

虚拟 workflow 约束的时间-精确率迭代归约优化算法

罗智勇^{*①②} 朱梓豪^① 尤波^② 刘嘉辉^①

^①(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)

^②(哈尔滨理工大学机械动力工程学院 哈尔滨 150080)

摘要: 针对复杂产品生产业务调度这一问题, 该文运用 workflow 技术并以完工时间为约束, 提出一种虚拟迭代归约算法, 能较好地在完工时间约束下优化生产精确率。通过将各制约任务抽象虚拟成一个虚拟节点, 采用逆向迭代的求解方式, 确定了一条兼顾完工时间与生产精确率的调度路径。对比发现, 虚拟迭代归约算法对全局生产精确率有较大幅度的提高, 且通过改变截止期、任务数等参数可以提高算法的效率。

关键词: 生产调度; workflow; 虚拟迭代; 优化精确率

中图分类号: TP393

文献标识码: A

文章编号: 1009-5896(2018)08-2013-07

DOI: 10.11999/JEIT171038

Virtual Workflow Constrained Time-accuracy Optimization Algorithm Scheduling by Iterative Reduction

LUO Zhiyong^{①②} ZHU Zihao^① YOU Bo^② LIU Jiahui^①

^①(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

^②(School of Mechanical Engineering, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: For the problem of the production of complex operations, this paper uses workflow technology and takes the completion time as constraint, and proposes a Virtual Iterative Reduction Algorithm (VIRA) to achieve better production accuracy in the constraint completion time. By virtualizing tasks in mutual constraint into a virtual node, the algorithm uses inverse iterative way to determine a path that completion time and production accuracy get balance. By comparison, the virtual iterative reduction algorithm can increase the production accuracy in the constraint completion time, and it is found to improve the accuracy of the algorithm by changing the deadline, the number of tasks and other parameters.

Key words: Production scheduling; Workflow; Virtual iterative; Accuracy optimization

1 引言

workflow 作为一种在业务和生产中使用的利用计算机实现全自动化或半自动化的技术, 已经普遍被应用在相关的领域中。根据服务质量 QoS (Quality of Service) 的不同以及用户不同的业务需求来确定执行的服务为 workflow 的核心任务。工程项目的完工时间与精确率的平衡是一种 NP 难题^[1]。在 workflow 系统中, 每个任务的服务选择极为重要, 对整个业务的完工起着关键性影响。因此能否合理地选择任务中对应的服务, 是 workflow 优化调度的本质所在^[2]。目前的业务流程通过任务之间的约束关系构建 workflow 模型, 并对其中的每个任务分配对应的服务^[3]。追求局

部最优解的算法策略并不能解决全局最优解的问题, 为此需要设计更加合理的优化算法, 在充分考虑完工时间的约束下, 有效地提高业务流程的生产精确率^[4], 实现充分利用现有资源, 又能使得时间与精确率平衡。

workflow 包含着相关任务及其所对应的服务。由于每个服务又包含多个属性, 因此在对 workflow 调度时应综合考虑这些属性间的制约关系, 并实现动态平衡, 诸如完工时间、投资成本和生产精确率等。文献[5]针对用户自定义 QoS 分析了现有科学 workflow 在云平台 and 网格计算中的费用, 为日后研究人员提供了便利, 但主要还是费用优化方面的问题; 文献[6]在基于截止期时间下, 最小化 workflow 代价, 他们提出的方法可以在截止期充裕条件下, 降低云费用, 但不适用于网格 workflow; 文献[7]针对基于搜索策略的算法的高时间复杂度问题, 提出了启发式调度算法 DBCS, 并且仅有平方的时间复杂度, 但其研究的依旧是通常的 QoS 限制参数时间和费用; 文献[8]

收稿日期: 2017-11-03; 改回日期: 2018-04-16; 网络出版: 2018-05-11

*通信作者: 罗智勇 luozhiyongemail@sina.com

基金项目: 国家自然科学基金青年项目(61403109)

Foundation Item: The National Natural Science Foundation of China (61403109)

在模拟的云服务上进行 workflow 调度, 具有良好的节省货币成本效果, 并且和著名的启发式调度算法在相同的截止期约束下能节约同样的成本, 局限性则是并未在现实的云环境中实施。

在国内, 近些年也开展了一些关于在单目标约束下优化其他目标的研究工作。文献[9]针对原有时间约束网资源不足的问题, 拓展了时间约束网的定义, 并且调整了时间约束策略, 使得 workflow 调度更合理; 文献[10]将同平台的功能模块形式化成服务应用, 并结合基于 Markov 链技术的工作流策略有效地降低了成本; 文献[11]提出了双向调度算法, 前向考虑尽可能较早地执行任务, 后向以最迟开始时间进行调度, 为本文设置任务自由度参数提供一定参考; 文献[12]考虑了任务执行之间的时序关系提出了改进的蚁群混合算法, 而本文将局部时序关系抽象化虚拟成一个虚拟节点; 文献[13]对关键路径进行了粒子初始化并且在搜索阶段实施了筛选处理, 提出了基于粒子群算法的最佳调度方法, 同时有效地降低了搜索的计算时间。

本文的主要贡献在于提出了一种在截止期限定内, 实现整体业务流程生产精确率得到优化的虚拟迭代归约算法(VIRA)。该算法通过增加虚拟节点从而达到简化 workflow 模型的目的, 并采用逆向层层迭代的方式解决了以局部最优解换取全局最优解的问题, 较好地实现了影响复杂产品生产业务调度各因素的动态平衡。本文所讨论的如何在完工时间约束下对生产精确率进行全局优化也是一个多任务调度的研究热点。

2 问题描述

2.1 工作流的相关定义

定义 1 任务池 N : 表示为 $N = \{n_1, n_2, \dots, n_n\}$, 指整个业务所包含的全部任务集合, 其中, λ_i^* 表示任务 n_i 的出度, $^*\lambda_i$ 表示任务 n_i 的入度。

定义 2 条件集合 C : 表示为 $C = \{c_1, c_2, \dots, c_z\}$, 指任务之间存在的完成必备条件集合。

定义 3 服务池 S_i : 表示为 $S_i = \{s_1, s_2, \dots, s_m\}$, 指能完成某个任务 n_i ($n_i \in N$) 的所有服务组成的集合。

定义 4 服务属性 q_{ij} : 表示为 $q_{ij} = (a_{ij}, t_{ij})$, 指任务 n_i 对应的服务集合 S_i 中, 每个服务 s_j 所具有的属性。其中, a_{ij} 表示任务 n_i 对应的服务集合 S_i 中 s_j 服务完成 n_i 能够达到的生产精确率, t_{ij} 表示所用的完成时间。

定义 5 工作流模型 W : 表示为 $W =$

$\langle N, E, C, DP \rangle$, 是一个网格环境下的有向图 DAG。其中, N 表示任务集合; E 表示模型的有向边集合; C 表示各任务之间序列条件的集合, 即同一条路径上只有当前驱任务 n_{i-1} 全部完成, 任务 n_i 才可以执行; DP 表示检测节点, 当精确率高于 σ 时允许通过, 否则返回重新加工, 期间所消耗的生产时间应该累加。此外, 工作流中设置虚拟任务 n_s 和 n_e , 分别表示整个业务的开始和结束。

定义 6 任务完成时间 T_i : 表示为

$$T_i = \text{Max} \{ \dots, T_j, \dots \} + t_i, \quad k = 1, 2, \dots, ^*\lambda_i \quad (1)$$

指业务进行到任务 n_i 所花费的累积时间。其中, T_j 为任务 n_i 的直接前驱任务 n_j 的累积执行时间, $j < i$; t_i 表示完成任务 n_i 自身花费的时间。

定义 7 生产精确率 A_i : 表示为

$$A_i = \prod_{n_j \in N'} a_j \quad (2)$$

指业务进行到任务 n_i 所累积的生产精确率。其中, N' 表示业务进行到任务 n_i 时的所有已完成任务的集合, a_j 表示任务 n_j 自身的生产精确率且 $a_j \in [0, 1]$, 显然 $a_s = a_e = 1$ 。

定义 8 截止期 ψ : 指工作流的最晚完成时间, 工作流由任务集合 $N' = \{n_1, n_2, \dots, n_i\}$ 组成, 其中, $N' \in N$, 且当 $i = n$ 时, 表示整个业务流程的最晚完成时间。

2.2 工作流模型生成算法

工作流模型能够有效地描述生产加工型业务的工序情况, 对求得约束时间下的有效工序路径起着关键性作用^[14]。本文在以往研究的有向无环图 DAG 基础上, 结合工作流中任务的偏序关系, 设计生成工作流模型算法 WFG 的步骤如下:

- (1) 根据业务流程的需要, 搜集任务和该任务所对应的序列条件, 并分别加入集合 N 和 C 中;
- (2) 提取集合 N 中的任务 n_i , 在 C 中查找该任务的序列条件 c_i , 并在集合 N 和 C 中删除 n_i 和 c_i ;
- (3) 根据任务 n_i 的序列条件 c_i , 查找后继节点, 并加以连接, 对存在需要检测的任务, 添加检测节点 DP , 完成循环结构的组建;
- (4) 重复步骤(2), (3), 直到集合 N 中元素为空;
- (5) 输出工作流模型 W 。

经分析, 工作流模型生成算法 WFG 的时间复杂度为 $O(n^2)$ 。

2.3 工作流模型实例

某机械厂需要利用供应链技术实现生产资源的合理分配, 进而提高整个业务生产效率, 该调度系

统包括生产、组装等多个模块。有些模块必须串行且存在相互制约反馈关系，从而完成整个生产。通过对该零件加工流程中添加检测点，提高其生产精确率；有些模块由多个部门相互配合且可并行执行业务流程，经分析该公司的生产调度过程存在 20 个任务节点，制约条件抽象为集合 C ，每个任务 n_i 对应一个服务池 S_i 。将这些信息输入工作流模型生成算法 WFG 中，可得图 1 所对应的工作流模型，且约定截止期 $\psi = 30$ 。

该工作流模型各任务的服务参数如表1所示。

表 1 各任务的服务池

任务 n_i	服务池 S (生产精确率 $\times 100\%$, 时间(d))
n_1	(0.73,1),(0.86,2),(0.90,4)
n_2	(0.72,1),(0.91,2)
n_3	(0.78,2),(0.87,3),(0.89,4)
n_4	(0.96,1),(0.97,2)
n_5	(0.95,2),(0.97,3),(0.99,4)
n_6	(0.92,2),(0.94,4),(0.98,6)
n_7	(0.94,3),(0.97,5)
n_8	(0.92,4),(0.95,5),(0.99,6)
n_9	(0.96,3),(0.98,4),(0.99,6)
n_{10}	(0.95,2),(0.97,4),(0.99,9)
n_{11}	(0.95,1),(0.97,3)
n_{12}	(0.84,2),(0.90,5)
n_{13}	(0.81,3),(0.86,5)
n_{14}	(0.93,3),(0.97,4),(0.99,7)
n_{15}	(0.96,2),(0.98,3)
n_{16}	(0.94,2),(0.96,3),(0.98,5)
n_{17}	(0.93,2),(0.95,3),(0.96,8)
n_{18}	(0.91,6),(0.93,7)
n_{19}	(0.93,4),(0.95,5),(0.99,6)
n_{20}	(0.96,4),(0.99,5)

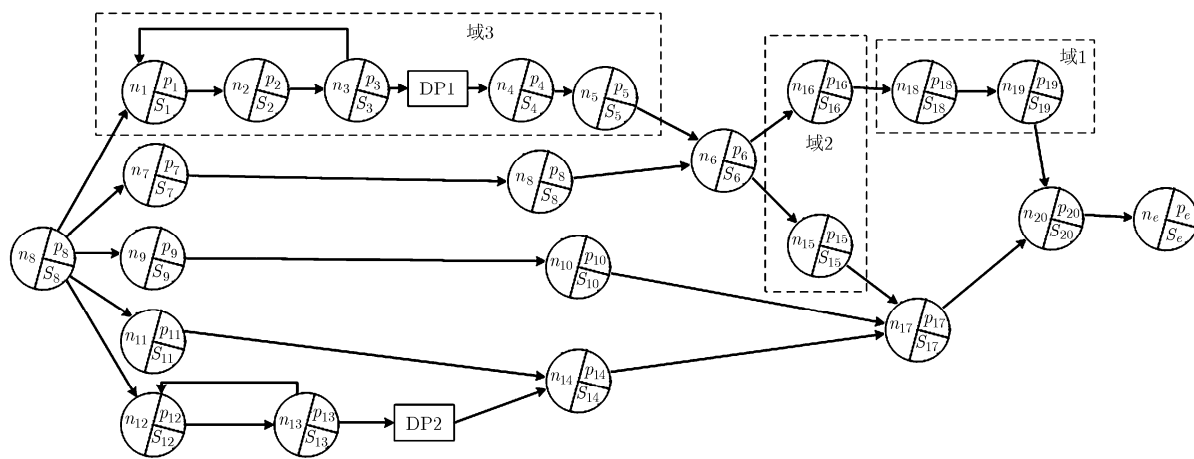


图1 工作流模型实例

2.4 工作流模型的约束分析

在图1所示的模型中，需增加部分约束，才可在有效地完成截止期 ψ 下对生产精确率 A_k 优化调度的目标。这些约束如式(3)和式(4)所示。

$$\left. \begin{aligned} A_q &= \prod_{n_i \in N'} l_{ij} a_{ij} \\ \text{s.t. } \sum_{j=1}^m l_{ij} &= 1, S_i = \{s_1, s_2, \dots, s_m\} \\ l_{ij} &\in \{0, 1\} \end{aligned} \right\} \quad (3)$$

式中， N' 表示业务进行到任务 n_q 时的所有已完成任务的集合； l_{ij} 表示当执行到任务 n_i 时，是否选用其对应服务池集合 S_i 中的服务 s_j (s_j 具有唯一性)来执行；若 $q = e$ ，即 $n_q = n_e$ 时，表示整个业务执行完毕并用 A_e 表示整个业务能够达到的生产精确率。式(3)为本文的启发目标函数，为优化对象。

$$T_q = \text{Max} \{ \dots, T_p, \dots \} + l_{qj} t_{qj} < \psi, k = 1, 2, \dots, * \lambda_q \quad (4)$$

式中， T_p 为任务 n_q 的直接前驱任务 n_p 的累积执行时间， $p < q$ ； ψ 表示业务流程的截止期。若 $q = e$ ，即 $n_q = n_e$ 时，表示整个业务执行完毕并用 T_e 表示整个业务所用时间，且满足 $T_e \leq \psi$ 的约束关系。式(4)为约束条件，可辅助式(3)确定截止期下生产精确率 A_e 得到优化的路径。

3 传统最小关键路径算法调度工作流

定义 9 任务自由度 $DFn_i \in [BEn_i, ENn_i]$ ：指由若干个任务组成的业务流程，在截止期 ψ 下每个任务的开始时间，其中 BEn_i 表示任务 n_i 的最早开始执行时间， ENn_i 表示任务 n_i 的最迟开始执行时间。 BEn_i 和 ENn_i 可由式(5)求得

$$\left. \begin{aligned} \text{BE}n_i &= \text{Max} \{ \text{BE}n_{i-1} + \text{Min}(t_{ij}) \} \\ \text{BE}n_1 &= 0, \text{BE}n_{i-1} \in \{ \dots, \text{BE}n_q, \dots \} \\ \text{EN}n_i &= \text{Min} \{ \text{EN}n_{i+1} - \text{Min}(t_{ij}) \} \\ \text{EN}n_n &= \psi, \text{EN}n_{i+1} \in \{ \dots, \text{EN}n_p, \dots \} \end{aligned} \right\} \quad (5)$$

式中, $k = 1, 2, \dots, \lambda_i$; $l = 1, 2, \dots, \lambda_i^*$; $\text{BE}n_q$ 为任务 n_i 的直接前驱任务 n_q 的最早开始执行时间; $\text{EN}n_p$ 为任务 n_i 的直接后继任务 n_p 的最迟开始执行时间。

在工作流 W 中, 在截止期 ψ 约束下, 通过最小完工时间 T_e 确定每个任务 n_i 的服务 s_j 选择, 以此完成整个业务流程, 该方法即为传统的最小关键路径算法。在此策略的服务分配下, 利用文献[12]的关键路径算法CPM的式(6)可求得每个任务 n_i 的最早开始时间 α_i 与最迟完工时间 β_i 。

$$\left. \begin{aligned} \alpha_i &= \text{BE}_i \\ \beta_i &= \text{Min} \{ \text{BE}n_j \}, \text{BE}n_j \in \{ \dots, \text{BE}n_j, \dots \} \end{aligned} \right\} \quad (6)$$

其中, $k = 1, 2, \dots, \lambda_i$; $\text{EN}n_j$ 为任务 n_i 的直接后继任务 n_j 的最迟开始执行时间。

4 虚拟迭代归约算法调度 workflow

虚拟迭代归约算法的核心思想是考虑生产业务的生产精确率 A_i 和完成时间 T_i 的制约关系, 在满足截止期 ψ 的约束下, 通过组合循环任务或者并行任务集合为虚拟节点, 从而简化 workflow 调度过程, 并采用归约算法进行迭代求解, 找到一条生产精确率 A_e 与时间 T_e 的动态平衡路径。

4.1 相关定义

定义10 虚拟任务 $n_{[i-j]}$ 或 $n_{[i,j]}$: 表示对多个任务进行组合, 进而看作整体, 成为一个虚拟任务。 $n_{[i-j]}$ 表示任务 n_i 至任务 n_j 之间多任务的组合, $n_{[i,j]}$ 表示任务 n_i , n_j 两任务的组合。在对循环局部 workflow 进行组合时, 应满足初始生产精确率为1且是串行路径的约束条件。

性质1 处于检测节点 DP 前的任务集合所构成的虚拟节点在相同截止期下生产精确率 A_i 取值越大越好。

证明 对于虚拟节点在一个截止期下取得的生产精确率 A_i , 则若 A_i 低于检测节点 DP 的通过率 σ , 则进行循环, 由此产生的失效精确率为 $(1 - A_i)$, 经过重新加工后的生产精确率为 $A_i(1 - A_i)$, 最终该虚拟节点的累积生产精确率为 $A_i(1 - A_i) + A_i = -A_i^2 + 2A_i$ 。由函数 $f(A_i) = -A_i^2 + 2A_i$ 的第1象限的曲线图

可知, 在区间 $[0, 1]$ 内满足单调递增关系。因此, 生产精确率 A_i 取值越大最终的生产精确率也越大。

证毕

定义11 异路径SDP: 指 workflow 模型 W 中某任务 n_i 不能与其他任务 n_j 进行组合构成虚拟任务的那部分出度路径称之为异路径。例如, workflow 模型 W 的域1中, 可将任务 n_{15} , n_{16} 进行组合, 但由于任务 n_{16} 有另一条出度路径 n_{18} , 即域2, 因此任务集合 $\{n_{18}, n_{19}\}$ 即为异路径。

定义12 执行域 ω : 表示为 $\omega = [\psi_{\min}, \psi_{\max}]$, 其中, ψ_{\min} 为最小截止期, ψ_{\max} 为最大截止期。由集合 N' 组成的业务流程构成的时间区间为集合 N' 的执行域, 任务集合 N' 是任务集合 N 的任意子集。任务或虚拟任务 n_i , 其所对应的执行域 ω 可由式(7)求得

$$\left. \begin{aligned} \psi_{\min} &= \max \{ \text{BE}n_j - \text{BE}n_i \} \\ \psi_{\max} &= \min \{ \text{EN}n_j - \text{BE}n_i \} \end{aligned} \right\} \quad (7)$$

4.2 算法描述

算法采用分层的策略, 对 workflow 模型中的节点 n_i 划分任务自由度 $\text{DF}n_i$ 。在分析 workflow 模型 W 后, 进行整体结构优化, 将由反馈环结构或并行结构组成的任务集合, 转换为虚拟节点, 对该节点设置执行域 ω 进行计算, 通过层层迭代求解方式, 确定截止期 ψ 下对整体生产精确率 A_e 优化的路径。

假设存在 n 个任务, 在截止期 ψ 下, 令 $f(n_i, t)$ 表示任务 n_i 在 t 时刻开始时能够达到的最大生产精确率, 则对于任务 n_n , 在 $\text{DF}n_n$ 内, 可通过式(8)求得不同时刻开始时的一个生产精确率最大值 a_n 。

$$\left. \begin{aligned} f(n_i, t) &= \max \{ a_{ij} \} \\ t &\in [\text{BE}n_i, \text{EN}n_i], t + t_{ij} \leq \psi \end{aligned} \right\} \quad (8)$$

假设任务 n_{i-1} 为任务 n_i 的直接前驱, 则在任务自由度 $\text{DF}n_{i-1}$ 中, 利用式(8)计算最后一层任务 n_n 的最大生产精确率 a_n 再结合式(9)可反向求得任务 n_{n-1} 在不同时刻开始时的最大生产精确率 $a_n \times a_{n-1}$, 进一步反向层层迭代可求得 workflow W 中每个任务在不用时刻开始时能够达到的最大准确率 $a_n \times \dots \times a_i$, 进而求得在截止期 ψ 下整个 workflow W 能够达到的最大生产精确率 A_e 。

$$\left. \begin{aligned} f(n_{i-1}, t_{i-1}) &= \max \{ f(n_i, t_{i-1} + t_{i-1j}) \times a_{i-1j} \} \\ t_{i-1} &\in [\text{BE}n_{i-1}, \text{EN}n_{i-1}] \end{aligned} \right\} \quad (9)$$

定理1 对于一个任务集合 N' , 若存在 $f(n, t)$ 为截止期 ψ 下的最大生产精确率, 则 $f(n, t + t_0)$ 为在截止期 $\psi - t_0$ 下的最大生产精确率。

证明 已知在截止期 ψ 下的最大生产精确率为 $f(n, t)$, 假设 $\psi - t_0$ 下的最大生产精确率为 $f(n, t + t_1)$, 其中 $(t_1 < t_0)$, 若 $f(n, t + t_0) = f(n, t + t_1)$, 则定理1成立; 若 $f(n, t + t_0) < f(n, t + t_1)$, 则最终的完工时间 ψ' 必定大于 $\psi - t_0$, 假设与规定矛盾。假设 $\psi - t_0$ 下的最大生产精确率为 $f(n, t + t_2)$, 其中 $(t_2 > t_0)$, 则 $f(n, t + t_0) \geq f(n, t + t_2)$, $f(n, t + t_0)$ 为最大生产精确率。综上所述, 定理1成立。

基于上述策略, 归纳虚拟迭代归约算法VIRA的步骤如下:

(1)根据总截止期 ψ 并结合式(5)求出 workflow 模型 W 中每个任务自由度 DFn_i ;

(2)从开始任务 n_s 遍历全部任务集 N , 将出度 λ^* 和入度 λ 大于1的任务加入到队列中, 其中循环结构里面的任务不加入;

(3)从队中找出第1个出度 λ^* 大于1的任务, 并查找与该任务最近的入度任务, 将其中间的任务集合组合成虚拟任务, 利用式(7)求得对应的执行域 ω , 并对不同截止期下每条串行路径求解最大生产精确率; 如遇循环结构, 则将循环结构的任务集合组合成新的虚拟任务, 并求出不同截止期下能够达到的最大生产精确率; 如遇到串行路径中的某个任务连接其他入度 λ 大于1的任务, 并将其后继路径划分为异路径SDP;

(4)执行步骤(2), 直到队列中没有入度 λ 或出度 λ^* 大于1的任务;

(5)对异路径SDP之外的所有任务序列求解在截止期下的最大生产精确率;

(6)对已完成的任务计算异路径的执行域, 并求解异路径截止期下的最大生产精确率, 最终确定整个业务流程的生产精确率。

通过分析, 时间复杂度为 $O(nm^n)$ 。

4.3 虚拟迭代归约算法局部实例

图1所示域3中的任务集合代表一段循环 workflow。通过式(5)和式(7)的计算, 该流程执行域 ω 中存在一个截止期为12, 现规定检测节点DP的通过率为0.95, 其中当生产精确率低于 $\sigma_1 = 0.5$ 时, 则需要多花费时间2进行加工; 高于 σ_1 且低于 $\sigma_2 = 0.95$ 则多花费时间1进行加工, 处理过程如下:

(1)将任务 n_1 、任务 n_2 、任务 n_3 组合成虚拟任务 $n_{[1-3]}$;

(2)调用式(5)计算任务 n_4 的自由度 $DFn_4 = [4, 9]$;

(3)利用任务自由度 DFn_4 并结合式(7)可得虚拟

节点 $n_{[1-3]}$ 的执行域为 $[4, 9]$, 进而求得虚拟任务的不同完成时间及其所能够达到的最大生产精确率:

求得的精确率分别为从时间点0开始的0.713, 从时间点1开始的0.697, 从时间点2开始的0.681, 从时间点3开始的0.610, 从时间点4开始的0.518, 从时间点5开始的0.410。

利用定理1可得如下所示的不同截止期下累积生产精确率 A_3 的计算结果:

截止期为7的精确率为0.99, 截止期为6的精确率为0.98, 截止期为5的精确率为0.95, 截止期为4的精确率为0.98。

综上所述, 对于虚拟组合任务 $n_{[1-3]}$, 存在3个虚拟服务, 即: $\{(7, 0.95), (8, 0.98), (9, 0.99)\}$ 。

(4)将虚拟组合任务与其他的任务进行归约。对于任务 $n_{[1-3]}$, n_4 , n_5 的任务自由度分别为 $[0, 2]$, $[7, 9]$, $[8, 10]$, 在截止期 $\psi = 12$ 的约束下结合算法VIRA, 求解最大生产精确率, 结果如下:

从时间点0开始的精确率为0.912, 时间点1开始的精确率为0.894, 时间点2开始的精确率为0.866。

综上, 在截止期 $\psi = 12$ 的约束下, 图1所示域3任务虚拟后的最大累积生产精确率 A_5 为0.912。对于虚拟组合任务 $n_{[1-5]}$, 存在3个虚拟服务, 即: $\{(10, 0.866), (11, 0.894), (12, 0.912)\}$, 自由度为 $[0, 2]$ 。

5 案例分析

为评估虚拟迭代归约算法VIRA的性能, 本文设计了图1所示 workflow 模型案例, 并与传统最小关键路径算法CPM进行了生产精确率提高方面的对比。案例假设生产业务流程的总截止期 $\psi = 30$, 定义检测点任务为DP ($\sigma_1 = 0.5$, $\sigma_2 = 0.95$) 并满足的要求是: 当生产精确率低于 σ_1 时, 则需要多花费时间2进行重新加工; 当生产精确率高于 σ_1 且低于 σ_2 则需多花费时间1进行重新加工; 当生产精确率高于 σ_2 则通过检测点任务, 开始执行后继任务。

5.1 传统最小关键路径算法分析

以优化最小完工时间为目标的传统 workflow 调度策略可节省更多的加工时间, 这种策略在 workflow 中每次选择最小的加工时间来进行调度。由最小关键路径算法CPM可知最小完工时间 $T_{c1} = 28$, 对非关键路径上的每个任务在对应的截止期下选择最大生产精确率服务, 可得最终生产精确率 $A_{c1} = 0.512$ 。

5.2 虚拟迭代归约算法分析

将案例 workflow 模型 W 及其所对应的服务池数据输入虚拟迭代归约算法 VIRA, 可计算出图 2 所示的归约过程:

图 2 显示, 算法 VIRA 先对能组合的任务序列进行处理, 然后求出合并后的最大生产精确率。对已确定服务的任务来计算异路径的执行域, 以此来求解异路径的最大生产精确率。图 2 计算出 workflow 模型 W 的最终完工时间 $T_{e2} = 30$, 所达到的最终生产精确率 $A_{e2} = 0.608$ 。

6 其他参数对算法性能的影响

6.1 任务数目的影响

workflow 模型的总体生产精确率是通过每个任务的生产精确率累乘所得, 随着任务数目的增多, 对模型的生产精确率以及算法的性能都会有影响。随机生成 $\{10, 20, 30, 40\}$ 个任务节点, 任务 n_i 的服务池 S_i 的个数取自区间 $[2, 5]$ 的任意整数, 服务池带有不同的属性参数, 以最小完工时间增加 10% 作为业务流程截期。通过对任务数的改变分析对算法 CPM 和算法 VIRA 的性能影响, 其对比变化规律如图 3 所示。

图 3 显示, 随着任务数目的增加, 两种算法的总体生产精确率均在下降。但图 3 同时显示, 随着任务数目的变化, 本文提出的算法 VIRA 较算法 CPM 在最终最大生产精确率 A_e 上有所提高, 分别为: 6.7%, 9.3%, 18%, 26%。

6.2 截止期的影响

截止期是业务流程的最迟完工时间, 通常情况随着截止期的增大, 生产精确率也会随之提高。随机生成 $\{10, 20, 30\}$ 个任务节点, 任务 n_i 的服务池 S_i 个数取自区间 $[2, 5]$ 的任意整数, 服务池带有不同的属性参数, 对每个任务序列分别以最小完工时间加 5%, 10%, 15%, 20%, 25%, 30% 作为截止期, 分析对算法 VIRA 性能的影响, 如图 4 所示。

图 4 显示, 在不同的截止期约束下, 各种业

务流程经过算法 VIRA 的优化, 使得生产精确率有所不同。随着截止期的增大, 算法 VIRA 的生产精确率明显提高。

通过 6.1 节与 6.2 节的对比发现: 算法 VIRA 相对于算法 CPM 的生产精确率总体提高 $K = (A_{e2} - A_{e1}) / A_{e1} \times 100\% = 18.7\%$, 显示了算法 VIRA 在约束时间下对生产精确率的优化效果。

6.3 循环结构数目的影响

循环结构数目的增加, 导致局部业务流程的生产精确率随之提高, 但同时也会消耗更多的生产时间, 进而延误工期。随机生成 $\{20, 30, 40\}$ 个任务节点, 任务 n_i 的服务池 S_i 的个数取自区间 $[2, 5]$ 的任意整数, 服务池带有不同的属性参数, 以最小完工时间增加 10% 作为业务流程截止期, 分别对模型 W 添加 1 至 5 个循环结构, 得出循环结构数对算法 VIRA 的影响如图 5 所示。

图 5 显示, 随着循环结构数目的增加, 在不超过截止期约束下不同业务流程的生产精确率也随之提高。

7 结束语

针对 workflow 约束时间下难于提高生产精确率的问题, 本文先将业务流程中的任务集合抽象为 workflow 模型, 在该模型上通过算法 CPM 对业务流程进行求解, 针对该算法不能充分利用完工时间的弊端, 提出了在约束时间下的虚拟迭代归约算法 VIRA。算法 VIRA 对复杂结构进行组合, 抽象为虚拟任务, 求出其执行域, 进而求解不同截止期下的最大生产精确率, 最终通过层层迭代简化 workflow 模型以达到用串行执行的模式来优化整体 workflow 路径的目的。模拟数据表明, 算法 VIRA 相对于算法 CPM 在性能方面提高了 18.7%。此外, 本文还讨论了影响算法 VIRA 性能的其他参数。本文的不足之处在于未将生产成本加入 workflow 模型中, 未找到一条在约束时间下时间、成本和生产精确率平衡的路径, 笔者将在后续的研究过程中加以解决。

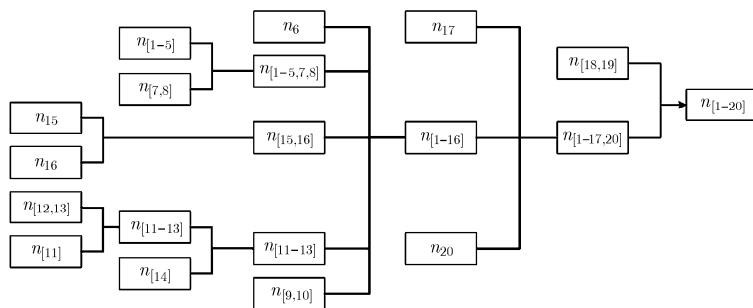


图 2 案例 workflow 模型 W 的归约过程

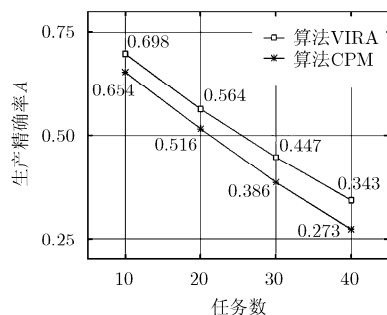


图3 不同任务数目下算法VIRA的性能提高率

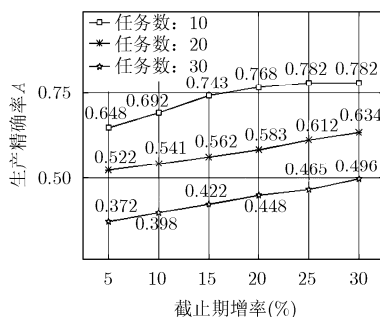


图4 不同截止期对算法VIRA性能的影响

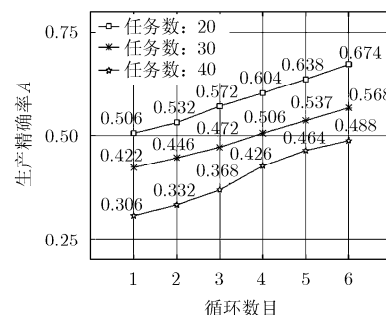


图5 循环结构数目对算法VIRA性能的影响

参考文献

- [1] DE P, DUNNE E J, GHOSH J B, et al. Complexity of the discrete time-cost trade-off problem for project networks[J]. *Operations Research*, 1997, 45(2): 302-306. doi: 10.1287/opre.45.2.302.
- [2] KUMAR A, DIJKMAN R, and SONG M. Optimal resource assignment in workflows for maximizing cooperation[C]. 11th International Conference, BPM 2013, Beijing, China, 2013: 26-30. doi: 10.1007/978-3-642-40176-3_20.
- [3] BUYYA R, GIDDY J, and ABRAMSON D. An evaluation of economy-based resource trading and scheduling on computational power grids for parameter sweep applications [C]. Proceedings of the 2nd International Workshop on Active Middleware Services, Pittsburgh, USA, 2000: 221-230. doi: 10.1007/978-1-4419-8648-1_19.
- [4] DELDARI A, NAGHIBZADEH M, and ABRISHAMI S. CCA: A deadline-constrained workflow scheduling algorithm for multicore resources on the cloud[J]. *The Journal of Supercomputing*, 2017, 73(2): 756-781. doi: 10.1007/s11227-016-1789-5.
- [5] ALKHANAK E N, LEE S P, REZAEI R, et al. Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues[J]. *Journal of Systems and Software*, 2016, 133: 1-26. doi: 10.1016/j.jss.2015.11.023.
- [6] VIRIYAPANT K and SMANCHAT S. A deadline-constrained scheduling for dynamic multi-instances parameter sweep workflow[C]. 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), Okayama, Japan, 2016: 1-6. doi: 10.1109/ICIS.2016.7550820.
- [7] ARABNEJA H, BARBOSA J G, and PRODAN R. Low-time complexity budget-deadline constrained workflow scheduling on heterogeneous resources[J]. *Future Generation Computer Systems*, 2016, 55: 29-40. doi: 10.1016/j.future.2015.07.021.
- [8] VERMA A and KAUSHAL S. Cost-time efficient scheduling plan for executing workflows in the cloud[J]. *Journal of Grid Computing*, 2015, 13(4): 1-12. doi: 10.1007/s10723-015-9344-9.
- [9] 冯复剑. 时间约束工作流的可调度性分析[J]. *计算机工程与应用*, 2016, 52(12): 26-30. doi: 10.3778/j.issn.1002-8331.1511-0313.
- [10] 武星, 卓少剑, 张武. 成本最优化工作流技术驱动的研发协同软件即服务应用[J]. *计算机集成制造系统*, 2013, 19(8): 1748-1754.
- [11] WU Xing, ZHUO Shaojian, and ZHANG Wu. Cost optimization workflow-driven SaaS for collaborative research and development[J]. *Computer Integrated Manufacturing Systems*, 2013, 19(8): 1748-1754.
- [12] 张佩云, 凤麒. 一种云计算环境下的工作流双向调度算法[J]. *计算机科学*, 2015, 42(11): 425-430.
- [13] ZHANG Peiyun and FENG Qi. Method of workflow bi-directional scheduling in cloud computing environment[J]. *Computer Science*, 2015, 42(11): 425-430.
- [14] 梁合兰, 杜彦华, 李苏剑. 时序约束下科学工作流的动态调度研究[J]. *系统工程理论与实践*, 2015(9): 2410-2421. doi: 10.12011/1000-6788(2015)9-2410.
- [15] LIANG Helan, DU Yanhua, and LI Sujian. Research on dynamic scheduling of scientific workflows with temporal constraints [J]. *Systems Engineering-Theory & Practice*, 2015(9): 2410-2421. doi: 10.12011/1000-6788(2015)9-2410.
- [16] 曹斌, 王小统, 熊丽荣, 等. 时间约束云工作流调度的粒子群搜索方法[J]. *计算机集成制造系统*, 2016, 22(2): 372-380.
- [17] CAO Bin, WANG Xiaotong, XIONG Lirong, et al. Searching method for particle swarm optimization of cloud workflow scheduling with time constraint[J]. *Computer Integrated Manufacturing Systems*, 2016, 22(2): 372-380.
- [18] 刘中金, 卓子寒, 何跃鹰, 等. 一种基于动态配额的虚拟网带宽公平调度算法[J]. *电子与信息学报*, 2016, 38(10): 2654-2659. doi: 10.11999/JEIT151485.
- [19] LIU Zhongjin, ZHUO Zihan, HE Yaoying, et al. Dynamical weighted scheduling algorithm supporting fair bandwidth allocation of virtual networks[J]. *Journal of Electronics & Information Technology*, 2016, 38(10): 2654-2659. doi: 10.11999/JEIT151485.
- [20] 罗智勇: 男, 1978年生, 副教授, 研究方向为供应链调度、生产工艺优化和生产安全控制.
- [21] 朱梓豪: 男, 1994年生, 硕士生, 研究方向为供应链调度.
- [22] 尤波: 男, 1962年生, 教授, 研究方向为生产工艺优化、机器人控制.
- [23] 刘嘉辉: 男, 1974年生, 副教授, 研究方向为供应链调度、生产安全控制.