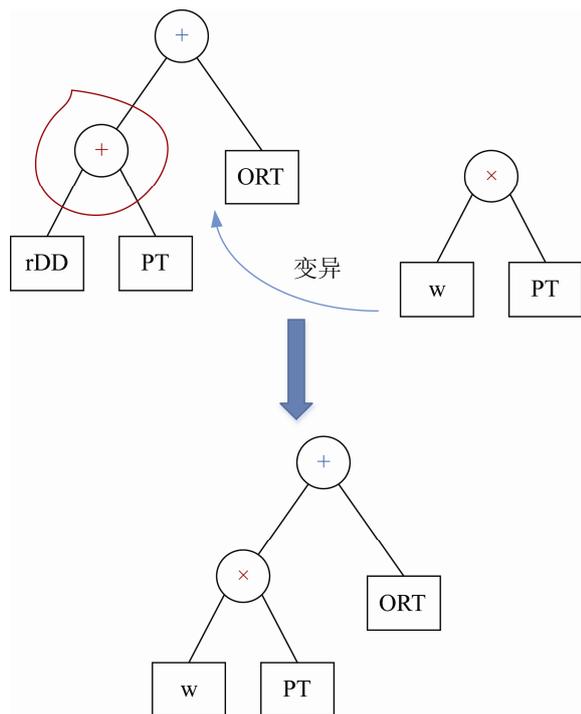


图 6 GP 树的变异操作  
Fig. 6 Mutation of GP tree

### 3 仿真实验设计

#### 3.1 仿真环境构建

动态作业车间调度问题仿真实验具体参数设定如表 2 所示, 车间内有 10 台机器, 工件数量为 5 000 个, 1 000 个预热工件<sup>[21]</sup>, 每个工件服从泊松分布随机到到达车间, 且每个工件的工序数是服从正态分布  $U(1,10)$ , 每个工序的加工时间也服从正态分布  $U(1,99)$ 。在每一次的仿真中, 每个工件都有不同的权重, 符合 1:2:4 的原则<sup>[22]</sup>, 分别表示没有那么重要的工件、重要的工件和非常重要的工件。

表 2 动态作业车间调度问题仿真模型参数

Tab. 2 Parameter of simulation model for Dynamic Job Shop Scheduling Problem

参数	设定
机器个数	10
工件个数	5 000
预热工件个数	1 000
工件的工序数机器分布	$U(1,10)$
工件达到时间及其分布	泊松分布
工件的加工时间及其分布	$U(1,99)$
工件的权重	1, 2, 4
机器的利用率	85%, 95%
交货期宽裕度系数	2, 4, 6

机器利用率和交货期宽裕度系数是影响动态作业车间调度规则性能的重要因素<sup>[23]</sup>。其中, 机器利用率就是机器忙于处理工序操作的时间比例 (机器利用率=机器工作时间/机器开机时间), 设定机器的利用率可以有效防止机器堵塞的现象, 从而提高调度规则的性能并提高算法的搜索和计算的效率。为了有效平衡每台机器的加工状态, 实验设定工序在任何一台机器上处理的概率都是相同的<sup>[23]</sup>, 但是连续的工序不在同一台机器上处理。

每个工件的交货期被定义为<sup>[24]</sup>:

$$dd_i = ar_i + \alpha \times \sum_i \sum_j \sum_k pt_{ijk} \quad (3)$$

式中:  $at_i$  为第  $i$  个工件的到达时间;  $pt_{ijk}$  为第  $i$  个工件的第  $j$  道工序在机器  $M_k$  的加工时间;  $\alpha$  为交货期宽裕度系数, 交货期宽裕度系数越大, 交货期越紧张<sup>[25]</sup>。

在实验中, 采用 2 种不同机器利用率(85%, 95%)和 3 种不同的交货期宽度系数(2,4,6), 分别代表不太紧急工件、紧急工件和非常紧急工件<sup>[26-27]</sup>, 从而产生了  $2 \times 3 = 6$  种不同的生产情景, 并将其表示为 <机器利用率, 交货期宽裕度系数>。将上述 6 种生产情景, 重复模拟仿真 30 次, 得到  $6 \times 30 = 180$  次重复, 对调度规则进行综合评估。考虑到车间状态的稳定性和仿真实验的有效性, 对于每一次的仿真的开始, 需要 1 000 个工件进行预热, 然后对接下来的 5 000 个工件进行性能评估统计。

### 3.2 GP 算法参数设计

GP 算法仿真实验中所使用的具体参数设置如表 3 所示。该实验使用 Ramped half and half 的方法来产生初始化种群<sup>[28]</sup>，并将其分成了 2 部分。其中，50%是随机生成的最大深度树，其深度为 7；另外的 50%则产生深度从 1~7 不等的 GP 树。该实验还通过复制操作、交叉操作和变异操作来保持种群的多样性<sup>[29]</sup>。

表 3 遗传规划算法的参数设定  
Tab. 3 Parameter Setting of Genetic Programming

参数	设定
种群大小	1 024
遗传代数	60
初始化方法	Ramped-half-and-half
个体的最大深度	6
复制率	5%
变异率	10%
交叉率	85%
选择方法	锦标赛选择法
精英保留个体	10

## 4 仿真结果分析

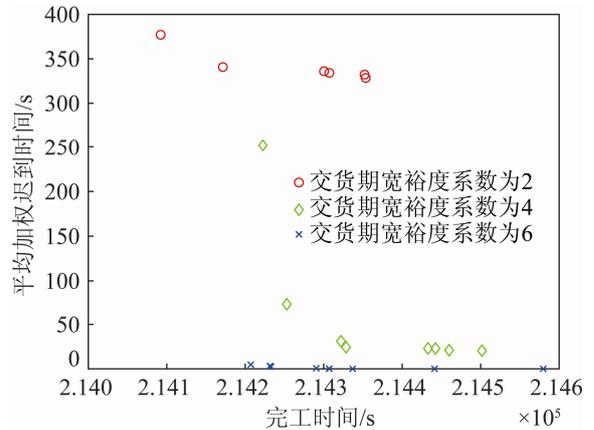
考虑启发式遗传算法的运算需求，仿真运行环境为：Windows 10，Intel(R) Core(TM) i7-7500U CPU 2.90 GHz 处理器，16 G 内存；编程语言为：java 语言；开发环境：Eclipse IDE for Java Developers -2019-12。

### 4.1 模型参数对调度规则的影响

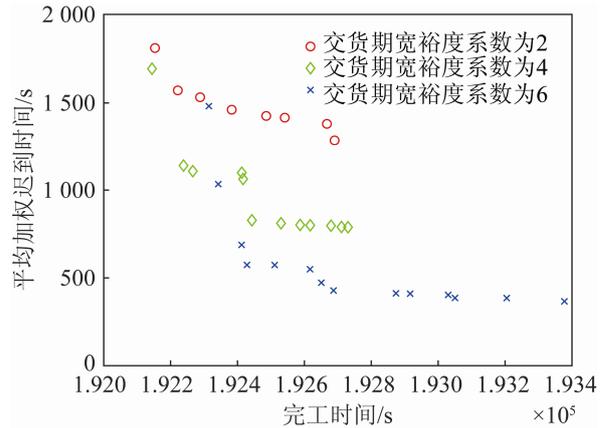
为了分析研究模型参数对调度规则的影响，实验设定如表 2 的动态作业车间调度问题仿真模型参数设定所示。通过对不同的机器利用率和不同的交货期宽裕度系数进行仿真，该实验遗传规划算法在 6 种不同的生产环境下独立运行了 6 次所得到的非支配解集。

从图 7 可以看出，当机器利用率一定，交货期宽裕度系数增大时，算法所得到的调度规则的平均加权迟到时间也会随之降低，而完工时间的变化不

大。因为完工时间与交货期宽裕度系数的大小无关。然而，交货期宽裕度系数越大，平均加权迟到时间越小，这是因为交货期宽裕度系数越大，工件越紧急，调度规则在交货期相关性能方面会更好。



(a) 机器利用率为 85%



(b) 机器利用率为 95%

图 7 遗传规划算法在不同生产情景下的非支配解集  
Fig. 7 Non-Dominated set of Genetic Programming in different scenarios

当交货期宽裕系数一定，机器利用率增大时，算法所得到的调度规则的完工时间会随之降低，然而，平均加权迟到时间的变化并不明显。这是由于机器利用率越高，机器加工越频繁，空闲时间越少，从而完工时间就会提前。

### 4.2 演化调度规则与基准调度规则对比

选择<85%，4>生产情景下的演化调度规则与表 4 所示的基准调度规则进行实验仿真分析。表

4 的上半部分是已有文献中性能良好的简单调度规则<sup>[6]</sup>, 下半部分是已有文献中性能良好的手工设置的复合调度规则<sup>[5]</sup>。本文将<85%, 4>的非支配解集按照最大完工时间进行升序排序, 选择第一个、最后一个和中间一个, 分别代表偏向最大完工时间、偏向平均加权迟到时间和同时考虑 2 个目标<sup>[30-31]</sup>。

表 4 基准调度规则  
Tab. 4 Benchmark Dispatching Rules

基准调度规则	含义
FIFO	先进先出
EDD	最早交货期
SPT	最短加工时间
SL	所剩宽裕时间
ct+PT	当前时间+加工时间
SPT+PW+FDD	最小加工时间+工件剩余时间+ 最晚交货期
max {SL, RNOs}	max {所剩宽裕时间, 工件剩余工序数}

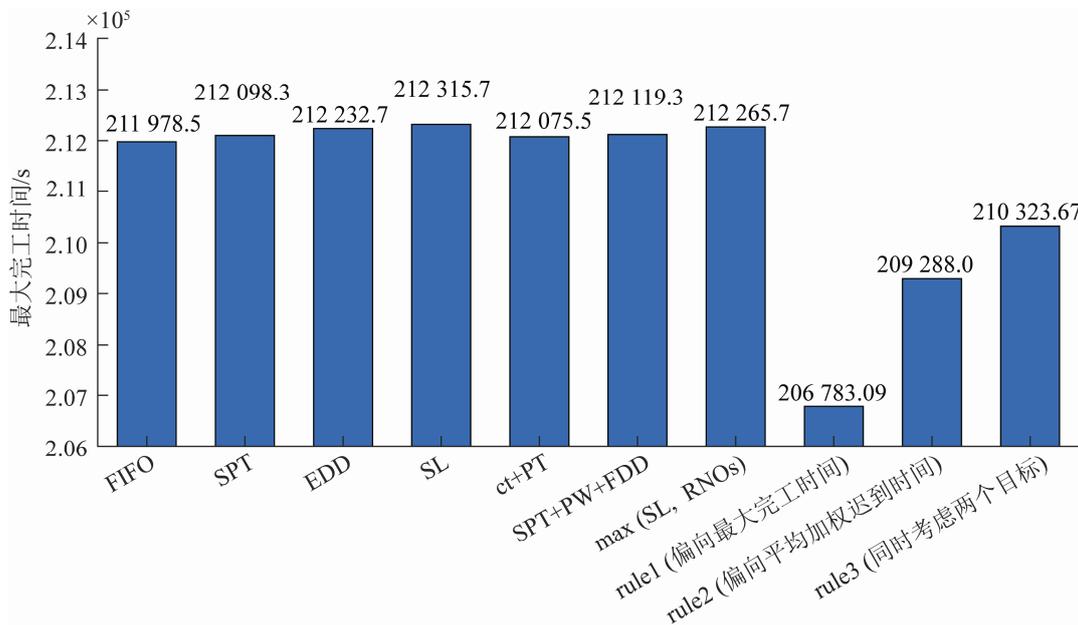


图 8 在<85%, 4>下调度规则的最大完工时间  
Fig. 8 Makespan of the rules in the <85%, 4>

图 9 为不同调度规则在<85%, 4>下的平均加权迟到时间, 横坐标为各个调度规则, 纵坐标为平均加权迟到时间。

图 9 可以发现, 在优化平均加权迟到时间时,

图 8 为不同调度规则在<85%, 4>下最大完工时间, 横坐标为各个调度规则, 纵坐标为最大完工时间。

从图 8 中发现, 在已有的 4 个简单调度规则中, FIFO 的性能最好, SPT、EDD 和 SL 次之。由于 FIFO 是选取队列中到达时间最早的工件。较早到达的工件比较晚到达的工件更有可能被选中进行处理。因此, 缩短了完工时间。复合调度规则“ct+PT”的完工时间仅次于 FIFO, 其考虑了队列中工序的完成时间, 选择队列中能够最先完成的工序优先加工, 缩短了完工时间。相比于简单调度规则与人工复合调度规则, 用 GP 演化的调度规则 rule1 和 rule2 有明显的优势, 极大程度的降低了完工时间, 节约了成本。这说明用遗传规划算法演化调度规则的方法是切实可行的且有效的。

FIFO 的性能相对较弱。因为先进先出的原则无法确保工件的交货期。SL 在最小化平均加权迟到时间方面比 FIFO 有更好的结果, 但与其他规则相比, 其结果仍然较差。SPT 的性能是最好的, 因

为它挑选队列中加工时间最短的工序优先加工，降低了迟到时间。复合调度规则“ct+PT”在最小化平均加权迟到时间方面性能仍然良好。从图 9 中明显可以看出，GP 演化出来的 rule2 和 rule3 的性能优于其余的规则，极大程度上降低了平均加权迟到时间。

表 5~6 为分别在 <85%, 4> 和 <95%, 4> 下遗传

规划演化出来的针对不同目标的调度规则。

图 10~11 为不同调度规则分别在 <85%, 4> 和 <95%, 4> 下的性能指标。

实验结果表明，针对不同的生产情景，遗传规划算法演化调度规则是有效的，能够大幅度的提高调度规则的性能，降低作业车间的生产成本。

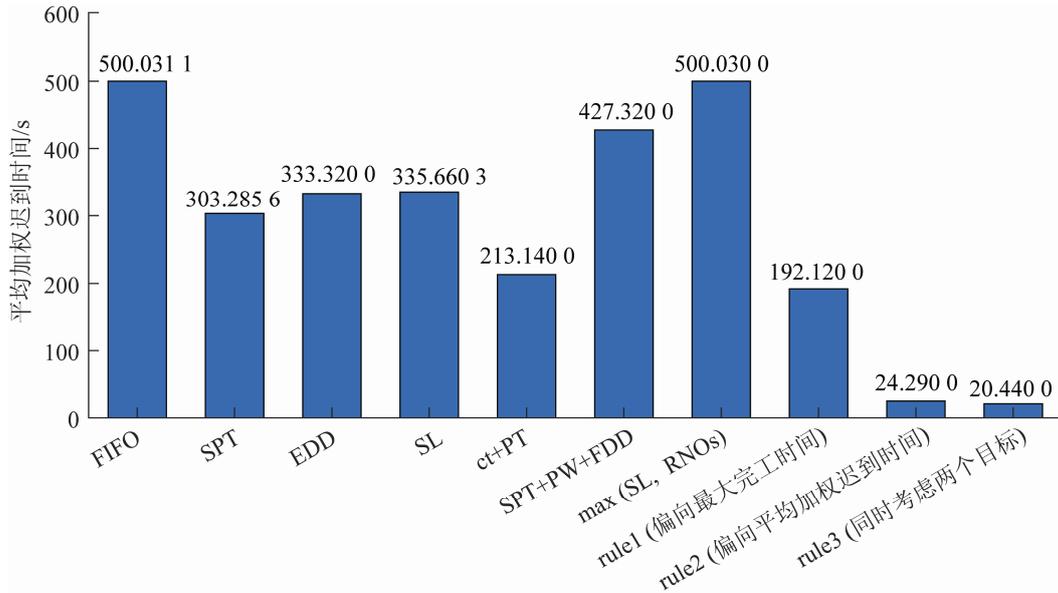


图 9 在 <85%, 4> 下调度规则的平均加权迟到时间  
Fig. 9 Mean Weighted Tardiness of rules in <85%, 4>

表 5 <85%, 4> 的 3 个演化调度规则  
Tab. 5 Three Dispatching Rules of <85%, 4>

优化目标	调度规则
rule1 (偏向最大完工时间)	$rDD/WKR-NPT$
rule2 (偏向平均加权迟到时间)	$\max\{PT \times WKR (PT - W), rDD/WKR, \min\{PT, OWT\}\}$
rule3 (同时考虑两个目标)	$\max\{\min\{NPT + WKR, NOR + SL\}, PT \min\{rDD/WKR, WKR\}, PT^2 rDD/WKR, WKR\}$

表 6 <95%, 4> 的 3 个演化调度规则  
Tab. 6 Three Dispatching Rules of <95%, 4>

优化目标	调度规则
rule1 (偏向最大完工时间)	$PT \max\{NOR, PT, NPT/PT, (W + NOR)SL/WKR\} + \min\{WKR - NOR \times rDD, (W + PT)SL/2(PT + NOR)\}$
rule2 (偏向平均加权迟到时间)	$PT + NOR + \max\{2PT \times SL / \max\{PT, NOR \times rDD\}, \min\{\max\{NOR, PT\} / \max\{PT, NOR \times rDD\}, NOR\}\}$
rule3 (同时考虑两个目标)	$PT + NOR + \max\{(2PT + NOR) \times SL / \max\{PT, NOR \times rDD\}, W / (NOR + PT)\}$

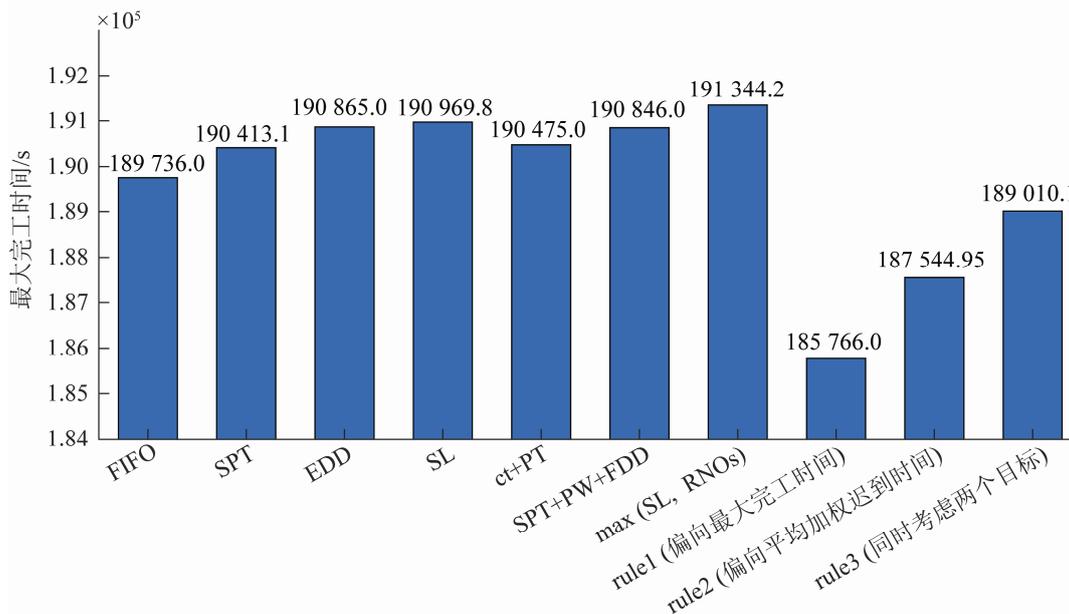


图 10 在<95%, 4>下调度规则的最大完工时间  
Fig. 10 Makespan of rules in <95%, 4>

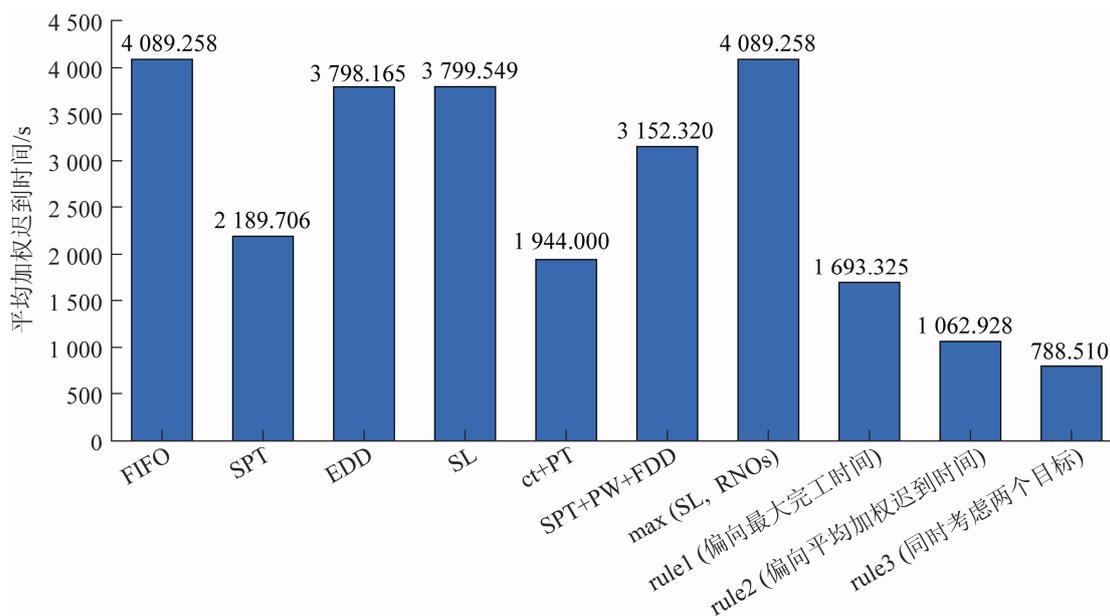


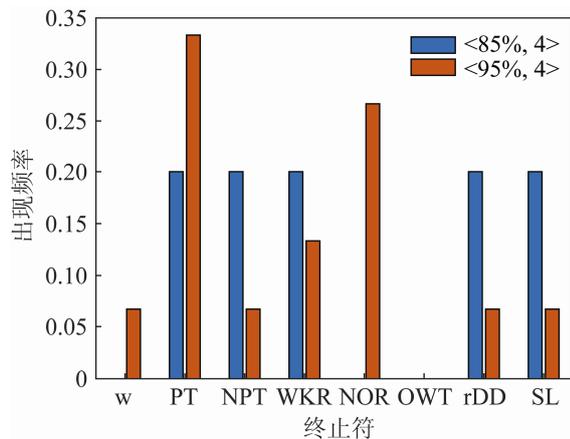
图 11 在<95%, 4>下调度规则的平均加权迟到时间  
Fig. 11 Mean Weighted Tardiness of rules in <95%, 4>

### 4.3 演化调度规则语义分析

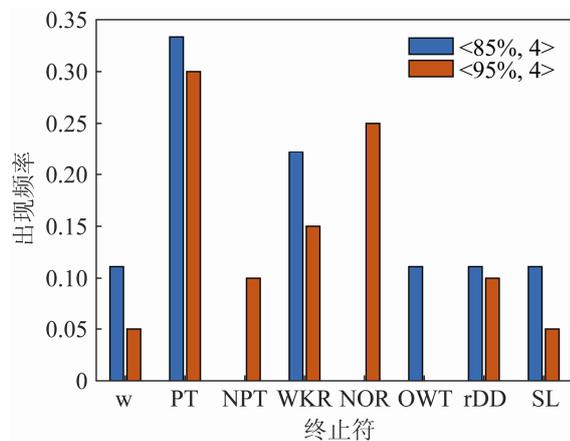
为了进一步分析调度规则的工作机理, 仿真独立运行 30 次后, 计算终止符出现频率的平均值, 并对调度规则进行语义分析。如图 12 所示, 考虑不同目标偏向规则的终端属性出现频率。Rule1 拥有三者中最小的最大完工时间和最大的平均加权

迟到时间, 更倾向于优化最大完工时间。从 rule1 的语义中可以看出, PT 的使用频率很高, 对工件的优先级计算产生重要的影响。NOR 和 WKR 的出现频率较高, 机器偏向于选择加工时间短, 剩余工序数少且剩余的工作量少的工件优先加工。简单来说, 机器偏向于选择能尽快完成的工件优先加

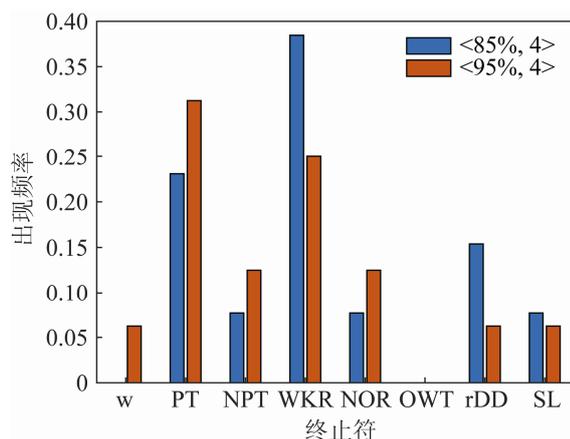
大, 符合缩短完工时间的逻辑。



(a) 偏向最大完工时间的 rule



(b) 偏向平均加权迟到时间的 rule



(c) 同时考虑两个目标的 rule

图 12 考虑不同目标偏向规则的终止符出现频率

Fig. 12 Frequency of terminator with different targets

Rule2 拥有三者中最大的最大完工时间和最小的平均加权迟到时间, 更倾向于优化平均加权

迟到时间。从 Rule2 语义中可以看出, PT 对于较小迟到时间同样重要, 机器选择加工时间短的工件优先加工, 可以尽可能的缩短工件的完成时间, 减小工件迟到的可能性。rDD 的出现频率较高, 机器优先选择 rDD 小的工件, 也就是说距离交货期越近的工件越优先加工, 大大降低了工件的迟到时间。

Rule3 同时考虑了最大完工时间和平均加权迟到时间, 在两个优化目标之间取得了平衡。机器更倾向于选择 PT 小, rDD 小, SL 大的工件优先加工。

## 5 结论

本文基于超启发式遗传规划算法求解动态作业车间调度问题, 针对不同生产情景自动演化调度规则, 考虑车间里机器队列中工件加工的优先级, 优化目标为最大完工时间和平均加权迟到时间。本文详细介绍了超启发式遗传规划算法自动演化调度规则的框架, 并给出了仿真实验具体的参数设定。实验结果表明了通过 GP 演化的调度规则比已有的人工设置的调度规则在优化性能上有着明显的优势。GP 能够针对特定的生产情景, 自动演化出合适的调度规则, 并不是一成不变的。本文还分析研究了模型参数(机器利用率和交货期宽裕度系数)对调度规则生成的影响。最后本文对调度规则进行了语义分析, 分析终端属性针对不同优化目标倾向的调度规则的作用。

本文只考虑了机器队列上工件的排序选择, 并没有考虑到工件对机器的路径选择。在今后的研究中, 将从路径与排序两个方面来研究遗传规划求解动态作业车间的调度问题, 进一步优化遗传规划算法, 与实际作业车间相结合, 提高解决实际问题的效率。

## 参考文献:

- [1] Pickardt C W, Hildebrandt T, Branke J, et al. Evolutionary Generation of Dispatching Rule Sets for Complex Dynamic Scheduling Problems[J]. International

- Journal of Production Economics (S0925-5273), 2013, 145(1): 67-77.
- [2] Lourenco H R. Job-shop Scheduling: Computational Study of Local Search and Large-Step Optimization Methods[J]. *European Journal of Operational Research* (S0377-2217), 1995, 83(2): 347-364.
- [3] Balas E, Simonetti N, Vazacopoulos A. Job Shop Scheduling with Setup Times, Deadlines and Precedence Constraints[J]. *Journal of Scheduling* (S1094-6136), 2008, 11(4): 253-262.
- [4] Kundakci N, Kulak O. Hybrid Genetic Algorithms for Minimizing Makespan in Dynamic Job Shop Scheduling Problem[J]. *Computers & Industrial Engineering* (S0360-8352), 2016, 96(6): 31-51.
- [5] Branke J, Nguyen S, Pickardt C W, et al. Automated Design of Production Scheduling Heuristics: A Review[J]. *IEEE Transactions on Evolutionary Computation* (S1089-778X), 2016, 20(1): 110-124.
- [6] Tay J C, Ho N B. Evolving Dispatching Rules Using Genetic Programming for Solving Multi-Objective Flexible Job-shop Problems[J]. *Computers & Industrial Engineering* (S0360-8352), 2008, 54(3): 453-473.
- [7] Fan H L, Xiong H G, Jiang G Z, et al. Generation of Dispatching Rules for Dynamic Job Shop Scheduling Problem Based on Genetic Programming Algorithm[J]. *Computer Integrated Manufacturing Systems* (S1006-5911), 2018, 24(4): 876-885.
- [8] Jayamohan M S, Rajendran C. Development and Analysis of Cost-Based Dispatching Rules for Job Shop Scheduling[J]. *European Journal of Operational Research* (S0377-2217), 2004, 157(2): 307-321.
- [9] Kanet J J, Li X. A Weighted Modified Due Date Rule for Sequencing to Minimize Weighted Tardiness[J]. *Journal of Scheduling* (S1094-6136), 2004, 7(4): 261-276.
- [10] Burke E K, Gendreau M, Hyde M, et al. Hyper-heuristics: a Survey of the State of the Art[J]. *Journal of the Operational Research Society* (S0160-5682), 2013, 64(12): 1695-1724.
- [11] 张国辉, 高亮, 李培根. 基于遗传规划的作业车间调度算法研究[J]. *控制与决策*, 2008(8): 924-928.  
Zhang Guohui, Gao Liang, Li Peigen. Job-Shop Scheduling Based on Genetic Programming Algorithm[J]. *Control and Decision*, 2008(8): 924-928.
- [12] 范华丽, 熊禾根, 蒋国璋, 等. 基于遗传规划的动态作业车间调度规则生成[J]. *计算机集成制造系统*, 2018, 24(4): 876-885.  
Fan Huali, Xiong Hegen, Jiang Guozhang, et al. Generation of Dispatching Rules for Dynamic Job Shop Scheduling Problem Based on Genetic Programming Algorithm[J]. *Computer Integrated Manufacturing Systems*, 2018, 24(4): 876-885.
- [13] Nguyen S, Zhang M J, Johnston M, et al. Genetic Programming for Job Shop Scheduling[M]. Switzerland: Evolutionary and Swarm Intelligence Algorithms, 2018.
- [14] Nie L, Shao X, Gao L, et al. Evolving Scheduling Rules with Gene Expression Programming for Dynamic Single-machine Scheduling Problems[J]. *The International Journal of Advanced Manufacturing Technology* (S0268-3768), 2010, 50(5/8): 729-747.
- [15] Zhou H, Cheung W, Leung L C. Minimizing Weighted Tardiness of Job-Shop Scheduling Using a Hybrid Genetic Algorithm[J]. *European Journal of Operational Research* (S0377-2217), 2009, 194(3): 637-649.
- [16] Koza J R. Genetic Programming: on the Programming of Computers by Means of Natural Selection[M]. Cambridge: MIT Press, 1992.
- [17] Mei Y, Nguyen S, Xue B, et al. An Efficient Feature Selection Algorithm for Evolving Job Shop Scheduling Rules with Genetic Programming[J]. *IEEE Transactions on Emerging Topics in Computational Intelligence* (S2471-285X), 2017, 1(5): 339-353.
- [18] Zitzler E, Thiele L. Multi-objective Evolutionary Algorithms: a Comparative Case Study and the Strength Pareto Approach[J]. *IEEE Transactions on Evolutionary Computation* (S1089-778X), 1999, 3(4): 257-271.
- [19] Deb K, Pratap A, Agarwal S, et al. A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II[J]. *IEEE Transactions on Evolutionary Computation* (S1089-778X), 2002, 6(2): 182-197.
- [20] Kanet J J, Li X. A Weighted Modified Due Date Rule for Sequencing to Minimize Weighted Tardiness[J]. *Journal of Scheduling* (S1094-6136), 2004, 7(4): 261-276.
- [21] Wu C C, Liu S C, Zhao C, et al. A Multi-machine Order Scheduling with Learning Using the Genetic Algorithm and Particle Swarm Optimization[J]. *The Computer Journal* (S0010-4620), 2018, 61(1): 14-31.
- [22] Nguyen S, Zhang M J, Johnston M, et al. A Computational Study of Representations in Genetic Programming to Evolve Dispatching Rules for the Job Shop Scheduling Problem[J]. *IEEE Transactions on Evolutionary Computation* (S1089-778X), 2013, 17(5): 621-639.
- [23] Nguyen S, Zhang M J, Johnston M, et al. Automatic Design of Scheduling Policies for Dynamic

- Multi-objective Job Shop Scheduling via Cooperative Coevolution Genetic Programming[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2014, 18(2): 193-208.
- [24] Branke J, Nguyen S, Pickardt C W, et al. Automated Design of Production Scheduling Heuristics: a Review[J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2015, 20(1): 110-124.
- [25] Nguyen S, Zhang M J, Johnston M, et al. Automatic Programming via Iterated Local Search for Dynamic Job Shop Scheduling[J]. IEEE Transactions on Cybernetics (S2168-2267), 2015, 45(1): 1-14.
- [26] Jamrus T, Chien C F, Gen M, et al. Hybrid Particle Swarm Optimization Combined With Genetic Operators for Flexible Job-Shop Scheduling Under Uncertain Processing Time for Semiconductor Manufacturing[J]. IEEE Transactions on Semiconductor Manufacturing (S0894-6507), 2018, 31(1): 32-41.
- [27] 曹东风, 王艳. 基于组合权重的制造系统能效贝叶斯评价方法[J]. 系统仿真学报, 2018, 30(11): 290-299.  
Cao Dongfeng, Wang Yan. Bayesian Evaluation Method for Energy Efficiency of Manufacturing System Based on Combined Weights[J]. Journal of System Simulation, 2018, 30(11): 290-299.
- [28] Nguyen S, Mei Y, et al. A Hybrid Genetic Programming Algorithm for Automated Design of Dispatching Rules[J]. Evolutionary Computation (S1063-6560), 2018, 27(3): 367-496.
- [29] Hildebrandt T, Branke J, Jürgen. On Using Surrogates with Genetic Programming[J]. Evolutionary Computation (S1063-6560), 2015, 23(3): 343-367.
- [30] Zhou Y, Yang J J, Zheng L Y. Hyper-Heuristic Coevolution of Machine Assignment and Job Sequencing Rules for Multi-Objective Dynamic Flexible Job Shop Scheduling[J]. IEEE Access (S2169-3536), 2018, 11(7): 68-88.
- [31] Zhang F F, Mei Y, Zhang M J. A Two-Stage Genetic Programming Hyper-Heuristic Approach with Feature Selection for Dynamic Flexible Job Shop Scheduling[C]. Genetic & Evolutionary Computation Conference. Czech Republic: Association for Computing Machinery, 2019.