

面向云计算的花朵差分授粉 workflow 多目标优化算法研究

罗智勇 朱梓豪 谢志强 孙广路

(哈尔滨理工大学计算机科学与技术学院 黑龙江哈尔滨 150080)

摘要: 为解决云计算环境下 workflow 多目标难于优化的问题, 本文提出了一种花朵差分授粉 workflow 多目标调度优化算法. 该算法将 workflow 中任务和虚拟机建模成花粉, 将完整的调度序列建模成花朵. 依据任务的偏序关系进行离散花朵授粉过程. 仿真结果表明较算法 NSGA-II 和 MEOA/D, 该算法能在限定的截止期和预算的条件下具有更高的资源利用率.

关键词: workflow 调度 花朵授粉算法 多目标优化 云计算

中图分类号: TP393

文献标识码: A

文章编号: 0372-2112 (2021) 03-0470-07

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20191211

A Multi-objective Workflow Scheduling Algorithm Based on Flower Pollination Cloud Environment

LUO Zhi-yong ZHU Zi-hao XIE Zhi-qiang SUN Guang-lu

(School of Computer Science and Technology Harbin University of Science and Technology Harbin Heilongjiang 150080 China)

Abstract: In order to solve the problem that the multi-objective workflow scheduling is difficult to optimize in the cloud computing environment, this paper proposes a differential flower pollination algorithm for the multi-objective workflow scheduling. The algorithm models the tasks and virtual machines in the workflow into pollen and models the complete scheduling sequence into flowers. Then it adopts a discrete flower pollination process according to the partial order relationship of the task. The simulation results show that compared with the algorithms NSGA-II and MEOA/D, the algorithm can have higher resource utilization under the limited deadline and budget.

Key words: workflow scheduling; flower pollination algorithm; multi-objective optimization; cloud computing

1 引言

目前随着中国工业制造 2025 的提出, 相关企业对计算和大量存储资源的需求与日俱增. 由于云计算具有超大规模和廉价等特点, 逐渐被应用在各领域复杂的工作流调度问题上^[1]. 云计算环境下的工作流调度问题已经被证明是 NP 完全问题^[2], 可以想象, 当有 n 个任务和 m 个设备时, 存在 m^n 种建立任务到设备的映射调度方案, 当 n 和 m 比较大时, 通过穷举法的方式则会耗费非常大的代价. 用户希望尽可能的减少调度时间和费用, 服务提供商则希望以最小的代价获得最大化的资源利用率, 而这些工业目标又是相互冲突的, 因此整个工作流的调度过程是一个多目标优化问题^[3].

花朵授粉算法是 Yang 于 2012 年提出的新型元启发式算法, 该算法继承了其余智能算法的优点, 并且参数少易调节, 而且具有灵活, 适应性强, 可扩展和优化方法简单实现过程相对容易等特点, 一经提出就引起了不同领域学者的浓厚兴趣并对其进行做出相关研究.

本文在以往的研究基础上^[4], 提出了一种花朵差分授粉算法 DFPA (Differential Flower Pollination Algorithm), 所做出的贡献如下: (1) 基本的花朵授粉算法是用于解决数值优化问题, 本文通过改进其授粉更新方程, 使之适用于调度这类组合优化问题; (2) 工作流调度存在任务之间的依赖关系, 智能启发式算法随机生成初始种群存在产生不可行解的问题, 本文通过提出种群初始化算法以保证任务间的次序关系不被破坏;

(3) 随着迭代次数的增加, 种群的多样性会降低导致算法易陷入局部最优, 本文提出的算法在授粉更新操作之后加入了变异操作, 保证了种群的多样性, 让种群间差异性不会随着迭代次数的增加而减少, 更容易获得全局最优。

2 相关工作

2.1 工作流调度研究

近年来, 云计算下的工作流调度技术形成了一些科研成果, 但主要围绕时间和费用两大目标进行优化。文献[5]提出了一种基于模拟退火的双目标差分进化算法, 能够减少数据中心的任务丢失率的同时提高中心的利润。文献[6]提出了基于基因算法的双目标优化算法, 优化目标为系统的能耗和系统的信赖度, 实验结果表明由于 MOHEFT 和 MODE 算法。文献[7]提出一种改进的蚁群算法, 通过改进信息素更新规则, 优化了启发式策略, 避免蚂蚁陷入单目标优化同时增加了学习策略, 实验结果表明取得了较好的效果。文献[8]提出了一种模糊支配的双目标优化算法, 通过改进 HEFT 的排序机制, 实验表明在科学工作流上, 可以取得调度费用和执行时间的相对平衡。文献[9]提出了一种改进的引力搜索算法, 引入了 HEFT 的部分策略, 通过 MCR 和 SLR 调度评价准则比较优于对比算法。

2.2 花朵授粉算法研究

花朵授粉算法是一种基于群体的优化技术, 其令人印象深刻的特性引起了许多优化领域中研究人员的关注。文献[10]使用了基于动态转移概率的 FPA, 通过编码应用于全局和局部搜索, 对于发电系统中的经济分配问题, 与其余算法相比, 燃油成本更低。

3 问题描述

3.1 工作流的相关定义

定义 1 虚拟机 M 。IaaS 服务通常用虚拟机进行抽象表示其计算资源, 用户可以租用虚拟机去执行工作流任务, 虚拟机被形式化为四元组 $M = (P, C, B, F_{start})$, 其中 P 表示虚拟机计算能力; C 表示租用虚拟机单位时间费用; B 表示带宽; F_{start} 表示虚拟机启动标志, 当 $F_{start} = 1$ 时, 虚拟机已经启动, 当 $F_{start} = 0$ 时, 虚拟机未启动。

定义 2 工作流模型 G 。工作流被表示为有向无环图 (DAG), 可形式化为四元组 $G = (T, E, M, W)$, 其中 $T = \{t_1, t_2, \dots, t_N\}$ 表示任务集合; E 表示任务之间的依赖关系; M 表示虚拟机; W 表示任务和虚拟机的映射关系。

定义 3 执行时间 $T_{exe}(t_i, m_j)$ 。表示为工作流中任务 t_i 在虚拟机 m_j 上的执行时间, 可以由式 (1) 计算, 其中 $C_{mips}(t_i)$ 表示任务的操作指令数, 即任务的规模

大小。

$$T_{exe}(t_i, m_j) = \frac{C_{mips}(t_i)}{P(m_j)} \quad (1)$$

定义 4 传输时间 $T_{trans}(t_i, t_j)$ 。表示了任务 t_i 和任务 t_j 之间的数据传输时间, 可以由式 (2) 计算, 当其处于同一虚拟机上时, 传输时间可忽略不计并置为 0。

$$T_{trans}(t_i, t_j) = \begin{cases} \frac{D_{data}(t_i^{output})}{\min(B_i, B_j)}, & i \neq j \\ 0, & i = j \end{cases} \quad (2)$$

其中 t_i^{output} 表示任务 t_i 的传出数据大小, 单位为比特; B_i 和 B_j 分别表示任务 t_i 和 t_j 所在虚拟机的带宽, 单位为比特/s。

定义 5 虚拟机 m_j 的就绪时间 $T_{run1}(m_j)$ 。表示虚拟机 m_j 的当前最早可以开始执行任务的时间, 可以由式 (3) 计算:

$$T_{run1} = \begin{cases} \min\{T_{exe}(t_i, m_j)\}, & m_j.F_{start} = 1 \\ 0, & m_j.F_{start} = 0 \end{cases} \quad (3)$$

定义 6 任务 t_i 的就绪时间 $T_{run2}(t_i, m_j)$ 。表示任务 t_i 在虚拟机 m_j 上的最早可以开始执行的时间, 可以由式 (4) 计算:

$$T_{run2}(t_i, m_j) = \max_{t_j \in S_{parent}(t_i)} (T_{trans}(t_j, t_i) + T_{exe}(t_j, m_j)) \quad (4)$$

其中 $S_{parent}(t_i)$ 表示任务 t_i 的父任务集合, m_j 表示父任务 t_j 执行的虚拟机。

定义 7 任务 t_i 的实际开始时间 $T_{start}(t_i, m_j)$ 。表示任务 t_i 实际在虚拟机 m_j 上开始执行的时间, 可以由式 (5) 计算:

$$T_{start}(t_i, m_j) = \max\{T_{run1}(m_j), T_{run2}(t_i, m_j)\} \quad (5)$$

定义 8 完成时间 $T_{end}(t_i, m_j)$ 。表示任务 t_i 的在虚拟机 m_j 上的完成时间, 可以由式 (6) 计算:

$$T_{end}(t_i, m_j) = T_{start}(t_i, m_j) + T_{exe}(t_i, m_j) \quad (6)$$

3.2 调度优化目标

差分授粉算法将时间、费用和资源利用率作为优化目标, 可以具体形式化定义如下, 时间表示为整个任务的调度时间, 费用为租用虚拟机的总花费, 资源利用率为虚拟机集群忙碌状态下的占比, 用平衡度表示。

定义 9 调度时间 $T_{makespan}$ 。表示整个工作流的调度执行时间, 可以由式 (7) 计算:

$$T_{makespan} = T_{end}(t_{exit}) - T_{start}(t_{begin}) \quad (7)$$

其中 t_{begin} 和 t_{exit} 表示开始和结束任务。

定义 10 执行费用 $C_{exe}(t_i, m_j)$ 。表示在虚拟机 m_j 上执行任务 t_i 的费用, 可以由式 (8) 计算:

$$C_{exe}(t_i, m_j) = T_{exe}(t_i, m_j) * C(m_j) \quad (8)$$

其中 $C(m_j)$ 表示了单位时间内运行虚拟机 m_j 的价格。

定义 11 平衡度 D 。表示是否均衡使用虚拟机的程度, 平衡度数值越小表示虚拟机使用越均衡, 可以由

式(9)计算:

$$D(m_j) = \frac{T_{\max} - T_{\min}}{T_{\text{avg}}} \quad (9)$$

其中 T_{\max} 和 T_{\min} 分别表示虚拟机最大和最小执行时间, T_{avg} 表示虚拟机平均执行时间.

本文的调度是在用户设定截止期和预算的情况下,尽可能减少整个工作流的调度时间 T_{makespan} ,即总执行时间 f_{TET} ;减少执行整个工作流的费用 C_{total} ,即总执行费用 f_{TEC} ;减少虚拟机空闲等待时间,降低虚拟机平衡度 D ,即总虚拟机平衡度 f_{DI} . 优化目标可由式(10)计算:

$$\min F(x) = \{f_{\text{TET}}, f_{\text{TEC}}, f_{\text{DI}}\} \quad (10)$$

其中 $f_{\text{TET}} = T_{\text{makespan}}$, $f_{\text{TEC}} = \sum_{i=1}^N C_{\text{exe}}(t_i)$, $f_{\text{DI}} = \sum_{j=1}^M D(m_j)$.

多目标差分授粉算法调度目标为 $\min(f_{\text{TET}}, f_{\text{TEC}}, f_{\text{DI}})$ 通过求出其帕累托全局最优解来获得最佳的调度方案.

3.3 花朵差分授粉工作流调度优化算法

传统的花朵授粉算法是用于连续数值优化,其中迭代的解都是实数,而工作流调度问题都是组合优化,解是由一组整数序列组成,针对这个问题,对迭代过程中的调度解进行重新设计,具体如下:

在算法设计中,每个调度解是由两行的矩阵组成,该矩阵称为花朵个体. 矩阵的列数跟任务数一样表示一个花粉,矩阵的第一行为任务调度顺序,第二行为任务对应的虚拟机分配,整个花朵种群构成了解空间.

对于一个有 N 个任务和 M 个虚拟机的工作流调度问题而言,总共有 M^N 种不同的解,用两行矩阵可以表示调度所需信息,第一行表示任务的 ID,第二行表示虚拟机的选择,如具有六个花粉的花朵个体可表示为:

$$\begin{pmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\ m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \end{pmatrix}$$

其中: t_i 为任务编号, m_j 为虚拟机编号.

对于具有 N 个任务的工作流,第 i 个花朵的迭代过程可以表示为:

$$\begin{aligned} X^i &= \begin{pmatrix} X_1^i \\ X_2^i \end{pmatrix} = \begin{pmatrix} x_{11}^i & x_{12}^i & x_{13}^i & x_{14}^i & x_{15}^i & x_{16}^i \\ x_{21}^i & x_{22}^i & x_{23}^i & x_{24}^i & x_{25}^i & x_{26}^i \end{pmatrix} \\ &= \begin{pmatrix} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 \\ m_1 & m_2 & m_3 & m_4 & m_5 & m_6 \end{pmatrix} \end{aligned}$$

该矩阵表示整个种群中的第 i 个解.

为执行更新操作,需要将矩阵分成两个部分,矩阵的第二行虚拟机分配向量 $X_2^i = (x_{21}^i, x_{22}^i, x_{23}^i, x_{24}^i, x_{25}^i, x_{26}^i)$ 执行全局授粉和局部授粉操作.

由于传统的花朵授粉算法存在易陷入局部最优的问题,在完成授粉操作之后,需要对所有花朵个体矩阵

的第二行虚拟机分配向量 $X_2^i = (x_{21}^i, x_{22}^i, x_{23}^i, x_{24}^i, x_{25}^i, x_{26}^i)$ 执行变异杂交操作,相应变异杂交用式(11)计算:

$$\begin{cases} X_2^{i,j+1} = \begin{cases} V_2^{i,j+1}, & \text{if } \text{rand}(0,1) \leq p_{\text{CR}} \\ X_2^{i,j}, & \text{otherwise} \end{cases} \\ V_2^{i,j+1} = X_2^{h,j} + \alpha (X_2^{p,j} - X_2^{q,j}) \end{cases} \quad (11)$$

其中: p_{CR} 表示变异杂交概率, α 为变异因子, $V_2^{i,j+1}$ 表示第 t 次迭代过程中花朵 i 完成变异之后的新位置; $X_2^{h,j}$, $X_2^{p,j}$ 和 $X_2^{q,j}$ 表示随机从种群中选取互不相同的花朵 h , p 和 q .

在整个差分授粉过程完成后,将任务向量 X_1^i 和虚拟机分配向量 X_2^i 重新组成花朵个体 X_2^i ,通过式(10)来计算花朵个体 X^i 的适应值 $f_{\text{TET}}, f_{\text{TEC}}, f_{\text{DI}}$,然后通过帕累托准则来决定是否更新花朵的位置. 如果 $X^{i,j+1}$ 帕累托支配 $X^{i,j}$,那么花朵个体 X^i 的位置就更新成 $X^{i,j+1}$,否则花朵个体 X^i 的位置仍然保持 $X^{i,j}$.

在设计差分授粉工作流调度算法之前,需要初始化种群,因此设计种群初始化算法 Init_population 的伪代码如算法 1 所示.

算法 1 种群初始化算法

INPUT: 工作流 G , 种群规模 NP

OUTPUT: 相应规模的花朵种群 pop

1. int $i = 0$;
2. while $i < \text{NP}$ do
3. 根据任务集合 T 随机生成向量
4. while $j < \text{length}$ do
5. 检查任务 j 的父结点是否全都出现在其前面
6. if TRUE
7. Then $j++$
8. else
9. 从任务 j 的父结点中选一个未出现在其前面的结点 k
10. 交换 j 和 k 的值
11. $m = 0$
12. while $m < \text{length}$
13. do = rand(0, D)
14. 将向量组合成调度方案

完成初始化种群后,调用花朵差分授粉工作流调度算法 DFPA 进行优化调度,设计算法 DFPA 的伪代码如算法 2 所示.

算法 2 DFPA 算法

INPUT: 工作流 G , 种群规模 NP, 最大迭代次数 maxiter, 转移概率 p

OUTPUT: 最优调度方案

1. 根据式(1)和(2)计算出任务的执行时间和传输时间
2. 通过算法 2-1 获得初始种群
3. int $t = 0$;
4. while $t < \text{maxiter}$ do

5. 根据式(3)~(6)对种群中的个体进行调度
6. 根据式(7)~(9)计算出花朵个体的执行时间、执行费用 and 平衡度
7. 根据式(10)计算当前种群最优个体
8. for $i = 1$ to pop. size do
9. 将矩阵转换成两个向量
10. for do
11. if $\text{rand}() < p$ do
12. 根据式(11)~(12)执行全局授粉
13. else 根据式(13)执行局部授粉 end if
14. 根据式(14)执行杂交变异操作 end for
15. 将向量组合成调度方案
16. 根据式(10)判断是否更新当前个体
17. end for
18. end while
19. 输出最优个体

若任务数为 n 种群规模为 m 迭代次数为 p , 则算法 DFPA 的时间复杂度为 $O(m^* p + m^* n)$, 因为种群规模 $n \ll$ 迭代次数 p , 所以总的算法时间复杂度为 $O(m^* p)$.

4 仿真实验

4.1 实验环境

本次实验通过使用 workflowsim 仿真平台模拟了一个数据中心, 其中包括五种类型的虚拟机. 虚拟机的配置依据目前亚马逊 EC2 提供的虚拟机配置信息, 如表 1 所示. 仿真实验环境的物理配置为 Inter Core I5-7200U (2.5GHz) 和 8GB 内存. 算法的相关参数设计如表 2 所示.

表 1 实验中的虚拟机配置情况

型号	处理能力 (MIPS)	带宽 (比特/s)	费用 (万元/h)
m1.small	4400	39000	0.03
m1.large	17600	85000	0.12
m1.xlarge	35200	131000	0.24
c1.medium	22000	85000	0.15
c1.xlarge	88000	131000	0.60

表 2 算法的参数设置

参数	取值
种群大小	20
迭代次数	150
初始化种群	默认初始化
变异概率	0.2
初始转移概率	0.8

4.2 优化调度

本实验采用 Pegasus 中的四种科学 workflow 调度模型, 即模型 Sipht, LIGO, CyberShake 和 Epigenomic, 其模

型结构如图 1 所示. 将该数据集和四种科学 workflow 模型带入花朵差分授粉算法 DFPA 进行优化调度, 得出如表 3~表 6 的调度过程及优化结果.

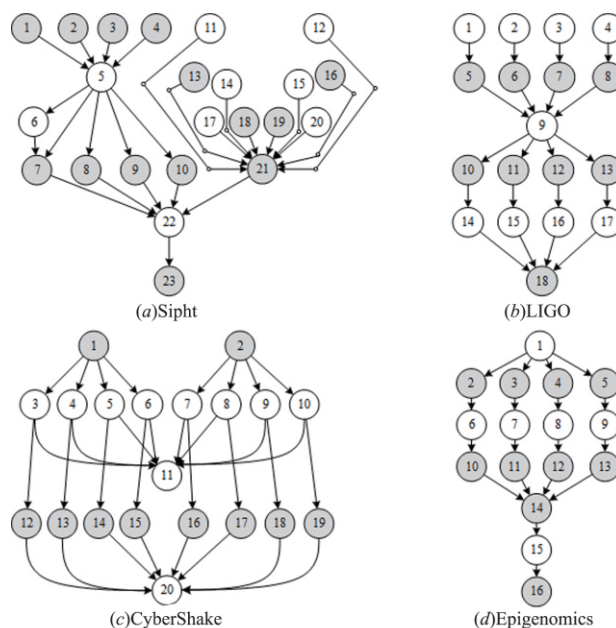


图 1 四种常见的工作流调度模型

在优化工作流模型 Sipht 时, 仿真实验采用该模型的任务平均长度约为 6000M, 传输数据大小约为 1000M, 设定的截止期为 1.5h, 则工作流模型 Sipht 被算法 DFPA 优化过程及结果如表 3 所示.

表 3 Sipht 调度实验过程

迭代次数	执行结果
1	任务调度顺序为 0 3 5 1 4 2 7 6 8 9, 对应执行的虚拟机为 2 4 4 2 3 2 2 3 2 2. TEC = 10.52 万元, TET = 2.46h, DI = 2.32. TET > 1.5h 不满足继续迭代
2	任务调度顺序为 0 1 2 5 4 3 6 7 8 9, 对应执行的虚拟机为 4 4 2 3 1 2 2 3 2 2. TEC = 11.84 万元, TET = 2.09h, DI = 2.26. TET > 1.5h 不满足继续迭代
...	...
50	任务调度顺序为 0 4 1 3 2 5 8 7 6 9, 对应执行的虚拟机为 4 4 2 3 1 2 4 2 3 4. TEC = 9.52 万元, TET = 1.46h, DI = 1.38. TET < 1.5h 满足停止迭代

表 3 显示, 工作流模型 Sipht 被算法 DFPA 优化时, 当迭代 50 次的时候达到全局最优解, 其中总执行费用为 9.52 万元, 总执行时间为 1.46h, 平衡度为 1.38.

在优化工作流模型 LIGO 时, 仿真实验采用该模型的任务平均长度约为 4000M, 传输数据大小约为 5000M, 设定的截止期为 2.35h, 则工作流模型 LIGO 被算法 DFPA 优化过程及结果如表 4 所示.

表4 LIGO 调度实验过程

迭代次数	执行结果
1	任务调度顺序为 0,1,5,3,2,6,4,8,7,9,对应执行的虚拟机为 2,4,4,2,3,1,3,3,2,2. TEC = 2.4 万元, TET = 4.16h, DI = 3.35. TET > 2.35h 不满足继续迭代
2	任务调度顺序为 0,3,1,2,5,6,4,8,7,9,对应执行的虚拟机为 4,4,2,3,1,1,3,3,2,2. TEC = 2.28 万元, TET = 2.88h, DI = 3.21. TET > 2.35h 不满足继续迭代
...	...
34	任务调度顺序为 0,1,4,5,3,2,8,6,7,9,对应执行的虚拟机为 4,4,4,2,3,2,2,4,2. TEC = 1.66 万元, TET = 2.34h, DI = 2.57. TET < = 2.35h 满足停止迭代

表4显示, workflow模型 LIGO 被算法 DFPA 优化时,当迭代34次的时候达到全局最优解,其中总执行费用为1.66万元,总执行时间为2.34h,平衡度为2.57.

在优化 workflow模型 CyberShake 时,仿真实验采用该模型的任务平均长度约为24000M,传输数据大小约为12000M,设定的截止期为9.15h,则 workflow模型 CyberShake 被算法 DFPA 优化过程及结果如表5所示.

表5 CyberShake 调度实验过程

迭代次数	执行结果
1	任务调度顺序为 0,5,2,6,1,4,3,7,8,9,对应执行的虚拟机为 2,2,4,2,1,1,4,2,4,2. TEC = 7.38 万元, TET = 9.74h, DI = 3.15. TET > 9.15h 不满足继续迭代
2	任务调度顺序为 0,5,1,4,2,6,3,7,8,9,对应执行的虚拟机为 4,2,3,1,3,2,4,2,4,2. TEC = 7.17 万元, TET = 9.35h, DI = 3.36. TET > 9.15h 不满足继续迭代
...	...
28	任务调度顺序为 0,1,4,5,3,2,8,6,7,9,对应执行的虚拟机为 4,4,4,2,3,2,2,4,2. TEC = 6.96 万元, TET = 9.13h, DI = 2.69. TET < = 9.15h 满足停止迭代

表5显示, workflow模型 CyberShake 被算法 DFPA 优化时,当迭代28次的时候达到全局最优解,其中总执行费用为6.96万元,总执行时间为9.13h,平衡度为2.69.

在优化 workflow模型 Epigenomics 时,仿真实验采用该模型的任务平均长度约为40000M,传输数据大小约为14000M,设定的截止期为13h,则 workflow模型 Epigenomics 被算法 DFPA 优化过程及结果如表6所示.表6显示, workflow模型 Epigenomics 被算法 DFPA 优化时,当迭代36次的时候达到全局最优解,其中总执行费用为9.51万元,总执行时间为12.71h,平衡度为2.93.

表6 Epigenomics 调度实验过程

迭代次数	执行结果
1	任务调度顺序为 0,1,4,5,3,2,7,6,8,9,对应执行的虚拟机为 4,3,3,3,4,2,2,2,3,2. TEC = 10.02 万元, TET = 20.36h, DI = 3.49. TET > 13h 不满足继续迭代
2	任务调度顺序为 0,1,4,5,2,6,3,7,8,9,对应执行的虚拟机为 4,2,2,2,4,2,4,3,2,2. TEC = 9.9 万元, TET = 19.4h, DI = 3.12. TET > 13h 不满足继续迭代
...	...
36	任务调度顺序为 0,1,4,5,3,7,8,2,6,9,对应执行的虚拟机为 2,1,1,1,4,4,4,2,2,2. TEC = 9.51 万元, TET = 12.71h, DI = 2.93. TET < = 13h 满足停止迭代

5 算法对比

围绕优化目标总执行费用 f_{TEC} 、总执行时间 f_{TET} 和平衡度 f_{DI} , 本文将算法 DFPA 与传统多目标优化算法 NSGA-II 和 MOEA/D 在优化结果上进行了对比. 当设定100个结点时,四种科学 workflow 分别在优化算法 DFPA、NSGA-II 和 MOEA/D 下总执行时间 TET 的对比情况如图2所示.

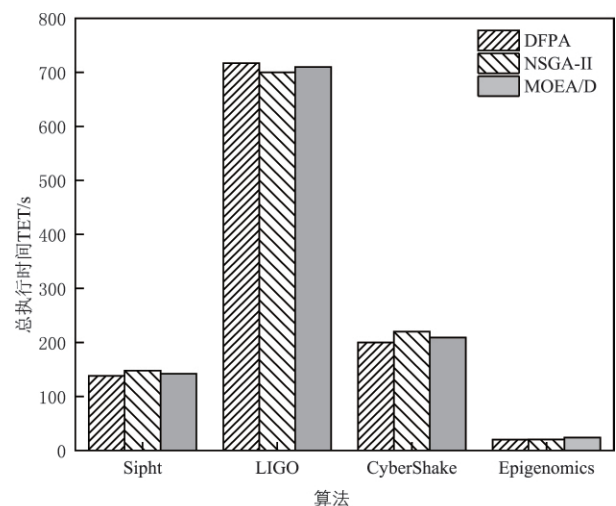


图2 总执行时间TET的柱状图

分析图2发现,在 Sipht 和 CyberShake workflow 上调度结果,DFPA 平均调度时间要低于 NSGA-II 和 MOEA/D. 在 Epigenomics 上,三种调度算法的平均调度时间没有明显差别,而在 LIGO workflow 上,NSGA-II 算法的表现得较为稳定. 当设定100个结点时,四种科学 workflow 分别在优化算法 DFPA、NSGA-II 和 MOEA/D 下的平衡度 DI 对比情况如图3所示.

分析图3发现,在 CyberShake workflow 调度的结果上,DFPA 算法远优于 MOEA/D,在 Sipht、LIGO 和 Epigenomics workflow 调度的结果上,DFPA 与 NSGA-II 和 MOEA/D

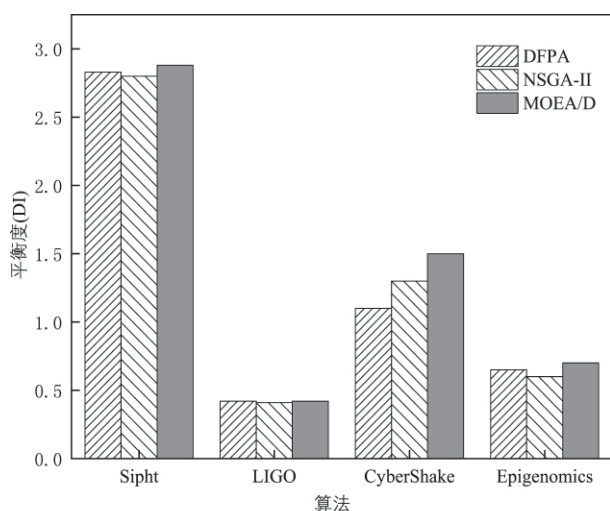


图3 平衡度DI的柱状图

等算法的性能差以不大,此结果可以看出,在工作流为串并行结构的情况下,DFPA 得到的平衡度较低,相较于其他算法更能充分利用资源。当设定 100 个结点时,四种科学工作流分别在优化算法 DFPA、NSGA-II 和 MOEA/D 下总执行费用 TEC 的对比情况如图 4 所示。

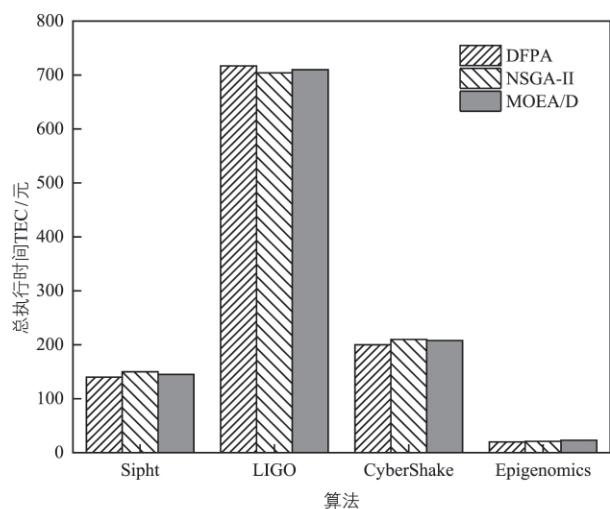


图4 总执行费用TEC的柱状图

分析图 4 发现,在 Epigenomics 上,三种算法所花费开销基本一致,除去 LIGO 提出的 DFPA 在剩余两种工作流上略优于 NSGA-II 和 MOEA/D 算法。综上所述,结合总的执行费用 TEC,平衡度 DI 和总的执行时间 TET 三种指标来看,提出的 DFPA 算法在 CyberShake、Sipht 和 Epigenomics 等结构的工作流上可以取得较好效果,在像 LIGO 结构的不规则工作流调度上,表现情况一般。

6 结论

工作流调度问题是一个 NP-Hard 问题,尤其是在复

杂的云计算环境下的多目标工作流调度,现有的相关研究略显不足。本文针对这个不足提出了花朵差分授粉的多目标工作流调度算法,该算法根据工作流任务之间的次序关系建立花朵模型。随后,结合差分变异杂交操作对花朵个体进行更新,通过不停的迭代更新种群,最终得到最佳的任务-虚拟机映射序列。通过和算法 NSGA-II 及 MOEA/D 进行对比,该算法具有更好的多目标优化效果,值得进一步研究。

参考文献

- [1] Choudhary A, Gupta I, Singh V, et al. A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing [J]. Future Generation Computer Systems 2018, 83(6): 14-26.
- [2] MANASRAH A M, BA ALI H. Workflow scheduling using hybrid GA-PSO algorithm in cloud computing [J]. Wireless Communications and Mobile Computing, 2018, 2018(1): 1-16.
- [3] Yao G, Ding Y, Jin Y, et al. Endocrine-based coevolutionary multi-swarm for multi-objective workflow scheduling in a cloud system [J]. Soft Computing, 2017, 21(15): 4309-4322.
- [4] 罗智勇, 朱梓豪, 尤波, 等. 基于虚拟归约工作流三层决策模型的时间-质量优化算法 [J]. 电子学报, 2019, 47(01): 245-251.
Luo Zhiyong, Zhu Zihao, You Bo, et al. A time-quality optimization algorithm of three-layer decision model based on virtual iterative workflow [J]. Acta Electronica Sinica, 2019, 47(01): 245-251. (in Chinese)
- [5] Yuan H, Bi J, Zhou M, et al. Biobjective task scheduling for distributed green data centers [EB/OL]. <https://xplore.staging.ieee.org/document/8951255>, 2020.07.
- [6] Zhang L, Li K, Li C, et al. Bi-objective workflow scheduling of the energy consumption and reliability in heterogeneous computing systems [J]. Information Sciences, 2017, 379: 241-256.
- [7] Chen Z, Zhan Z, Lin Y, et al. Multiobjective cloud workflow scheduling: A multiple populations ant colony system approach [J]. IEEE Transactions on Cybernetics, 2018, 49(8): 2912-2926.
- [8] Zhou X, Zhang G, Sun J, et al. Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT [J]. Future Generation Computer Systems, 2019, 93(4): 278-289.
- [9] Liu J, Liu L, Li Y. A differential evolution flower pollination algorithm with dynamic switch probability [J]. Chinese Journal of Electronics, 2019, 28(04): 737-747.
- [10] Putra P H, Saputra T A. Modified flower pollination algo-

rithm for nonsmooth and multiple fuel options economic dispatch [A]. 2016 8th International Conference on Infor-

mation Technology and Electrical Engineering, ICITEE 2016 [C]. Yogyakarta, Indonesia: IEEE 2016, 11: 1-5.

作者简介



罗智勇(通信作者) 男,1978 年出生于山东,哈尔滨理工大学计算机科学与技术学院、博士、教授.研究方向为企业智能计算与调度系统,数据处理,网络优化.
E-mail: luozhiyongemail@sina.com



谢志强 男,1962 年出生于黑龙江,哈尔滨理工大学计算机科学与技术学院、博士、教授,CCF 高级会员.研究方向为企业智能计算与调度系统,数据处理,网络优化.
E-mail: xiezhiqiang@hrbust.edu.cn



朱梓豪 男,1994 年出生于浙江,哈尔滨理工大学计算机科学与技术学院硕士生.研究方向为 workflow 智能调度.
E-mail: tzuhaozhu@gmail.com



孙广路 男,1979 年出生于黑龙江,哈尔滨理工大学计算机科学与技术学院、博士、教授.研究方向为机器学习与智能信息处理,计算机网络与信息安全.
E-mail: guanglu_sun@163.com