第 35 卷 第 2 期
2020 年 4 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 35 No. 2
Apr. 2020

DOI:10.13364/j.issn.1672-6510.20180149

基于遗传算法的批处理科学工作流任务调度算法的改进

熊聪聪，陈长博，赵 青，林 颖
（天津科技大学人工智能学院，天津 300457）

摘 要：随着云计算应用的不断深入以及对大数据处理需求的不断提升，越来越多的企业选择使用云平台处理海量的数据. 由于云计算的商业性，这就对云计算中的任务调度提出了更加严苛的要求，如何合理且经济地完成任务调度成为了研究云计算的关键问题之一. 批处理科学工作流是大数据时代的一种新型工作流建模形式，近两年已引起业内的重视，但当前仍处于起步阶段. 本文首先对当前传统的任务调度算法进行分析，并指出其中的不足之处，从而改进了基于遗传算法的批处理科学工作流任务调度算法 BIGA（batch scientific workflow task scheduling based on improved genetic algorithms），在满足固定截止期的条件下，以任务调度成本最优优化目标，分别对独立任务调度与非独立任务调度进行研究实验. 最后在 Matlab 中进行模拟实验，结果表明：本文的改进算法在满足任务截止期的情况下与按比例划分截止期经典调度算法相比，在一定任务规模下，完成任务调度所需成本更低，更加符合云资源的使用特征与用户需求.

关键词：云计算；任务调度；遗传算法；批处理
中图分类号：TP311 **文献标志码**：A **文章编号**：1672-6510（2020）02-0074-07

Improving Task Scheduling of Batch Workflow Applications Based on Genetic Algorithm

XIONG Congcong，CHEN Changbo，ZHAO Qing，LIN Ying
（College of Artificial Intelligence，Tianjin University of Science & Technology，Tianjin 300457，China）

Abstract：With the continuous development of cloud computing application and the increasing demand for big data processing，more and more enterprises choose to use the cloud platform to process massive amounts of data. The commercial nature of cloud computing is giving task scheduling in cloud computing more stringent requirements，and how to reasonably and economically complete the task scheduling is one of the key problems of cloud computing. Batch scientific workflow is a new type of workflow model in big data era，which has attracted the attention of industry in the past two years，but is still in its infancy. This article analyzes the current task scheduling algorithm，points out its shortcomings，and then puts forward a new batch BIGA task scheduling algorithm based on genetic algorithm（batch scientific workflow task scheduling based on improved genetic algorithms）. Under the condition of meeting the deadline and aimed at task scheduling cost optimization，experiments of both independent task scheduling and dependent task scheduling were conducted. Finally，in Matlab simulation experiments，the results show that the improved BIGA algorithm，compared with the classic task scheduling algorithm，has the advantage of lower cost and is more in line with the cloud resources characteristics. It can also better meet the user' requirements.

Key words：cloud computing；task scheduling；genetic algorithm；batch task

基于批处理的工作流广泛应用于多个领域，尤其是在数据分析应用中，例如移动领域、电子商务和科学研究，其原理是通过一系列的连接操作来处理大量数据[1]. 科学领域的计算任务通常被建模为科学工作

收稿日期：2018-05-17；**修回日期**：2019-04-03
基金项目：天津市教委科研计划资助项目（2017KJ035）；天津市自然科学基金资助项目（17JCQNJC00400）
作者简介：熊聪聪（1961—），女，四川泸州人，教授；**通信作者**：赵 青，讲师，zhaoqing@tust.edu.cn

流形式，其任务根据数据流与计算相关性形成链式结构，由于科学计算领域通常存在数据量巨大和计算复杂度高的问题，需要高性能计算环境支持运行[2]. 两者最主要的区别为：科学工作流主要以数据为中心，而传统工作流更注重业务过程的自动化. 云计算作为近年来最火热的网络服务方式为科学工作流提供了高效、高性价比、可扩展的运行支撑环境[2−3]. 当前，基于云平台的科学工作流的调度研究已成为一个研究热点，国内外已有很多成果面世. 例如，2018 年 Anwar 等[4]提出了一种基于工作流程(DSB)的任务包的动态调度策略，用于调度科学的工作流，目的是在用户定义的期限约束下最小化租赁虚拟机的租用成本. Liu 等[5]提出了一种新的基于任务回填的科学工作流调度策略，使用 task backfill 算法在虚拟机实例上聚合多个任务，使用合适的性能，并利用单个任务填充虚拟机的空闲时间槽，在不影响整体性能的情况下提高资源利用率. Zhao 等[6−7]针对异构云环境，提出了一种基于数据依赖聚类和递归划分的数据聚类算法，并将数据大小和固定位置的因素结合起来. 然后，提出了一种启发式的树对树数据布局策略，以使频繁的数据移动发生在高带宽的信道上. 之后，又在此基础上提出了面向科学工作流模型的任务调度算法[8]，可以在提高执行效率的同时，提高计算资源利用率，减少能源消耗.

然而，随着大数据时代的到来，一种新型的科学工作流模型——批处理科学工作流逐渐引起人们的注意. 为了提高大数据时代数据密集型应用的计算效率，工作流中的很多任务需要通过数据划分为可并行处理的任务组，从而在原始简单的有向无环图(directed acyclic graph，DAG)上形成了一个个包含批处理操作的宽节点结构.

当前，关于批处理科学工作流的云调度研究正处于起步阶段，传统科学工作流调度中的 deadline 划分、虚机映射等方法并不适合于批处理科学工作流模型. 因此，本文将重点关注批处理科学工作流的特征，在研究批处理科学工作流任务调度过程中，尽可能满足用户所提供的 deadline 的情况，寻找调度成本最低的虚拟机资源分配方式. 云批处理科学工作流的任务调度包括两个层次：一是任务包与 VM 之间的映射，二是在单个 VM 上的顺序执行[9]. 本文重点关注第一个层次，在满足或相对满足任务截止期的情况下，工作流调度的成本最优化，寻找最优调度方案.

相关研究中，Mao[10]开发了一个工作流计算系统 GreePipe，该系统可以将工作流描述自动转换成一系列基于 Hadoop 的 MapReduce 任务，实现生物研究领域复杂的数据分析逻辑. 该工作流属于不可拆分工作流. 对于可拆分批处理工作流，大多数工作都直接采用更细的粒度进行建模，并直接采用一些非线性的 DAG 图进行表示，并作为多个单独的 MapReduce 任务处理，这样的建模方式，丢失了批结构信息. 虽然很多针对一般工作流的调度算法可以用于批处理工作流，但是，由于没有考虑批处理工作流的批结构特点，容易导致较低的资源利用率.

Cai 等[11]提出了一种最少新租赁时间区间优先规则、最便宜执行成本优先规则和剩余时间片长度和执行时间匹配度优先规则的混合式启发算法. 但是，该算法初始虚拟机资源分配方案的原则是选用最多的性能最差的虚拟机类型，并以最大并行度的方式执行批处理任务调度. 如果调度时间超出了用户提出的 deadline，就会升级关键路径上任务节点使用的虚拟机类型. 因此，最终所得解倾向于租用大量低成本的虚拟机类型，容易陷入局部最优.

与上述已有方法相比，本文改进了遗传算法中的初始种群生成过程，使初始种群覆盖面更广；优化适应度函数，使其更适合评估批处理科学工作流下的个体适应度值；引入非关键路径虚拟机使用数量收缩操作，在不影响关键路径的前提下进一步减少任务调度成本.

1　批处理科学工作流任务调度模型任务模型

为了方便描述，对批处理科学工作流进行如下建模：

(1)图 1 为一个标准的批处理科学工作流 DAG，对于一个标准的 DAG 图来说，入口节点是其他所有任务的前驱节点，其优先度是最高的，故在所有任务节点中入口节点应该最先获得调度，出口节点与之同理.

(2)需要进行任务调度的任务批为 $T=(t_1,t_2,\cdots,t_n)$，其中 t_i 代表编号为 i 的任务节点. 在每一个任务节点上包含若干个子任务，即 $t_i=(t_i(1),t_i(2),...,t_i(n))$，$t_i(k)$ 代表 t_i 任务节点上的第 k 个任务包.

(3)定义批处理科学工作流为有向无环图 DAG，任务节点 t_i 为有向无环图 DAG 上的一个节点.

(4)假设云平台提供 m 种不同类型的虚拟机，将任务节点 t_i 上的 q 个子任务调度到不同虚拟机上的

期望运行时间($\boldsymbol{t}$)是一个 $q \times m$ 的矩阵，其计算公式见式(1). 矩阵的第 i 行表示任务节点 i 的所有子任务在每一种虚拟机上的预测执行时间；而第 j 列表示每个任务节点的所有任务在单个第 j 类虚拟机上运行所需的预估时间. 预估时间由两部分组成，一部分为单个虚拟机执行任务所必需的时间，第二部分为虚拟机启动和安装软件的时间.

$$\boldsymbol{t}=\begin{Bmatrix} t_{11} & \cdots & t_{1m} \\ \vdots & \vdots & \vdots \\ t_{n1} & \cdots & t_{nm} \end{Bmatrix} \tag{1}$$

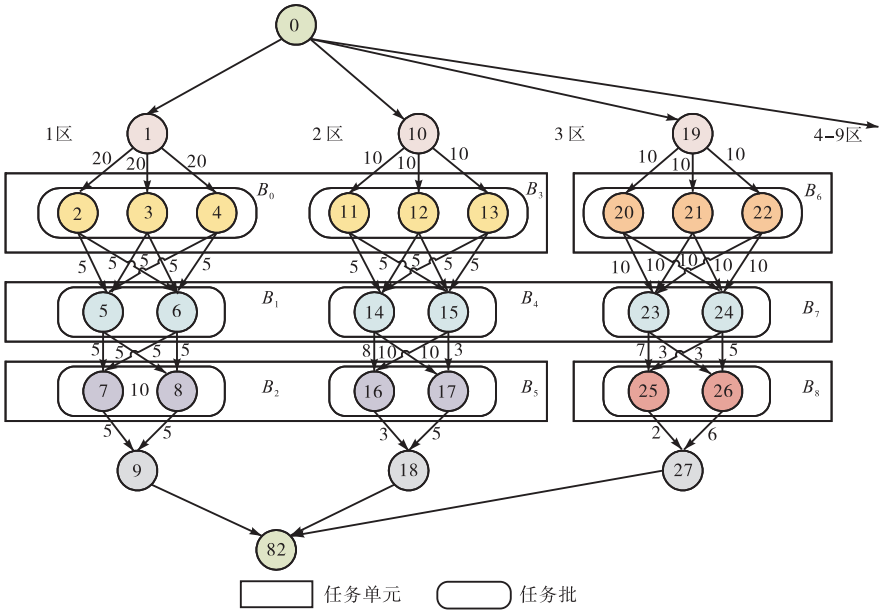

图 1　标准批处理科学工作流 DAG 图

Fig. 1　DAG diagram of standard batch processing science workflow

2　批处理科学工作流任务调度

遗传算法是受达尔文进化论启发进而产生的一种全局搜索和优化算法. 它把每一个总群内的个体看作一种解，通过模仿自然界中优胜劣汰的法则，使优秀的个体留下，并进行交叉变异继续寻找更优解，同时将劣质的个体淘汰，最终通过不断的算法迭代得到最优解[12].

本文采用基于改进遗传算法的智能启发式算法，以尽可能搜索到在尽量满足工作流整体 deadline 要求前提下的任务调度总成本最优的虚拟机资源分配方案，对可随意分割的批处理科学工作流任务调度和不可随意分割的批处理科学工作流任务调度进行论述. 可随意分割的批处理科学工作流是指：在该科学工作流上的每一个节点上的任务包可以根据虚拟机数量随意分割进行并行处理，使得每个虚拟机上的负载是均衡的. 不可随意分割的批处理科学工作流是指：在该科学工作流上的每一个节点上有一定数量的任务包，任务包不可分割. 在本文中假设每个任务包的处理时间是相同的.

2.1　可随意分割批处理科学工作流调度

2.1.1　*编码方式*

本文采用整数编码的方式，编码长度取决于任务节点数. 每一条染色体都对应一个任务解. 假设任务节点数 $N=5$，可用虚拟机类型数量为 $M=6$，则染色体(1，3，2，4，3，3，4，1，5，2)代表在任务节点一上使用 3 个 1 号虚拟机并行执行，在任务节点二上使用 4 个 2 号虚拟机并行执行，在任务节点三上使用 3 个 3 号虚拟机并行执行，在任务节点四上使用 1 个 4 号虚拟机串行执行，在任务节点五上使用 2 个 5 号虚拟机并行执行.

2.1.2　*优化初始种群生成*

传统遗传算法的初始种群生成可能存在初始种群生成覆盖度不够广、初始种群生成重复率高的缺点，从而导致算法无法收敛到全局最优解. 本文对遗传算法的初始化种群处理如下：

为了保证初始解距离最优解距离不至于太远，在生成初始解的过程中引入时间成本比概念，即采用满足任务执行需求的最低端配置虚拟机类型，在增加虚

拟机使用数量的同时与串行执行相比，调度时间减少量与成本增加量之间的比值.

(1) 在生成初始解时选取三分之一初始解为每个任务节点选取一个时间减少量与成本增加量之间的比值，并选取比该比值小的所有虚拟机使用类型方案，按照任务节点顺序以全组合方式生成初始解. 其主要目的是使这一部分生成的初始种群先天就相对较优，提升算法的收敛速度和准确度.

(2) 在批处理科学工作流任务调度每一个任务节点内所包含的任务包对应需求的虚拟机类型可能不同，例如有些任务节点所包含的任务为数据密集型任务，有的则为计算密集型，因此在云提供商平台中会提供各种侧重点不同的虚拟机类型. 为了进一步保证初始解相对较优，三分之一初始解按照每个任务节点所对应需求类型的虚拟机类型进行随机生成.

为了保证初始解的随机性，三分之一初始解按照完全随机的方式进行初始解生成.

2.1.3　适应度函数

在遗传算法中，适应度函数是用来衡量一个解的好坏的，本文以调度成本最优以及调度时间尽量满足截止期为优化目标，由于在该模型下，每个任务节点开始进行任务调度的时间依赖于前序节点的结束时间且任务节点与任务节点之间可以并行执行，因此整个批处理科学工作流的任务调度完成时间取决于该批处理科学工作流关键路径上的任务节点进行任务调度的总时间. 而对于同一个批处理科学工作流来说，关键路径的选择取决于虚拟机的具体配置情况，比如：给定一个批处理科学工作流，在该科学工作流中至少存在一条或多条路径可以从初始节点通往最终节点，而在某种虚拟机类型和数量的配置下对于该科学工作流，其从初始节点通往最终节点耗时最长的一条路径为该科学工作流的关键路径，耗时即为该科学工作流的调度总时长.

第 y 个任务节点使用 j 个 i 类虚拟机的执行任务所需时间 (t_y) 的计算公式见式 (2).

$$t_y = \frac{t_{yi} - t_{bi}}{j} + t_{bi} \tag{2}$$

式中：t_{yi} 为单个 i 类虚拟机执行第 y 个任务节点所需的预估时间；t_{bi} 为虚拟机的启动以及执行软件的安装时间.

任务调度总时间为该批处理科学工作流上关键路径节点任务调度所需时间之和.

第 y 个任务节点使用 j 个 i 类虚拟机的执行任务所需的总费用 (C_y) 的计算公式见式 (3).

$$C_y = \left\lceil \frac{t_y}{a} \right\rceil \cdot c_i \cdot j \tag{3}$$

式中：a 为虚拟机的租用周期；c_i 为第 i 类虚拟机单个租用周期的单价.

总费用 C_{total} 为每个任务节点的费用的连续累加和，其计算式见式 (4).

$$C_{\text{total}} = \sum_{i=1}^{n} c_i \tag{4}$$

由上，定义适应度函数 *fitness* 为

$$fitness = \frac{1}{w_1 \cdot C_{\text{total}} + w_2 \cdot \text{Max}(0, T_{\text{totaltime}} - T_{\text{deadline}})} \tag{5}$$

式中：w_1 与 w_2 为两个实数，两数相加之和为 1；T_{deadline} 为用户所提供的调度该批任务所要求的 deadline 与任务批开始进行调度的时间差值，单位为 min；$T_{\text{totaltime}}$ 为任务调度总时间，单位为 min；Max 代表取括号内两数中较大的. 下文在交叉概率函数中用 f 简化表示.

2.1.4　遗传算法的交叉变异

(1) 由于编码方式为每两位数字代表一组信息，因此本文的交叉掩码取值为随机偶数. 交叉概率函数 (P) 的计算公式见式 (6).

$$P = \begin{cases} k_1 \cdot (f_{\max} - f') / (f_{\max} - f_{\text{av}}) & f' \geqslant f_{\text{av}} \\ k_2 & f' < f_{\text{av}} \end{cases} \tag{6}$$

式中：k_1、k_2 为常数；$f_{\max}$ 为种群最大适应度值；f_{av} 为群体平均适应值；f' 为要交叉的两个个体中较大个体的适应度值. 当 f' 大于等于 f_{av}，应该让交叉概率 P 较小，防止适应度值大的个体统治群体，陷入局部最优解；反之，则应该让交叉概率较大，以重组出新的个体，扩展搜索空间.

(2) 变异操作：本文的变异操作采用的是基本位变异，选取一定百分比的个体随机变异染色体编码的偶数位，即代表使用某种虚拟机类型数量的表示位，变异方式是将该位数字随机加减 1，本文选取 3%. 这样做的好处是变异后的染色体适应度值变得更好的概率较高.

2.1.5　非关键路径虚拟机使用数量的收缩

为了进一步优化遗传算法通过迭代次数所得出的最优解，提出了一种非关键路径虚拟机使用数量收缩的方法. 在遗传算法通过多次迭代得出一个最优解后，在不改变关键路径的情况下，将非关键路径上使用的虚拟机数量按照单价由高到低不断收缩，直到

再次收缩将会改变该解的关键路径便停止收缩并输出最优解.

2.2　不可随意分割批处理科学工作流调度

与可随意分割批处理科学工作流相比，不可随意分割批处理科学工作流每个任务节点所包含的任务包由于内部的数据关系，任务包不可随意分割. 也就是说在执行调度时，每个任务包只能放在一个虚拟机上进行处理，所以在把任务调度到虚拟机上时，与可随意分割批处理科学工作流有区别. 调度算法具体区别如下：

(1) 在本文中，为了更好地使最终解所得到的调度成本最低以及时间尽量满足用户所给出的截止期，采用任务平均分配至每个虚拟机上的分配方式. 具体分配方式如图 2 所示. 由于任务包不可随意分割，调度到每台虚拟机上的任务包数量可能不同，所以每个任务节点所需的调度总时间取决于该任务节点上所使用的耗时最长的虚拟机.

(2) 由于不可随意分割批处理科学工作流每个任务节点子任务的不可随意分割性，所以在改进遗传算法的变异过程中所采取的基本位变异就有了上限，即虚拟机使用数量不能超过子任务数量，所以当变异位数已经达到上限，则默认虚拟机使用数量变异只能减 1.

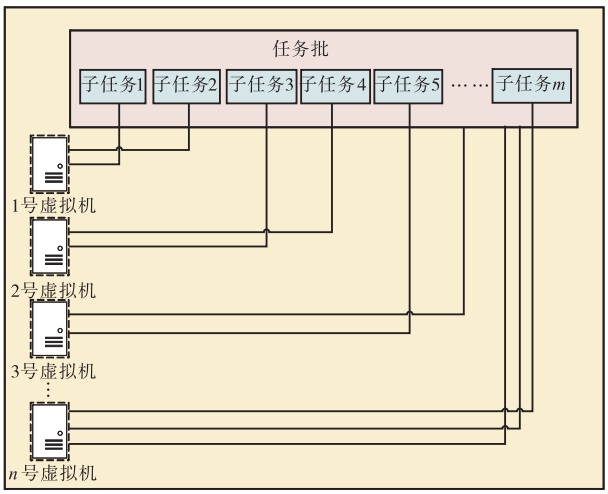

图 2　虚拟机分配方式

Fig. 2　Virtual machine allocation mode

3　仿真实验

使用 Matlab 验证本文所提模型以及算法的有效性. 所有进行实验的批处理科学工作流均为随机生成，虚拟机类型与单周期租赁费用为亚马逊云上选取的实例，共选取了五类具有代表性的虚拟机类型. 将 T_{deadline} 设置为 100 min. 亚马逊云真实实例具体数据见表 1.

表 1　亚马逊云资源真实实例数据表

Tab. 1　Amazon cloud resources real instance data sheet

实例类型	vCPU	内存/GiB	专用带宽/Mbps	价格/($·h^{-1})
t2.small	1	2	0	0.1
m5.large	2	8	最高 2 120	0.4
m5.xlarge	4	16	最高 2 120	0.8
c5.large	2	4	最高 2 250	0.2
C5.xlarge	4	8	最高 2 250	0.8

表中数据全部来源于亚马逊云官方提供的云资源的真实数据，其中 vCPU 代表虚拟机 CPU，每个虚拟机都有内存，单位为 GiB，带宽主要影响虚拟机之间的传输速度，单位为 Mbps.

本文在 Matlab 平台上模拟实现了改进的遗传算法的可随意分割批处理科学工作流任务调度与不可随意分割批处理科学工作流任务调度算法，同时还原了任务可随意分割按比例划分截止期经典调度算法与任务不可随意分割按比例划分截止期经典调度算法，并与之进行对比. 用于实验的批处理科学工作流是通过 Cai 等[11]使用的批处理科学工作流生成器完全随机生成的. 按比例划分截止期经典调度算法在执行虚拟机资源配置时主要是根据待执行任务量大小，即任务节点内任务执行时间预测值与任务个数的乘积，然后按照所占总任务量的百分比进行分配截止期的长短，最终按照所分得的时间配置虚拟机使其满足相应分得的截止期. 4 种算法的调度成本如图 3 所示.

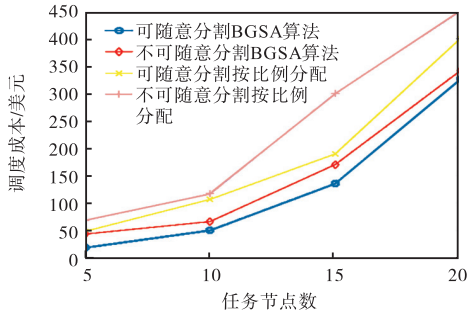

图 3　调度成本对比图

Fig. 3　Scheduling cost comparison chart

任务调度成本是在某一特定数目任务节点时生成的批处理科学工作流下，算法运行 100 次所取的平均值. 由于按比例划分截止期经典调度算法划分截止期的方式简单的按照任务规模按比例分配截止期，且配置虚拟机类型与数量只是简单按照是否满足截

止期配置，因此耗费的成本就会相对较高. 本文所提的改进后的遗传算法的不可随意分割批处理科学工作流任务调度算法，由于其搜索空间广且能并重虚拟机升级与并行度，且在算法得出最优解后进一步对非关键路径上所使用的虚拟机数量进行收缩，进一步缩减成本，所以解的效果要好一些；而基于遗传算法的可随意分割任务调度算法，由于模型相对理想化，所以在使用虚拟机类型时算法会比较偏向选择价格低但并行度高的方式处理任务. 亚马逊云资源的实际标价并不是线性增加的，因此可随意分割 BIGA 算法的调度成本会低于不可随意分割 BIGA.

为了证明算法的有效性，图 4 为改进遗传算法的成本调优图，具体实验数据为在任务节点数为 5 的随机 10 个批处理科学工作流生成 1 000 次初始解，其中每个批处理科学工作流生成 100 次，并计算这些初始解的平均调度成本，然后进行算法迭代，观察随着算法迭代次数的不断增加，解的平均调度成本的变化. 从图 4 可以看出改进遗传算法通过一定次数的迭代，与最初始解相比成本缩减了近 90%.

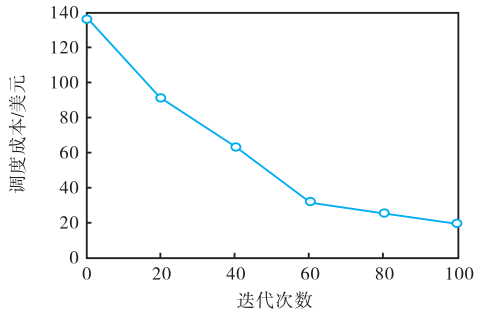

图 4　成本调优图

Fig. 4　Cost tuning diagram

为了检验本文种群初始化算法的有效性，将经过本文所提的初始化种群优化方法的可随意分割批处理科学工作流任务调度算法和不可随意分割批处理科学工作流任务调度算法与其他部分不变初始化种群为随机生成的两组算法所得的最终解的调度成本进行对比，图 5 和图 6 为基于改进遗传算法的可随意分割批处理科学工作流任务调度与不可随意分割批处理科学工作流任务调度算法进行 1 000 次初始种群优化与不进行初始种群优化的效果对比图.

由图 5 和图 6 可以看出：两种算法经过初始种群优化后所得的最终解均要大幅优于未经初始种群优化的算法，其原因主要是经过初始种群优化后所生成的初始种群不仅具有随机性，同时与随机生成的初始种群相比，大部分个体都相对更为优秀，这就使得在这样的种群中进行交叉变异更容易产生优秀解，一定程度上避免了算法陷入局部最优的可能性.

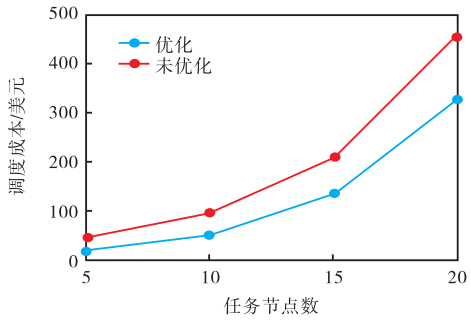

图 5　可随意分割 BIGA 初始种群优化与未优化最终成本对比

Fig. 5　Final cost comparison of optimized and unoptimized freely split BIGA initial population

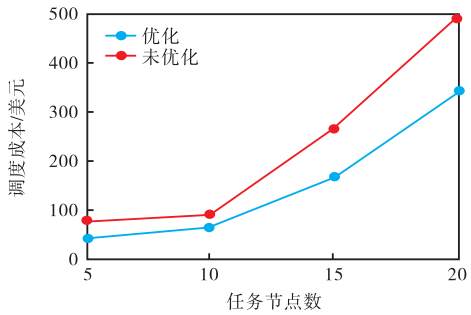

图 6　不可随意分割 BIGA 初始种群优化与未优化最终成本对比

Fig. 6　Final cost comparison of optimized and unoptimized not freely split BIGA initial population

最后，针对本文算法的执行效率问题，通过实验对比了 4 种算法的执行效率，结果如图 7 所示.

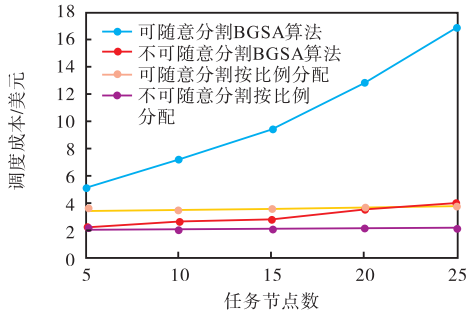

图 7　执行效率对比

Fig. 7　Comparison of performance efficiency

由图 7 可以看出：两种任务不可随意分割的任务调度算法的执行效率要明显高于另外两种可随意分割的任务调度算法，且两种任务可随意分割的任务调度算法随着任务节点数的增加其执行效率也会降低，而任务不可随意分割的两种任务调度算法则没有显

著影响. 其原因是在进行任务调度虚拟机资源配置计算时，由于任务不可随意分割的两种任务调度算法其任务组内的子任务不可随意分割，导致虚拟机配置数量上限受到子任务数影响，所以其解搜索空间相对较小；而任务可随意分割的两种任务调度算法则相反，其解空间随任务节点数的增加指数增大，所以相应的耗时也就更大. 另外，由于随着任务节点数的增大，其搜索空间是指数增大，所以耗时也会越来越多.

4　结　语

通过以上理论和实验分析，按比例划分截止期经典调度算法解的生成方式简单，且资源分配方案随机性太大，而所提的两种任务调度算法优化了初始解的生成，使得初始解不至于离全局最优解太远，加快了算法的收敛速度. 同时，通过并重虚拟机升级与并行度调整两种操作使最终解更趋近于全局最优，因此在任务调度成本上要比前者少. 下一步将重点研究数据密集型批处理工作流任务关联度聚类与本文方法的结合，以缩减网络传输耗时，从而更全面地解决批处理工作流的云调度问题.

致谢：本研究同时受到国家自然科学基金（11503051，61402325）、国家自然科学基金委员会-中国科学院天文联合基金（U1531111，U1531115，U1531246，U1731125，U1731243）资助，一并致谢.

参考文献：

[1] Benjamas N，Uthayopas P. Impact of I/O and execution scheduling strategies on large scale parallel data mining[C]//2012 6th International Conference on New Trends in Information Science，Service Science and Data Mining（ISSDM2012）. Information Science and Service Science and Data Mining. New York：IEEE，2012：654–660.

[2] 邹永贵，万建斌. 云计算环境下的资源管理研究[J]. 数字通信，2012，39（4）：39–43.

[3] 房秉毅，张云勇，程莹，等. 云计算国内外发展现状分析[J]. 电信科学，2010，26（S1）：1–6.

[4] Anwar N，Deng H. Elastic scheduling of scientific workflows under deadline constraints in cloud computing environments[J]. Future Internet，2018，10（1）：5.

[5] Liu S，Ren K，Deng K，et al. A task backfill based scientific workflow scheduling strategy on cloud platform[C]//2016 Sixth International Conference on Information Science and Technology（ICIST）. New York：IEEE，2016：105–110.

[6] Zhao Q，Xiong C，Wang P. Heuristic dataplacement for data-intensive applications in heterogeneous cloud[J]. Journal of Electrical & Computer Engineering，2016，2016（13）：1–8.

[7] Zhao Q，Xiong C，Zhao X，et al. A data placement strategy for data-intensive scientific workflows in cloud[C]// 15th IEEE/ACM International Symposium on Cluster，Cloud，and Grid Computing. New York：IEEE，2015：928–934.

[8] Zhao Q，Xiong C，Yu C，et al. A new energy-aware task scheduling method for data-intensive applications in the cloud[J]. Journal of Network and Computer Applications，2016，59：14–27.

[9] Rodriguez M，Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds[J]. IEEE Trans Cloud Computing，2014，2（2）：222–235.

[10] Mao M. Auto-scaling to minimize cost and meet application deadlines in cloud workflows[C]// Proceedings of 2011 International Conference for High Performance Computing，Networking，Storage and Analysis. New York：IEEE，2011：1–12.

[11] Cai Z，Li X，Gupta J N D. Heuristics for provisioning services to workflows in XaaS clouds[J]. IEEE Transactions on Services Computing，2016，9（2）：250–263.

[12] 秦勇，梁旭. 基于混合遗传算法的并行测试任务调度研究[J]. 国外电子测量技术，2016，35（9）：72–75.

责任编辑：郎婧