



ELSEVIER

Contents lists available at ScienceDirect

Engineering

journal homepage: [www.elsevier.com/locate/eng](http://www.elsevier.com/locate/eng)



Research  
Intelligent Manufacturing—Article

## 基于网络图拓扑结构的 MILP 模型求解智能制造系统中的工艺规划问题

刘齐浩, 李新宇\*, 高亮

State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China

### ARTICLE INFO

#### Article history:

Received 29 October 2019

Revised 21 March 2021

Accepted 27 April 2021

Available online 5 May 2021

#### 关键词

智能工艺规划

工艺网络图

混合整数线性规划

CPLEX

### 摘要

智能工艺规划是智能制造系统中的重要组成部分。从制造流程角度出发, 工艺规划 (process planning, PP) 连接着产品设计和实际生产, 有着承上启下的关键作用。PP 属于非确定性多项式时间困难 (NP-hard) 问题, 现有的问题模型都是非线性形式, 因此不能够通过求解现有模型来得到问题的精确解。从工艺网络图的拓扑结构出发, 本文提出了一个全新的混合整数线性规划 (mixed-integer linear programming, MILP) 数学模型, 并通过三种优先关系矩阵讨论了网络图中工序的优先关系。该模型能够凭借常用的数学模型求解器, 如 CPLEX、Gurobi 等, 来搜寻并获得大部分算例的最优解。该模型通过在 5 组公开的著名数据集上的测试, 证明了其通用性和有效性。实验结果有力地说明了所提模型能够有效地解决工艺规划问题, 并获得比当前最先进算法更好的解。

© 2021 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

## 1. 引言

智能制造包含智能制造技术和智能制造系统[1–3]。智能工艺规划 (process planning, PP) 属于重要的智能制造技术, 在智能制造系统中扮演着重要的角色[4,5]。PP 能够有效地缩短生产周期, 提高产品质量, 并进一步降低资源和能源的消耗[6]。因此, PP 在工业应用与研究中获得了较多的关注。然而, 因为 PP 问题内部存在着大量的柔性, 导致在实际生产场景中很难得到最优的工艺规划路线[7]。

解决传统的 PP 问题通常需要三个步骤: 工艺选择、资源分配和工序排序[5]。通过第一步的工艺选择来确定不同加工特征所需要的工艺及工序, 并基本确定工艺

路线的总长度。然后, 通过分配机器、刀具等资源, 来确定各工序的加工资源分配, 预获取相关加工参数, 例如加工时间等。最关键的工序排序一般是最后执行, 且排序过程必须遵循该工件工艺网络图中的优先关系约束[8]。例如, 粗铣工序和粗磨工序必须安排在精铣工序和精磨工序之前执行, 镗孔工序必须排在钻工工序后[9]。

PP 问题已经被证明是非确定性多项式时间困难 (nondeterministic polynomial-time-hard, NP-hard) 问题[6,10], 这也意味着仅仅依靠传统的梯度下降法、图论法和仿真方法很难实现 PP 问题的求解。因此, 多数研究者试图引入元启发式算法来求解 PP 问题[11]。当前主要的求解方法包括遗传算法 (genetic algorithm, GA)[7,12]、禁忌搜索算法 (tabu search, TS)[13]、粒子群优

\* Corresponding author.

E-mail address: [lixinyu@mail.hust.edu.cn](mailto:lixinyu@mail.hust.edu.cn) (X. Li).

化算法 (particle swarm optimization, PSO) [8]、蚁群优化算法 (ant colony optimization, ACO) [14] 和蜜蜂配合优化算法 (honey-bees mating optimization, HBMO) [10]。这些算法能够在较短时间内获得近优解, 更侧重算法的求解效率而不是解的最优性。

在此研究背景下, 大部分公开PP问题的最优解至今还未找到[8,15]。其中一个重要的原因是当前研究更多关注的是如何提高智能算法的性能, 而不是去研究问题的数学模型, 尤其是混合整数线性规划 (mixed-integer linear programming, MILP) 模型[14]。有效的MILP模型能够通过数学规划求解器求解, 如CPLEX、Gurobi等, 并在一定条件下稳定地获得最优解。但是, 模型的求解效率与问题的规模相关, 随着问题规模的逐渐增大, 模型的求解效率会迅速降低, 甚至在相当长的计算时间内无法获得可行解[16]。这也是当前PP的研究重点不在模型上的重要原因。此外, 当前PP的MILP研究还处于一种“妥协”状态, 即当前模型都是基于工艺特征建立的[8,14,17]。这种类型的模型通过将原始工艺图转换成工艺特征表来简化表达, 但是这种转化会改变工艺网络图的原始信息, 甚至改变原问题的解空间结构而导致最优解的丢失 (该部分的讨论将在第3.3节展开)。当然, 也有一些建模方法能够规避上述情况的发生, 但它需要复杂的前序处理操作[18], 而当工件的工艺网络图较为复杂时, 这种前序处理操作会非常繁琐费时。

为了添补当前研究工作的空白, 本文提出了一个全新的基于PP网络图拓扑结构的MILP数学模型, 本文的主要贡献如下:

(1) 与现有研究中的模型相比, 本文所提的基于网络图OR节点的MILP模型不需要任何工艺特征转换或前序处理操作。

(2) 工序的优先关系在本文中进行了详细的讨论, 并提出了三种优先关系模型来分析论述工序的优先关系约束。

(3) 通过通用代数建模系统 (general algebraic modeling system, GAMS) 调用CPLEX求解器对所提模型进行求解, 成功得到了现有文献中大部分开放问题的最优解[8,15,16]。

本文的后续部分组织如下: 第2节介绍相关的研究; 第3节介绍所提的模型细节和上文所提的相关分析; 第4节是验证实验部分, 验证所提模型的优越性; 第5节则进行了总结和对未来工作的展望。

## 2. 相关理论与研究介绍

关于PP问题的研究主要可以分为两类: 算法相关的研究与模型相关的研究。Xu等[6]和Leo Kumar [4]在PP问题上已经做了很全面的综述工作。智能算法, 如GA、模拟退火 (simulated annealing, SA) 算法、TS算法和PSO算法已经在解决PP问题上展现出了足够的优势, 并且已经广泛地应用到该问题的求解中。为了求解复杂棱柱形零件的PP问题, Li等[19]提出了一种由GA和SA组成的混合算法。通过基于Hamming距离的解搜索和选择策略, 该混合算法的局部搜索能力得到了有效的加强。Hua等[20]基于GA提出了一种合成算法来搜索PP问题的最优或近优解。Li等[15]提出了一种有效的基因编程 (genetic programming, GP) 算法, 来操作工艺网络图中的OR节点支路。考虑刀具和进给速度等制造资源的选择, Shin等[21]提出了一种共生进化算法来优化三个目标, 包括机器负载平衡、零件移动最小化和换刀次数最小化。Wang等[22]提出了一种包含两类局部搜索策略的PSO算法求解PP问题, 也取得了不错的效果。后来, Li等[8]也提出了改进的PSO算法来求解考虑工件转运时间的PP问题。Liu等[14]将ACO算法与问题的约束矩阵和状态矩阵结合, 并成功应用于求解两类柱状零件的PP问题。

虽然智能算法解决PP问题方面取得了一些成果, 但由于元启发式算法不能保证解的最优性, 所以在大多数现有PP实例中, 解的质量还可以通过数学模型进一步提高甚至优化到最优。此外, 数学模型作为一种问题的表述方法, 能够帮助研究者进一步地理解和探索问题[23]。因此, 进行PP问题的模型相关研究意义深远。Floudas和Lin [24]基于时间表示方法, 分析研究了几种PP的混合整数规划 (mixed-integer programming, MIP) 模型, 并提出了几种有效的优化方法来提高模型的计算效率。着眼于PP与调度的互补属性, Li等[25]建立了PP与调度集成的数学模型。Xia等[26]基于工艺特征的思路提出了可重构PP问题的数学模型。同样是基于工艺特征的方式, Jin和Zhang [16]建立了考虑机器间转运约束的PP问题的数学模型。

据目前调研所知, 现有的PP问题数学模型都是基于工艺特征建立的[8,14,17]。虽然这种表示方法能够描述大部分类型工件的工艺, 但是仍然不可避免地加入一些本来不存在的工序紧前关系约束[16]。与之相比, 本文所提的基于网络图的表示法比基于特征的表示法更能

表示原有的工艺信息，而不会添加额外信息。本文提出了一个直接基于工艺网络图拓扑结构的MILP模型，该模型能够描述所有类型的PP制造柔性，而不添加或遗漏任何约束。通过求解所提模型，成功获得了一些著名数据集的新最优解。

### 3. 工艺规划 MILP 数学模型

#### 3.1. 问题描述

PP问题中包含的柔性类型有工艺柔性、机器选择柔性和工序序列柔性，常用的表示方法有Petri网络[27]、工艺特征表[28]、AND/OR图和网络图等[29,30]。图1为某工件的制造柔性通过网络图表示的示例，网络图由5类节点构成：虚拟的开始节点和终止节点，表示加工的

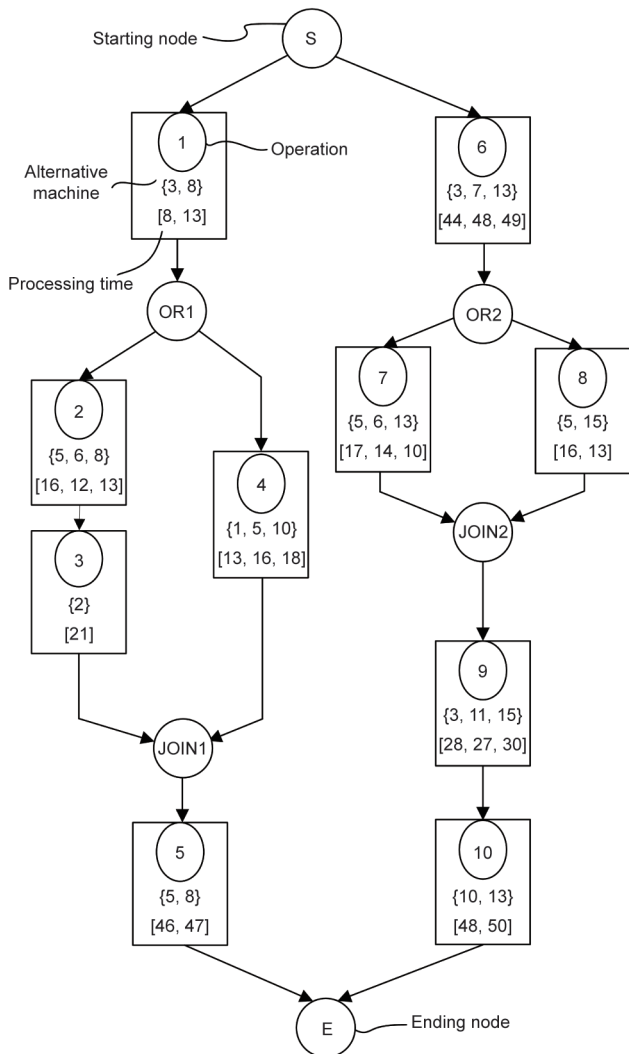


图1. 工艺网络图示例。OR1、OR2、JOIN1、JOIN2分别表示1号OR节点、2号OR节点、1号JOIN节点和2号JOIN节点。

开始和结束；表示工序的中间节点；以及配合起来表示工艺约束的OR节点和JOIN节点[15]。中间节点包含工序序号、可选加工机器号以及对应的加工时间三种信息。例如，中间节点6表示的是工序6能够在序号为3、7和13的机器上加工，且对应的加工时间分别为44、48和49。这里的加工时间单位按照原始数据被省去。被箭头链接的两两工序之间有着优先关系约束[21]，如图1中节点2和3之间的箭头表示工序2必须在工序3之前加工。而没有箭头链接的工序之间不存在这种优先关系约束。此外，介于一对OR节点和JOIN节点的支路只能有一条被选择，包含在支路里面的工序配合其他工序组成一条可行的加工路线[29]。以图1为例，如果节点OR1和JOIN1之间选择支路2 → 3，节点OR2和JOIN2之间选择工序7，则其中一条可行的加工路线可以是：1 → 2 → 3 → 5 → 6 → 7 → 9 → 10。

#### 3.2. 工序间的优先关系

根据模型中0-1二元变量的定义，PP问题的建模方式可以分为三种[31]：

$$\theta_{jt} = \begin{cases} 1, & \text{if operation } j \text{ is processed at period } t \\ 0, & \text{otherwise} \end{cases}$$

$$\rho_{jt} = \begin{cases} 1, & \text{if operation } j \text{ is the } t\text{th one of the sequence to} \\ & \text{be processed} \\ 0, & \text{otherwise} \end{cases}$$

$$q_{jj'} = \begin{cases} 1, & \text{if operation } j \text{ is processed before operation } j' \\ 0, & \text{otherwise} \end{cases}$$

变量 $\theta_{jt}$ 会引入比其他两种类型更多的派生变量和约束，因此现有的PP问题建模研究中鲜少提及。基于位置的变量 $\rho_{jt}$ 描述的是工序在一条序列中的位置序号[16,32,33]，被较多地应用在现有的模型相关研究[24,34,35]中。作为对比，本文所提的MILP模型则是首个基于第三种类型变量 $q_{jj'}$ 建立的模型。

以图2中网络图表示的工件为例，其中一条可行的加工路线为：1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9。因为工序1在其他所有工序的前面加工，所以根据变量 $q_{1j'}$ 的定义可以得出： $q_{1j'} = 1$ ，其中 $j' = 2, \dots, 9$ 。工序2在工序3到工序9前面加工，因此 $q_{2j'} = 1$ ，其中 $j' = 3, \dots, 9$ 。以此类推，矩阵 $Q = [q_{jj'}]$ 中等于1的变量总数为 $(n-1)n/2$ ，其中 $n$ 表示工序总数，同样也等于该工序序列的总长度。图3展示了由工序序列生成 $Q$ 矩阵的过程。

在一条已经确定加工顺序的序列中，任意两两工序之间都存在一个优先关系。因此，可以通过一个 $n \times n$

矩阵来表示一条工序序列中包含的所有优先关系，即上述的矩阵 $Q$ 。矩阵 $Q$ 可以被看作是工序序列优先关系的完全表达矩阵（completely expressing matrix, CEM），因为它包含了一条序列中所有的优先关系。通过观察和分析发现，CEM  $Q$ 的特征约束可以总结如下：

(1) CEM  $Q$ 对角线上的元素都等于0。

$$q_{jj} = 0, \forall j \quad (1)$$

(2) 关于对角线对称的两元素之和等于1。

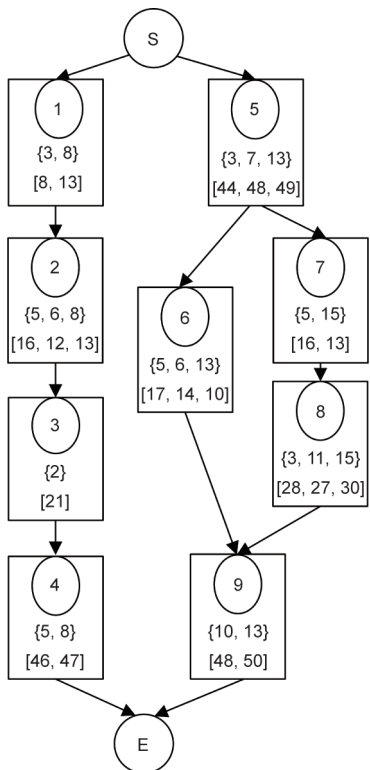


图2. 工艺网络图实例。

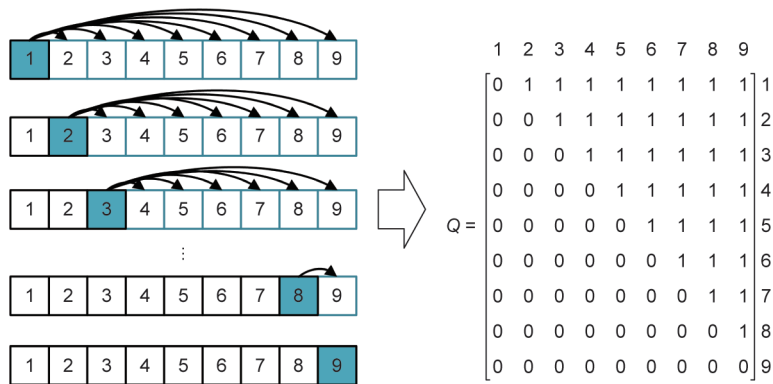


图3. 优先关系矩阵转换示例。

$$q_{ij} + q_{j'i} = 1, \forall j \neq j' \quad (2)$$

(3) 任意两列元素之和互不相等。

$$\sum_j q_{ij} \neq \sum_j q_{i'j}, \forall j' \neq j'' \quad (3)$$

式(1)指代的是优先关系只存在于两个不同工序之间。式(2)指代的是两工序之间只能存在一种优先关系。CEM  $Q$ 中的每列元素之和等于该列对应工序在序列中所处的位置序号，根据位置的唯一性推断出式(3)成立。CEM  $Q$ 矩阵包含了一条序列的所有优先关系，可以根据序列的CEM  $Q$ 来判断该序列是否满足网络图中的优先关系约束。

引入符号 $s_{ij}$ 来表示图2中网络图的优先约束：

$$s_{ij} = \begin{cases} 1, & \text{if operation } j \text{ should be processed before } j' \text{ according} \\ & \text{to the network} \\ 0, & \text{otherwise} \end{cases}$$

根据定义，对应的矩阵 $S = [s_{ij}]$ 如图4所示，其中 $S$ 矩阵中的“1”对应于网络图中的箭头，表示优先关系约束。如果一条工序序列满足工艺网络图中的优先约束，则其序列的CEM  $Q$ 应该与矩阵 $S$ 有如下关系：

$$q_{ij} \geq s_{ij}, \forall j, j' \quad (4)$$

矩阵 $S$ 可以从网路图中生成，但是在序列未定前，CEM  $Q$ 则是未知的。由此，式(4)是用来约束未知的CEM  $Q$ 的。

CEM中等于“1”的元素数量为 $(n-1)n/2$ ，但是并不意味着定义一条工序序列需要 $(n-1)n/2$ 个“1”。实际上，定义一条序列所需的最小“1”总数为 $n-1$ 个。以序列 $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ 为例，只需要定义两两相邻的元素之间的优先关系，就可以确定

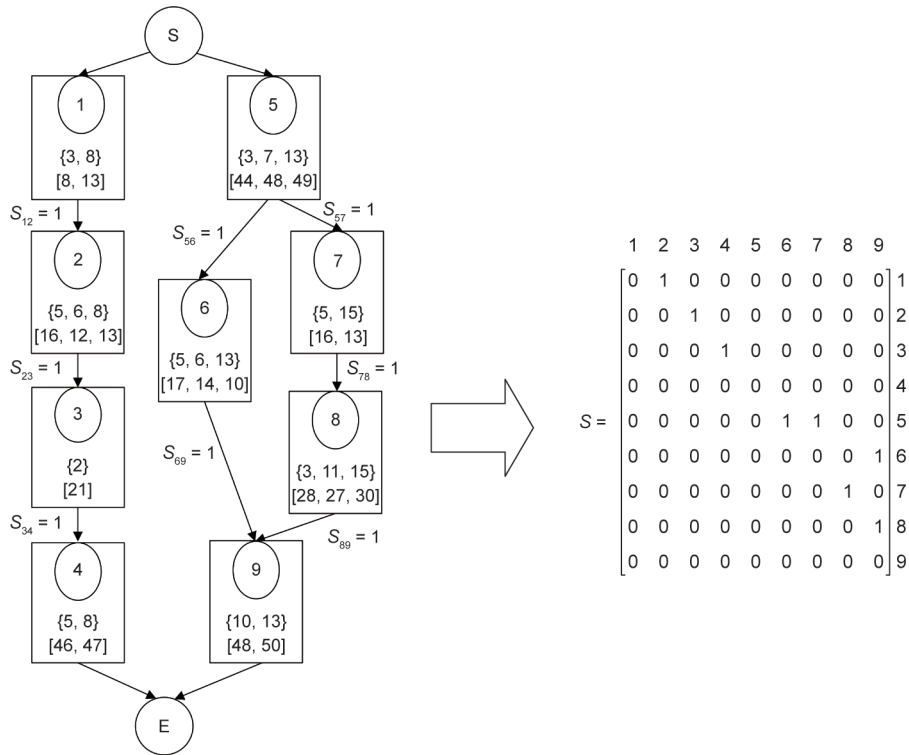


图4. 网络图中的优先约束和对应的矩阵S。

该序列，即 $q_{12}$ 、 $q_{23}$ 、 $q_{34}$ 、 $q_{45}$ 、 $q_{56}$ 、 $q_{67}$ 、 $q_{78}$ 、 $q_{89}$ 等于1即可。从这点出发，定义一个包含等于1的变量数最少的矩阵 $V = [v_{ij}]$ ，有利于简化工序优先关系的研究。基于上述讨论，矩阵可看作是精确表达矩阵 (exactly expressing matrix, EEM)，包含定义一条工序序列所需要的最精确和最少的等于“1”的变量。上述工序序列的EEM  $V$ 如图5所示。

因为EEM  $V$ 包含所有紧邻工序之间的优先关系约束，所以EEM  $V$ 同时也能被称作紧前关系矩阵。根据直接紧邻关系，可以较为容易地通过依次读取EEM  $V$ 中等于1的元素来获得一条工艺路线，完成矩阵到序列的转化。EEM  $V$ 的特征约束如下：

(1) EEM  $V$ 中等于1的元素总数等于 $n - 1$ 。

$$\sum_j \sum_{j'} v_{jj'} = n - 1 \tag{5}$$

(2) EEM  $V$ 的行和列中最多只能有1个元素等于1。

$$\sum_j v_{ij} \leq 1, \forall j' \tag{6}$$

$$\sum_{j'} v_{ij'} \leq 1, \forall j \tag{7}$$

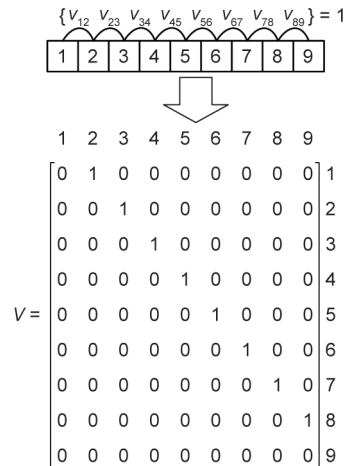


图5. 序列的精确表达矩阵。

(3) CEM  $Q$ 与EEM  $V$ 之间的关系如下。

$$v_{ij'} \leq q_{ij'}, \forall j, j' \tag{8}$$

EEM  $V$ 是CEM  $Q$ 的简化表达形式，并且二者都能够定义一条唯一的工序序列。由于矩阵 $S$ 中对序列优先关系的不完整表示，使其不能定义一条唯一的工序序列。因此，在本文中矩阵 $S$ 也可以被称作优先关系的部分表达矩阵 (partly expressing matrix, PEM)。不难得出，一般情况

下不止一条工序序列可以满足由PEM  $S$ 表达的优先关系约束。CEM  $Q$ 、EEM  $V$ 、PEM  $S$ 的对比如图6所示。

### 3.3. 关于工艺特征表示法与工艺网络图表示法的讨论

现有文献[8,16]中关于PP问题的模型都是基于工艺特征表示法建立的，如图7 (a) 中的以工艺特征表示的算例。实际上，该算例是由文献[29]中的第18个算例通过转换得到的，如图7 (b) 所示。图7 (a)、(b) 表达的是同一件工件的工艺信息。

在特征表[8,16]中，工艺特征 $F_2$ 、 $F_3$ 和 $F_9$ 有着可选的加工工序或加工工序集合，分别与网络图中的三对OR及JOIN节点相对应。然而不同的是，特征表中同一工艺特征下的工序集合会被紧前关系约束，而这种约束

在原始的网络图中却是不存在的。例如，如果工艺特征 $F_2$ 选择了通过工序集合 $O_4 \sim O_5$ 来实现，那么在生成的工序序列中 $O_5$ 必须在 $O_4$ 完成加工后立刻开始加工。但是，根据网络图图7 (b) 传达的工艺信息，是不存在这种立刻开始加工的必要的。这种多余的捆绑可能改变原始的解空间结构，导致失去找到最优解的可能；而实际上这个算例就是如此。表1分别展示了通过基于特征的模型和基于网络图的模型所求得的最优解工序序列，其中带数字下标的字母 $M$ 表示的是工序的加工机器号。

观察表1，由基于工艺网路图模型得到的序列可知，工序 $O_4$ 和 $O_5$ 之间不存在紧前关系约束。此外，基于工艺网络图模型获得的最优解356优于基于工艺特征模型获得的最优解357。其实，因为受到了紧前关系的约束，

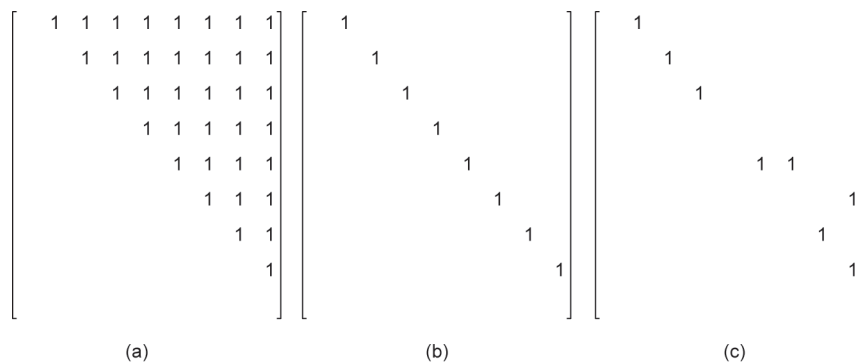


图6. 三类优先关系矩阵。(a) CEM  $Q$ ; (b) EEM  $V$ ; (c) PEM  $S$ 。

Feature	Operations	Precedence constraints of the features
$F_1$	$O_1$	Before $F_2, F_3$
$F_2$	$O_2 \sim O_3$ $O_4 \sim O_5$	Before $F_3$
$F_3$	$O_6$	Before $F_4, F_5, F_7$
$F_4$	$O_7$	Before $F_6, F_7$
$F_5$	$O_8$ $O_9$	Before $F_7$
$F_6$	$O_{10}$	Before $F_7$
$F_7$	$O_{11}$	Before $F_8, F_{10}, F_{11}$
$F_8$	$O_{12}$	Before $F_{10}, F_{11}$
$F_9$	$O_{13}$ $O_{14} \sim O_{15}$	Before $F_{11}$
$F_{10}$	$O_{16}$	Before $F_{11}$
$F_{11}$	$O_{17}$	

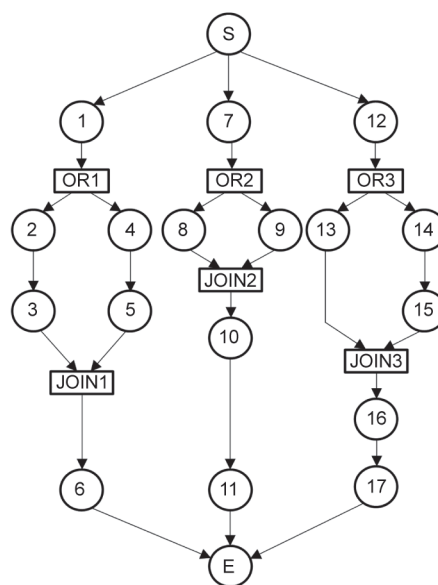


图7. 同一算例的两种表现形式。(a) 工艺特征表; (b) 工艺网络图。

图7. 同一算例的两种表现形式。(a) 工艺特征表; (b) 工艺网络图。

表1 工艺特征模型与工艺网络图模型求得的最优解

Result	Feature-based model	Network-based model
Optimal process plan	$O_7(M_3)-O_1(M_3)-O_{12}(M_3)-O_4(M_5)-O_5(M_9)-O_{13}(M_9)-O_9(M_{15})-O_{10}(M_{15})-O_{16}(M_{15})-O_6(M_8)-O_{17}(M_{10})-O_{11}(M_{10})$	$O_7(M_3)-O_1(M_3)-O_4(M_1)-O_8(M_{13})-O_{12}(M_{13})-O_{10}(M_3)-O_5(M_9)-O_{13}(M_9)-O_{16}(M_4)-O_6(M_8)-O_{11}(M_{10})-O_{17}(M_{10})$
Production time	357	356

通过工艺特征构建的问题解空间是原始网络图表示的解空间的子集。由此可知，直接根据工艺网络图建立问题的模型要好于基于工艺特征表的方式。

通过工艺特征表表示的PP问题的工艺柔性是通过同一加工特征下可选的工序和工序集合实现的，而工艺网络图表示的PP问题则是通过选择连接在OR节点上的支路实现。每条支路对应于工艺柔性的一种选择。如图8所示，OR节点与JOIN节点的序号是自上而下、自左而右的。如果选择了OR节点1的支路1，那么工序2、3和4则一定会被选择，此时无论OR节点2选择支路1还是2，工序5、6、7和8都不会被选择。这是因为，工序5、6、7和8不仅被OR节点2控制，还被OR节点1的支路2所控制，而OR节点1的支路2并没有被选择。OR节点支路选择有效的前提是该节点位于的支路被选择。

引入二进制参数 $w_{jrl}$ 来描述OR节点对工序节点的控制功能，参数 $w_{jrl}$ 定义如下：

$$w_{jrl} = \begin{cases} 1, & \text{if operation } j \text{ is controlled by the } l\text{th link of} \\ & \text{the } r\text{th OR node} \\ 0, & \text{otherwise} \end{cases}$$

以图8为例，对应的 $w_{jrl}$ 参数值如表2所示。

综上，可将OR节点的控制功能归纳如下：当且仅当控制工序 $j$ 的所有支路都被选择，工序 $j$ 才会被选择；否则，只要当其中一条控制支路未被选择，工序 $j$ 就不会被选择。引入0-1二元变量 $u_{rl}$ 来描述OR节点支路的选择状态，同时引入0-1二元变量 $x_j$ 来描述工序的选择状态。变量 $u_{rl}$ 和 $x_j$ 的定义如下：

$$u_{rl} = \begin{cases} 1, & \text{if the } l\text{th link of } r\text{th OR node is selected} \\ 0, & \text{otherwise} \end{cases}$$

$$q_{jr} = \begin{cases} 1, & \text{if operation } j \text{ is processed before operation } j' \\ 0, & \text{otherwise} \end{cases}$$

第3.4节介绍的模型是基于上述两种变量 $u_{rl}$ 和 $x_j$ 以及参数 $w_{jrl}$ 展开的。

### 3.4. PP问题的数学模型

本文中MILP模型是基于优先关系和网络图OR节点的。大部分的工艺优化目标都是时间相关[8]或者成本

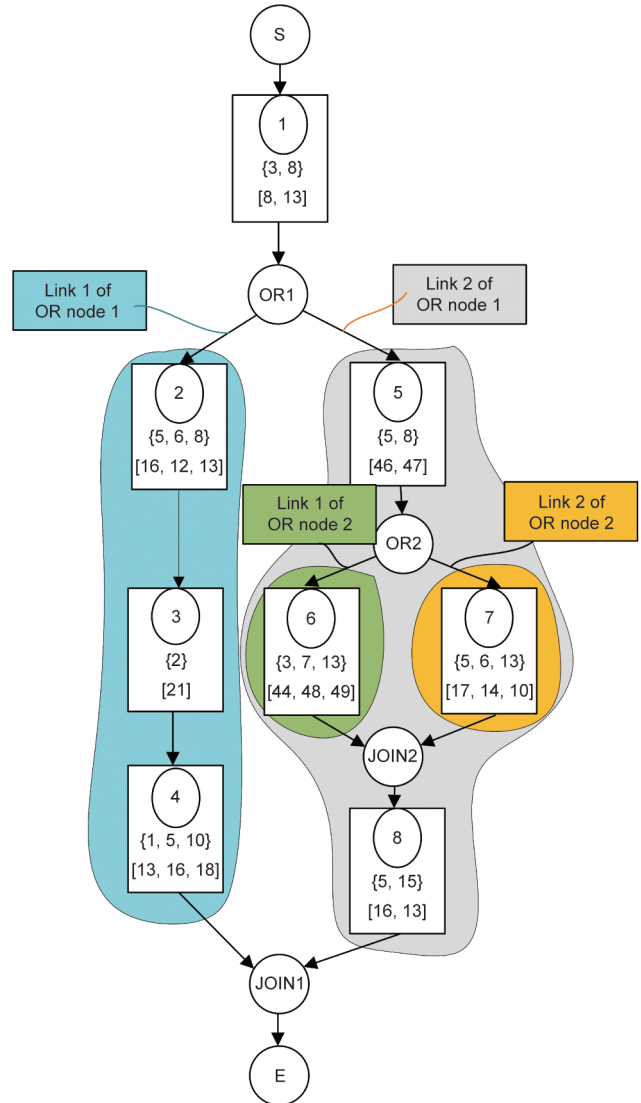


图8. OR节点控制功能的讨论示例。

相关[14,18]的，本文的优化目标为最小化总生产时间，并且考虑工件在机器间的转运时间约束。模型的集合、下标、参数以及变量展示如表3所示。

总生产时间为模型的目标函数：

$$\min TT = \sum_j MT_j + \sum_j \sum_k T_{jk} \cdot y_{jk} \quad (9)$$

式(9)右边的第一部分是工件的总转运时间，第

表2 参数 $w_{jrl}$ 值

Values of $r$ and $l$	$w_{jrl}$							
	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$
$r=1, l=1$	0	1	1	1	0	0	0	0
$r=1, l=2$	0	0	0	0	1	1	1	1
$r=2, l=1$	0	0	0	0	0	1	0	0
$r=2, l=2$	0	0	0	0	0	0	1	0

表3 PP问题数学模型的集合、下标、参数以及变量定义

Items	Definition
Sets and subscripts	
$j, j'$	Operations, $j = 1, 2, \dots, n$ , where $n$ is the total number of operations
$k, k'$	Machines, $k = 1, 2, \dots, K$ , where $K$ is the total number of machines
$r$	OR-nodes, $r = 1, 2, \dots, R$ , where $R$ is the total number of OR nodes
$l$	Links, $l = 1, 2, \dots, L$ , where $L$ is the total number of links of one node
Parameters	
$s_{jj'}$	The precedence constraints in the network: If operation $j$ is supposed to be processed before operation $j'$ , then $s_{jj'} = 1$ ; otherwise, $s_{jj'} = 0$
$z_{jk}$	1, machine $k$ is optional for operation $j$ ; 0, otherwise
$w_{jrl}$	1, operation $j$ is controlled by the $l$ th link of the $r$ th OR node; 0, otherwise
$T_{jk}$	The processing time of operation $j$ on machine $k$
$TK_{kk'}$	The transmission time from machine $k$ to $k'$
$M$	A large enough positive number
Variables	
$q_{jj'}$	1, operation $j$ is processed before operation $j'$ ; 0, otherwise
$v_{jj'}$	1, operation $j$ is processed directly before operation $j'$ ; 0, otherwise
$y_{jk}$	1, operation $j$ is processed on machine $k$ ; 0, otherwise
$u_{rl}$	1, the $l$ th link of the $r$ th OR node is selected; 0, otherwise
$x_j$	1, operation $j$ is selected to process the part; 0, otherwise
$MT_j$	The transmission time from the machine processing operation $j$ to the next machine to process the operation
$TT$	The total production time

二部分是工件的总加工时间。模型的约束如下：

(1) OR节点约束。

$$x_j \leq M \cdot (1 - w_{jrl}) + M \cdot u_{rl}, \forall j, r, l \quad (10)$$

$$x_j \geq 1 - M \cdot \sum_r \sum_l w_{jrl} \cdot (1 - u_{rl}), \forall j \quad (11)$$

$$\sum_l u_{rl} = 1, \forall r \quad (12)$$

式(10)表示的是被OR节点控制的工序不被选择的前提条件：只要有一条控制工序 $j$ 的支路未被选择，工序 $j$ 就不会被选择。式(11)表示的则是工序被选择的前提条件：工序不被任何OR节点控制；或者，控制该工序的所有支路都被选择。式(12)表示的则是工艺柔性只能选择连接在一个OR节点上的所有支路中的一条。

(2) 工序优先约束。

$$q_{jj} = 0, \forall j \quad (13)$$

$$q_{jj'} + q_{j'j} \leq 1 + M \cdot (2 - x_j - x_{j'}), \forall j \neq j' \quad (14)$$

$$q_{jj'} + q_{j'j} \geq 1 - M \cdot (2 - x_j - x_{j'}), \forall j \neq j' \quad (15)$$

$$M \cdot (2 - x_j - x_{j'}) + \sum_{j''} q_{jj''} - \sum_{j''} q_{j''j} \neq 0, \forall j \neq j' \quad (16)$$

$$q_{jj'} + q_{j'j} \leq M \cdot x_j, \forall j, j' \quad (17)$$

$$(2 - x_j - x_{j'}) \cdot M + q_{jj'} \geq s_{jj'}, \forall j, j' \quad (18)$$

式(13)~(18)表示的是工序之前的优先约束。



与式(1)~(4)不同的是,在这组约束里加入了工序被选择的前提。式(13)对应式(1),式(14)、(15)对应式(2),式(16)对应式(3)。式(17)表示如果工序*j*和*j'*未被选择,那么 $q_{jj'}$ 和 $q_{j'j}$ 都被设置为0。式(18)则是确保被选中的工序遵循网络图中的优先关系约束。

(3) EEM *V*与CEM *Q*约束。

$$\sum_j \sum_{j'} v_{jj'} = \sum_j x_j - 1 \tag{19}$$

$$\sum_{j'} v_{jj'} \leq 1, \forall j \tag{20}$$

$$\sum_j v_{jj'} \leq 1, \forall j' \tag{21}$$

$$v_{jj'} \leq q_{jj'}, \forall j, j' \tag{22}$$

因为矩阵*V*只包含两相邻工序的优先关系,工件相邻工序的转运时间可以借此计算。式(19)~(22)描述的是矩阵*V*的自身属性,及矩阵*V*和*Q*之间的关系。

(4) 机器选择约束。

$$\sum_k z_{jk} \cdot y_{jk} = x_j, \forall j \tag{23}$$

式(23)表示的是被选工序只能选择一台机器执行

加工,且未被选择的工序无须分配机器加工。

(5) 转运时间约束。

$$MT_j \leq TK_{kk'} + M \cdot (1 - v_{jj'}) + M \cdot (2 - y_{jk} - y_{j'k'}), \forall j, j', k, k' \tag{24}$$

$$MT_j \geq TK_{kk'} - M \cdot (1 - v_{jj'}) - M \cdot (2 - y_{jk} - y_{j'k'}), \forall j, j', k, k' \tag{25}$$

式(24)和(25)表述了工序*j*从当前处理机器到下一个机器的转运时间。

### 4. 实验与讨论

通过5组由著名数据集组成的数字实验来验证MILP模型,所得结果直接与文献报道的其他方法所得的结果比对。实验平台配置3.7 GHZ和16 GB 随机存取存储器(RAM)的个人计算机,模型通过GAMS编码并调用CPLEX进行求解。引入Gap值(%)来评价模型所求解的质量,该值表示所得解的相对偏差,定义为(BF - BP)/BP,其中BF为目标值的当前最好解,BP为当前下界。Gap值越小,表示当前解越接近理论下界,即质量越好。求解时间设置为3600 s,如果在该时间内未搜索到最优解,计算进程就会被中止,并且当前解会被输

表4 机器间的转运时间矩阵[8,18]

Machine	Machine														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	5	7	9	10	11	7	6	14	13	12	10	5	6	9
2	5	0	3	4	5	7	2	7	6	4	7	12	10	7	8
3	7	3	0	6	5	4	3	7	2	4	3	5	6	8	9
4	9	4	6	0	4	4	6	7	4	10	12	13	14	15	16
5	10	5	5	4	0	10	12	7	8	9	10	12	11	10	8
6	11	7	4	4	10	0	4	4	5	5	6	7	6	7	8
7	7	2	3	6	12	4	0	5	6	6	6	7	7	7	8
8	6	7	7	7	7	4	5	0	4	2	3	4	2	4	3
9	14	6	2	4	8	5	6	4	0	5	7	4	7	6	8
10	13	4	4	10	9	5	6	2	5	0	8	10	12	14	7
11	12	7	3	12	10	6	6	3	7	8	0	7	10	14	10
12	10	12	5	13	12	7	7	4	4	10	7	0	10	12	10
13	5	10	6	14	11	6	7	2	7	12	10	10	0	8	8
14	6	7	8	15	10	7	7	4	6	14	14	12	8	0	9
15	9	8	9	16	8	8	8	3	8	7	10	10	8	9	0

出。实验1、2、4和5中机器间的转运时间数据来自文献[8,18], 如表4所示。

#### 4.1. 实验 1

实验1中的3个算例来自Jin和Zhang [16], 他们提出类似动态规划 (DP)-like的启发式算法。本文中MILP模型获得的结果与DP-like启发式算法所得结果对比如表5所示。第一个对比算例中MILP模型获得的解目标值为357, 好于DP-like启发式算法获得的解目标值360。此外, MILP模型能够获得并证明全部3个算例的最优解。

#### 4.2. 实验 2

实验2的4个算例来自不同的文献: 算例1来自Zhang和Lee [36], 算例2~4来自Li和McMahon [37]。算例的详细数据可见原文献。算例的对比结果为Li等[8]提出的改进PSO算法是当前求解此类组合优化问题最有效的算法之一。对比结果如表6所示, MILP模型能够获得目标值更好的解, 并且以较短的计算时间 (小于1 s) 获得了算例2~4的最优解。

#### 4.3. 实验 3

本组实验的两个算例来自Li等[15]提出的GP算法, 机器转运时间矩阵如表7所示。两个算例唯一的区别是算例2中的机器2被移除。实验对比结果如表8所示, GP

算法和MILP算法都能找到两个算例的最优解。

#### 4.4. 实验 4

本组实验的17个算例来自著名的Kim数据集[29], 由18个工件组成, 其对比的算法和结果来自文献[8]。由于文献[8]中工件4的算例数据有问题, 因此在本组实验中不予展示。MILP模型与改进的PSO算法、GA和SA算法的对比结果如表9所示。在可接受计算时间内, MILP模型能够找到本组17个算例中的13个最优解。对于未找到最优解的算例, 模型所得解的质量也好于对比算法解的质量, 如算例3、6、12和15。

#### 4.5. 实验 5

实验5的11个算例来自另一组有名的Shin数据集[21], 其对比计算结果同样来自Li等[8]。因为同样的数据问题, 略去了工件9、10、12、13、15、17和18的对比。MILP模型与改进的PSO算法、GA和SA算法的对比结果如表10所示。在可接受计算时间内, MILP模型能够找到本组11个算例中的9个最优解。对于未找到最优解的算例, 模型所得解的质量也好于对比算法解的质量, 如算例3。

#### 4.6. 实验讨论与分析

通过本节实验发现, MILP模型能够在可接受计算

表5 实验1的结果对比

Cases	DP-like heuristic [16]	MILP
1 Process plan	$O_{12}(M_5)-O_7(M_3)-O_1(M_3)-O_4(M_{10})-O_5(M_9)-O_{13}(M_9)-O_9(M_{15})-O_{10}(M_{15})-O_{16}(M_{15})-O_6(M_8)-O_{17}(M_{10})-O_{11}(M_{10})$	$O_7(M_3)-O_1(M_3)-O_{12}(M_5)-O_4(M_5)-O_5(M_9)-O_{13}(M_9)-O_9(M_{15})-O_{10}(M_{15})-O_{16}(M_{15})-O_6(M_8)-O_{17}(M_{10})-O_{11}(M_{10})$
Production time	360	357 <sup>a</sup>
Computing time	—	3546.74
Gap	—	0%
2 Process plan	$O_{13}(M_1)-O_4(M_1)-O_2(M_1)-O_1(M_1)-O_7(M_1)-O_8(M_1)-O_9(M_1)-O_{12}(M_1)-O_{10}(M_1)-O_{11}(M_1)-O_5(M_1)-O_6(M_1)-O_3(M_1)$	$O_{13}(M_1)-O_2(M_1)-O_4(M_1)-O_1(M_1)-O_3(M_1)-O_{10}(M_1)-O_{11}(M_1)-O_7(M_1)-O_8(M_1)-O_9(M_1)-O_5(M_1)-O_6(M_1)-O_{12}(M_1)$
Production time	222	222 <sup>a</sup>
Computing time	—	12.16
Gap	—	0%
3 Process plan	$O_1(M_2)-O_3(M_2)-O_5(M_2)-O_6(M_2)-O_9(M_2)-O_4(M_2)-O_7(M_2)-O_8(M_4)-O_2(M_4)$	$O_1(M_2)-O_3(M_2)-O_5(M_2)-O_6(M_2)-O_4(M_2)-O_9(M_2)-O_7(M_2)-O_8(M_4)-O_2(M_4)$
Production time	212	212 <sup>a</sup>
Computing time	—	0.44
Gap	—	0%

<sup>a</sup> Indicates that the optimal solution is found.

表6 实验2的结果对比

Cases	Modified PSO [8]	MILP
1 Process plan	$O_{12}(M_4)-O_2(M_4)-O_{15}(M_4)-O_{16}(M_4)-O_3(M_2)-O_{13}(M_1)-O_8(M_1)-O_9(M_1)-O_{10}(M_1)-O_4(M_1)-O_1(M_1)-O_{17}(M_1)-O_{14}(M_1)-O_5(M_1)-O_6(M_1)-O_7(M_1)-O_{11}(M_1)-O_{18}(M_1)$	$O_{16}(M_4)-O_{15}(M_4)-O_2(M_4)-O_{12}(M_4)-O_1(M_2)-O_8(M_2)-O_9(M_2)-O_{10}(M_2)-O_3(M_2)-O_5(M_2)-O_6(M_1)-O_7(M_1)-O_{14}(M_1)-O_4(M_1)-O_{13}(M_1)-O_{17}(M_1)-O_{18}(M_1)-O_{11}(M_1)$
Objective value	359	358
Computing time	—	3600.00
Gap	—	4.71%
2 Process plan	$O_1(M_4)-O_5(M_4)-O_{18}(M_4)-O_2(M_4)-O_6(M_4)-O_{11}(M_4)-O_{12}(M_4)-O_{13}(M_4)-O_{14}(M_4)-O_7(M_4)-O_4(M_4)-O_{17}(M_4)-O_{15}(M_4)-O_{16}(M_4)-O_8(M_4)-O_9(M_4)-O_{10}(M_4)-O_{19}(M_4)-O_{20}(M_4)-O_3(M_4)$	$O_1(M_4)-O_{18}(M_4)-O_3(M_4)-O_5(M_4)-O_7(M_4)-O_8(M_4)-O_9(M_4)-O_{10}(M_4)-O_2(M_4)-O_{11}(M_4)-O_{17}(M_4)-O_4(M_4)-O_6(M_4)-O_{12}(M_4)-O_{13}(M_4)-O_{14}(M_4)-O_{19}(M_4)-O_{20}(M_4)-O_{15}(M_4)-O_{16}(M_4)$
Objective value	341	341 <sup>a</sup>
Computing time	—	0.20
Gap	—	0%
3 Process plan	$O_1(M_4)-O_2(M_4)-O_9(M_4)-O_{10}(M_4)-O_{11}(M_4)-O_5(M_4)-O_3(M_4)-O_6(M_4)-O_4(M_4)-O_7(M_4)-O_{12}(M_4)-O_{13}(M_4)-O_8(M_4)-O_{14}(M_4)$	$O_1(M_4)-O_2(M_4)-O_5(M_4)-O_4(M_4)-O_3(M_4)-O_6(M_4)-O_9(M_4)-O_{10}(M_4)-O_{11}(M_4)-O_7(M_4)-O_{12}(M_4)-O_{13}(M_4)-O_{14}(M_4)-O_{14}(M_4)$
Objective value	176	176 <sup>a</sup>
Computing time	—	0.44
Gap	—	0%
4 Process plan	$O_{16}(M_4)-O_3(M_4)-O_5(M_4)-O_1(M_4)-O_6(M_4)-O_7(M_4)-O_{14}(M_4)-O_{11}(M_4)-O_{15}(M_4)-O_9(M_4)-O_{12}(M_4)-O_4(M_4)-O_{13}(M_4)-O_2(M_4)-O_8(M_4)-O_{10}(M_4)$	$O_{16}(M_4)-O_6(M_4)-O_7(M_4)-O_3(M_4)-O_{11}(M_4)-O_{13}(M_4)-O_{12}(M_4)-O_9(M_4)-O_{14}(M_4)-O_1(M_4)-O_3(M_4)-O_2(M_4)-O_{15}(M_4)-O_4(M_4)-O_{10}(M_4)-O_8(M_4)$
Objective value	187	187 <sup>a</sup>
Computing time	—	0.39
Gap	—	0%

<sup>a</sup> Indicates that the optimal solution is found.

表7 机器间的转运时间矩阵

Machine	Machine									
	1	2	3	4	5	6	7	8	9	10
1	0	5	8	12	15	4	6	10	13	18
2	5	0	3	7	10	6	4	6	10	13
3	8	3	0	4	7	10	6	4	6	10
4	12	7	4	0	3	14	10	6	4	6
5	15	10	7	3	0	18	12	10	6	4
6	4	6	10	14	18	0	5	8	12	15
7	6	4	6	10	12	5	0	3	7	10
8	10	6	4	6	10	8	3	0	4	8
9	13	10	6	4	6	12	7	4	0	4
10	18	13	10	6	4	15	10	8	4	0

时间内，求得总计37个算例中28个算例的最优解。与现有高性能的启发式[16]和元启发式算法[8,15]对比，所提模型能够获得质量更好的解。第4组和第5组实验是在广泛使用的著名数据集[21,29]上验证的，实验结果表明了所提模型的优越性。

如表11所示，所提模型包含了4种类型的下标，即工序（operation）、机器（machine）、OR节点（OR node）和支路（link）；而文献[16]中的模型包含6种下标，即工艺特征（feature）、工序集合（operation set）、工序（operation）、机器（machine）、位序（position）和位置

表8 实验3的结果对比

Case	GP [15]	MILP
1 Process plan	$O_1(M_2)-O_2(M_7)-O_3(M_2)-O_4(M_2)-O_9(M_3)$	$O_1(M_2)-O_2(M_7)-O_3(M_2)-O_4(M_2)-O_9(M_3)$
Objective value	213	213 <sup>a</sup>
Computing time	—	0.16
Gap	—	0%
2 Process plan	$O_{10}(M_9)-O_{11}(M_3)-O_{15}(M_7)$	$O_{10}(M_8)-O_{11}(M_2)-O_{15}(M_6)$
Objective value	224	224 <sup>a</sup>
Computing time	—	0.13
Gap	—	0%

<sup>a</sup> Indicates that the optimal solution is found.

表9 实验4的结果对比

Case	Objective value				Gap	Computing time
	SA	GA	PSO	MILP		
1	303	303	303	292 <sup>a</sup>	0%	0.08
2	359	359	359	351 <sup>a</sup>	0%	191.50
3	502	502	498	486	45.5%	3600.00
5	314	314	314	280 <sup>a</sup>	0%	26.45
6	409	409	408	399	40.5%	3600.00
7	304	304	304	304 <sup>a</sup>	0%	27.02
8	358	358	358	353 <sup>a</sup>	0%	63.44
9	393	392	391	391	38.1%	3600.00
10	264	264	264	264 <sup>a</sup>	0%	1.27
11	271	271	271	264 <sup>a</sup>	0%	22.24
12	442	442	442	433	34.9%	3600.00
13	216	216	216	215 <sup>a</sup>	0%	10.69
14	269	269	269	244 <sup>a</sup>	0%	56.77
15	358	357	357	353	5.1%	3600.00
16	248	248	248	244 <sup>a</sup>	0%	3071.28
17	314	314	314	300 <sup>a</sup>	0%	3600.00
18	361	361	360	356 <sup>a</sup>	0%	3600.00

<sup>a</sup> Indicates that the optimal solution is found.

表10 实验5的结果对比

Case	Objective value				Gap	Computing time
	SA	GA	PSO	MILP		
1	267	267	267	266 <sup>a</sup>	0%	19.55
2	165	165	163	162 <sup>a</sup>	0%	30.06
3	299	297	296	268	33.7%	3600.00
4	268	268	267	267	20.2%	3600.00
5	204	204	204	199 <sup>a</sup>	0%	463.20
6	204	204	204	189 <sup>a</sup>	0%	268.33
7	137	137	137	116 <sup>a</sup>	0%	0.94
8	181	181	181	178 <sup>a</sup>	0%	0.20
11	151	150	149	149 <sup>a</sup>	0%	69.88
14	120	120	120	120 <sup>a</sup>	0%	11.06
16	170	167	167	167 <sup>a</sup>	0%	620.38

<sup>a</sup> Indicates that the optimal solution is found.

表11 模型下标

Subscripts	Definition
Proposed MILP model	
$j, j'$	Operations
$k, k'$	Machines
$r$	OR nodes
$l$	Links
Reported model in Ref. [16]	
$i, i'$	Features
$j, j'$	Operation sets
$k, k'$	Operations
$l, l'$	Machines
$r$	OR nodes
$h$	Place, where a feature can be assigned

(place)。所提出的模型中较少的下标使得模型的求解比文献[16]中的模型更有效率。此外，基于OR节点的建模方法使得所提模型具有更强的通用性。这也解释了所提模型能够获得Kim [29]和Shin [21]数据集大部分最优解的原因。

## 5. 结论及未来工作展望

从工艺网络图的拓扑结构出发，本文为PP问题提出了一种基于OR节点的MILP数学模型。首先，提出了三种优先矩阵来讨论工序间的优先关系，然后引入了符号 $w_{jrl}$ 、 $u_{rl}$ 和 $x_j$ 来描述OR节点的选择机制以拓展模型的通用性，最后将模型编码并调用CPLEX求解器在公开数据集上进行验证。实验比较结果验证了所提模型的正确性和优越性。

本研究首次提出了基于OR节点的建模方法，为PP问题研究展示了新的思路。本文对三个优先矩阵的分析也揭示了工序排序子问题的本质，有利于进一步理解PP问题。然而，PP模型的相关研究仍存在一些局限性，即少数情况下无法找到最优解，且计算效率有时也不尽如人意。这意味着本文所提出的方法还可以进一步改进，模型的简化和加速方法是未来研究工作的重点之一。

## 致谢

本工作得到了国家自然科学基金（51825502、51775216）以及华中科技大学学术前沿青年团队计划（2017QYTD04）的支持。

## Compliance with ethics guidelines

Qihao Liu, Xinyu Li, and Liang Gao declare that they have no conflict of interest or financial conflicts to disclose.

## References

- [1] Zhou J, Li P, Zhou Y, Wang B, Zang J, Meng L. Toward new-generation intelligent manufacturing. *Engineering* 2018;4(1):11–20.
- [2] Zhong RY, Xu X, Klotz E, Newman ST. Intelligent manufacturing in the context of industry 4.0: a review. *Engineering* 2017;3(5):616–30.
- [3] Zhou J, Zhou Y, Wang B, Zang J. Human–cyber–physical systems (HCPSs) in the context of new-generation intelligent manufacturing. *Engineering* 2019;5(4):624–36.
- [4] Leo Kumar SP. State of the art-intense review on artificial intelligence systems application in process planning and manufacturing. *Eng Appl Artif Intell* 2017;65:294–329.
- [5] Miljković Z, Petrović M. Application of modified multi-objective particle swarm optimisation algorithm for flexible process planning problem. *Int J Comput Integrated Manuf* 2017;30(2–3):271–91.
- [6] Xu X, Wang L, Newman ST. Computer-aided process planning—a critical review of recent developments and future trends. *Int J Comput Integrated Manuf* 2011;24(1):1–31.
- [7] Li X, Gao L, Pan Q, Wan L, Chao KM. An effective hybrid genetic algorithm and variable neighborhood search for integrated process planning and scheduling in a packaging machine workshop. *IEEE Trans Syst Man Cybern Syst* 2019;49(10):1933–45.
- [8] Li X, Gao L, Wen X. Application of an efficient modified particle swarm optimization algorithm for process planning. *Int J Adv Manuf Technol* 2013;67(5–8):1355–69.
- [9] Gan PY, Lee KS, Zhang YF. A branch and bound algorithm based process-planning system for plastic injection mould bases. *Int J Adv Manuf Technol* 2001;18(9):624–32.
- [10] Wen X, Li X, Gao L, Sang H. Honey bees mating optimization algorithm for process planning problem. *J Intell Manuf* 2014;25(3):459–72.
- [11] Zhang J, Xiao Mi, Gao L, Pan Q. Queuing search algorithm: a novel metaheuristic algorithm for solving engineering optimization problems. *Appl Math Model* 2018;63:464–90.
- [12] Liu Q, Li X, Gao L, Li Y. A modified genetic algorithm with new encoding and decoding methods for integrated process planning and scheduling problem. *IEEE Trans Cybern. Epub* 2020 Oct 15.
- [13] Li WD, Ong SK, Nee AYC. Optimization of process plans using a constraint-based tabu search approach. *Int J Prod Res* 2004;42(10):1955–85.
- [14] Liu X, Yi H, Ni Z. Application of ant colony optimization algorithm in process planning optimization. *J Intell Manuf* 2013;24(1):1–13.
- [15] Li XY, Shao XY, Gao L. Optimization of flexible process planning by genetic programming. *Int J Adv Manuf Technol* 2008;38(1–2):143–53.
- [16] Jin L, Zhang C. Process planning optimization with energy consumption reduction from a novel perspective: mathematical modeling and a dynamic programming-like heuristic algorithm. *IEEE Access* 2019;7:7381–96.
- [17] Jiang Z, Jiang Ya, Wang Y, Zhang H, Cao H, Tian G. A hybrid approach of rough set and case-based reasoning to remanufacturing process planning. *J Intell Manuf* 2019;30(1):19–32.
- [18] Jin L, Tang Q, Zhang C, Shao X, Tian G. More MILP models for integrated process planning and scheduling. *Int J Prod Res* 2016;54(14):4387–402.
- [19] Li WD, Ong SK, Nee AYC. Hybrid genetic algorithm and simulated annealing approach for the optimization of process plans for prismatic parts. *Int J Prod Res* 2002;40(8):1899–922.
- [20] Hua G, Zhou X, Ruan X. GA-based synthesis approach for machining scheme selection and operation sequencing optimization for prismatic parts. *Int J Adv Manuf Technol* 2007;33(5–6):594–603.
- [21] Shin KS, Park JO, Kim YK. Multi-objective FMS process planning with various flexibilities using a symbiotic evolutionary algorithm. *Comput Oper Res* 2011;38(3):702–12.
- [22] Wang YF, Zhang YF, Fuh JYH. A hybrid particle swarm based method for process planning optimisation. *Int J Prod Res* 2012;50(1):277–92.
- [23] Shabaka AI, ElMaraghy HA. A model for generating optimal process plans in RMS. *Int J Comput Integrated Manuf* 2008;21(2):180–94.
- [24] Floudas CA, Lin X. Mixed integer linear programming in process scheduling: modeling, algorithms, and applications. *Ann Oper Res* 2005;139(1): 131–62.
- [25] Li X, Gao L, Shao X, Zhang C, Wang C. Mathematical modeling and evolutionary algorithm-based approach for integrated process planning and scheduling. *Comput Oper Res* 2010;37(4):656–67.
- [26] Xia Q, Etienne A, Dantan J, Siadat A. Reconfigurable machining process planning for part variety in new manufacturing paradigms: definitions, models and framework. *Comput Ind Eng* 2018;115:206–19.

- [27] Lee K, Jung M. Petri net application in flexible process planning. *Comput Ind Eng* 1994;27(1-4):505-8.
- [28] Sormaz D, Sarkar A. Process sequencing problem in distributed manufacturing process planning. In: Goldengorin B, editor. *Optimization problems in graph theory*. Berlin: Springer; 2018. p. 293-324.
- [29] Kim YK, Park K, Ko J. A symbiotic evolutionary algorithm for the integration of process planning and job shop scheduling. *Comput Oper Res* 2003;30(8):1151-71.
- [30] Zhang S, Wong TN. Integrated process planning and scheduling: an enhanced ant colony optimization heuristic with parameter tuning. *J Intell Manuf* 2018;29(3):585-601.
- [31] Pan CH. A study of integer programming formulations for scheduling problems. *Int J Syst Sci* 1997;28(1):33-41.
- [32] Fattahi P, Saidi Mehrabad M, Jolai F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *J Intell Manuf* 2007;18(3):331-42.
- [33] Zhang L, Tang Q, Wu Z, Wang F. Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops. *Energy* 2017;138:210-27.
- [34] Özgüven C, Özbakır L, Yavuz Y. Mathematical models for job-shop scheduling problems with routing and process plan flexibility. *Appl Math Model* 2010;34(6):1539-48.
- [35] Liu Q, Li X, Gao L. Mathematical modeling and a hybrid evolutionary algorithm for process planning. *J Intell Manuf* 2021;32(3):781-97.
- [36] Zhang Y, Nee A. Application of genetic algorithms and simulated annealing in process planning optimization. In: Pham DT, editor. *Computational intelligence in manufacturing handbook*. Boca Raton: CRC Press; 2000. p. 243-68.
- [37] Li WD, McMahon CA. A simulated annealing-based optimization approach for integrated process planning and scheduling. *Int J Comput Integrated Manuf* 2007;20(1):80-95.