

基于自适应惩罚函数的云 workflow 调度协同进化遗传算法

徐健锐^{1,2} 朱会娟³

(江苏大学计算机科学与通信工程学院 江苏 镇江 212013)¹

(江苏联合职业技术学院镇江分院 江苏 镇江 212016)²

(中国科学院大学计算机与控制学院 北京 100049)³

摘要 云计算为大规模科学 workflow 应用的执行提供了更高效的运行环境。为了解决云环境中科学 workflow 调度的代价优化问题,提出了一种基于协同进化的 workflow 调度遗传算法 CGAA。该算法将自适应惩罚函数引入严格约束的遗传算法中,通过协同进化的方法,自适应地调整种群个体的交叉与变异概率,以加速算法收敛并防止种群早熟。通过 4 种科学 workflow 的仿真实验结果表明,CGAA 算法得到的调度方案在满足 workflow 调度截止时间约束与降低任务执行代价的综合性能方面优于同类型算法。

关键词 云计算,科学 workflow,任务调度,协同进化,遗传算法

中图分类号 TP393 文献标识码 A DOI 10.11896/j.issn.1002-137X.2018.08.019

Coevolutionary Genetic Algorithm of Cloud Workflow Scheduling Based on Adaptive Penalty Function

XU Jian-rui^{1,2} ZHU Hui-juan³

(School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu 212013, China)¹

(Zhenjiang Branch, Jiangsu Union Technical Institute, Zhenjiang, Jiangsu 212016, China)²

(School of Computer and Control Engineering, University of Chinese Academy of Sciences, Beijing 100049, China)³

Abstract The cloud computing provides a more efficient operation environment for the execution of large-scale scientific workflow application. To solve the cost optimization problem of the scientific workflow scheduling in the cloud environment, a workflow scheduling genetic algorithm based on coevolution was proposed. This algorithm introduces an adaptive penalty function into GA with the strict constraints. By the coevolutionary approach, it can adjust the crossover and mutation probability of population individuals adaptively to accelerate the convergence of the algorithm and prevent the prematurity of population. The simulation experiment results of four kinds of scientific workflow in reality show that the scheduling scheme obtained by the CGAA algorithm performs better in satisfying the comprehensive performance of the workflow scheduling deadline constraints and reducing the total execution cost of tasks compared with the same types of algorithms.

Keywords Cloud computing, Scientific workflow, Tasks scheduling, Coevolution, Genetic algorithm

1 引言

科学实验通常体现为 workflow 形式,其任务根据数据流与计算相关性形成链式结构,如常见的计算密集型和数据密集型 workflow 应用^[1];其特点是数据量和计算需求量巨大,需要更高性能的计算环境来支持运行。作为分布式计算、网格计算和并行计算的最新发展,云计算能够以效用形式提供动态且可扩展伸缩的计算资源,如日常生活中水、电、气的使用方式。

云计算最主要的特征是虚拟化^[2],云以虚拟机 VMs 的形式提供计算资源,而 workflow 中任务与计算资源间的映射问题即为 workflow 调度问题^[3]。云 workflow 调度包括两个层次:1)任

务与 VM 间的映射;2)在单个 VM 上任务的顺序执行^[4]。云环境中的 workflow 调度优化目标通常包括执行跨度 makespan、执行代价、吞吐量及负载均衡优化等。本文将重点关注截止时间约束下 workflow 调度执行代价的优化问题,利用协同进化遗传方法寻找最优的任务-虚拟机 VM 映射方案,使得 workflow 执行跨度在满足截止时间的同时,最小化总执行代价。

2 相关研究

在相关工作中,文献[4]提出一种基于 workflow 调度的约束条件下代价最优化 PSO 算法,该算法虽然充分考虑了云资源弹性与资源异构的特征,利用惩罚函数评估了可行粒子的优

到稿日期:2017-07-04 返修日期:2017-10-26 本文受国家自然科学基金项目(61302124),江苏省高校自然科学研究面上项目(16KJB520010)资助。

徐健锐(1975—),男,硕士,副教授,主要研究方向为复杂网络预测与评价、云计算与大数据分析、教育信息化及职业教育人才培养模式,E-mail: zjxjr@126.com(通信作者);朱会娟(1984—),女,博士生,主要研究方向为数据清洗、数据融合、数据分析。

劣问题,但仍未解决 PSO 中普遍的早熟问题。文献[5]提出一种前向 GA 算法,通过可靠性驱动的信誉值评估资源可靠性,以实现执行跨度与 workflow 执行可靠性的多目标优化。文献[6]提出退火与遗传算法相结合的任务调度多目标算法,利用任务的影响度进行种群初始化,并引入失败率因子,提高了调度可靠性;但该算法仅应用了静态适应度函数,且仅与传统 GA 进行比较,不具备说服力。文献[7]利用 GA 算法将约束条件引入目标适应度函数中,将约束最优化问题转换为非约束问题;但该算法在全局最优搜索与约束满意度方面无法设置合适与均衡的惩罚因子。文献[8]提出一种改进 GA 算法,其染色体不仅表示资源与任务分配方案,而且代表了虚拟机上任务的执行序列,当个体违背约束条件时,算法根据最优化目标进化个体,并基于约束目标进化种群,以此解决合适惩罚函数的设计问题;但该算法的进化代数过多,可能无法找到可行解。文献[9]利用关键路径的粒子初始化和搜索阶段的筛选机制,提出基于粒子群的最优化调度方案搜索方法,该方法可以有效降低调度费用开销,但忽略了云资源的动态提供特征。

在已有研究中,面对严格约束最优化问题,大多数进化算法通常使用静态惩罚函数,通过正比于约束违例度的方式来降低个体适应度,以惩罚不可行个体方案。然而,设置准确且适合的惩罚因子是比较困难的。另一种方法是在个体进化过程中移除不可行个体,但是,由于不可行个体的数量通常较少,容易使优良基因个体无法遗传至下一代。

鉴于以上问题,针对云资源动态提供与计算资源异构的特征,提出一种基于自适应惩罚函数的协同进化遗传算法,实现带有严格约束条件的云 workflow 调度优化算法。该算法将自适应惩罚函数应用于传统 GA 中,并在进化过程中对个体进行自适应调整,实现交叉与变异概率的调整。本文具体工作如下:1)考虑云资源动态提供与异构性的特征,设计了一种截止时间约束的云 workflow 调度代价最优化模型;2)定义了一种带有自适应惩罚的协同进化函数,可以有效防止种群早熟,并通过协同进化调度调整交叉与变异概率;3)不同于传统 GA,所提算法基于关键路径进行种群初始化,可有效防止早熟,提高 workflow 任务调度的截止时间约束满足率。

3 workflow 任务调度问题的描述

将 workflow 模型定义为有向无循环图 DAG(Direct Acyclic Graph),描述为 $G=(V, E)$ 。其中, $V=\{t_1, t_2, \dots, t_n\}$ 表示图的顶点,即 workflow 任务 t ; E 表示图的边,即任务间的执行依赖关系; n 表示 workflow 的任务总数。图的边表明任务间的执行顺序约束, t_x 至 t_y 间的有向边 $e_{x,y}$ ($x, y \in V$) 表明只有任务 t_x 完成后 t_y 才可以开始执行。此时,任务 t_x 称为任务 t_y 的父任务(或前驱任务),任务 t_y 称为任务 t_x 的子任务(或后继任务)。workflow DAG 中,不存在父任务的任务称为入口任务 t_{entry} ,而不存在子任务的任务称为出口任务 t_{exit} 。 d_w 表示用户定义的 workflow 执行的截止时间约束(Deadline),该约束决定了完成所有 workflow 任务的最长时间限制。图 1 给出了一种典型的工作流结构示例。

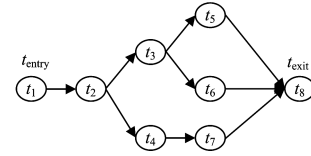


图 1 工作流实例

Fig. 1 Workflow instance

云环境中, $VM = \{VM_1, VM_2, \dots, VM_n\}$ 表示 IaaS 云提供的虚拟机服务,不同类型的 VM 提供不同的计算资源。根据虚拟机处理能力 P_{VM_i} 与单位时间代价 C_{VM_i} 定义 VM 类型, VMs 按单位时间 τ 收取费用,若 $\tau=60$ min,则当使用 61 min VM 时均按 2 小时(单位时间)支付费用;同时,假设每个 workflow 任务只在单个 VM 上执行,单个 VM 可执行多个 workflow 任务。

任务 t_i 在 VM_{t_i} 上的执行时间 $RT_{t_i}^{VM_{t_i}}$ 如式(1)所示,其中 s_{t_i} 表示任务 t_i 的大小, $P_{VM_{t_i}}$ 表示 VM_{t_i} 的处理能力。父任务 t_i 与其子任务 t_j 间的传输时间 $TT_{e_{i,j}}$ 如式(2)所示,其中 $d_{t_i}^{\text{out}}$ 表示 t_i 产生的输出数据大小, β 表示 VMs 间的带宽。若两个任务在相同的 VM 上执行,则传输时间为 0。

$$RT_{t_i}^{VM_{t_i}} = s_{t_i} / P_{VM_{t_i}} \quad (1)$$

$$TT_{e_{i,j}} = d_{t_i}^{\text{out}} / \beta \quad (2)$$

本文的目标是寻找 workflow 调度的最优方案,使得在满足截止时间约束的同时,任务的总执行代价达到最小。定义调度向量 $S = \{M, TEC, TET\}$,其中, M 表示任务与资源间的映射关系, TEC 表示 workflow 的总执行代价, TET 表示总执行时间。 M 由所有任务的 VMs 类型、开始时间和结束时间组成, $M = (m_{t_1}^{VM_{t_1}}, m_{t_2}^{VM_{t_2}}, \dots, m_{t_M}^{VM_{t_M}})$, $m_{t_i}^{VM_{t_i}} = (t_i, VM_{t_i}, ST_{t_i}, ET_{t_i})$ 表示任务 t_i 分配至 VM_{t_i} ,任务的开始时间与结束时间分别表示为:

$$ST_{t_i} = \begin{cases} LET_{VM_{t_i}}, & t_i = t_{\text{entry}} \\ \max(\max_{t_a \in \text{Parent}(t_i)} (ET_{t_a} + TT_{e_{a,i}}), LET_{VM_{t_i}}), & \\ \text{otherwise} \end{cases} \quad (3)$$

$$ET_{t_i} = ST_{t_i} + RT_{t_i}^{VM_{t_i}} \quad (4)$$

其中, $LET_{VM_{t_i}}$ 表示 VM_{t_i} 的租用结束时间,总执行代价 TEC 与总执行时间分别表示为:

$$TEC = \sum_{i=1}^{|VM|} C_{VM_{t_i}} \times \lceil \frac{RT_{t_i}^{VM_{t_i}}}{\tau} \rceil + \sum_{i,j \in T} TT_{e_{i,j}} \times TC_{VM_{t_i}} \quad (5)$$

$$TET = \max(ET_{t_i}, t_i \in V) \quad (6)$$

其中, $C_{VM_{t_i}}$ 表示 VM_{t_i} 的处理代价, $TC_{VM_{t_i}}$ 表示 VM_{t_i} 的数据传输代价。

最优化调度目标即寻找方案 S ,使得 TET 在不超过 workflow 截止时间 d_w 的情况下, TEC 达到最小化,形式化描述为:

$$\text{Min } TEC \quad (7)$$

$$\text{s. t. } TET \leq d_w$$

本文提出一种带自适应惩罚函数的协同进化遗传算法(Coevolutionary Genetic Algorithm Based on Self-Adaption, CGAA)对以上问题进行求解。表 1 列出了文中相关参数符号及其含义说明。

表 1 符号说明
Table 1 Symbol description

符号	参数含义
VM_i	虚拟机 i
P_{VM_i}	VM_i 的处理能力
C_{VM_i}	VM_i 的单位时间代价
$RT_{t_i}^{VM_i}$	任务 t_i 在 VM_i 的运行时间
s_{t_i}	任务 t_i 的大小
$TT_{e_i,j}$	父任务 t_i 与子任务 t_j 间的传输时间
$d_{t_i}^{out}$	任务 t_i 的输出数据大小
β	VM 间的带宽
TEC	工作流总执行代价
TET	工作流总执行时间
d_w	工作流执行的截止时间约束
ST_{t_i}	任务 t_i 的开始时间
ET_{t_i}	任务 t_i 的结束时间
LET_{VM_i}	VM_i 的租用结束时间(VM_i 的空闲时间)

4 协同进化遗传算法

4.1 算法机理

协同进化是指两个以上种群相互适应的过程,其算法的关键是一个种群的进化依赖于其他种群。本文将使用协同进化调整交叉与变异概率,在传统 GA 的基础上利用自适应惩罚函数对个体进行更新。CGAA 算法的进化结构如图 2 所示,其利用两种类型的种群,一类为单种群(标记为 $Population_2$),其大小为 M_2 ,用来自适应调整交叉与变异概率;另一类为多种群(标记为 $Population_{1,1}, Population_{1,2}, \dots, Population_{1,M_2}$),其大小为 M_1 ,以不同的交叉与变异概率进行进化,以寻找较好的决策解。 $Population_2$ 中的每个个体 B_j 代表一个交叉与变异概率因子集合,而 $Population_{1,j}$ 中的个体则对应一个决策解。

在每一代协同进化过程中,每个 $Population_{1,j}$ 通过 GA 算法以 $Population_2$ 中个体 B_j 获得的交叉与变异概率确定进化代数 G_1 ;然后,得到 $Population_2$ 中每个个体 B_j 的适应度。评估 $Population_2$ 中的所有个体后, $Population_2$ 通过 GA 算法进行进化。重复执行该协同进化过程,直至满足预先设定的结束条件即可(如达到协同进化最大代数 G_2)。

简言之,两类种群相互进化,带有自适应惩罚函数的 $Population_{1,j}$ 用来进化决策解,而 $Population_2$ 则用于对决策解的交叉与变异概率进行自适应修正。由于协同进化,不仅决策解可以通过进化获得,而且交叉与变异概率可以通过自适应方式调整,从而避免人工设置的反复尝试。

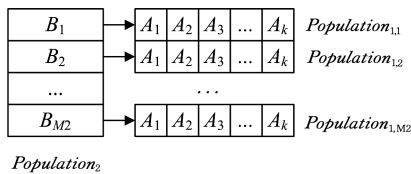


图 2 协同进化模型

Fig. 2 Coevolution model

4.2 自适应交叉与变异

传统 GA 中,交叉概率 p_c 与变异概率 p_m 越大,种群的多样性个体越多。然而,如果两种概率过大,将很容易损失较好

的基因个体;反之,若两种概率过小,将不利于产生新个体,且会降低算法的搜索速度。

通常,GA 的进化过程包括 4 种状态:初始状态、亚成熟状态、正成熟状态及成熟状态。概率 p_c 与概率 p_m 可以根据种群所处状态的不同进行调整,例如:当种群中拥有许多低适应度的劣质个体时,算法应在初始状态增加 p_c 与 p_m ;如果种群的适应度跨度过大,算法应在亚成熟与正成熟状态增加交叉概率 p_c ;如果种群拥有的个体均是良好适应度的完美个体,则应在成熟状态降低 p_c 与 p_m 。借助文献[10]的思想,本文设计了一种自适应交叉与变异概率调整机制:

$$p_c(i) = \omega_1 \times \cos\left(\frac{\pi}{2} \times \frac{1}{e^{(\sigma_1(i) + \sigma_2(i) + \dots + \sigma_m(i))}}\right) \quad (8)$$

$$p_m(i) = \omega_2 \times \frac{f_i}{f_p} \quad (9)$$

其中, $p_c(i)$ 表示个体 i 的交叉概率, $p_m(i)$ 表示变异概率, $\sigma_m(i) = |f_i - f_i(m)| / f_i$, $f_i(m)$ 表示离个体 i 第 m 近的个体适应度, f_i 表示个体 i 的适应度, f_p 表示种群中适应度最大的个体, ω_1 和 ω_2 表示交叉与变异概率因子,其值通过协同进化进行更新。

4.3 自适应惩罚适应度函数

定义惩罚适应度函数为:

$$F_i^p = F^a(x_i) = F(x_i) - P(x_i) = F(x_i) + \sum_{j=1}^m \lambda(j) \times E_j(x_i) \quad (10)$$

其中, F_i^p 表示惩罚后个体 i 的适应度函数, $F(x_i)$ 表示最优优化目标个体的适应度, $\lambda(j)$ 表示针对约束违例 j 的惩罚因子, m 表示约束数量, $E_j(x_i)$ 表示个体 i 的第 j 个约束违例。

惩罚函数应适用于所有不可行个体。如果惩罚过小,不可行个体可能拥有较高的惩罚适应度,种群将会沿着错误的方向进化至不可行区域;如果惩罚过大,拥有较好基因的个体将被移除,导致早熟式收敛。基于文献[11]的思想,设计一种自适应惩罚函数来追踪种群中可行个体的数量,以决定惩罚值。

首先,每个个体的适应度和约束违例由式(11)和式(12)进行标准化处理:

$$F'(x_i) = \frac{F(x_i) - F_{\min}}{F_{\max} - F_{\min}} \quad (11)$$

其中, $F'(x_i)$ 表示标准适应度, F_{\min} 与 F_{\max} 分别表示当前种群中所有个体适应度的最大值与最小值。通过该方式,每个个体的适应度将处于 $[0, 1]$ 之间,且每个不可行个体的标准约束违例 $E'(x_i)$ 可表示为:

$$E'(x_i) = \frac{1}{m} \sum_{j=1}^m \frac{E_j(x_i)}{E_j^{\max}} \quad (12)$$

其中, m 表示约束数量, E_j^{\max} 表示所有不可行个体中第 j 个约束的最大违例值。

进一步地,对于 GA 中不同的进化状态,惩罚规则将进行自适应调整。例如:如果种群中拥有较少的可行个体,则拥有较低约束违例的不可行个体将被惩罚较小值;另一方面,如果种群拥有较多可行个体,则拥有较低标准适应度的不可行个体将被惩罚较小值^[12]。适应度通过以下方式定义:

1) 如果当前种群中拥有至少 1 个可行个体, 则适应度为:

$$F1^a(x_i) = \begin{cases} F'(x_i), & i \text{ 为可行个体} \\ \sqrt{F'(x_i)^2 + E'(x_i)^2} + [(1-r_f)E'(x_i) + r_f F'(x_i)], & \text{否则} \end{cases} \quad (13)$$

其中, r_f 表示可行个体占种群数量的比例。通过这种方式, 拥有较低适应度与较低约束违例的个体将有可能被视为更优个体。而且如果种群中的可行概率 r_f 较小, 则越接近于可行空间的个体将越被视为较优个体, 否则拥有更低标准适应度的个体为较优个体。

2) 如果当前种群中没有可行个体, 则适应度为:

$$F1^a(x_i) = E'(x_i) \quad (14)$$

明显地, 拥有更小约束违例的个体将被优先考虑, 搜索将进入至约束违例之和较小的区域(可行区域边界)。

在 CGAA 算法中, $Population_{1,j}$ 的个体 i 的适应度可使用式(13)与式(14)进行计算。 $Population_2$ 的每个个体表示一个概率因子(w_1 和 w_2), 在 $Population_{1,j}$ 进化 G_1 代之后, $Population_2$ 中的个体 B_j 通过式(15)进化:

$$F2(B_j) = -\min(F1_j) + \frac{N_{infeasible}}{M_1} \quad (15)$$

其中, $F1_j$ 表示 $Population_{1,j}$ 中所有个体的适应度, $N_{infeasible}$ 表示 $Population_{1,j}$ 中不可行个体的数量, M_1 表示种群 $Population_{1,j}$ 的大小。

5 CGAA 算法的设计

5.1 算法模型

本文使用两类染色体对云计算环境中的 workflow 调度问题建立算法模型, 如图 3 所示。 $Population_{1,j}$ 中的 $chromosome_{i,j}$ 表示决策解, 即 workflow 中任务与资源间的映射方案。此时, 在 $chromosome_{i,j}$ 中, 每个基因的位置表示任务数量; 在 $chromosome_{i,j}$ 中, 每个基因的值表示 VM 的数量。因此, $chromosome_{i,j}$ 的维度表示 workflow 中任务的数量。在 $chromosome_{i,j}$ 中, 基因的范围由运行任务的可用资源量决定。图 3 给出了拥有 8 个任务与 5 个可用 VMs 的 workflow 映射模型。适应度函数用来决定决策解的优劣, 即最终计算得到的总执行代价 TEC 和总执行时间 TET 的约束最优化目标。 TEC 和 TET 的计算过程如算法 1 所示。

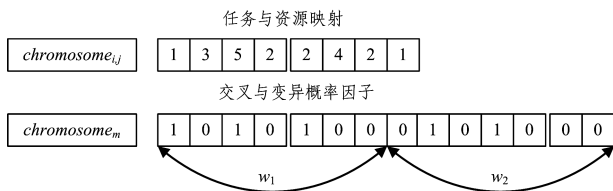


图 3 协同进化的染色体编码

Fig. 3 Chromosome coding of coevolution

算法 1 TEC 与 TET 的计算过程

Input: a set of workflow tasks T , a set of VMs \mathbf{VM} , a chromosome $_{k,j}$ in population $_{1,j}$

Output: TEC and TET

1. initialize VMs state matrix VS and task state matrix TS
2. calculate execution time $RT[|T| \times |\mathbf{VM}|]$ and transfer time $TT[|T| \times |T|]$
3. for $i=1$ to $|T|$
4. if $TS(T(i))$ is unscheduled
5. $t_i = T(i)$, $VM_{t_i} = vm_{chromosome_{k,j}(i)}$
6. if t_i has no parents
7. $ST_{t_i} = LET_{VM_{t_i}}$
8. else
9. $ST_{t_i} = \max(\max_{t_a \in parent(t_i)} (ET_{t_a} + TT_{e_{a,i}}), LET_{VM_{t_i}})$
10. for each child task t_c of t_i
11. if t_c is mapped to a VM different to VM_{t_i}
12. $TT(i) = TT(i) + TT(i, c)$
13. $RT_{t_i}^{VM_{t_i}} = RT(t_i, VM_{t_i})$
14. $ET_{t_i} = RT_{t_i}^{VM_{t_i}} + TT(i)$
15. update VS and TS, set the time period $[ST_{t_i}, ET_{t_i}]$ for VM_{t_i} is busy, set $TS(T(i))$ as scheduled
16. calculate TEC according to Equation (5)
17. calculate TET according to Equation (6)
18. return TEC and TET

$Population_2$ 中的 $chromosome_m$ 表示交叉与变异概率因子, 通过二进制编码定义。 w_1 的范围为 $(0, 1]$, 利用最前面的 7 个基因表示因子 w_1 , 则 $chromosome_m$ 中 w_1 的值为: $w_1 = 2^6 + 2^4 + 2^2 + 1/128 = 0.66$ 。 w_2 的范围为 $(0, 1]$, 利用后面 7 个基因表示因子 w_2 , $w_2 = 2^5 + 2^3 + 1/128 = 0.32$ 。

$Population_{1,j}$ 即为满足总执行时间 TET 约束下最小化总执行代价 TEC 时任务与资源匹配的进化决策解, $Population_2$ 则对进化解自适应调整交叉与变异概率。

5.2 TEC 和 TET 的计算

为了求解 workflow 调度最优化问题, 需要计算特定任务-虚拟机映射调度中 workflow 的运行时间和相应代价。算法 1 给出了计算 $Population_{1,j}$ 中 $chromosome_{i,j}$ 的总执行代价 TEC 和总执行时间 TET 的过程。 $chromosome_{i,j}(k)$ 的第 k 个位置的值表示任务 k 调度至 $VM_{chromosome_{i,j}(k)}$, $Population_{1,j}$ 中一个染色体即为一种任务-资源映射方案。

算法说明: 首先, 初始化 VMs 状态向量 VS 与任务状态向量 TS, 输入 workflow 任务集合 T 和 VMs 集合 \mathbf{VM} 。然后, 根据式(1)计算每个 workflow 任务 t_i ($t_i \in T$) 在每种 VM VM_i ($VM_i \in \mathbf{VM}$) 上的执行时间 $RT_{t_i}^{VM_i}$, 根据式(2)计算任务间的传输时间 $TT_{e_{i,j}}$, 即算法 1 中的步骤 2。

任务的起始时间 ST_{t_i} 有两种情况: 如果任务没有父任务, 只要分配至该任务的 VM 为空闲状态即可开始执行, 即算法 1 中的步骤 6—步骤 7; 否则, 只有等待其父任务完成并发送输出数据后才可开始执行, 即算法 1 中的步骤 8—步骤 9。如果该 VM 处于执行状态, 则需等待该 VM 空闲时才可开始执行。在算法 1 中, 如果分配至相同 VM 上的两个任务拥有相同的开始时间, 则 VM 将优先执行更小的任务。结束时间 ET_{t_i} 根据开始时间和执行时间 $RT_{t_i}^{VM_{t_i}}$ 由式(4)计算得到, 即算法 1 中的步骤 10—步骤 14。任务调度后, 需要更新 VS 和

TS , 设置任务 t_i 为调度状态并在 ST_{t_i} 至 ET_{t_i} 之间将 VM_{t_i} 设置为忙状态,即算法 1 中的步骤 15。最后,通过算法 1 中的步骤 16—步骤 17 计算 TEC 和 TET 。该过程执行至所有任务被调度后结束。

5.3 种群初始化

对于科学 workflow 而言,协同进化中的任务执行时间对 workflow 的总执行时间拥有更大的影响权重,同时,这些任务的执行代价又占总执行代价更小的比重。因此,将这些任务分配至高性能的 VM 上将极大地减少总执行时间,同时对总执行代价有更小的影响。

初始种群的多样性对 GA 的性能有着极大的影响,但是大多数的 GA 通过随机方式产生初始种群。为了改善解的质量和收敛速度,本文基于协同进化方法产生五分之一的初始种群,并将这些任务分配至具有更强处理能力的 VMs 上;初始种群的另外五分之一则分配至成本最低的 VMs 上,其他种群则以随机方式产生。

图 4 给出了 CGAA 算法的总体流程。首先,初始化得到两类种群, $Population_2$ 用于调整交叉与变异概率,使得 $Population_1$ 可以找到决策解。本文中, $Population_2$ 的进化是非约束最优化问题,无须设置惩罚函数。 $Population_1$ 利用自适应惩罚函数将约束化 workflow 调度问题转换为非约束最优化问题。 $Population_1$ 利用前文中的机制进行 m 种群初始化, $Population_2$ 则利用随机化方式产生。 $Population_1$ 中的每个子种群 $Population_{1,j}$ 同步进化 G_1 代数,来自 M_2 个子种群中的最优 M_2 个个体用来评估 $Population_2$ 中的相应个体。 $Population_2$ 进化 G_2 代数,以寻找最优交叉与变异概率以及最优决策解。

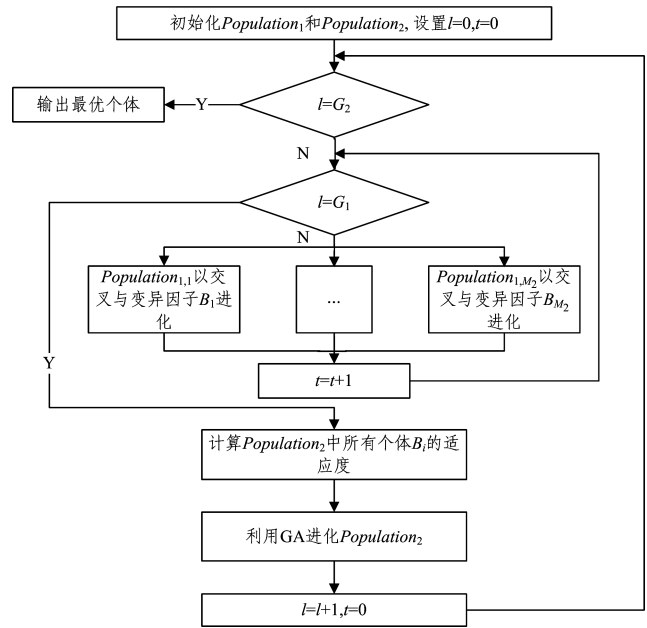


图 4 CGAA 算法的执行流程

Fig. 4 Execution flow of CGAA algorithm

6 实验分析

为了评估算法的性能,本文使用基于 CloudSim^[13] 支持的云 workflow 仿真工具包 WorkflowSim^[14] 对算法进行实验分析。仿真 workflow 使用 4 个著名的现实科学 workflow,包括 Epigenomics workflow (生物学)、Montage workflow (天文学)、LIGO workflow (引力物理学)和 CyberShake workflow (地震学),其 workflow 结构如图 5 所示^[15]。

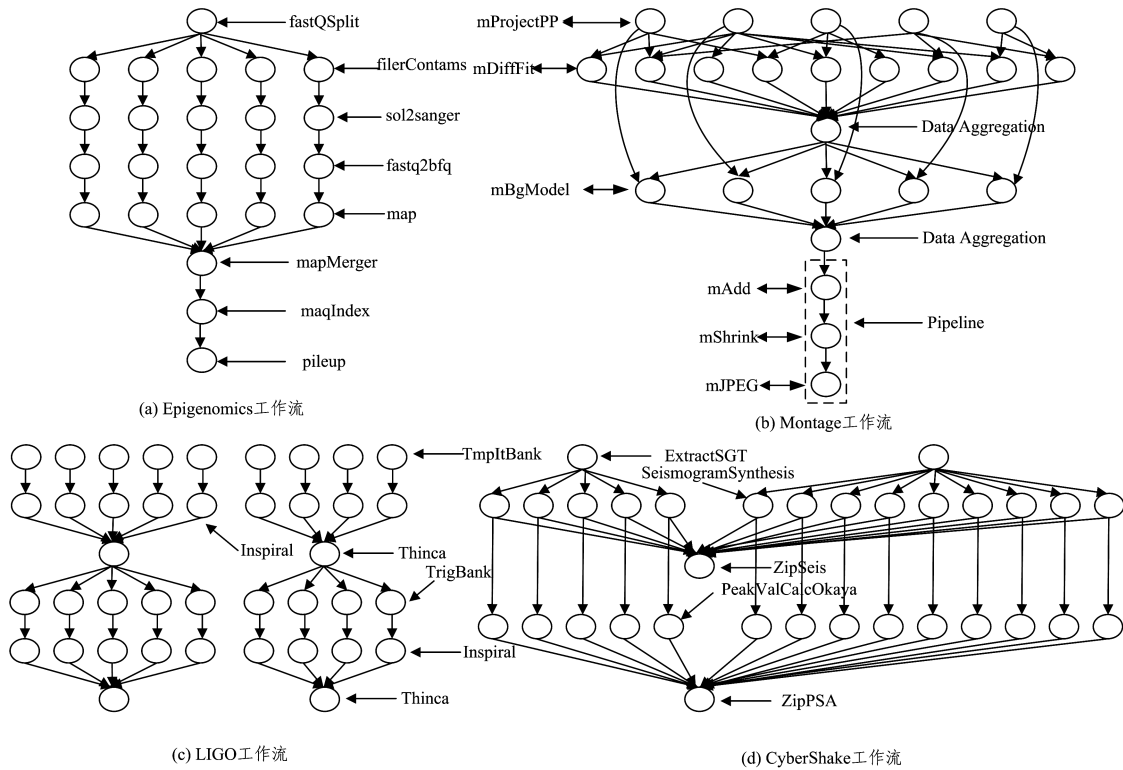


图 5 科学 workflow 结构图

Fig. 5 Structure of scientific workflow

Epigenomics 工作流为信息生物领域的工作流形式,任务以计算密集型为主,且对内存的要求较高;Montage 工作流为天文学领域中的工作流形式,任务以 I/O 密集型为主,对 CPU 计算能力要求不高;LIGO 工作流为物理学领域引力波工作流形式,任务以计算密集型为主,且对内存的要求较高;CyberShake 工作流为地震学领域的工作流形式,任务以数据密集型为主,且对计算能力和内存存储均有较高要求。

实验模拟了提供单个数据中心的 IaaS 云环境,支持 5 种 VMs,VMs 配置参考云应用 Amazon EC2^[16],参数如表 2 所列。

表 2 VMs 类型
Table 2 Types of VMs

类型名称	EC2 单元	处理能力	代价/小时(\$/h)
m1.small	1	44	0.03
m1.large	4	176	0.12
m1.xlarge	8	352	0.24
c1.medium	5	220	0.06
c2.xlarge	20	880	0.44

实验建立了 4 种不同的截止时间约束,截止时间处于最慢与最快时间之间,最慢时间设置为所有任务在所有 VMs 平均处理能力的单个 VM 上计算得到,最快时间设置为所有任务在最快处理能力的 VM 上计算得到。将最快时间与最慢时间之差以固定步长分成 10 等份,Deadline 1 为最慢时间减去 1 个步长时间,Deadline 2 为最慢时间减去 4 个步长时间,Deadline 3 为最快时间加上 2 个步长时间,Deadline 4 为最快时间加上 1 个步长时间。即约束严格度上有以下结果:Deadline 1 < Deadline 2 < Deadline 3 < Deadline 4。

CGAA 算法的其他参数分别设置为: $M_1 = 200, G_1 = 100, M_2 = 50, G_2 = 20$,实验运行 30 次,总执行时间 TET 和总执行时间 TEC 取平均值。为了符合现实科学工作流的随机化规模特点,本文在每种工作流中设置 3 种不同规模的任务结构:small(50 个任务)、medium(100 个任务)及 large(1000 个任务)。

使用以下 4 种求解约束最优化问题的常见工作流调度算法作为比较基准算法,包括随机算法 Randon、异构最早完成时间算法 HEFT^[17]、GA 算法^[8]以及截止时间约束下的云科学工作流调度 PSO 算法^[18]。

Randon 算法将就绪任务随机分配至空闲 VM 上。HEFT 算法给予拥有更高等级的工作流任务更高的优先级,该等级由每个任务的平均执行时间和两个连续任务在资源上的平均通信时间计算得到,协同进化中的任务拥有更高等级。然后,算法按等级的降序对任务进行排列,拥有更高等级的任务拥有更高的优先级。在资源选择阶段,任务按其优先级排序进行调度,每个任务分配至其完成时间最早的资源上执行。设 $|T_x|$ 为任务 T_x 的大小, R 为可用资源 VMs 集合,平均处理能力 $|R| = \sum |R_i| / n$,任务的平均执行时间为 $E(T_x) = |T_x| / |R|$ 。

6.1 截止时间约束性能

本节分析算法对工作流截止时间的满意程度,结果如图 6 所示。约束满意度的定义为:若执行时间小于或等于截止时间,则满意度为 100%;否则,满意度为截止时间与执行时间之比。对于 Epigenomics 工作流,HEFT 满足所有截止时

间约束。Random 算法在 Deadline 1 和 Deadline 2 上的满意度分别为 10% 和 3.3%,无法满足 Deadline 3 和 Deadline 4。GA 和 PSO 算法在 Deadline 1 和 Deadline 2 上的满意度为 100%,但当约束严格时,满意度越来越小;对于 Deadline 3,这两种进化算法的约束满意度为 93% 和 73.3%;对于 Deadline 4,其满意度分别为 26.7% 和 13.3%。对于 CGAA 算法,当约束变严格时,其仍能在截止时间内找到最优解。前 3 个 Deadline 的满意度均为 100%,Deadline 4 的满意度为 80%。对于 Montage 工作流,HEFT 算法满足所有截止时间约束,性能优于其他算法;Random 算法得到了与 Epigenomics 工作流类似的结果,其截止时间满意度最低;GA 与 PSO 算法仅在截止时间较宽松的 Deadline 1 和 Deadline 2 时性能更优;CGAA 算法即使在约束变得严格时,也能找到最优解,其满意度分别为:100%,100%,100% 和 60%。

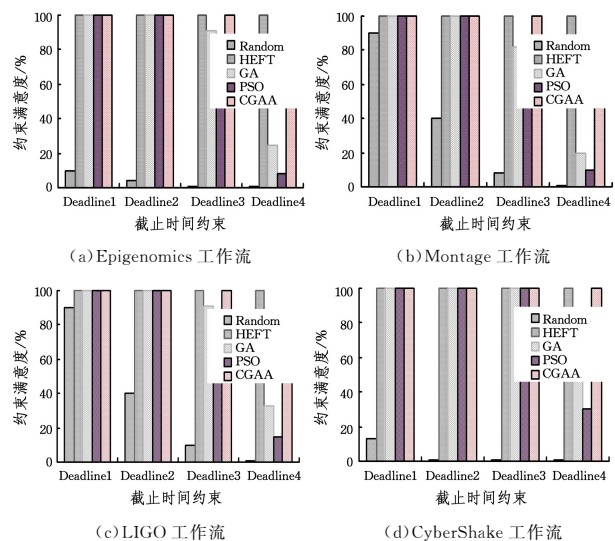


图 6 截止时间约束满意度

Fig. 6 Deadline constraint satisfaction

LIGO 工作流和 CyberShake 工作流的满意度与以上两种工作流类似。Randon 算法在所有约束下均很难得到可行解,而 HEFT 算法在所有约束下均能得到 100% 的满意度。对于 3 种进化算法,在前两种宽松约束下得到的性能是相似的,而 CGAA 算法则明显优于另外两种进化算法,其原因是 HEFT 算法始终将任务分配至完成时间最小的 VMs 上,仅考虑了整个工作流而不是重点关注拥有不同优先级的未映射独立任务;同时,算法未考虑执行代价约束。Random 算法将任务随机分配至 VMs 上,很难满足用户约束。对于 GA 和 PSO 等进化算法,利用静态惩罚函数或排除进化过程中约束违例个体的简单约束最优化处理方式,可能导致早熟收敛或使进化进入不可行区域的错误方向。

6.2 TEC 性能

图 7 给出了算法在不同工作流中得到的平均总执行代价。对于 Epigenomics 工作流,HEFT 算法的总执行时间在 4 种约束下均是最低的,但它的总执行代价高于其他进化算法。实验中,CGAA 算法在每种约束下均得到了最小代价,这表明算法比其他进化算法拥有更好的优化能力,尤其在约束严格的情况下。当约束严格时,GA 和 PSO 算法的解不仅无法满足截止时间约束,而且代价将更高,这表明静态类型的惩罚

函数可能导致种群的早熟与不可行。

对于 Montage 工作流, HEFT 算法的总执行时间在 4 种约束下均是最低的, 但它的总执行代价高于其他进化算法, 这主要是由于进化算法能够以更低的价格租用 VMs, 从而最小化总执行代价。而且, Montage 工作流中的任务相对更小, 这表明 HEFT 算法中的主机资源可以运行更少的时间但要支付全部的账单时间, 而选择更优秀的处理能力的 VM 将带来更高的代价。

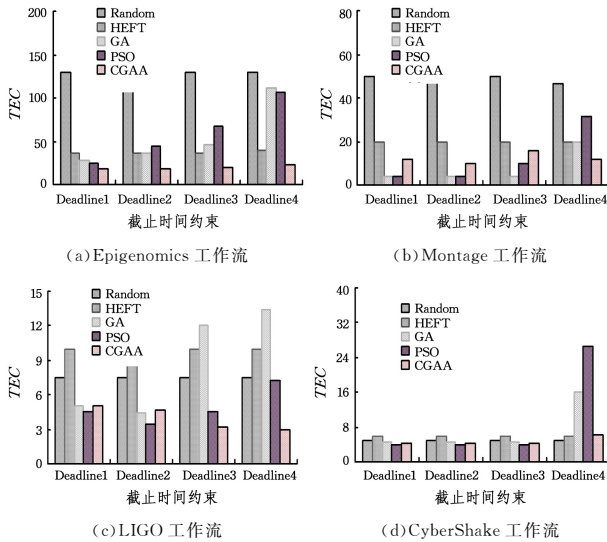


图 7 平均总执行代价

Fig. 7 Average total execution cost

在所有满足截止时间约束的算法中, GA 算法和 PSO 算法在宽松截止时间时可以得到较低的代价, 但当约束变得严格时, 算法得到的代价较高。而 CGAA 算法在较严格约束下其 TEC 仍可以满足截止时间, 并得到更优的解。综合以上结果可以明显地看出, 进化算法的代价优化优于 HEFT 算法, 而 CGAA 算法在较严格约束下仍不会出现约束违例, 并得到最少的代价。

对于 LIGO 和 CyberShake 工作流, 5 种算法的结果与前两种工作流是类似的, CGAA 算法仍然是 TEC 结果最好的, 尤其在约束较严格时。同时, 从图 7(c) 和图 7(d) 可以看出, 带有静态惩罚函数的 GA 算法和 PSO 算法在约束变得严格时, 其 TEC 增长较快, 而 CGAA 算法的结果未出现明显波动。而且, 尽管此时的 PSO 算法在宽松约束时的 TEC 结果优于 CGAA 算法, 但在严格约束时却无法得到满意解。

综合以上分析, 可以得出以下结论: Random 算法在多数情况下均无法满足截止时间约束, 而 HEFT 算法可以得到最低的执行时间跨度, 这是由于其优先将就绪任务分配至处理能力最强的 VMs 上。相比 HEFT 算法, 进化算法在执行代价方面是更优的。相比带有静态惩罚函数的 GA 算法和 PSO 算法, CGAA 算法在严格的截止时间约束下, 仍可以得到科学 workflow 调度的最优可行解。

结束语 为了解决云环境中科学 workflow 调度的代价优化问题, 提出一种基于自适应惩罚函数的协同进化遗传算法。针对现有进化算法的惩罚函数可能导致约束最优化问题求解的早熟收敛问题, 该算法利用自适应惩罚函数有效地调整交叉与变异概率, 从而防止种群早熟并改善 workflow 截止时间的

满意度。实验表明, 所提算法不仅可以在约束较严格时保证截止时间的满意度, 而且可以得到最小化的执行代价。

参考文献

- [1] KASHLEV A, LU S. A System Architecture for Running Big Data Workflows in the Cloud[C]// IEEE International Conference on Services Computing. IEEE, 2014: 51-58.
- [2] LIU L, ZHANG M, LIN Y, et al. A survey on workflow management and scheduling in cloud computing[C]// 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE, 2014: 837-846.
- [3] WU F, WU Q, TAN Y. Workflow scheduling in cloud: a survey[J]. Journal of Supercomputing, 2015, 71(9): 3373-3418.
- [4] RODRIGUEZ M A, BUYYA R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds[J]. IEEE Transactions on Cloud Computing, 2014, 2(2): 222-235.
- [5] WANG X, YEO C S, BUYYA R. Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm[J]. Future Generation Computer Systems, 2011, 27(8): 1124-1134.
- [6] HUANG T T, LIANG Y W. Improved simulated annealing algorithm of cloud workflow tasks scheduling[J]. Micro-electronics and Computer, 2016, 1(33): 42-46. (in Chinese)
黄婷婷, 梁意文. 云 workflow 任务调度的模拟退火遗传改进算法[J]. 微电子学与计算机, 2016, 1(33): 42-46.
- [7] FELLER E, RILLING L, MORIN C. Energy-aware ant colony based workload placement in clouds[C]// Proceedings of the 2014 IEEE/ACM 13th International Conference on Grid Computing. IEEE Computer Society, 2014: 26-33.
- [8] HUANG J. The workflow task scheduling algorithm based on the GA model in the cloud computing environment[J]. Journal of Software, 2014, 9(4): 873-880.
- [9] CAO B, WANG X T, XIONG L R, et al. Particle swarm searching method of cloud workflow scheduling under time constraint[J]. Computer Intergrated Manufacturing Systems, 2016, 22(2): 372-380. (in Chinese)
曹斌, 王小统, 熊丽荣, 等. 时间约束云 workflow 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.
- [10] LI Y L, SHAO W, WANG J T, et al. An improved NSGA-II and its application for reconfigurable pixel antenna design[J]. Radio Engineering, 2014, 23(2): 733-738.
- [11] ZHU Z, ZHANG G, LI M, et al. Evolutionary Multi-Objective Workflow Scheduling in Cloud[J]. IEEE Transactions on Parallel & Distributed Systems, 2016, 27(5): 1344-1357.
- [12] RAHMAN M, HASSAN R, RANJAN R, et al. Adaptive workflow scheduling for dynamic grid and cloud computing environment[J]. Concurrency Computation Practicew Experience, 2013, 25(13): 1816-1842.
- [13] RODRIGO C, RAJIV R, ANTON B, et al. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms[J]. Software: Practice and Experience, 2011, 41(1): 23-50.
- [14] CHEN W, DEELMAN E. WorkflowSim: a toolkit for simulating scientific workflows in distributed environments[C]// IEEE 8th

- International Conference on E-Science (e-Science), IEEE, 2012: 1-8.
- [15] JUVE G, CHERVENAK A, DEELMAN E, et al. Characterizing and profiling scientific workflows[J]. *Future Generation Computer Systems*, 2013, 29(3): 682-692.
- [16] Amazon. Amazon EC2 Pricing[EB/OL]. <http://aws.amazon.com/ec2/pricing>.
- [17] TOPCUOGLU H, HARIRI S, WU M Y. Performance-effective and low-complexity task scheduling for heterogeneous computing[J]. *IEEE Transactions on Parallel Distributed Systems*, 2012, 13(3): 260-274.
- [18] PANDEY S, WU L, GURU S M, et al. A particle swarm optimization based heuristic for scheduling workflow applications in cloud computing environments[C] // 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE, 2014: 400-407.
-
- (上接第 99 页)
- [6] NADERI M, PAETZOLD M, HICHERI R, et al. A Geometry-Based Underwater Acoustic Channel Model Allowing for Sloped Ocean Bottom Conditions[J]. *IEEE Transactions on Wireless Communications*, 2017, 4(16): 2394-2408.
- [7] GUTIERREZ C A, PATZOLD M. The Design of Sum-of-Cisoids Rayleigh Fading Channel Simulators Assuming Non-Isotropic Scattering Conditions[J]. *IEEE Transactions on Wireless Communications*, 2010, 9(4): 1308-1314.
- [8] BEYGI S, STROM E G, MITRA U. Geometry-based stochastic modeling and estimation of vehicle to vehicle channels[C] // *Speech and Signal Processing*, 2014. *IEEE International Conference on Acoustics*, 2014, 1: 4289-4293.
- [9] QIANG Z L, ZHU F L. Modeling and Simulation on Underwater Acoustic Communication Channel[J]. *Computer Knowledge and Technology*, 2013, 9(1): 140-145.
- [10] YING L, HUA Z X. Modeling and simulation on shallow water acoustic multipath channels[J]. *Ship Science and Technology*, 2010, 32(9): 120-124.
- [11] CHITRE M. A high-frequency warm shallow water acoustic communications channel model and measurements[J]. *Journal of the Acoustical Society of America*, 2007, 122(5): 2580-2586.
- [12] BREKHOVSKIKH L M, LYSANOV Y P. *Fundamentals of Ocean Acoustics*[M]. New York: Springer, 2002.
- [13] RADOSEVIC A, AHMED R. Adaptive OFDM Modulation for Underwater Acoustic Communications; Design Considerations and Experimental Results[J]. *IEEE Journal of Oceanic Engineering*, 2014, 39(2): 357-370.
- [14] DOMINGO M C. Overview of channel models for underwater wireless communication networks[J]. *Physical Communication*, 2008, 1(3): 163-182.
- [15] ZAJIC A G. Statistical Modeling of MIMO Mobile-to-Mobile Underwater Channels [J]. *IEEE Transactions on Vehicular Technology*, 2011, 60(4): 1337-1351.
- [16] MA Y, PÄTZOLD M. 2007 International Symposium on Wireless Personal Multimedia Communications Jaipur[C] // *IEEE International Conference on Acoustics Speech and Signal Processing*, 2007.
- [17] SKLAR B. *Digital communications*[M]. NJ: Prentice Hall, 2001.
- [18] MOLISCH A F, WIN M Z. MIMO systems with antenna selection[J]. *IEEE Microwave Magazine*, 2004, 5(1): 46-56.
- [19] TEAL P, ABHAYAPALA T, KENNEDY R. Spatial correlation in non-isotropic scattering scenarios[C] // *IEEE International Conference on Acoustics*. Orlando; IEEE, 2002; III-2833-III-2836.
- [20] CHO Y, LEE J H. Effect of fading correlation on the SER performance of M-ary PSK with maximal ratio combining[J]. *IEEE Communications Letters*, 2002, 3(7): 199-201.
- [21] LOYKA S L. Channel capacity of MIMO architecture using the exponential correlation matrix [J]. *Communications Letters IEEE*, 2001, 5(9): 369-371.
- [22] TSAI J, WOERNER B. The fading correlation function of a circular antenna array in mobile radio environment[J]. *Communications Letters IEEE*, 2002, 6(5): 178-180.
- [23] ZHOU Z, ISHIZAWA K, KIKUCHI H, et al. Generalized spatial correlation equations for antenna arrays in wireless diversity reception exact and approximate analyses[C] // *Proceedings of the 2003 International Conference on Neural Networks and Signal Processing*, 2003. IEEE, 2003: 180-184.
- [24] RADOSEVIC A, PROAKIS J G, STOJANOVIC M. Statistical characterization and capacity of shallow water acoustic channels [M]. *Bermen: OCEANS 2009-EUROPE*, 2009.
- [25] BOUVET P J, LOUSSERT A. Capacity analysis of underwater acoustic MIMO communications[M]. Sydney: Oceans, 2010.
- [26] HICHERI R, PATZOLD M, TALHA B. A study on the distribution of the envelope and the capacity of underwater acoustic channels[J]. *IEEE International Conference on Communication system*, 2015, 5(1): 394-399.
- [27] GUTIERREZ C A, PATZOLD M. Sum-of-Sinusoids-Based Simulation of Flat Fading Wireless Propagation Channels Under Non-Isotropic Scattering Conditions[C] // *IEEE Globecom Global Telecommunications Conference*. Washington; IEEE, 2007: 3842-3846.
- [28] GUTIÉRREZ C A, PÄTZOLD M. A generalized method for the design of ergodic sum-of-cisoids simulators for multiple uncorrelated rayleigh fading channels[C] // *International Conference on Signal Processing & Communication System*. Gold Coast; IEEE, 2011: 1-10.
- [29] PÄTZOLD M. *Mobile Fading Channels*[M]. New York: John Wiley & Sons, 2003.
- [30] ZAJIC A G, STUBER G L. Three-Dimensional Modeling, Simulation, and Capacity Analysis of Space-Time Correlated Mobile-to-Mobile Channels[J]. *IEEE Transactions on Vehicular Technology*, 2008, 57(4): 2042-2054.
- [31] MILLER I. *Probability, Random Variables, and Stochastic Processes*[J]. *Physics Today*, 1967, 20(1): 135-135.
- [32] PÄTZOLD M, TALHA B. On the statistical properties of mobile to mobile fading channels in cooperative networks under line of sight conditions[C] // *2007 Wireless Personal Communication*. 2007.
- [33] ZHOU J, WANG Y L, KIKUCHI H. Doppler power spectrum density and multi-antenna system performance in three-dimensional environment [J]. *Acta Physica Sinica*, 2014, 63(24): 240507.