

基于多变异位自适应遗传算法的 CPS 任务调度

焦晓璇, 景博, 黄以锋

(空军工程大学航空航天工程学院, 西安 710038)

摘要: 针对信息物理融合系统典型的分布式异构并行计算结构存在的任务调度问题, 以有向无环图为调度模型, 最小化任务完成时间为目标, 采用多变异位自适应遗传算法, 通过依赖矩阵、改进的交叉和变异算子确保基因个体的有效性, 同时采用多变异位和自适应的方法保证基因个体的多样性和算法的收敛性。仿真结果显示, 该算法比基于任务表的启发式调度算法更有效。

关键词: 信息物理融合系统; 任务调度; 有向无环图; 遗传算法; 表调度算法

中图分类号: TP316 **文献标志码:** A **文章编号:** 1671-637X(2015)02-0031-05

Task Scheduling for Cyber-Physical System Based on Multi-Mutation Adaptive Genetic Algorithm

JIAO Xiao-xuan, JING Bo, HUANG Yi-feng

(Engineering College of Aeronautics and Astronautics, Air Force Engineering University, Xi'an 710038, China)

Abstract: To solve the task scheduling problem of Cyber-Physical Systems (CPSs), which are typically distributed heterogeneous parallel computing architecture, the Directed Acyclic Graph (DAG) is adopted as the scheduling model, the minimum task implementation time is taken as the object, and multi-mutation adaptive genetic algorithm is used. The effectiveness of individual gene is ensured through dependency matrix, and improved crossover and mutation operators, while the diversity of individual gene and convergence of the algorithm are ensured by using multi-mutation and adaptive methods. Simulation results show that the algorithm is more efficient than list-scheduling algorithm.

Key words: Cyber-Physical System (CPS); task scheduling; Directed Acyclic Graph (DAG); genetic algorithm; list-scheduling algorithm

0 引言

随着实时通信技术的发展,人们可以通过高速网络将多个分散的有限计算能力的资源联系起来,形成分布式异构并行计算结构来处理复杂的并行分布式程序,但是这种异构计算结构给系统资源任务调度带来了巨大挑战,如云计算,信息物理融合系统(Cyber-Physical System, CPS)^[1]。中国科学院何积丰院士认为“CPS是在环境感知的基础上,深度融合了计算、通信和控制能力的可控、可信、可扩展的网络化物理设备系统,它通过计算进程和物理进程相互影响的反馈循环实现深度融合和实时交互来增加或扩展新的功能,以

安全、可靠、高效和实时的方式监测或者控制一个物理实体”^[2]。CPS中包括感知(如温度、湿度、应变等传感器)、传输(有线或者无线)、控制、计算等各种资源,它的正常运行是建立在资源合理有效的调度基础之上,如果调度过程中出现了问题,不仅仅会导致系统资源的浪费,更有甚者造成系统瘫痪。目前,我国针对CPS系统的调度算法仍无法满足实际的需要,这种不足在电力资源调度、车站调度等环境中体现得更为明显。

异构并行计算结构下任务调度的实质是将任务映射为多个具有顺序约束的子任务,考虑任务之间的通信开销,子任务在处理器上的运行时间等,将不同的子任务映射到不同的处理器节点,从而尽可能地缩短整个任务的执行时间。文献[3-4]已经证明任务调度是一个典型的NP完全问题。采用穷举法虽然可以获得最优解,但是当问题维数增加时,这种方法显然是不行的。国内外学者针对任务调度问题进行了大量的研

收稿日期:2014-04-10 修回日期:2014-04-28

基金项目:国家自然科学基金(51201182)

作者简介:焦晓璇(1990—),男,山西运城人,硕士生,研究方向为故障诊断、信息物理融合系统、智能算法。

究^[5-9],提出了基于启发式和智能搜索的调度策略。基于启发式的调度策略包括表调度算法^[10]、分簇调度^[11]、基于任务复制调度^[12];智能搜索调度策略包括模拟退火^[13-15]、蚁群优化^[16]、禁忌搜索算法^[17]等。基于表的任务调度分为两个过程,首先计算每个任务优先级,然后按照优先级排序分配计算资源,算法过程比较复杂,结果不一定是全局最优。文献[18]在任务调度过程中没有考虑任务在不同处理器之间的通讯延迟。

本文采用多变异位自适应遗传算法(Multi-Mutation Adaptive Genetic Algorithm, MMAGA)进行任务调度。在有向无环图(Directed Acyclic Graph, DAG)任务调度模型的基础上,明确了遗传算法的基因编码方式、种群初始化方法,以及交叉和变异算子,确保了种群个体的有效性,采用多变异位和自适应的方法保证基因个体的多样性和算法的收敛性,最后通过实例仿真,并且与 HEFT(Heterogeneous Earliest Finish Time, HEFT)和 LDGP(Longest Dynamic Critical Path, LDGP)算法进行了比较,验证了该算法的有效性。

1 问题描述与建模

任务调度分为动态调度和静态调度两种,静态调度是在系统的计算资源、任务依赖关系、通信成本以及处理器节点性能已知的前提下,在任务执行前完成任务的分配,反之即为动态调度。本文主要研究的是静态任务分配策略,由以下几部分组成。

1) 任务调度模型。DAG 采用三元组 $G(T, E, C)$ 来表示,如图 1 所示。其中: T 是 DAG 中节点集合,表示待处理的任务; E 是 DAG 中的有向边,表示任务之间的约束依赖关系; C 表示任务在不同资源上处理时的通讯延迟矩阵。

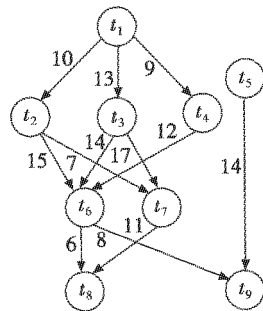


图 1 有向无环图

Fig. 1 Directed acyclic graph

2) m 个任务集合 $T = [t_1, t_2, t_3, \dots, t_m]$, $t_i (1 \leq i \leq m)$ 为第 i 个待处理的任务。

3) 依赖矩阵 $E_{m \times m} = [e_{ij}]$, 其中, $e_{ij} \in \{-1, 0, 1\}$, 表示任务 t_i 与任务 t_j 之间的约束关系。 $e_{ij} = 1$ 或者 $e_{ji} = -1$, 均表示任务 t_j 必须在任务 t_i 完成之后才能运行处

理; $e_{ij} = 0$, 表示两个任务之间没有依赖关系。

4) 通讯延迟矩阵 $C_{m \times m} = [c_{ij} | e_{ij} \in E, c_{ij} > 0]$, c_{ij} 为任务 t_i 与任务 t_j 在不同计算资源上处理时的通讯延迟矩阵。

5) n 个计算资源集合 $P = [p_1, p_2, p_3, \dots, p_n]$, $p_j (1 \leq j \leq n)$ 为第 j 个计算资源。

6) 任务在计算资源的执行时间矩阵 $W_{m \times n} = [w_{ij}]$, 计算资源的性能和处理能力不同, 用 w_{ij} 表示任务 t_i 在计算资源 p_j 上的执行时间。

定义 1 任务节点 t_i 的前驱节点

$$p_{\text{pre}}(t_i) = \{t_j | e_{ij} = -1 \text{ 或 } e_{ji} = 1\} \quad (1)$$

定义 2 任务节点 t_i 的后继节点

$$s_{\text{sue}}(t_i) = \{t_j | e_{ij} = 1 \text{ 或 } e_{ji} = -1\} \quad (2)$$

定义 3 任务 t_i 在资源 p_j 上的最早执行时间

$$T_{\text{test_begin}}(t_i, p_j) = \max\{T_{\text{pre_begin}}(t_i), T_{\text{fsh}}(p_j)\} \quad (3)$$

其中,

$$T_{\text{pre_begin}}(t_i) = \max\{T_{\text{fsh}}(t_j) + c'_{ji} | t_j \in p_{\text{pre}}(t_i)\} \quad (4)$$

式中: $T_{\text{fsh}}(t_j)$ 表示任务 t_j 的前驱任务完成时间; c'_{ji} 表示任务 t_i 与 t_j 之间的通讯延迟时间, 当 t_i 和 t_j 在同一计算资源时, 通讯延迟时间为 0。

$$c'_{ji} = \begin{cases} 0 & p_j = P_{\text{pre}}(t_j) \\ c_{ji} & \text{otherwise} \end{cases} \quad (5)$$

定义 4 $T_{\text{fsh}}(p_j)$ 表示执行任务 t_i 之前在资源 p_j 上运行任务的完成时间。

定义 5 任务 t_i 在资源 p_j 上的完成时间

$$T_{\text{fsh}}(t_i) = T_{\text{pre_begin}}(t_i) + w_{ij} \quad (6)$$

定义 6 整个调度任务完成时间

$$T_{\text{span}} = \max\{T_{\text{fsh}}(t_i) | 1 \leq i \leq m\} \quad (7)$$

2 多变异位自适应遗传算法

遗传算法是一种模拟自然界生物适者生存进化过程的全局优化概率搜索方法, 将问题的潜在解映射为基因编码, 通过适应度值选择较好的基因个体, 借助遗传算子进行交叉和变异, 最终通过多次迭代得到问题的最优解。多变异位自适应遗传算法是在自适应遗传算法中引入多位变异, 增加种群多样性, 扩大搜索范围, 同时采用自适应算法可以根据适应度值调整交叉概率和变异概率, 保证算法的收敛性。

多变异位自适应遗传算法流程如下所述(见图 2): 1) 初始化种群大小、迭代次数, 生成染色体基因编码; 2) 解码, 计算每个染色体的适应度值; 3) 通过轮盘赌策略选择适应度值大的染色体个体; 4) 根据自适应交叉算子进行交叉操作; 5) 根据适应度值产生变异位进行变异操作; 6) 判断终止条件, 如果满足条件, 则终止迭代, 输出最优值, 否则返回步骤 2)。

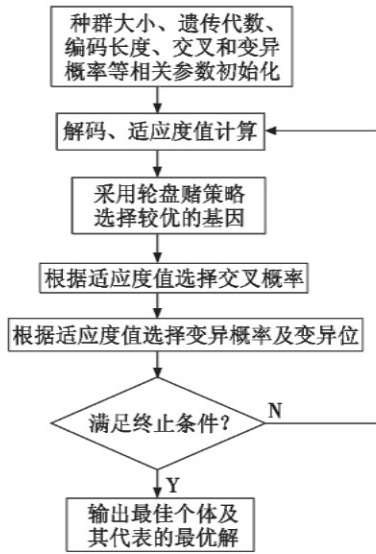


图 2 遗传算法流程

Fig.2 Flow chart of genetic algorithm

3 基于多变异位自适应遗传算法的任务调度

将 MMAGA 算法应用于任务调度,需要对算法以下部分进行改进说明。

3.1 染色体编码结构

任务调度的目标就是将 m 个具有顺序约束关系的任务分配到 n 个计算资源上,综合不同任务之间的通讯延迟和计算资源性能的差异,最终使得全部任务完成时间最短,因此染色体编码结构由 $t(\cdot)$ 和 $p(\cdot)$ 两部分组成,即 $[t_1 t_4 t_3 \dots t_m p_3 p_1 p_2 \dots p_n]$ 。

假设有 m 个任务,染色体的长度即为 $2 \times m$,染色体的第 k 位与 $m+k$ 位相对应,如示例中 t_4 和 p_1 ,表示任务 4 在资源 1 上执行。

3.2 种群初始化

种群个体的染色体编码结构由两部分组成,因此在初始化种群时要对左子串和右子串分别进行初始化。右子串表示的是不同的计算资源,资源之间没有依赖关系,只需要在 $m+1$ 和 $2 \times m$ 的每一位随机生成一个 $[1, n]$ 之间的随机数即可;左子串表示任务的调度顺序,必须满足 DAG 图中所示的依赖关系。研究人员普遍采用式(8)定义的深度值方法进行任务的调度,但是对于类似图 1 中存在不连通的子图时,采用深度值方法进行顺序调度就会漏掉一些有用的解,例如采用深度值的方法,任务 t_5 必须在任务 t_1, t_2, t_3, t_4 完成之后才能运行,但是实际上任务 t_5 的运行与 t_1, t_2, t_3, t_4 的执行没有关系。

$$H(t_i) = \begin{cases} 0 & p_{pm}(t_i) = \emptyset \\ 1 + \max_{t_k \in p_{pm}(t_i)} (H(t_k)) & \text{otherwise} \end{cases} \quad (8)$$

本文根据依赖矩阵 E ,任务之间的前驱和后继关

系,克服高度值排序的不足,以图 1 为例说明任务调度顺序的初始化。

1) 根据图 1 得到依赖矩阵 E ,如表 1 所示。

表 1 依赖矩阵

Table 1 Dependency matrix

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9
t_1	0	1	1	1	0	0	0	0	0
t_2	-1	0	0	0	0	1	1	0	0
t_3	-1	0	0	0	0	1	1	0	0
t_4	-1	0	0	0	0	1	0	0	0
t_5	0	0	0	0	0	0	0	0	1
t_6	0	-1	-1	-1	0	0	0	1	1
t_7	0	-1	-1	0	0	0	0	1	0
t_8	0	0	0	0	0	-1	-1	0	0
t_9	0	0	0	0	-1	-1	0	0	0

2) 判断依赖矩阵 E 中不包括 -1 的任务行($t_i \neq -1$),说明该任务没有前驱任务或者前驱任务已经执行完),将其所对应的任务提取并保存起来,去除已经执行过的任务,然后从中随机选择一个调度任务。表 1 中应该选择任务 t_1 或者 t_5 作为初始任务。

3) 将依赖矩阵 E 中,步骤 2) 执行过的任务对应的列全部更新为 0。

4) 判断染色体左子串是否已经选择了 m 个不同任务,如果小于 m ,返回 2),否则退出循环。

3.3 适应度函数

遗传算法以适应度值为搜索依据,适应度值越大,染色体个体被保留的概率越大,而任务调度的目标函数是尽可能减少任务执行的时间,因此定义适应度函数为

$$f(S_i) = \frac{1}{T_{span}(S_i)} \quad (9)$$

式中, S_i 为第 i 个染色体个体。

3.4 交叉和变异算子

交叉概率和变异概率是遗传算法中影响算法行为和性能的关键,直接影响算法的收敛性。交叉概率越大,新的染色体生成的速度越快,搜索的范围越大,最优解被破坏的可能性越大,但是交叉概率过小时,容易导致算法搜索速度过慢;变异概率过小时,不容易产生新的染色体编码,过大时遗传算法就完全变成了随机搜索算法。因此,采用自适应交叉概率 p_c 和变异概率 p_m 时,对于适应度值高的个体采取较低的交叉和变异概率,以保护较优的解进入下一代,反之,采用较大的交叉和变异概率,淘汰较差的解。

$$p_c = p_{c0} + \frac{p_{c0}(f_{max} - f)}{f_{max} - f_{avg}} \quad (10)$$

$$p_m = p_{m0} + \frac{p_{m0}(f_{max} - f')}{f_{max} - f_{avg}} \quad (11)$$

式中: p_{c0}, p_{m0} 为初始的交叉和变异概率; f_{max} 为最大的

适应度值; f_{avg} 为平均适应度值; f 为进行交叉的两个个体中较大的适应度值, f' 为变异个体的适应度值。

任务调度的交叉算子和变异算子, 必须满足经过交叉和变异后, 染色体中任务个数保持不变, 并且任务顺序满足任务之间的依赖关系的要求, 本文设计了如下的交叉和变异算子。

3.4.1 交叉算子

1) 采用轮盘赌转的方法选择待交叉的两个个体 S_i, S_j 。

2) 对于染色体的左子串和右子串采用不同的交叉方法, 左子串在交叉点前的任务保持不变, 交叉点后的任务按照待交换的染色体中的顺序进行重新排列; 右子串采用单点交叉方法, 即将交叉点后的计算资源进行互换。

3.4.2 变异算子

1) 采用轮盘赌转的方法选择染色体个体 S_i 。

2) 在染色体 S_i 上, 根据适应度值产生变异位, 如果变异位在左子串, 将变异位的任务在其前驱和后继任务之间进行迁移, 如果变异位在右子串, 将变异位的计算资源用其他随机产生的资源替代。

4 实例仿真与分析

4.1 实例仿真

例1 应用文献[19]中的案例来验证多变异位自适应遗传算法的有效性。已知任务调度的 DAG 结构如图3所示, 任务在计算资源的执行时间矩阵如表2所示。

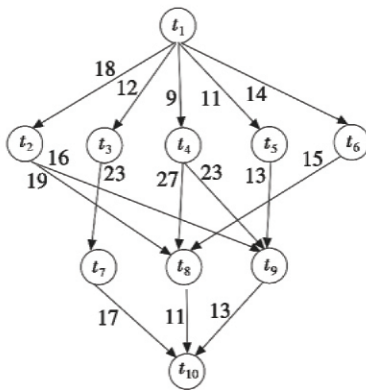


图3 DAG结构示意图

Fig. 3 Example of directed acyclic graph

表2 图3中任务在不同资源上的处理时间

Table 2 Execution time of tasks on different resource in Fig. 3

任务	p_1	p_2	p_3	任务	p_1	p_2	p_3
t_1	14	16	9	t_6	13	16	9
t_2	13	19	18	t_7	7	15	11
t_3	11	13	19	t_8	5	11	14
t_4	13	8	17	t_9	18	12	20
t_5	12	13	10	t_{10}	21	7	16

本文分别采用 HEFT, LDPC 和 MMAGA 进行任务调度, 设置 MMAGA 算法的参数为: 个体数目 $N = 80$, 最大进化代数 $M = 100$, 交叉概率 $p_{c0} = 0.3$, 变异概率 $p_{m0} = 0.02$, 以最小化任务完成时间为目标函数。图4为 MMAGA 算法任务完成时间与迭代次数的变化曲线, 从图中可以看出, 采用了多变异位和自适应方法增加了了解的多样性, MMAGA 算法在 50 代左右收敛于最优解。

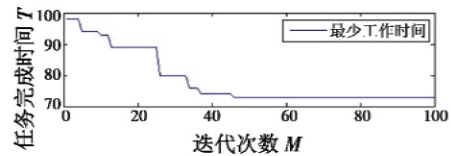


图4 MMAGA 适应度变化曲线

Fig. 4 MMAGA fitness curve

3 种不同算法的任务调度结果分别为: HEFT 为 [1 3 5 7 4 6 2 8 9 10 3 3 3 3 2 2 1 1 2 2], $T_{span} = 80$; LDPC 为 [1 3 6 7 4 5 2 8 3 10 3 3 3 3 2 2 1 1 2 2], $T_{span} = 76$; MMAGA 为 [1 2 6 4 5 3 7 9 8 10 3 3 3 2 2 1 1 2 2], $T_{span} = 73$ 。甘特图如图5所示, MMAGA 算法通过多次迭代之后不同任务之间基本上没有通讯的延迟, 因此调度结果最优。

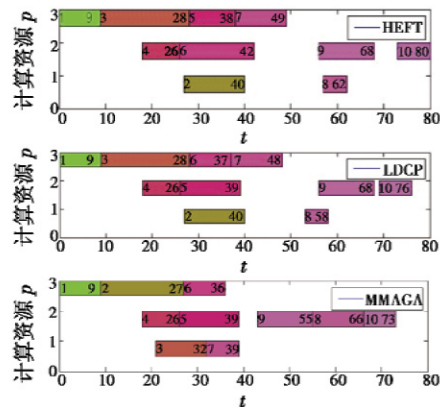


图5 不同调度算法结果甘特图

Fig. 5 Gant chart of scheduling result with different algorithms

例2 某公司有 8 项任务需要交付给 3 个不同的工厂进行加工, 每个任务的加工顺序和任务在不同工厂间的运输成本如图6所示, 任务在不同工厂的资源成本如表3所示。

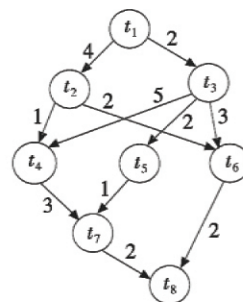


图6 任务 DAG 图

Fig. 6 DAG of tasks

表 3 任务在不同工厂消耗的资源成本

Table 3 The consumption of resource costs of tasks on different plants

任务	p_1	p_2	p_3	任务	p_1	p_2	p_3
t_1	8	13	11	t_5	10	9	13
t_2	15	11	13	t_6	15	13	17
t_3	12	17	15	t_7	9	11	15
t_4	17	13	8	t_8	13	8	9

本文采用 MMAGA 对任务进行优化,结果如图 7 所示。

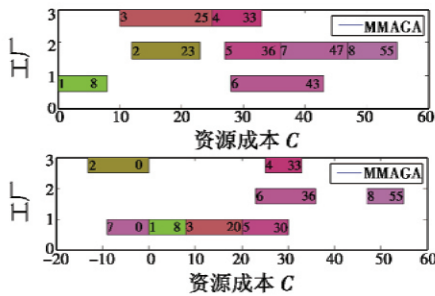


图 7 任务优化结果

Fig. 7 Task optimization results

从图 7 中可以得到任务序列分别为 [1 2 3 4 5 7 6 8 1 2 3 3 2 2 1 2]、[1 2 3 6 5 4 7 8 1 3 1 2 1 3 1 0], Cost 为 55。从仿真结果可以看出,本文设计的智能搜索算法能够适用于具有多个最优解的问题。

4.2 算法性能分析

进行算法的复杂度分析,设任务个数为 m ,计算资源个数为 n ,进化代数为 M ,种群个体数目为 N ,遗传算法需要计算每个个体的适应度值 f ,还要进行自然选择、交叉、变异,其时间复杂度为 $O(MN^2)$ 。基于启发式表调度算法的复杂度较低,HEFT 算法的时间复杂度为 $O(nm^2)^{[20]}$,LDPC 的时间复杂度为 $O(nm^3)$ 。基于启发式的调度算法结构简单、复杂度较低,但是其没有考虑当前任务对前驱和后继任务节点的影响,可能增加了任务之间的通讯费用,其调度结果可能是局部最优。

本文将 HEFT,LDPC 以及 MMAGA 3 种算法针对不同的 DAG 系统进行了对比,分别研究了任务与处理资源关系为 $10 \times 5, 30 \times 10, 50 \times 20$ 的系统。设任务处理时间区间为 $[5, 20]$,通讯延迟时间区间为 $[5, 15]$ 。在英特尔奔腾双核处理器 E6600,内存 2 GB 和 Windows XP, Matlab2009a 环境下编写算法程序。3 种算法分别对不同规模系统进行 10 次试验,然后取平均值,统计结果显示,随着规模的增大,MMAGA 算法运行时间增加,但是仍在可接受范围内,而 HEFT,LDPC 算法的运行时间相对较小,但是对于任务的调度,MMAGA 算法结果总是优于其他两种算法。

5 结束语

CPS 是物理过程与信息世界紧密交互的分布式异构处理系统,高效的资源任务调度算法能够提高其任务完成的可靠性。本文采用多变异位自适应遗传算法进行智能搜索,改进了染色体编码方式,保证初始种群的有效性,同时采用多变异位和自适应的方法,避免了算法的早熟和陷入局部最优的缺陷。仿真结果显示,该算法具有良好的全局搜索能力,而且算法的复杂度、计算时间在可接受的范围内,因此能够应用于 CPS 系统的任务调度。

参考文献

- [1] CHEN H, XIONG P C, SCHWAN K, et al. A cyber-physical integrated system for application performance and energy management in data centers[C]//Green Computing Conference, 2012:1-10.
- [2] 何积丰. Cyber-physical systems[J]. 中国计算机学会通讯, 2010, 6(1):25-29. (HE J F. Cyber-physical systems [J]. Communications of the China Computer Federation, 2010, 6(1):25-29.)
- [3] MICHAEL R G, JOHNSON D S. Computers and intractability: A guide to the theory of NP-completeness[M]. San Francisco: WH Freeman & Co., 1979.
- [4] ULLMAN J D. NP-complete scheduling problems[J]. Journal of Computer and System Sciences, 1975, 10(3):384-393.
- [5] ERGU D, KOU G, PENG Y, et al. The analytic hierarchy process: Task scheduling and resource allocation in cloud computing environment[J]. The Journal of Supercomputing, 2013, 64(3):835-848.
- [6] WANG L, KHAN S U, CHEN D, et al. Energy-aware parallel task scheduling in a cluster[J]. Future Generation Computer Systems, 2013, 29(7):1661-1670.
- [7] GUO F, YU L, TIAN S W, et al. A workflow task scheduling algorithm based on the resources' fuzzy clustering in cloud computing environment[J]. International Journal of Communication Systems, 2014. doi:10.1002/dac.2743.
- [8] SAVITHA P, REDDY J G. A review work on task scheduling in cloud computing using genetic algorithm[J]. International Journal of Scientific & Technology Research, 2013, 2(8):241-245.
- [9] SHEIKH H F, AHMAD I. Dynamic task graph scheduling on multicore processors for performance, energy, and temperature optimization[C]//Green Computing Conference, 2013:1-6.
- [10] NESMACHNOW S, ITURRIAGA S, DORRONSORO B, et al. List scheduling heuristics for virtual machine mapping in cloud systems[C]//VI High Performance Computing Latin America Symposium, 2013:37-48.

- tring radar [J]. *Science China Information Sciences*, 2010, 53(11):2310-2322.
- [5] CHEN C Y, VAIDYANATHAN P P. A subspace method for MIMO radar space-time adaptive processing [C]// *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007. doi: 10. 1109/ICASSP. 2007. 366388.
- [6] 张西川, 张永顺, 王永良, 等. 机载 MIMO 雷达空时二维杂波建模及特性分析[J]. *空军工程大学学报: 自然科学版*, 2010, 11(5):35-38. (ZHANG X C, ZHANG Y S, WANG Y L, et al. Modelling and characteristics analysis of two-dimensional space-time clutter for airborne MIMO radar[J]. *Journal of Air Force Engineering University: Natural Science Edition*, 2010, 11(5):35-38.)
- [7] 严韬, 谢文冲, 王永良. 机载 MIMO 雷达杂波建模及杂波特性分析[J]. *雷达科学与技术*, 2010, 8(4):289-295. (YAN T, XIE W C, WANG Y L. Model and clutter characteristics analysis for airborne MIMO radar[J]. *Radar Science and Technology*, 2010, 8(4):289-295.)
- [8] 胡文琳, 王永良, 王首勇. Log-normal 分布杂波背景下有序统计恒虚警检测器性能分析[J]. *电子与信息学报*, 2007, 29(3):517-520. (HU W L, WANG Y L, WANG S Y. Performance of ordered statistic CFAR detector under Log-normal distributed clutter[J]. *Journal of Electronics & Information Technology*, 2007, 29(3):517-520.)
- [9] 胡文琳, 王首勇, 王永良, 等. 多分布类型杂波背景下恒虚警处理器性能分析[J]. *系统工程与电子技术*, 2007, 29(6):883-886. (HU W L, WANG S Y, WANG Y L, et al. Performance analysis of CFAR processor in multi-distribution clutter background[J]. *Systems Engineering and Electronics*, 2007, 29(6):883-886.)
- [10] 杜鹏飞. 机载预警雷达恒虚警率检测方法研究[D]. 长沙: 国防科学技术大学, 2003. (DU P F. A study on constant false alarm rate detection of airborne early-warning radar[D]. Changsha: National University of Defense Technology, 2003.)
- [11] KARIMIAN A, YARDIM C, GERSTOFT P, et al. Multiple grazing angle sea clutter modeling[J]. *IEEE Transactions on Antennas and Propagation*, 2012, 60(9):4408-4417.
- (上接第 35 页)
- [11] 郭凤羽, 禹龙, 田生伟, 等. 云计算环境下对资源聚类的工作流任务调度算法[J]. *计算机应用*, 2013, 33(8):2154-2157. (GUO F Y, YU L, TIAN S W, et al. Workflow task scheduling algorithm based on resource clustering in cloud computing environment[J]. *Journal of Computer Applications*, 2013, 33(8):2154-2157.)
- [12] 谢志强, 韩英杰, 齐永红, 等. 基于关键路径和任务复制的多核调度算法[J]. *国防科学技术大学学报*, 2014, 36(1):172-176. (XIE Z Q, HAN Y J, QI Y H, et al. A scheduling algorithm for multi-core based on critical path and task duplication[J]. *Journal of National University of Defense Technology*, 2014, 36(1):172-176.)
- [13] CHAI S, LI Y, WANG J, et al. A list simulated annealing algorithm for task scheduling on network-on-chip[J]. *Journal of Computers*, 2014, 9(1):176-182.
- [14] SHIVASANKARAN N, KUMAR P S, NALLAKUMARASAMY G, et al. Repair shop job scheduling with parallel operators and multiple constraints using simulated annealing[J]. *International Journal of Computational Intelligence Systems*, 2013, 6(2):223-233.
- [15] CHAI S, LI Y B, WU C, et al. A comparison of genetic algorithm, particle swarm optimization and simulated annealing in real-time task scheduling on CMP[J]. *Advanced Materials Research*, 2013, 679:77-81.
- [16] ZHAO J, QIU H. Genetic algorithm and ant colony algorithm based energy-efficient task scheduling[C]// *International Conference on Information Science and Technology (ICIST)*, IEEE, 2013:946-950.
- [17] GONZALEZ-SIEIRA A, BUGARIN A, MUCIENTES M, et al. A tabu search optimization module for scheduling: Design and integration in the open source tool LibrePlan for project management[C]// *The 8th Iberian Conference on Information Systems and Technologies*, IEEE, 2013:1-6.
- [18] 潘玉林, 陈勇, 姚新宇, 等. 多层次分布实时仿真系统的静态调度策略[J]. *系统仿真学报*, 2011, 23(1):85-88. (PAN Y L, CHEN Y, YAO X Y, et al. Task matching and scheduling for multi-level distributed real-time simulation system[J]. *Journal of System Simulation*, 2011, 23(1):85-88.)
- [19] 王小乐, 黄宏斌, 邓苏. 处理顺序约束的信息物理融合系统静态任务表调度算法[J]. *自动化学报*, 2012, 38(11):1870-1879. (WANG X L, HUANG H B, DENG S. List scheduling algorithm for static task with precedence constraints for cyber-physical systems[J]. *Acta Automatica Sinica*, 2012, 38(11):1870-1879.)
- [20] DAOU M I, KHARMA N. A high performance algorithm for static task scheduling in heterogeneous distributed computing systems[J]. *Journal of Parallel and Distributed Computing*, 2008, 68(4):399-409.