

# 一种 DAG 云任务多目标调度算法

郑宇超<sup>1</sup>, 夏学文<sup>1</sup>, 艾冬梅<sup>2</sup>

(1 华东交通大学 软件学院, 江西 南昌 330013;

2 北京科技大学 数理学院, 北京 100083)

**摘 要:** 提出了一种基于模糊占优排序的 DAG 任务调度离散粒子群算法 FDS-PSO. 首先, 构建了预算与截止期限双 QoS 约束的多目标调度优化模型, 模型构建了任务执行时间、执行代价及调度可靠性的多目标优化指标; 然后, 设计了一种离散 PSO 算法求解多冲突目标的均衡最优优化问题, 算法通过模糊占优排序的形式量化多目标解的相对适应度, 进而得到满足帕累托最优的调度均衡解集合; 最后, 通过随机 workflow 结构实验仿真, 与同类智能群体多目标调度算法进行性能比较. 仿真结果证明, FDS-PSO 算法不仅求解调度解的收敛速度更快, 且解空间分布更均匀, 能以更高的效率实现多目标的均衡优化.

**关键词:** 云计算; 任务调度; 粒子群算法; 模糊占优排序; 多目标优化

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1000-7180(2018)08-0048-07

## A Multi-objective Scheduling Algorithm of DAG Cloud Tasks

ZHENG Yu-chao<sup>1</sup>, XIA Xue-wen<sup>1</sup>, AI Dong-mei<sup>2</sup>

(1 School of software, East China Jiaotong University, Nanchang 330013, China;

2 School of Mathematics and Physics, University of Science and Technology Beijing, Beijing 100083, China)

**Abstract:** A DAG tasks scheduling algorithm FDS-PSO of discrete particle swarm optimization based on the fuzzy dominance sort is presented. First, the multi-objective optimization model of workflow scheduling under bi-QoS with the budget and deadline constraint is established, which introduces three optimization objectives, including the execution makespan, the execution cost and the scheduling reliability. Second, a discrete particle swarm optimization algorithm is designed to solve this three conflicting objectives optimization. FDS-PSO can quantify the relative fitness of multi-objective solutions through the fuzzy dominance sorting and obtain the solutions set of workflow scheduling satisfying Pareto optimal. Finally, through the simulation experiments of generated randomly workflow, compared with the same types of intelligent swarm multi-objective scheduling algorithms. Experimental results show that the scheduling solution obtained by FDS-PSO not only has better convergence, but has better and uniform spacing distribution among solutions, which can better achieve multi-objective trade-off optimization.

**Key words:** cloud computing; tasks scheduling; particle swarm optimization; fuzzy dominance sort; multi-objective optimization

### 1 引言

工作流调度通常需要巨大的计算与通信代价, 尤其科学工作流形式, 对调度环境性能要求更高, 而云计算的出现则为这类工作流调度提供了更加高效与适用的环境支撑<sup>[1]</sup>. DAG 云任务调度即为工作流

调度问题, 与传统的独立任务的并行执行不同, 工作流任务间具有约束与依赖性, 其调度本质上是实现相互依赖与关联的任务至可用云资源间的映射与分配, 同时需要满足用户对任务执行时关于费用与时间上的相关 QoS 约束. 云环境中, 不同能力的资源提供具有不同的使用价格, 使用模式是即付即用的,

且资源能力越高,价格越高,对应的使用代价也越高,因此,调度方案的不同将导致最终的工作流执行时间与执行代价均不同.因此,云工作流任务调度需要同步考虑预算和代价约束,进而实现执行时间与执行代价的最小化.

传统的工作流任务调度是 NP 问题,常规的最小执行时间算法 MCT、最小最小算法 MIN-MIN 和最大最小算法 MAX-MIN 等均是启发式方法,属于尽力服务类调度.另外,启发式线性调度是另一种常用调度算法,如:异构最早完成时间算法 HEFT<sup>[2]</sup>、资源关键路径算法 CPOP<sup>[2]</sup>、基于任务动态分级的调度算法 DLS<sup>[3]</sup>以及基于动态关键路径的调度算法 DCP<sup>[4]</sup>等.该类算法相同之处在于均以最小化执行时间为优化目标,未考虑工作流执行时需向资源提供方支付的代价,这并不适用于云计算环境中的资源即付即用的使用与提供模式.

多目标调度中,文献[5]提出 NSGA-II 算法,考虑了双相互冲突目标的优化:执行时间和可靠性,并得到了近似 Pareto 最优边界的解.文献[6]提出多目标线性调度算法 MOLS,可得到满足 Pareto 最优的占优解,但同样是优化了双目标.文献[7]利用 Pareto 占优的概念同步考虑时间与代价优化,得到代价最小的工作流调度方案,但算法未考虑预算与截止期限等约束条件.文献[8]对 MOHEFT 进行扩展,实现时间与能耗双目标优化.文献[9]则扩展 HEFT 实现联合云中时间/代价帕累托均衡优化.文献[10]则设计预算与截止期限约束多目标算法 BHEFT,但算法中资源数量相对固定,不符合云资源动态可扩展特点.为解决该问题,文献[11]提出多目标调度算法 MOPSO,文献[12]实现双约束下时间与代价双目标的最小化调度方案.

综合相关研究,现有大多数多目标工作流任务调度算法仅适用于传统分布式计算环境,应用于云环境中的多目标调度算法中,多数仅着眼于执行时间和代价的双目标优化,忽略了工作流调度时用户的 QoS 约束,更重要的是忽略了资源出现失效时的调度可靠性问题.本文将结合已有的双目标调度优化,在算法中融入截止时间与预算的双 QoS 约束,并将资源可靠性考虑至工作流作各调度问题中,以三目标优化为算法,并求解云工作流调度的 Pareto 最优解集,实现多目标间的均衡优化.

## 2 多目标系统模型

DAG 任务即工作流应用,可表达为有向无循环

图模型  $DAG = (V, G)$ ,  $V$  为图的顶点集,表示可执行于任意可用资源的  $n$  个任务  $t_i \in V, 1 \leq i \leq n$ .  $E$  为图的边集,边  $e_{ij} = (t_i, t_j) \in E, 1 \leq i \leq n, 1 \leq j \leq n, i \neq j$ ,表示任务  $t_i$  与  $t_j$  间的依赖,  $t_i$  为父任务,  $t_j$  为子任务.无前戏的称为入口任务,无后继的称为出口任务.分配于任务  $t_i$  的权重  $w(t_i)$  为任务执行的指令数,分配于边  $e_{ij}$  的权重  $w(e_{ij})$  为调度于不同资源上的任务  $t_i$  与  $t_j$  间的数据传输量.

云资源表示为集合  $R = \{r_1, r_2, \dots, r_m\}$ ,由  $m$  个全连通的计算资源组成,资源拥有不同的存储能力、处理能力、价格和失效率,资源间的通信链路拥有不同的带宽.利用  $m \times m$  数据传输时间矩阵存储资源间的通信时间,  $B_{s,j}$  用于存储资源  $r_s$  与  $r_j$  间传输单位数据的时间.因此,每个资源  $r_j$  应具有以下属性:(1) 属性  $\gamma_j$ :表示计算速率;(2) 属性  $c_j$ :表示资源  $r_j$  的单位时间的使用代价;(3) 属性  $\lambda_j$ :表示资源失效率.假设资源失效率服从泊松分布,可根据过去的系统性能日志或统计预测技术得到.

云环境中的工作流任务调度问题即为分配具有依赖约束的 DAG 任务至不同的可用云资源上.本文中,每个调度解表示为任务分配串,对应于调度顺序串.任务分配串即为每个任务至资源的可用时间槽的分配,调度顺序串则对调度任务进行编码.调度顺序串中的任务序列必须满足任务间的依赖关系.为了描述任务  $t_i$  至合适资源  $r_j$  间的调度  $S: V \rightarrow R$ ,定义  $EST(t_i)$  表示任务  $t_i$  在资源  $r_j$  上的最早开始执行时间,  $EFT(t_i)$  表示任务  $t_i$  在资源  $r_j$  上的最早完成时间,则有:

$$EST(t_i) = \max_{t_p \in \text{pred}(t_i)} \{EFT(t_p) + DCT_{p,i}\} \quad (1)$$

$$EFT(t_i) = EST(t_i) + w(t_i) \times \gamma_j \quad (2)$$

式中,  $\text{pred}(t_i)$  为任务  $t_i$  的直接前驱任务集;  $DCT_{p,i}$  为任务  $t_p$  (调度于资源  $r_s$ ) 与任务  $t_i$  (调度于资源  $r_j$  上) 间数据传输总时间,且:

$$DCP_{p,i} = w(e_{pi}) \times B_{s,j} \quad (3)$$

对于入口任务  $t_{\text{entry}}$  有:

$$EST(t_{\text{entry}}) = 0 \quad (4)$$

对于 DAG 中的其他任务, EST 和 EFT 通过公式(1)和(2)从入口任务开始进行递归计算.

令  $k$  为资源  $r_j$  上执行的任务数量,  $FT_j$  为资源  $r_j$  上最后完成任务的完成时间,则:

$$FT_j = \max_{1,2,\dots,k} \{EFT(t_k)\} \quad (5)$$

由于云资源的使用需要按使用时间支付固定费用,令  $C_{i,j}$  为资源  $r_j$  上执行  $t_i$  的代价,则:

$$C_{i,j} = c_j \times w(t_i) \times \gamma_j \quad (6)$$

双目标 workflow 调度问题: 考虑 workflow 调度两个冲突目标的最优化问题, 即最小化执行时间和执行代价, 形式化为:

$$\min \text{Time}(S) = \text{EFT}(t_{\text{exit}}) \quad (7)$$

$$\min \text{Cost}(S) = \sum C_{i,j} \quad (8)$$

$$\text{s. t } \text{Cost}(S) < B \text{ and } \text{Time}(S) < D \quad (9)$$

式中,  $B$  为预算约束;  $D$  为截止时间约束。

三目标 workflow 调度问题: 除了考虑执行时间和执行代价最优之外, 将资源失效带来的可靠性考虑进来。workflow 调度的可靠性即为所有 DAG 任务在分配资源上成功执行的概率, 可以资源正常工作的概率进行描述。令  $\lambda_j$  为资源  $r_j$  的失效率, 则调度  $S$  中资源  $r_j$  能够成功完成所有分配任务的概率为:

$$R_S^i = \exp^{-\lambda_j \times \text{FT}_j} \quad (10)$$

因此, 调度  $S$  的可靠性  $R_S$  为任务在所有资源上成功执行的概率的乘积:

$$R_S = \prod_{j=1}^m R_S^i = \exp^{-\sum_{j=1}^m \lambda_j \times \text{FT}_j} \quad (11)$$

最大化可靠性需要最小化调度  $S$  的可靠性因子  $RI$ , 定义为:

$$RI = \sum_{j=1}^m \lambda_j \times \text{FT}_j \quad (12)$$

则最大化可靠性可形式化为可靠性因子  $RI$  的最小化, 即:

$$\min \text{Reliability Index}(S) = RI \quad (13)$$

则三目标优化即为公式(7)、(8)和(13)的同步最小化, 约束条件为公式(9)。

### 3 多目标最优化问题

多目标最优化问题 MOP(Multi-objective Optimization Problem) 即为多个相互冲突目标的同步最优化问题, 目标是寻找均衡解, 即以非占优解集合的形式表示的 Pareto 最优解, 该集合中的每一单个解均为一个可行解。以最小化形式表示为:

$$\min f_i(s), \quad i = 1, 2, \dots, M \quad (14)$$

式中,  $f_i$  为第  $i$  个目标函数;  $s$  为一个解的决策向量;  $M$  为优化的目标数量。MOP 中存在两种关系: 常规 Pareto 占优和  $\epsilon$ -Pareto 占优。

常规 Pareto 占优: 令  $f(s) = (f_1(s), f_2(s), \dots, f_M(s))$  由  $M$  个目标组成, 考虑两个解向量  $s_1$  和  $s_2$ , 如果以下两个条件满足, 则称  $s_1$  常规占优  $s_2$ , 表示为  $s_1 < s_2$ :

(1) 对于任意  $i \in \{1, 2, \dots, M\}$ ,  $f_i(s_1) \leq f_i(s_2)$ ;

(2) 存在  $j \in \{1, 2, \dots, M\}$ ,  $f_j(s_1) < f_j(s_2)$ 。

$\epsilon$ -Pareto 占优: 如果以下两个条件满足, 则称  $s_1 \epsilon$ -Pareto 占优于  $s_2$ :

(1) 对于任意  $i \in \{1, 2, \dots, M\}$ ,  $f_i(s_1 \setminus \epsilon_i) \leq f_i(s_2 \setminus \epsilon_i)$ ;

(2) 存在  $j \in \{1, 2, \dots, M\}$ ,  $f_j(s_1) \setminus \epsilon_j < f_j(s_2) \setminus \epsilon_j$ 。

对于给定的集合, 解之间的非占优关系是已知的。将非占优解映射至目标解空间中得到的边界即为 Pareto 最优边界。

### 4 基于模糊占优排序的离散 PSO 算法

多目标粒子群优化算法的主要问题是选择全局和局部吸引粒子, 使得种群能向着 Pareto 最优边界的方向移动, 同时保持种群多样性。为解决这一问题, 本文使用模糊占优排序机制选择全局吸引粒子, 在粒子速率更新时, 基于模糊占优机制, 粒子朝着拥有更低分级的粒子(全局吸引粒子)运行。如此, 可以产生种群的新位置, 先前位置则存储于内存文档中。以下分几个部分逐步介绍基于模糊占优排序的离散 PSO 算法机理。

#### 4.1 离散 PSO 算法机理

问题编码: 为了优化云 workflow 调度问题, 产生任务分配串  $S$ , 用于映射任务  $t_i$  与资源  $r_j$  间的调度关系。如图 1 所示为云 workflow 的 DAG 任务结构图, 边上的权重值为任务间的通信代价, 图 1 给出了对应于该 DAG 任务结构的一种可行任务分配串。

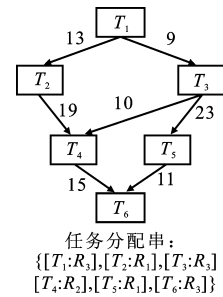


图 1 工作流结构及对应的任务分配串

任务分配串被映射至  $m \times n$  矩阵中, 定义为位置矩阵,  $m$  表示可用资源数量,  $n$  表示任务数量。令  $\mathbf{X}^k()$  为第  $k$  个种群粒子的位置矩阵,  $\mathbf{X}^k() = [X_{(i,j)}^k()]$ , 则:

$$\mathbf{X}_{(i,j)}^k() = \{0, 1\} (\forall i, j), i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\} \quad (15)$$

式中,  $\mathbf{X}_{(i,j)}^k = 1$  表示任务  $j$  执行于资源  $i$  上。此时, 矩阵的每一列仅有一个元素 1, 其它元素均为 0。表 1 是以 PSO 的编码形式将任务分配串映射至粒子位

置矩阵的结果.

表1 粒子位置矩阵

| 资源    | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ |
|-------|-------|-------|-------|-------|-------|-------|
| $R_1$ | 0     | 1     | 0     | 0     | 1     | 0     |
| $R_2$ | 0     | 0     | 0     | 1     | 0     | 0     |
| $R_3$ | 1     | 0     | 1     | 0     | 0     | 1     |

粒子  $k$  的速度矩阵为  $\mathbf{V}^k() = [\mathbf{V}_{(i,j)}^k()]$ , 则:

$$\mathbf{V}_{(i,j)}^k() \in \{-\mathbf{V}_{\max}, \mathbf{V}_{\max}\} (\forall i, j), i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\} \quad (16)$$

粒子  $k$  访问的最优位置为  $Pbest_{(i,j)}^k, Gbest_{(i,j)}^k$  表示粒子  $k$  及其邻居粒子访问的最优位置. 在算法每次迭代中,  $Pbest_{(i,j)}^k$  和  $Gbest_{(i,j)}^k$  的更新方法是利用后文 4.2 节提出的模糊占优排序机制进行.

速度更新: 根据公式(17) 更新粒子速度矩阵, 根据公式(18) 更新粒子位置矩阵:

$$\begin{aligned} \mathbf{V}_{(i,j)}^k(t+1) = & \omega \mathbf{V}_{(i,j)}^k(t) + c_1 r_1 (Pbest_{(i,j)}^k(t) - X_{(i,j)}^k(t)) \\ & + c_2 r_2 (Gbest_{(i,j)}^k(t) - X_{(i,j)}^k(t)) \end{aligned} \quad (17)$$

$$\begin{aligned} X_{(i,j)}^k(t+1) = & \begin{cases} 1, & \text{if } V_{(i,j)}^k(t+1) = \max\{V_{(i,j)}^k(t+1)\}, \\ & \forall i \in \{1, 2, \dots, m\} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (18)$$

式中,  $c_1$  和  $c_2$  为学习因子;  $c_1$  值越大, 粒子在搜索空间中将出现更大的背离; 而  $c_2$  值越大, 粒子将朝着其全局最优方向收敛. 迭代  $t$  的  $c_1$  和  $c_2$  更新机制为:

$$c_1 = (c_{1f} - c_{1i}) \frac{t}{\max\_t} + c_{1i} \quad (19)$$

$$c_2 = (c_{2f} - c_{2i}) \frac{t}{\max\_t} + c_{2i} \quad (20)$$

式中,  $c_{1f}$  和  $c_{2f}$  为学习因子的最终值;  $c_{1i}$  和  $c_{2i}$  为初始值.  $r_1$  和  $r_2$  为  $[0, 1]$  间的随机数,  $\omega$  为惯性权重, 用于控制当前速度对粒子行为的影响. 当  $\omega$  较大时, 算法具有较强全局搜索能力, 当  $\omega$  较小时, 算法具有较强局部搜索能力. 引入惯性权重函数为:

$$\omega = (\omega_{\max} - \omega_{\min}) \frac{\max\_t - t}{\max\_t} + \omega_{\min} \quad (21)$$

在位置矩阵的每一列中, 利用公式(18), 数值 1 被分配为元素, 其对应的元素在速度矩阵中拥有其对应列中的最大值. 如果速度矩阵的一列拥有多于一个元素, 则随机选择一个元素, 同时将数值 1 分配至其对应的位置矩阵的元素中.

适应度函数: 本文的目标是 workflow 执行跨度 makespan、执行代价和可靠性的多目标优化, 则定

义适应度函数  $F_{\text{makespan}}(S)$ 、 $F_{\text{cost}}(S)$  和  $F_{\text{reliability}}(S)$  评估种群个体, 计算方式如公式(7)、(8) 和(13) 所示. 若出现截止时间和预算约束违例时, 则需要将惩罚值添加至各自的目标函数中.

#### 4.2 模糊占优机理

假设 MOP 的目标是最小化  $M$  个目标函数  $f_i(s), i = 1, 2, \dots, M$ . 包含所有可能的解向量的集合表示为  $\Psi \subseteq R^n, n$  表示维度.

一个解的模糊  $i$ -占优定义为: 给定单调非递减函数  $\mu_i^{\text{dom}}()$ , 其范围为  $[0, 1], i \in \{1, 2, \dots, M\}$ , 一个解  $u \in \Psi$  视为  $i$ -占优解  $v \in \Psi$ , 当且仅当  $f_i(u) < f_i(v)$  成立时. 该关系可定义为:  $u >_i^F v$ . 如果  $u >_i^F v$ , 模糊  $i$ -占优度等于  $\mu_i^{\text{dom}}(f_i(v) - f_i(u)) \equiv \mu_i^{\text{dom}}(u >_i^F v)$ . 模糊占优可视为  $u$  与  $v$  间的模糊关系  $u >_i^F v$ .

一个解的模糊占优定义为: 解  $u \in \Psi$  视为模糊占优解  $v \in \Psi$ , 当且仅当对于  $i \in \{1, 2, \dots, M\}, u >_i^F v$  均成立时. 该关系可表示为:  $u >^F v$ . 模糊占优度的计算方式为:

$$\mu^{\text{dom}}(u >^F v) = \bigcap_{i=1}^M \mu_i^{\text{dom}}(u >_i^F v) \quad (22)$$

传统的模糊占优实现过程中, 对于负值参数, 用于计算模糊  $i$ -占优的隶属函数  $\mu_i^{\text{dom}}()$  被定义为 0. 因此, 如果  $f_i(u) > f_i(v)$ , 模糊占优度  $u >_i^F v$  必须为 0. 本文将改进为非零值, 即通过利用  $\epsilon$  值, 非占优解可无须分配零值. 隶属函数设置为梯形的, 定义为:

$$\mu_i^{\text{dom}}(\Delta f_i) = \begin{cases} 0, & \Delta f_i \leq -\epsilon \\ \Delta f_i / \Delta_i, & -\epsilon < \Delta f_i < \Delta_i - \epsilon \\ 1, & \Delta f_i \geq \Delta_i - \epsilon \end{cases} \quad (23)$$

式中,  $\Delta f_i = f_i(v) - f_i(u)$ .

种群的模糊占优: 给定一个种群解  $S \in \Psi$ , 一个解  $v \in S$  可视为  $S$  中的模糊占优解, 当且仅当该解被任一解  $u \in S$  模糊占优时. 此时, 模糊占优度可通过在每个可能的  $\mu^{\text{dom}}(u >^F v)$  上执行合并操作得到, 即:

$$\mu^{\text{dom}}(S >^F v) = \bigcup_{u \in S} \mu^{\text{dom}}(u >^F v) \quad (24)$$

通过该方法, 可以为每个解分配一个度量其在种群中与其它个体占优度的衡量值, 集合中越优解分配越小模糊占优值, 通过该排序方法, 在每次迭代中可选择最优的全局粒子.

#### 5 FDS-PSO 算法设计

为了求解多目标调度优化, FDS-PSO 算法步骤如下:

步骤 1: 初始化种群大小  $N$ , 设置迭代  $t = 0$

① 随机初始化  $m \times n$  维种群粒子,  $X_{(i,j)}^k(0), k \in$

$\{1, 2, \dots, N\}$ .

② 初始化所有粒子速度  $V_{(i,j)}^k(0) = 0$ , 个体最优位置  $Pbest_{(i,j)}^k(0) = X_{(i,j)}^k(0)$ .

步骤 2: 根据适应度函数评估种群粒子, 基于模糊占优和周长对种群粒子进行排序, 并将排序结果初始化为文档  $A_{(i,j)}^k(0)$ .

步骤 3: 更新迭代  $t, t = t + 1$

步骤 4: 基于 PSO 操作算子, 重复以下循环过程

① 利用二进制竞标选择法初始化文档中粒子  $k$  的全局最优位置  $Gbest_{(i,j)}^k(t)$

② 根据公式(17) 更新粒子  $k$  的速度  $V_{(i,j)}^k(t)$

③ 根据公式(18) 更新粒子  $k$  的位置  $X_{(i,j)}^k(t)$

④ 粒子位置进行自适应变异

⑤ 在所有粒子上重复执行循环

步骤 5: 评估种群中的每个粒子  $X_{(i,j)}^k(t)$

步骤 6: 联合当前粒子位置与前次迭代中文档粒子位置, 形成  $2N$  个粒子位置. 基于模糊占优排序和周长机制, 选择最优的  $N$  个解, 最后更新文档

步骤 7: 更新每个粒子的  $Pbest_{(i,j)}^k(t)$  和  $Gbest_{(i,j)}^k(t)$

步骤 8: 增加循环计数. 如果计数小于迭代最大值  $\max\_t$ , 返回步骤 3; 否则, 输出模糊占优解.

### 5.1 外部文档更新规则

引入 elite 文档存储多目标优化算法搜索过程中找到的非占优粒子. 对目标函数进行评估后, 算法检查每个粒子与其他种群成员的模糊占优关系, elite 文档存储算法找到的  $N$  个最优模糊占优解. 联立当前代中的解与前一代文档中的解组成  $2N$  个解, ( $N$  为文档大小). 根据模糊占优关系将  $2N$  个解作升序排列. 当多个解同时拥有相同的模糊占优值时, 则将周长  $I(\cdot)$  分配至多个解, 且  $I(\cdot)$  值越高, 该解越视为更优解. 根据以上  $2N$  个解的模糊占优排序和周长, 选择最优的  $N$  个解对 archive 文档进行更新.

### 5.2 分配周长

如果多个解同时具有相同的模糊占优值, 此时引入解的周长概念进行区分. 对于任意解  $v$  的周长  $I(v)$  可表示为:

$$I(v) = \sum_{i=1}^M \frac{f_i(u) - f_i(w)}{\max(f_i) - \min(f_i)} \quad (25)$$

式中,  $u$  和  $w$  分别表示联合解根据第  $i$  个目标升序排列后解  $v$  的邻近解, 且为处于边界的解分配无限值. 解的周长  $I(v)$  值越高, 算法越偏好选择该解, 原因在于此时沿该解  $v$  的区域越稀少, 选择该解可进一步保持解的多样性. 最后, 将所有  $N$  个解根据  $M$  个

目标函数进行排序.

### 5.3 更新粒子 $Pbest$ 和 $Gbest$

对于  $Gbest$ , 其解的选择方式为: 在当前 elite 文档解中, 根据解之间的非模糊占优关系及二进制竞赛选择的周长进行排序. 而对于  $Pbest$ , 其解选择为将粒子当前位置与前一代中粒子的最佳位置进行比较后择优选择, 即: 选择非占优解为  $Pbest$ . 如果解之间是相互非占优关系, 则选择粒子当前位置为  $Pbest$ .

### 5.4 粒子的自适应变异

多目标离散 FDS-PSO 算法在搜索过程中为了避免陷入局部最优, 引用变异算子提高空间搜索效率. FDS-PSO 算法的变异概率  $P(\text{Mutation})$  为:

$$P(\text{Mutation}) = 1 - t / \max\_t \quad (26)$$

式中,  $\max\_t$  为最大迭代次数;  $t$  为当前迭代次数. 算法执行过程中为每个粒子配置一个随机数  $m\_rand$ , 且  $m\_rand \in [0, 1]$ . 如果该随机数  $m\_rand < P(\text{Mutation})$ , 则从种群粒子中随机选择一个粒子作变异操作.

## 6 仿真实验

### 6.1 仿真模型

通过仿真环境 CloudSim 模拟工作流的依赖约束任务关系, workflow 结构利用随机生成的任务 DAG 模型. 实验中通过设置不同数量的任务改变 DAG 任务的规模.

资源包括八个虚拟节点, 每个虚拟节点包括八个处理器的异质分布式计算系统. 资源间的链接通过路由器连接, 因此资源间可发生直接通信. 不同资源的处理器速率随机产生于区间  $[1000\text{MIPS}, 10000\text{MIPS}]$ , 使得最快资源均为最慢资源处理速率的 10 倍, 相应执行时间则减少为 1/10. 单个资源的失效率范围为  $[10^{-5}, 10^{-7}] \text{failures/sec}$ .

为了产生满足截止时间与预算约束的有效调度解, 截止时间  $D$  和预算  $B$  的产生方式为:

$$D = \text{Time}_{\max} - 0.2 \times (\text{Time}_{\max} - \text{Time}_{\min}) \quad (27)$$

$$B = \text{Cost}_{\max} - 0.2 \times (\text{Cost}_{\max} - \text{Cost}_{\min}) \quad (28)$$

式中,  $\text{Time}_{\min}$  为最早完成时间算法 HEFT 得到的最小完成时间 makespan, 其对应的为最大执行代价  $\text{Cost}_{\max}$ .  $\text{Time}_{\max}$  为贪婪代价算法得到最大执行时间, 其对应的为最小执行代价  $\text{Cost}_{\min}$ .

选择同类型的非占优排序遗传算法 NSGA-II<sup>[5]</sup> 和多目标粒子群算法 MOPSO<sup>[11]</sup> 进行性能对比. 相关参数设置为: 种群大小为任务数的三倍,  $c_1 = 2.5 \rightarrow 0.5$ ,  $c_2 = 0.5 \rightarrow 2.5$ ,  $w = 0.9 \rightarrow 0.1$ , GA 中的交叉概率

= 0.8,变异概率 = 0.5.

6.2 实验结果

以随机产生的任务 DAG 结构进行测试,任务规模分别设置为 40、60 和 100,分别对应于小、中和大规模并行应用.资源间的通信代价随机产生于均值为平均通信代价与通信计算比 CCR 的均匀分布中,实验设置 CCR=0.5,表示计算密集型任务应用.任务  $t_i$  在资源  $r_j$  上的计算代价随机产生于均值为平均计算代价的两倍的均匀分布中.

对于双目标优化,选择执行时间 makespan 和执行代价为优化目标.图 2 是针对双目标优化算法得到 Pareto 最优解的分布.可以看出,FDS-PSO 算法得到的大部分解均处于更优边界上,同时解之间维持着更加均匀的分布,主要原因是 FDS-PSO 算法采用的模糊占优排序可以选择更接近于非占优边界的解.相比较而言,NSGA-II 和 MOPSO 算法得到的解的分布更加分散,在两个优化目标上的结果也不如 FDS-PSO 算法.

表 1 双目标优化性能指标

| 任务数量 |         | FDS-PSO 算法 |         | MOPSO 算法 |         | NSGA-II 算法 |         |
|------|---------|------------|---------|----------|---------|------------|---------|
|      |         | 均值         | 标准差     | 均值       | 标准差     | 均值         | 标准差     |
| 40   | GD      | 0.009 3    | 0.000 2 | 0.012 2  | 0.004 4 | 0.012 3    | 0.001 1 |
|      | Spacing | 0.038 5    | 0.004 2 | 0.045 6  | 0.006 5 | 0.045 3    | 0.006 1 |
|      | Time    | 0.184 6    | 0.02    | 0.189 3  | 0.03    | 0.215 5    | 0.03    |
| 80   | GD      | 0.008 3    | 0.000 2 | 0.014 4  | 0.000 3 | 0.013 3    | 0.000 3 |
|      | Spacing | 0.026 8    | 0.003 5 | 0.045 5  | 0.006 7 | 0.038 7    | 0.005 2 |
|      | Time    | 0.386 1    | 0.03    | 0.397 4  | 0.04    | 0.458 1    | 0.03    |
| 100  | GD      | 0.005 5    | 0.000 1 | 0.016 3  | 0.000 2 | 0.015 7    | 0.000 3 |
|      | Spacing | 0.025 6    | 0.002 6 | 0.038 8  | 0.005 1 | 0.032 3    | 0.004 8 |
|      | Time    | 0.733 4    | 0.03    | 0.737 3  | 0.04    | 0.832 2    | 0.04    |

而对于解的间隔分布,FDS-PSO 算法的间隔更小,性能更优,这是由周长分配和占优得到的结果.同时,FDS-PSO 算法还可以以相对更小的计算时间得到更好的收敛性和均匀间隔.

对于执行时间、执行代价和可靠性的三目标优化,算法的 Pareto 解如图 3 所示.结果表明,FDS-PSO 算法比较 MOPSO 和 NSGA-II 的大部分解均更好的处于最小化区域中,且拥有更好的均匀分布间隔.同时,表 2 的结果也证明 FDS-PSO 算法可以以更小的计算时间获得收敛,并得到更均匀的间隔分布.

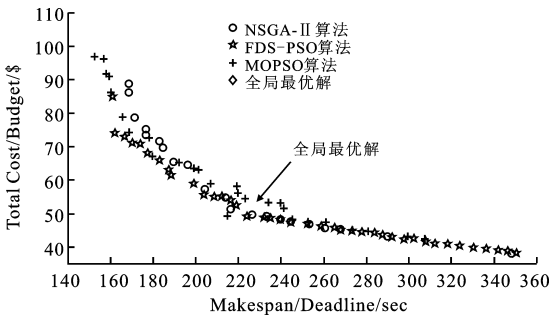


图 2 双目标优化

表 1 是算法三种指标的变化.从结果可以看出,FDS-PSO 算法的收敛指标 GD 的值小于另外两种算法,这导致其解更优,得到的解集合更接近于实际的 Pareto 边界.同时,随着任务规模的增大,算法性能将随着更大数量的非占优解的出现也随之提高.通过模糊占优排序,更接近于非占优边界的解将被选择,求解时得到更好的收敛性.

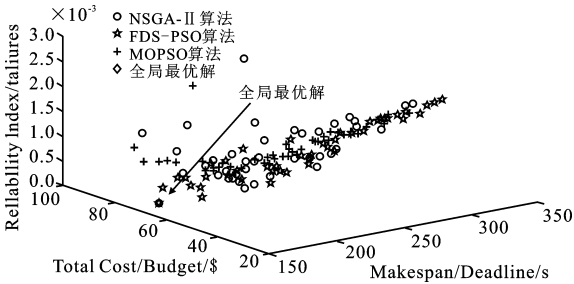


图 3 三目标优化

表 2 三目标优化性能指标

| 任务数量 |         | FDS-PSO 算法 |         | MOPSO 算法 |         | NSGA-II 算法 |         |
|------|---------|------------|---------|----------|---------|------------|---------|
|      |         | 均值         | 标准差     | 均值       | 标准差     | 均值         | 标准差     |
| 40   | GD      | 0.006 5    | 0.000 1 | 0.036 5  | 0.003 1 | 0.028 8    | 0.002 3 |
|      | Spacing | 0.074 1    | 0.003 5 | 0.078 1  | 0.004 0 | 0.088 1    | 0.003 5 |
|      | Time    | 0.338 8    | 0.03    | 0.346 6  | 0.04    | 0.403 4    | 0.03    |
| 60   | GD      | 0.004 5    | 0.000 1 | 0.014 3  | 0.000 1 | 0.011 1    | 0.000 1 |
|      | Spacing | 0.033 2    | 0.005 2 | 0.083 4  | 0.006 5 | 0.052 3    | 0.006 1 |
|      | Time    | 0.745 3    | 0.03    | 0.761 1  | 0.04    | 0.878 7    | 0.04    |
| 100  | GD      | 0.002 7    | 0.000 1 | 0.006 5  | 0.001 9 | 0.006 1    | 0.001 2 |
|      | Spacing | 0.035 4    | 0.003 1 | 0.052 2  | 0.003 4 | 0.040 2    | 0.004 3 |
|      | Time    | 1.137 8    | 0.04    | 1.134 5  | 0.05    | 1.327 6    | 0.04    |

7 结束语

为了解决截止期限与预算双 QoS 约束条件下 DAG 云任务的执行时间、执行代价和调度可靠性的多目标同步优化问题,设计了一种基于模糊占优排序的离散粒子群优化算法. 算法利用改进的粒子群进化机制及解之间的模糊占优关系可以求解满足帕累托最优的调度解集合. 通过仿真实验,证明算法在求解 DAG 云任务多目标调度优化问题时,收敛性更好,解分布更加均匀,计算效率更高,其性能是优于同类型智能算法的.

参考文献:

[1] Smachet S, Viriyapant K. Taxonomies of workflow scheduling problem and techniques in the cloud[J]. Future Generation Computer Systems, 2015, 52(3):1-12.

[2] Topcuoglu H, Hariri S, Wu M Y. Performance-effective and low-complexitytask scheduling for heterogeneous computing[J]. IEEE Transactions Parallel Distributed Systems, 2012,13(3):260-274.

[3] Parikh S, Sinha R. Double level priority based optimization algorithm for task scheduling in cloud computing[J]. Journal of Computer Applications, 2013, 62(20):33-37.

[4] 李静梅, 尤晓非, 韩启龙. 基于任务复制的多关键路径任务调度算法[J]. 计算机工程与设计, 2014, 35(5):1639-1645.

[5] Garg R, Kumar Singh A. Multi objective optimization to workflow grid scheduling using reference point based evolutionary algorithm[J]. Journal of Computer Applications, 2012, 22(6):44-49.

[6] Fard H, Prodan R, Barrionuevo J, et al. A multi-ob-

jective approach for workflow scheduling in heterogeneous environments[C]// Ieee/acm International Symposium on Cluster, Cloud and Grid Computing. Ottawa, Canada, ACM, 2012:300-309.

[7] Su S, Li J, Huang Q, et al. Cost-efficient task scheduling for executing large programs in the cloud[J]. Parallel Computing, 2013, 39(4/5):177-188.

[8] Durillo J, NaeV, Prodan R. Multi-objective energy-efficient workflow scheduling using list-based heuristics [J]. Future Generation Computer Systems, 2014, 36(3):221-236.

[9] Durillo J, Prodan R, Barbosa G. Pareto tradeoff scheduling of workflows on federated commercial Clouds[J]. Simulation Modelling Practice & Theory, 2015, 58(1):95-111.

[10] Zheng W, Sakellariou R. Budget-deadline constrained workflow planning for admission control[J]. Journal of Grid Computing, 2013, 11(4):633-651.

[11] Beegom A, Rajasree M. A particle swarm optimization based pareto optimal task scheduling in cloud computing[C] // International Conference in Swarm Intelligence. Hefei, China, Springer International Publishing, 2014:79-86.

[12] Verma A, Kaushal S. Cost-time efficient scheduling plan for executing workflows in the cloud[J]. Journal of Grid Computing, 2015, 13(4):1-12.

作者简介:

郑宇超 男, (1986-), 硕士, 讲师. 研究方向为云计算及应用. E-mail: yucaoz@sina.com.

夏学文 男, (1974-), 博士, 副教授. 研究方向为计算智能及应用.

艾冬梅 女, (1970-), 博士, 高级工程师. 研究方向为计算生物和生物信息学、人工智能、数学建模.