

基于串归约的时间约束下 workflow 精确率优化算法

罗智勇^{1,2}, 朱梓豪¹, 尤波², 苗世迪¹

(1. 哈尔滨理工大学 计算机科学与技术学院 黑龙江 哈尔滨 150080;

2. 哈尔滨理工大学 机械动力工程学院 黑龙江 哈尔滨 150080)

摘要: 复杂产品的业务调度依赖于完工时间和精确率等属性,追求单一的目标不能保证这两个方面的平衡。传统的算法经常是以花费时间最小或者保证完工质量最佳为目标,这样会导致产品质量过低或者完成产品所用的时间过多。针对这种弊端,提出了约束时间下精确率串归约优化算法 SRA,通过约束任务得到活动区间来进行优化路径。最后在典型案例中,分别利用了传统的单向目标算法和串归约算法求解对应的路径,并对算法 SRA 优化效果的其他参数进行了研究。研究表明,这种算法具有简单、高效、方便执行等优点。

关键词: 工作流调度; 时间一致性; 准确率优化; 截止期

DOI: 10.15938/j.jhust.2018.05.012

中图分类号: TP393 **文献标志码:** A **文章编号:** 1007-2683(2018)05-0068-07

Optimization Algorithm of Workflow's Accuracy Based on Serial Reduction under Constraint Time

LUO Zhi-yong^{1,2}, ZHU Zi-hao¹, YOU Bo², MIAO Shi-di¹

(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: Complex business scheduling depends on the completion of the time and accuracy and other attributes, the pursuit of a single goal cannot guarantee the balance of the aspects. Traditional algorithms often take the least time or the best quality as goal, it leads to the low quality low or the more time taken to complete the product. Aiming at this drawback, the SRA of the exact rate with string reduction optimization under the constraint time proposed, and obtains the active interval by the constraint task. Finally, in the typical case, the traditional one-way target algorithm and the string reduction algorithm are used to solve the corresponding path respectively, and analyzed the other parameters that affect the performance of SRA. The research shows that this algorithm has the advantages of simple, efficient and convenient implementation.

Keywords: workflow scheduling; time consistent; accuracy optimization; deadline

收稿日期: 2016-10-26

基金项目: 国家自然科学基金青年基金(61403109)。

作者简介: 朱梓豪(1994—)男,硕士研究生;

尤波(1962—)男,教授,博士研究生导师。

通信作者: 罗智勇(1978—)男,博士研究生,副教授,硕士研究生导师, E-mail: luozhiyongemail@163.com.

0 引言

复杂产品工作流是以网络流为基础,结合调度技术,并以服务为基本构成单位的一种架构技术^[1-2],主要用于表征多服务相互协调工作来完成整个产品加工的过程。工作流将业务流程抽象化,划分为诸多工序,结合工序中服务属性,确定最佳完工路径。在工作流中,每个任务中服务的选择极为重要,它决定着整个业务的效率及完工质量。能否合理地选择任务中对应的服务,对整个业务流程的有效执行起着重要的影响。

目前企业在复杂项目的实施过程中,工作流模型所需要的资源类型在模型的构建中就已经确定了,在实际的项目下发后,企业也能够及时地分配好任务,并利用对应的服务展开工作。但是项目流程作为一个整体,并被分成几个工序,往往因为一个或者几个工序存在问题,诸如追求效率从而忽略了服务质量或者追求服务质量而拖延了完工时间,使得整个项目的完工时间和完工质量失衡。因此单方面地追求效率,将会导致准确率过低,相反单方面追求服务准确率,反而影响复杂产品的完工时间。为此,应该使用更合理的方法,在业务流程规定的截止期内,选择一条能够有效提升复杂项目准确率的路径,达到既能充分利用资源,又能使效率和准确率实现平衡的目的^[3]。本文提出了一种在限定完工时间内,通过串归约算法(serial reduction algorithm, SRA)来优化生产精确率的解决办法,达到时间和生产精确率的相对平衡。

1 相关工作

工作流模型主要由任务和服务组成,其中任务代表了加工工序,而服务则代表了完成该任务所用的成本、工期和生产精度等属性^[4],合理平衡服务中的各属性参数对优化工作流模型中的任务调度具有关键性作用^[5-6]。随着工作流技术的不断完善,已不再以单纯的优化工期为研究目的,而是向着动态平衡工期和生产精度等更深层次优化提出了严格的要求。文[7]提出在时间限制条件下优化成本的问题实际上是一种 NP-hard 问题;文[8]基于时间约束下提出了3种降低成本的启发式算法;文[9]对截止期约束下的优化成本问题提出了正向分层的策

略,在每个环节的约束时间内选择费用最小的服务;文[10]提出了串行和并行的实例集合,在网络模型的基础上进行建模;文[11]在业务的网络图中,将整个业务分解为多个环节,对每个分量做出了时间约束并进行优化;文[12]中 Pawel 通过对工作流程中成本与时间的分析,对每个任务选择合适的服务,优化了工作流的线性组合;文[13]通过 AMPL 模型实现了约束时间下成本的最小化;文[14]通过并行方式进行资源整合,降低了成本并且提高了质量。

基于在约束时间下降低成本的问题,国内也开展了相应的研究。文[15]采用动态资源选择策略,在相对较小的完工时间范围内,实现业务成本的最小化。文[16]运用了基于 Markov 链技术实现了工作流成本最小化。可见在工作流模型的服务属性中,确实存在着时间、成本和服务质量问题,如何保证在业务流程约束时间下对完工质量进行优化也是一个研究热点^[17]。

2 问题描述

2.1 相关定义

定义1 工作流模型。若用 M 表示模型,则 M 可形式化为二元组 $M = (R, W_r)$, 其中 R 为任务池,由工序所涉及到的所有工作任务构成的集合,可表示为 $R = (r_1, r_2, \dots, r_i, \dots, r_n)$, n 为任务数; W_r 为任务 r_i 的服务池,是由完成该任务可选用的服务构成的集合,可表示为 $W_i = (w_1, w_2, \dots, w_j, \dots, w_m)$, m 为服务数;在具体的业务过程中,任务 r_i 与任务 r_j 之间($i \neq j$)存在顺序关系,且每个任务 r 对应着服务池 W_r ,通过每个任务对应服务的合理搭配,实现整个工作流;任务池 r 中任意任务 r 所对应的 W_r 中的数目称为服务池的秩,可表示为 z_r 。

定义2 工作流图 F 。是一个有向无环图 DAG,且 $F = \{w_{rk}, rh_{rk}, L\}$, 其中: w_{rk} 代表任意任务 r 选择服务池 W_r 中第 k 个服务来加工; rh_{rk} 为服务属性且 $rh_{rk} = (h_{rk}, e_{rk})$, 代表着服务 w_{rk} 全部的属性参数, h_{rk} 表示选择 W_r 中第 k 个服务 w 来加工任务 r 所用的时间参数, e_{rk} 则表示生产精确率; L 为有向边集合,代表各服务 w_i 之间的偏序关系,且只有前驱层服务完成相关的任务 r_i 后才允许开始后继层任务 r_j , $r \in R, 0 < k \leq z_r$ 。图 F 中,用结点 Begin 表示开始结点,其所对应的虚拟任务用 r_B 表示,虚拟服务池用 W_B 表示;用结点 Final 表示结束结点,其所对应的

虚拟任务用 r_F 表示, 虚拟服务池用 W_F 表示。

定义3 任务自由度。指模型 M 中任务 r 的活动工作区间, 表示为 $FD_r = [IM_r, AL_r]$, 其中: IM_r 为任务 r 的最早启动时间, AL_r 为任务 r 的最晚启动时间, 可由式(1)求得:

$$\left. \begin{aligned} IM_r &= \sum_{i=1}^{r-1} \min(h_{ik}) \quad 0 < k < = z_r, IM_1 = 0 \\ AL_r &= DL - \sum_{i=n}^r \min(h_{ik}) \quad 0 < k < = z_r \end{aligned} \right\} \quad (1)$$

定理1 workflow图 F 中, 任意任务 r 其自由度 FD_r 有且只有一个。

证明: 存在 $FD_r = [IM_r, AL_r]$, 有任务结点 r, r' , 并满足顺序关系, 即 r 在 r' 之前执。

1) 证明最早开始时间 IM_r 有且只有一个: 此时假设任务 r 的第 k 个服务的运行时间最短, 即 h_{rk} , 则任务 r 最早开始时间 $IM_r = IM_{r'} + h_{rk}$, 依次递推, 即每个任务的最早开始时间, 则任务 r 的最早开始时间有且只有一个。

2) 证明最晚开始时间 AL_r 有且只有一个: 反证法假设第 r 的任务的最迟开始时间为 $AL_{r'}$, 假设存在 $AL_{r'} < AL_r$, 任务 r 的最大服务时间的 $\max\{h_{rk}\}$, 则 $AL_{r'} + h_{rk} < DL$, 此时必不是任务 r 最迟开始时间。反之若 $AL_{r'} > AL_r$, 则 $AL_{r'} + h_{rk} > DL$, 此时完工时间超过截止期, 因此假设不成立。

综上所述, 任务 r 的最早开始时间和最晚开始时间有且只有一个, 所以任务自由度 FD_r 也有且只有一个。

2.2 workflow图生成算法

workflow图能有效地模拟出业务流程中的任务及对应服务, 对管理人员详细掌握所有工作路径有着至关重要的作用。本文在以往的研究基础上^[18-19], 结合 workflow图特点, 给出生成 workflow图的步骤如下:

1) 划分业务流程的任务集合 R , 对 R 中所有任务 r 安排顺序, 并对任务 r 规划出能够完成该任务的服务 W_r ;

2) 在结点 $Begin$ 处开始, 添加任务层, 将第一个任务的服务放到同一层中;

3) 查询下一个任务, 并将其对应的所有服务添加到同一层中, 与前一层的任务形成顺序关系;

4) 重复步骤3), 直至没有任务为止, 此时将最后一层服务集合连接到最终状态 $Final$;

5) 对所有的服务 w_{rk} 添加属性 rh_{rk} ;

6) workflow图完成。

根据上述的策略, 可设计的工作流程图生成算法

WGGA 的伪代码如下:

输入: 参数 $w_{rk}; rh_{rk}; Begin; Final; len; z_r; // len$ 为集合 R 中的任务数目

输出: workflow图 F

算法 WGGA

$Service_queue = NULL, F = NULL, len = R.length;$

$For(intr = 1; r < = len; i++) \{ For(intrk = 1; k < = z_r; k++) \{ Insert(Service_queue, w_{rk}) \};$

$r = 1; For(k = 1; k < = z_r; k++) \{ Delete(Service_queue, w_{rk}) \}; // 把任务1的服务出队$

$ADD w_{rk} TO F; // 把服务加入 workflow图 F 中;$

$Contact(Begin, w_{rk}); // 连接开始结点 Begin$ 和任务1的所有服务

$For(r = 2; r < = len; r++) \{ For(k = 1; k < = z_r; k++) \{ Delete(Service_queue, w_{rk}) \};$

$ADD w_{rk} TO F; // 把对应的服务加入到 F 中$

$Contact(w_{r-1k}, w_{rk}); // 把前一层的任务的所有服务连接到下一层的所有服务$

$Contact(w_{rk}, Final); // 把最后一层服务连接到结束结点 Final$

$ADD rh_{rk} TO w_{rk}; // 把服务属性添加进去$

$Return F.$

该算法的时间复杂度为 $O(m^n)$ 。

2.3 workflow图生成算法 WGGA 举例

假设 workflow模型 M 的任务集合 R 为 $\{\alpha, \beta, \gamma, \delta\}$, 且每个任务都有对应的服务池 $W_\alpha, W_\beta, W_\gamma, W_\delta$, 用 workflow图生成算法 WGGA 输入基本参数后, 生成该实例的工作流程图 F 及相应参数如图1所示。

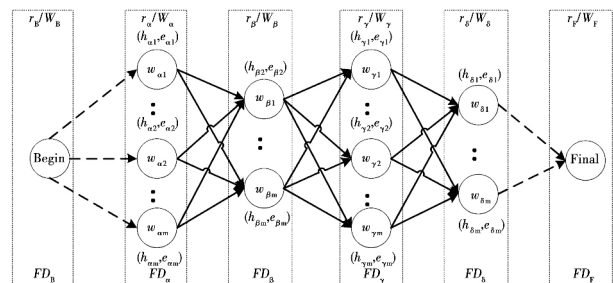


图1 workflow图示例

2.4 workflow图 F 所满足的约束

workflow图 F 中各任务加工需满足一定的约束条件才能达到工业需求, 式(2)为 workflow图 F 用于计算截止期下生产精确率的公式及遵循的约束条件:

$$\left. \begin{aligned} E &= \prod c_{rk} e_{rk}, H = \sum c_{rk} h_{rk} \leq DL \\ \text{s. t. } \sum_{k=1}^{z_r} c_{rk} &= 1, r \in R, c_{rk} \in \{0, 1\}, 0 < k \leq z_r \end{aligned} \right\} (2)$$

式中: E 为生产精确率, 指沿着某路径所经过的不同任务 r 选择对应的服务 w_{rk} 加工产品所得到的精确率; H 为加工时间, 指沿着某路径加工产品所花费的时间; DL 为截止期, 指加工产品的最晚完工时间; c_{rk} 为互斥变量取值为 1 或 0。

复杂产品的加工要求在不超过截止期 DL 内, 以式(2)为目标函数, 计算各 workflow 调度策略的加工时间 H 和精确率 E , 比较优化效果和性能指标。

3 单指标传统化调度算法^[20]

3.1 基于单指标 H 的传统优化调度策略 ODSH

以优化最小完工时间 H_{\min} 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 图 F 中每次选择最小的加工时间 h 来进行调度, 式(3)为在这种调度策略下求得的生产精确率 E :

$$\left. \begin{aligned} H_{\min} &= \sum \min(h_{rk}) \\ E &= \prod e_{rk}, 0 < k \leq z_r \end{aligned} \right\} r \in R \quad (3)$$

3.2 基于单指标 E 的传统优化调度策略 ODSE

以优化最大加工精确率 E_{\max} 为目标的传统 workflow 调度策略可获得最高的加工精确率, 这种策略在 workflow 图 F 中每次选择最大的加工精确率 e_{rk} 来进行调度, 式(4)为在这种调度策略下求得的完工时间 H :

$$\left. \begin{aligned} E_{\max} &= \prod \max(e_{rk}) \\ H &= \sum h_{rk}, 0 < k \leq z_r \end{aligned} \right\} r \in R \quad (4)$$

分析以上两种单指标优化调度策略发现其存在如下弊端: ①实现完工时间 H_{\min} 最小化, 则该路径完工精确率 E 可能会降低; ②实现完工精确率 E_{\max} 最大化, 则该路径所用时间可能超过截止期 DL , 不能满足工业需求。

4 基于串归约的时间-精确率优化策略 SRA

串归约分层技术的核心思想是充分考虑业务流程的完工时间和完工精确率, 即在截止期 DL 范围内, 对每层任务划分活动区间, 在其中寻找最优的服务, 将复杂产品的生产工序转变为局部结点的执行, 通过迭代的方式, 找到一条完工时间和完工精确率

平衡的路径。

若某 workflow 模型中的加工任务数为 n , 则令函数 $\Phi(r, h)$ 代表任务 r 从时刻 h_r 开始在其活动区间 $[IM_r, AL_r]$ 内所能达到的最大生产精确率。函数 $\Phi(r, h)$ 的值可通过式(5)进行计算:

$$\left. \begin{aligned} \varphi(r, h_r) &= \max\{e_{rk}\}, r = n, 0 < k \leq z_r \\ h_r &\in [IM_r, AL_r], h_r + h_{rk} \leq DL \end{aligned} \right\} \quad (5)$$

假设任务 r 的前驱为 r' , 则 workflow 模型整体被归约后, 可通过先计算最后层任务的精确率反向求得前层任务的各自最大精确率, 最终求得整个业务的最佳路径, 此过程可通过式(6)来进行计算:

$$\left. \begin{aligned} \varphi(r', h_{r'}) &= \max\{\varphi(r, h_r) + h_{rk} \times e_{rk}\} \\ h_{r'} &\in [IM_{r'}, AL_{r'}] \end{aligned} \right\} \quad (6)$$

综上所述, 基于串归约的时间-精确率优化算法 SRA 如下:

- 1) 调用算法 WGGGA 生成 workflow 图 F ;
- 2) 利用业务截止期并结合式(1)求得每个任务的自由度 FD_r ;
- 3) 调用式(5)求得在最后一层任务的自由度内, 在不同时刻开始达到的最大精确率;
- 4) 调用式(6), 采用迭代的方式, 求出每个服务在不同时间开始能达到的最大精确率;
- 5) 比较最终的精确率, 达到最大精确率的路径即为最佳路径。

5 案例描述及算法应用

5.1 案例描述

经调研某卡车装配的组装业务的流程分为车架总成、总装配流水线、布设电线、安装零件、加注油料、安装轮胎、膨胀水箱、调试下线等 8 个环节, 经分析可将申请和竣工环节当作虚拟任务, 作为开始与结束, 则该业务环节构成了 workflow 里面的任务集合 R , 可抽象化表示为 $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$, 虚拟的申请、竣工环节为 r_B, r_F ; 每个任务环节 r 对应一个服务池 W_r 。已知各服务集合中的任一元素的属性 rh_{rk} 结合服务可表示为 $w_{rk}(h_{rk}, e_{rk})$, 如 $w_{11}(3, 0.95)$ 表示审批环节 r_1 用该服务集合 W_1 中的第一个服务 w_{11} 来完成需要的时间为 3, 达到的准确率为 0.95。其他任务对应的服务及其工作时间和准确率如表 1 所示。

表1 服务时间及准确率

任务集合 R	任务对应的服务池 W
r_1	$w_{11}(3, 0.95); w_{12}(4, 0.98); w_{13}(6, 0.99);$
r_2	$w_{21}(3, 0.97); w_{22}(4, 0.99);$
r_3	$w_{31}(5, 0.93); w_{32}(6, 0.95); w_{33}(9, 0.98);$
r_4	$w_{41}(2, 0.92); w_{42}(5, 0.97);$
r_5	$w_{51}(3, 0.95); w_{52}(5, 0.98);$
r_6	$w_{61}(4, 0.95); w_{62}(5, 0.97); w_{63}(7, 0.99);$
r_7	$w_{71}(3, 0.94); w_{72}(4, 0.96);$
r_8	$w_{81}(4, 0.93); w_{82}(5, 0.95); w_{83}(6, 0.97);$

综合表1中数据,并调用 workflow 模型生成算法 WGGA,卡车装配的工作流程图如图2所示。

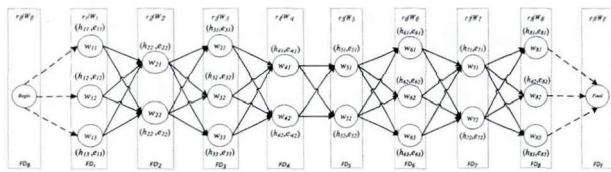


图2 卡车装配工作流程图

5.2 算法分析

若规定业务流程的截止期 $DL = 30$, 则图2所示工作流程图在不同优化算法下的调度过程如下:

1) 算法 ODSH 的优化。调用算法 ODSH 可求业务流程的工作流为 $w_{11} \rightarrow w_{21} \rightarrow w_{31} \rightarrow w_{41} \rightarrow w_{51} \rightarrow w_{61} \rightarrow w_{71} \rightarrow w_{81}$, 优化后的 $H_{\min} = h_{11} + h_{21} + h_{31} + h_{41} + h_{51} + h_{61} + h_{71} + h_{81} = 3 + 3 + 5 + 2 + 3 + 4 + 3 + 4 = 27 < DL$, 所对应的加工准确率 $E = e_{11} \times e_{21} \times e_{31} \times e_{41} \times e_{51} \times e_{61} \times e_{71} \times e_{81} = 0.95 \times 0.97 \times 0.93 \times 0.92 \times 0.95 \times 0.95 \times 0.94 \times 0.93 = 0.622$ 。

2) 算法 ODSE 的优化。调用算法 ODSE 可求业务流程的工作流为 $w_{13} \rightarrow w_{22} \rightarrow w_{33} \rightarrow w_{42} \rightarrow w_{52} \rightarrow w_{63} \rightarrow w_{72} \rightarrow w_{83}$, 优化后的 $E_{\max} = e_{13} \times e_{22} \times e_{33} \times e_{42} \times e_{52} \times e_{63} \times e_{72} \times e_{83} = 0.99 \times 0.99 \times 0.98 \times 0.97 \times 0.98 \times 0.99 \times 0.96 \times 0.97 = 0.842$, 所对应完工时间 $H = h_{13} + h_{22} + h_{33} + h_{42} + h_{52} + h_{63} + h_{72} + h_{83} = 6 + 4 + 9 + 5 + 5 + 7 + 4 + 6 = 46 > DL$, 不满足限定条件。

3) 算法 SRA 的优化。调用算法 SRA 可得主要任务的自由度为: $FD_{r1} = [0, 3], FD_{r2} = [3, 6], FD_{r3} = [6, 9], FD_{r4} = [11, 14], FD_{r5} = [13, 16], FD_{r6} = [16, 19], FD_{r7} = [20, 23], FD_{r8} = [23, 26]$, 在各自任务的自由度 FD_i 内, 进一步计算业务开始时刻所能获得的最大准确率如下:

调试下线 r_8 : $\Phi(8, 23) = \max\{0.93, 0.95,$

$0.97\} = 0.97; \Phi(8, 24) = \max\{0.93, 0.95, 0.97\} = 0.97; \Phi(8, 25) = \max\{0.93, 0.95\} = 0.95; \Phi(8, 26) = \max\{0.93\} = 0.93;$

膨胀水箱 r_7 : $\Phi(7, 20) = \max\{\Phi(8, 23) \times 0.94, \Phi(8, 24) \times 0.96\} = 0.931;$

$\Phi(7, 21) = \max\{\Phi(8, 24) \times 0.94, \Phi(8, 25) \times 0.96\} = 0.912;$

$\Phi(7, 22) = \max\{\Phi(8, 25) \times 0.94, \Phi(8, 26) \times 0.96\} = 0.893;$

$\Phi(7, 23) = \max\{\Phi(8, 26) \times 0.94\} = 0.874;$

安装轮胎 r_6 : $\Phi(6, 16) = \max\{\Phi(7, 20) \times 0.95, \Phi(7, 21) \times 0.97, \Phi(7, 23) \times 0.99\} = 0.865;$

$\Phi(6, 17) = \max\{\Phi(7, 21) \times 0.95, \Phi(7, 22) \times 0.97\} = 0.866;$

$\Phi(6, 18) = \max\{\Phi(7, 22) \times 0.95, \Phi(7, 23) \times 0.97\} = 0.848;$

$\Phi(6, 19) = \max\{\Phi(7, 23) \times 0.95\} = 0.830;$

加注油料 r_5 : $\Phi(5, 13) = \max\{\Phi(6, 16) \times 0.95, \Phi(6, 18) \times 0.98\} = 0.831;$

$\Phi(5, 14) = \max\{\Phi(6, 17) \times 0.95, \Phi(6, 19) \times 0.98\} = 0.823;$

$\Phi(5, 15) = \max\{\Phi(6, 18) \times 0.95\} = 0.806;$

$\Phi(5, 16) = \max\{\Phi(6, 19) \times 0.95\} = 0.789;$

安装零件 r_4 : $\Phi(4, 11) = \max\{\Phi(5, 13) \times 0.92, \Phi(5, 16) \times 0.97\} = 0.765;$

$\Phi(4, 12) = \max\{\Phi(5, 14) \times 0.92\} = 0.757;$

$\Phi(4, 13) = \max\{\Phi(5, 15) \times 0.92\} = 0.742;$

$\Phi(4, 14) = \max\{\Phi(5, 16) \times 0.92\} = 0.726;$

布设电线 r_3 : $\Phi(3, 6) = \max\{\Phi(4, 11) \times 0.93, \Phi(4, 12) \times 0.95\} = 0.711;$

$\Phi(3, 7) = \max\{\Phi(4, 12) \times 0.93, \Phi(4, 13) \times 0.95\} = 0.705;$

$\Phi(3, 8) = \max\{\Phi(4, 13) \times 0.93, \Phi(4, 14) \times 0.95\} = 0.690;$

$\Phi(3, 9) = \max\{\Phi(4, 14) \times 0.93\} = 0.675;$

总装配流水线 r_2 : $\Phi(2, 3) = \max\{\Phi(3, 6) \times 0.97, \Phi(3, 7) \times 0.99\} = 0.698;$

$\Phi(2, 4) = \max\{\Phi(3, 7) \times 0.97, \Phi(3, 8) \times 0.99\} = 0.684;$

$\Phi(2, 5) = \max\{\Phi(3, 8) \times 0.97, \Phi(3, 9) \times 0.99\} = 0.669;$

$\Phi(2, 6) = \max\{\Phi(3, 9) \times 0.97\} = 0.655;$

车架总成 r_1 : $\Phi(1, 0) = \max\{\Phi(2, 3) \times 0.95, \Phi(2, 4) \times 0.98, \Phi(2, 6) \times 0.99\} = 0.670;$

$$\Phi(1,1) = \max\{\Phi(2,4) \times 0.95, \Phi(2,5) \times 0.98\} = 0.656;$$

$$\Phi(1,2) = \max\{\Phi(2,5) \times 0.95, \Phi(2,6) \times 0.98\} = 0.642;$$

$$\Phi(1,3) = \max\{\Phi(2,6) \times 0.95\} = 0.622;$$

经逆向求解后可得审批任务 r_1 的加工准确度集合为 $(0.670, 0.656, 0.642, 0.622)$,在集合中选择最大准确率,即 $\Phi(1,0) = 0.670$ 。由此,可通过算法 SRA 得到的优化路径为 $w_{12} \rightarrow w_{22} \rightarrow w_{32} \rightarrow w_{41} \rightarrow w_{51} \rightarrow w_{61} \rightarrow w_{71} \rightarrow w_{82}$,所对应完工时间 $H = h_{12} + h_{22} + h_{32} + h_{41} + h_{51} + h_{61} + h_{71} + h_{82} = 4 + 3 + 6 + 2 + 3 + 4 + 3 + 5 = 30 = DL$,优化后的 $E_{\max} = e_{12} \times e_{21} \times e_{32} \times e_{41} \times e_{51} \times e_{61} \times e_{71} \times e_{82} = 0.670$ 。

5.3 算法对比分析

将 ODSH、ODSE 和 SRA 3 种算法对图 2 所示 workflow 模型优化效果进行对比,其对比结果如图 3 所示。

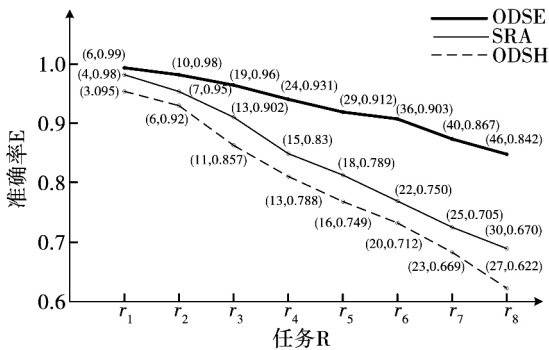


图 3 各算法优化效果对比图

分析图 3 可见,使用算法 ODSE 进行优化得到的完工准确率最大,但由于加工所用时间大于截止期 DL 而不能采用;采用算法 ODSH 和算法 SRA 进行优化,最终的完工时间均未超过截止期 DL ,因此可以采用且所得最大完工准确率为 $E_1 = 0.622$ 、 $E_2 = 0.670$ 。对于算法 ODSH,算法 SRA 的优化提高率 $IR = (E_2 - E_1) / E_1 \times 100\% = 7.71\%$ 。因此,算法 SRA 较传统单目标优化算法具有更好的优化效果。

6 不同截止期 DL 对算法性能的影响

截止期 DL 是加工任务必须遵守的完工限定时间,是 workflow 中重要的参数。一般情况下,截止期的变大使得最佳路径的完工准确率有所提高。算法 SRA 正是基于此参数的约束前提下,对各生产任务

进行的优化。若改变 DL 值,则 workflow 中各加工任务 r 的自由度 FD_r 也将进行改变,这使得自由度变大的任务可以选择准确率更高的服务,相反,自由度变小的任务由于不能超过最晚完工时间只能选择准确率较小的服务。对于案例所述 workflow 模型,不同截止期 DL 对算法性能影响的数据如表 2 所示。

表 2 DL 值对算法性能的影响

DL	ODSH (E_1)	ODSE (E_2)	SRA (E_3)	相对 E_1 提高 (IR_1)	相对 E_2 降低 (IR_2)
30	0.622	0.842	0.670	7.71%	20.4%
32	0.622	0.842	0.689	10.7%	18.2%
34	0.623	0.843	0.715	15.0%	15.2%
35	0.623	0.843	0.726	16.7%	13.9%
37	0.622	0.842	0.738	18.6%	12.4%
39	0.622	0.843	0.751	20.7%	10.9%

分析表 2 可知,增大 DL 值,相对以时间最小化和以准确率最大化为目标的算法,算法 SRA 的提高率随之增大。因此,可通过适当增加工程要求的截止期,来达到增加算法 SRA 优化性能的目的。

7 结 语

针对时间约束下 workflow 生产精确率难于优化等问题,本文提出了一种串归约的时间-精确率优化算法 SRA。该算法采用分层的思想并由后向前循环迭代,求得每层任务的自由度。在每个任务的自由度范围内,算法选择最佳服务,从而达到优化整体加工准确率的目的。案例分析表明,先对于传统优化算法 ODSH 和 ODSE,算法 SRA 的性能提高了 7.71%。未来的工作将围绕时间、精确率和成本等三方面以上因素来优化 workflow 调度。

参 考 文 献:

[1] I FOSTER C, KESSELMAN J, M Nick, et al. Grid Service for Distributed System Integration [J]. IEEE Computer 2002, 35(6): 37-46.

[2] DEELMAN L E, BLYTHEL J. Mapping Abstract Complex Workflows onto Grid Environments [J]. Journal of Grid Computing, 2003, 1(1): 25-39.

[3] EDMONDS J, KARPR M. Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems [J]. Journal of the ACM 2002, 19(2): 248-264.

[4] ALONSO G, CASATIF, KUNO H, et al. Web Service Concepts, Architectures and Applications[M]. Berlin: Springer 2004.

- [5] DUSTDAR S ,SCHREINER W. A Survey on Web Services Compositions[J]. International Journal of Web and Grid Services ,2005 , 1(1) : 1 - 30.
- [6] CHELLAMMAL S ,GOPINATH G ,MANIKANDAN S R. An Approach for Selecting Best Available Services Through a New Method of Decomposing QoSconstraints [J]. Service Oriented Computing and Applications 2015 9(2) : 107 - 138.
- [7] DE P ,DUNNER E J ,GHOSH J B ,Wells C E. Complexity of the Discrete Time-cost Trade-off Problem for Project Networks [J]. Operations Research ,1997 ,45(2) : 302 - 306.
- [8] BUYYA R ,DAVID ABRAMSON ,JONATHAN GIDDY ,et al. Economic Models for Resource Management and Scheduling in Grid Computing. Concurrency and computation [J]. Practice and Experience Journal , Special Issue on Grid Computing Environment , 2002 ,14(13/15) : 1507 - 1542.
- [9] ABRAMSON D ,BUYYA R ,GIDDY J. A Computational Economy for Grid Computing and Its Implementation in the Nimrod-G Resource Broker [J]. Future Generation Computer Systems(FGCS) Journal 2002 ,18(8) : 1061 - 1074.
- [10] ADNENE G ,FRANCOIS C. Multiple Instantiation in a Dynamic Workflow Environment [C]//Proceedings of 16th International Conference on Advanced Information Systems Engineering ,2004: 175 - 188.
- [11] AKKAN C ,DREX I A ,KIMMS A. Network Decomposition-based Bench Mark Results for the Discrete Time-cost Tradeoff Problem [J]. European Journal of Operational Research ,2005 , 165(2) : 339 - 358.
- [12] PAWEL CZAMUL. Modeling , Run-time Optimization and Execution of Distributed Workflow Applications in the JEE-based Beesy-Clusterenvironment [J]. The Journal of Supercomputing , 2013 , 63(1) : 46 - 71.
- [13] MACIEJ M ,KAMIL F ,MARIAN B ,et al. Cost Optimization of Execution of Multi-level Deadline-Constrained Scientific Workflows on Clouds [C]// Parallel Processing and Applied Mathematics , 2014 8384: 251 - 260.
- [14] WOO-JOONG K ,DONG-KI K ,SEONG-HWAN K ,et al. Cost adaptive VM Management for Scientific Workflow Application in Mobile Cloud [C]// Mobile Networks and Application , 2015 20(3) : 328 - 336.
- [15] 张伟 ,秦臻 ,苑迎春. 网格环境下工作流的费用 - 时间调度算法[J]. 计算机工程 2006 32(16) : 97 - 99.
- [16] 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.
- [17] 陈成 ,薛恒新 ,张庆民. 基于本体与多 Agent 的可靠供应链网络设计模型 [J]. 计算机集成制造系统 ,2011 ,17(1) : 142 - 150.
- [18] 罗智勇 ,孙广路 ,刘嘉辉 ,等. 攻击图算法在入侵防御系统中的应用[J]. 云南大学学报 ,2012 34(3) 271 - 275.
- [19] 罗智勇 ,尤波 ,许家忠 ,等. 基于三层攻击图的入侵意图自动识别模型[J]. 吉林大学学报 ,2014 ,44(5) : 1392 - 1397.
- [20] 苑迎春 ,李小平 ,王茜 ,等. 基于逆向分层的网格工作流调度算法[J]. 计算机学报 2008 ,31(2) : 282 - 290.

(编辑: 温泽宇)