

## 融合安全的网格依赖任务调度双目标优化模型及算法\*

朱海<sup>+</sup>, 王宇平

(西安电子科技大学 计算机学院, 陕西 西安 710071)

### Integration of Security Grid Dependent Tasks Scheduling Double-Objective Optimization Model and Algorithm

ZHU Hai<sup>+</sup>, WANG Yu-Ping

(School of Computer Science and Technology, Xidian University, Xi'an 710071, China)

+ Corresponding author: E-mail: zhu\_sea@163.com, <http://www.xidian.edu.cn>

**Zhu H, Wang YP. Integration of security grid dependent tasks scheduling double-objective optimization model and algorithm. *Journal of Software*, 2011, 22(11): 2729-2748. <http://www.jos.org.cn/1000-9825/3900.htm>**

**Abstract:** In order to solve the security threats that dependent tasks scheduling problems face under the heterogeneous grid environment, this paper takes into account the inherent safety and behavior of security of the grid resource node and the reliability of measurement functions in the grid resource node. In addition, the behavior in credibility assessment strategies are also constructed. In order to establish the subordinate relationship between the security requirements of the task nodes and resources security attributes, security benefits of the membership functions are defined. Hence, a grid task scheduling model for security integration is established. On this basis, the requirement representation model and the grid resource topology model are defined; thus, the models of double-objective optimization of grid task scheduling are proposed. In order to solve this model, the definition of depth values and the sort of coupling is introduced when dealing with the constraints between tasks. A particle evolution equation is re-defined and re-designed to consider the specific characteristics of the grid task scheduling problem. At the same time, a selection strategy is defined, based on the uniformly distributed vector and concentration of particles. Thus, this paper presents a multi-objective optimization of grid task scheduling particle algorithm, and the algorithm is proved to be viable by applying the relevant knowledge of a probability theory. Simulation results show that compared with similar algorithms, under the same conditions, this algorithm has a faster convergence speed and a better performance in double-objective optimization.

**Key words:** grid computing; dependent tasks scheduling; security integration model; double-objective optimization; evolution equation; uniform distribution; discrete particle swarm optimization algorithm

**摘要:** 为了解决异构网格环境下依赖任务调度问题面临的安全威胁,综合考虑网格资源节点的固有安全性和行为安全性,分别构建了一个网格资源节点身份可靠性度量函数和行为表现信誉度评估策略.同时,为了确立任务安全

\* 基金项目: 国家自然科学基金(61103143, 60873099); 国家教育部博士点基金(20090203110005)

收稿时间: 2009-12-28; 修改时间: 2010-03-30; 定稿时间: 2010-06-10

CNKI 网络优先出版: 2010-12-29 13:50, <http://www.cnki.net/kcms/detail/11.2560.TP.20101229.1350.000.html>

需求与资源节点安全属性之间的隶属关系,定义了安全效益隶属度函数,从而建立一个网格任务调度的安全融合模型.以此为基础,提出一个时间-安全驱动的双目标优化网格依赖任务调度模型.为了求解该模型,处理任务间约束关系时引入深度值和关联耦合度的排序定义,再结合网格任务调度问题的具体特点,重新定义和设计新的粒子进化方程.同时,基于均匀分布向量和粒子浓度定义了选择策略,从而提出一种双目标优化的网格依赖任务调度粒子群进化算法,并运用概率论的有关知识证明算法的收敛性.最后,对所提出的离散粒子群进化算法进行了多角度分析和大规模仿真实验,其仿真结果表明,该算法与同类算法相比,不仅具有较好的收敛速度和单目标优化性能,而且在任务调度长度和安全满意度方面具有更好的双目标优化综合性能.

**关键词:** 网格计算;依赖任务调度;安全融合模型;双目标优化;进化方程;均匀分布;离散粒子群算法

**中图法分类号:** TP311      **文献标识码:** A

网格将互联网中的计算资源、存储资源、信息资源、知识资源等多种资源组织在一个统一的框架下<sup>[1]</sup>,为各种复杂的应用任务提供服务资源.因此,应用任务在各种资源间的有效调度就成为网格应用系统获得高性能的关键因素之一.然而,由于网格环境具有异构性、分布性、开放性、不确定性及动态性等特征,对传统的调度策略提出了新的挑战.任务在网格平台调度执行时,除了满足传统调度策略中的时间性能需求以外,还必须面临网格环境的开放特性可能造成敏感数据在执行或传输过程中被泄密、篡改或冒名顶替等风险<sup>[2,3]</sup>;同时,由于网格资源节点的行为属性失效<sup>[4]</sup>任务需要重新调度或延迟执行等,因此,应用任务在网格平台的安全可靠调度成为一个新的必须要考虑的因素.针对此问题,本文提出了一个网格任务调度的安全融合模型.该模型综合考虑异构网格环境下资源节点的固有安全性和行为安全性,根据资源节点本身所采用的加密算法、Hash 函数及身份认证技术类型构造资源节点的身份可靠性度量函数,同时根据资源节点历史行为表现构建节点的信誉度动态评估策略,并且定义安全效益隶属度函数,使任务安全需求和资源节点安全属性间关系得以确立.

当前,网格任务调度的研究大多针对独立任务或元任务的特殊形式<sup>[5,6]</sup>,忽视了任务间的数据关联与优先约束关系,不能反映应用任务间的实际特征.具有依赖关系的任务调度在传统调度策略中的研究由来已久<sup>[7]</sup>,但通常很少考虑任务间的通信关系以及资源节点的异构性带来的链路竞争等影响.典型的依赖任务调度模型都是建立在图的基础之上<sup>[8,9]</sup>,也就是任务图优先模型.根据任务图的基本信息、处理单元本身及其拓扑结构的基本信息是否在应用任务调度前获得、已经调度好的任务是否可以迁移等因素,将任务调度分为静态调度和动态调度两大类.本文对具有依赖关系的静态网格任务调度问题,基于安全融合模型将不同网格任务的时间性能和安全性能等需求和网格资源服务水平结合起来,使网格任务调度在满足任务间依赖关系的约束下将任务调度长度和安全效益值达到最优,从而建立了一个双目标优化新模型.

如何把复杂的应用任务合理调度到网格资源节点上有效执行,已被证明是一个 NP 难完全问题.不少学者将遗传算法、模拟退火算法和蚁群算法等<sup>[10,11]</sup>应用到任务调度,并取得了很好的效果.粒子群算法作为一种新的群智能算法<sup>[12]</sup>,具有收敛速度快、容易克服早熟缺陷和分布式运算等优点,并且操作简单,易于实现,因此一经提出就受到广泛关注,目前已经在函数优化领域取得了成功的应用.但对粒子群算法的研究目前主要集中在连续优化问题的方面<sup>[13]</sup>,在离散组合优化方面,特别分配调度问题的研究非常有限<sup>[14]</sup>,而且多为求解单目标优化问题.另外,粒子群算法在求解离散组合优化问题时,由于当前粒子与最优粒子位置难以用定量的方式表示,导致粒子的速度和位置不能参照标准粒子群进化方程更新,此时基本 PSO 失效.为了求解本文提出的新模型,结合双目标优化的网格任务调度问题具体特点,运用粒子群算法运行机理对粒子的位置、速度及其运算规则进行了重新表示,设计了新的粒子进化方程,并基于均匀分布和浓度定义了新的选择策略来保证双目标非劣解集的分布均匀性;同时,为了处理依赖任务间约束关系在算法设计中引入任务深度值和关联度的排序定义,从而提出一种双目标优化离散粒子群算法(discrete particle swarm optimization,简称 DPSO),并运用概率论的有关知识证明算法的收敛性.

本文第 1 节对相关工作进行阐述.第 2 节给出双目标优化的网格资源任务调度体系框架和网格任务调度过程的简要描述.第 3 节着重对网格资源安全融合模型进行探讨研究;在第 4 节提出网格任务调度的双目标优化

模型.根据任务调度组合优化问题的具体特点,在第5节提出一种双目标优化的粒子群进化算法.第6节进行仿真实验与性能分析.最后一节对本文所做工作进行总结和展望.

## 1 相关工作

传统的任务调度策略,比如DSL,TCS等<sup>[15,16]</sup>,虽然都比较成功地应用到调度系统中,但通常忽略了任务调度对安全的需求,也没有考虑到网格资源节点的开放性、动态性等特征,从而使调度系统难以在开放、动态、真实的网格环境中有效运行.近年来,有不少学者也开始关注研究此问题.如,文献[17]对同构网格环境下的实时任务调度应用融入了安全因素,但没有考虑网格节点异构性和不确定性等因素;文献[18]针对网格资源节点的不确定性,参考社会人际关系信任模型建立了网格节点信任推荐机制,有效提高了任务在信任方面的服务质量需求,但是对网格资源节点本身固有的安全性欠缺考虑;文献[19]对传统批作业调度问题进行信任扩展,考虑了信任机制和调度机制的关系,但对信任表达并没有给出具体量化处理办法,在实际网格中无法应用.目前的这些相关研究工作大多数仅仅关注资源的可靠性而忽视安全性,特别是对任务需求和资源属性间的关系缺乏把握,调度机制和安全机制的分离,导致调度系统难以在开放、动态真实的网格环境中有效运行.本文基于现有工作基础并针对其不足提出了一种安全融合模型,综合考虑了网格资源节点的固有安全性和行为安全性,采用安全效益函数使任务安全需求和资源节点安全属性间关系得以确立,较好地解决了此问题.

随着网格应用研究的深入,对网格任务调度,特别是依赖网格任务调度问题的研究已有不少相关工作.如文献[20]对传统同构调度方案向异构进行了扩展,同时考虑任务的执行时间以及任务之间的通信时间.但由于其在计算过程中使用平均计算时间和平均通信时间,因而忽略了资源的差异性,不能适用于资源个体差异较大的网格环境中.文献[21]根据资源的能力分配任务从而实现资源的负载平衡,进而消除资源差异造成的影响.但异构资源节点间通信及资源本身安全性考虑欠缺,没有从根本上解决资源差异性问题.文献[22]将网格依赖任务可能的资源分配方案表示成任务资源分配图,在此基础上提出了优化选取的依赖任务调度模型,把依赖任务调度问题转化为图的优化选取问题,在资源差异较大及存在大量数据通信传输时取得了较好的效果,但由于它也没有考虑网格资源节点及链路本身的安全可靠性,不能直接适用于动态开放的网格环境.上述这些依赖网格任务调度的研究主要针对时间性能指标建立单目标优化模型,在一定程度上解决了网格异构性面临的问题,但对网格环境开放性带来的安全性欠缺考虑.鉴于这些不足,将本文提出的安全模型融合到依赖网格任务调度问题中,提出了时间-安全优化的双目标调度模型,并引入深度值和关联耦合度的排序定义,较好地处理了任务间的依赖约束关系.

对网格任务调度算法的研究,目前主要在QoS需求建模基础上,采用调度启发式来直接构造最终调度方案.已有不少研究者进行了基于多QoS约束的网格任务调度算法研究.文献[23]提出了一种QoS-Min-min启发式调度算法,采用的调度任务分为高、低两种服务质量需求,通过优先调度高服务质量的任务来满足用户.但由于其解决独立任务调度问题,并不适合解决依赖关系任务调度问题.文献[24]根据负载的情况进行调度,解决了任务调度的NASH均衡问题,但未考虑不同节点执行效率的问题,因而不能将调度长度降到最低.文献[25]将任务与聚类资源相匹配并实施调度,提出一种网格资源多维性能聚类任务调度算法,但没有在全局范围比较最优解,不能保证调度算法的收敛性.文献[26]提出了一个多QoS模型,基于这个模型比较了QSMTS-IP,GA,Least Slack First和Sufferage等几种启发式算法的性能.以上这些启发式调度策略考虑了任务的多个目标需求,将多目标聚合成单一目标函数处理.这种方法简单而有效,因此被广泛研究和采用,但其自身也存在一些固有的缺陷:由于多目标优化问题的解并非唯一的,而是存在一个最优解集,集合中的解称为Pareto最优或非劣解.而单目标优化算法仅能根据聚合函数得到决策空间的一个可行解,降低了最终解的质量,缺乏灵活性和扩展性.更为合理的途径是采用多目标组合最优化算法来解决这个问题,基于进化理论的粒子群算法是其中较为有效的方法之一.

粒子群主要被设计为在连续论域中搜索数值函数的最优值,并且实验证明它是一个非常有效的工具,但在离散论域中PSO算法的研究非常少.文献[27]对电网机组离散量的高维非线性控制问题,用基于二进制编码的PSO处理机组开工控制,同时用实数编码PSO求解经济负载分配问题,两个算法同时运行组成混合PSO算法,

取得了较好的效果.但由于其基于连续空间生成的连续解与目标函数评价值存在多对一映射,不能完全反映解的质量.文献[28]针对 TSP 问题用所有城市的一个排列表示粒子的一个位置,并引入了基本交换序的概念,在求解 14 个城市的 TSP 问题时只搜索问题空间的 0.064%就找到了最优解.文献[29]把任务分配问题映射为一个  $N$  维空间的搜索问题,由于位置在速度作用下会产生负数且不是整数,采用了舍去符号和小数方法,使速度作用的位置依然是合法搜索空间的位置,并重新定义了粒子方程,对单目标优化取得了不错的效果.本文在这些工作的基础上并针对其不足,结合网格任务调度的双目标组合优化问题具体特点,运用粒子群算法进化机理对粒子的位置、速度及其运算规则进行了重新表示,设计了新的粒子进化方程,同时基于均匀分布向量和粒子浓度定义了新的选择策略,从而提出一种双目标优化的网格任务调度粒子群进化算法.

## 2 网格资源任务调度体系框架

双目标优化的任务调度器可以设计成一个中间件,然后插入到网格资源系统中,从而使网格任务能够有效分配到目标性能达到最优的节点上执行.这样,一方面有效降低了任务调度长度,另一方面尽可能地提高了任务执行的完全性和可靠性,较好地满足了网格应用的服务质量需求.网格资源任务调度的体系架构如图 1 所示.

网格资源任务调度体系架构的最底层为网格资源层(包括节点局部调度器),第 2 层为网格中间件层,第 3 层为任务全局调度器层,最上一层为网格应用任务客户端.在全局调度器中,调度顾问为基于安全模型的离散粒子群算法.

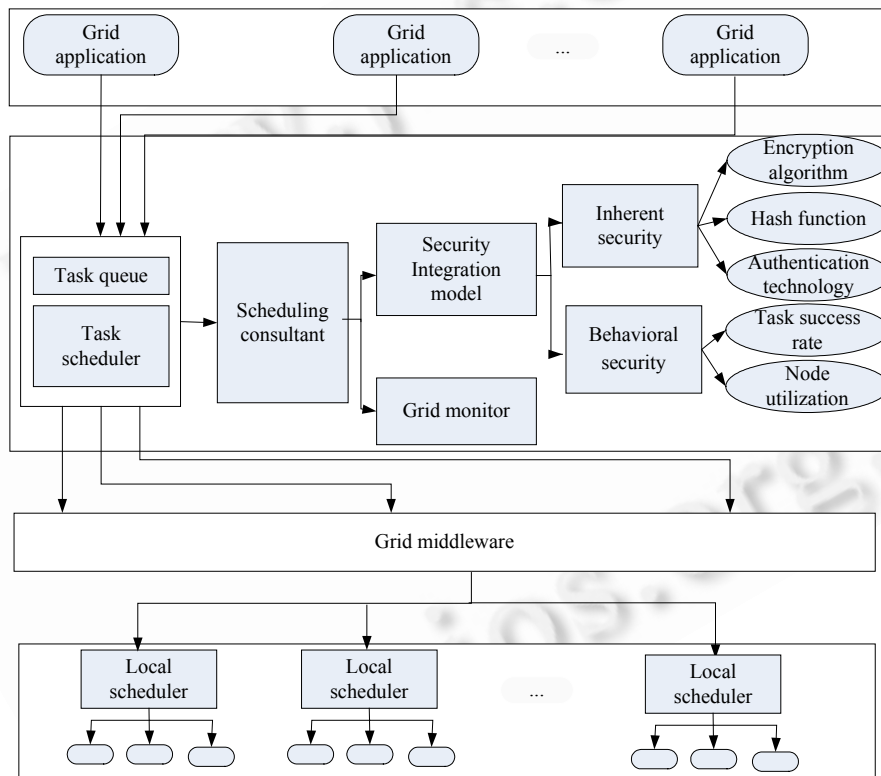


Fig.1 Architecture of grid resource task scheduling

图 1 网格资源任务调度体系框架图

在基于安全融合模型的双目标优化网格资源任务调度系统框架下,一个网格用户提交并执行一个网格任务的过程如下:

- (1) 应用客户端将网格应用提交到任务队列进行排队.
- (2) 任务调度器(全局)从任务队列中取出一个网格应用中所有具有依赖关系的子任务,同时与调度顾问进行交互.
- (3) 调度顾问与网格监视器和安全融合模型交互.
- (4) 网格监视器将网络目前链路状态和可用资源信息反馈给调度顾问.
- (5) 安全融合模型通过分析网格监视器反馈的可用资源信息,对资源节点的调度成功率及利用率进行分析获得行为安全性;同时,根据资源节点加密水平等获得固有安全性,然后将网格资源的安全信息反馈给调度顾问.
- (6) 调度顾问将网格监视器和安全融合模型反馈的信息传递给任务全局调度器,通过网格中间件将所有子任务分配到较优资源节点集执行.
- (7) 局部调度器对其控制范围内网格资源节点上任务进行最优调度.

### 3 网格资源安全融合模型

#### 3.1 资源节点的身份可靠性度量函数

网格资源节点的身份可靠性确保主要是由其采用的算法或方法决定的:如保密性一般主要依赖不同加密算法(如 IDEA,DES,RC5,Blowfish 和 RC4 等)实现,预防任务在网格环境中执行或传输时暴露给未经授权的其他用户或进程,避免信息泄露;完整性主要通过不同的 Hash 函数(如 MD4,RIPEMD,SHA-1 和 Tiger 等)来实现,确保任务在网格平台下执行时不会被非法的用户修改或篡改;真实性使用不同的数字签名技术(HMAC-MD5, HMAC-SHA-1 和 CBC-MAC-AES 等)对通讯实体身份的真实性进行鉴别,保证所处理的任务是从合法的用户提交的.下面以保密性为例,说明如何合理设计网格资源节点的保密性安全水平值.假定在网格环境下所有资源节点共采用了 8 种加密算法<sup>[3]</sup>,见表 1.

Table 1 Encrypt algorithms and their security value

表 1 加密算法及其安全水平值

Encrypt algorithm	Algorithm performance (KB/ms)	Security level
SEAL	168.75	0.08
RC4	96.43	0.14
Blowfish	37.5	0.36
Knufu/Khafre	33.75	0.40
RC5	29.35	0.46
Rijndael	21.09	0.64
DES	15	0.90
IDEA	13.5	1.00

将性能最低的 IDEA 对应的保密性安全水平值假定为 1,其他任一网格资源节点  $r_u$  上加密算法的安全水平值根据其性能可由公式(1)计算获得:

$$SL_{r_u}^e = 13.5 / \mu_{r_u}^e(k), 1 \leq k \leq 8 \quad (1)$$

这样,根据加密算法的不同性能将其安全值设定为 0.08~1 之间的值.可以看出,加密算法的保密性安全水平值与它的性能成反比例关系,安全水平值越低,其性能一般越高.这也符合一般逻辑思维,在同等条件下,没有人会选一个安全水平低而其性能也低的加密机制.

同理,网格资源节点  $r_u$  根据其采用的 Hash 函数和数字签名技术类型,得到其完整性和真实性水平值分别为  $SL_{r_u}^c$  和  $SL_{r_u}^a$ .

**定义 1(节点的身份可靠性).** 节点的身份可靠性指其上执行任务时所能够提供的保密性、完整性和真实性等安全服务保障能力.对任一网格资源节点  $r_u$ ,若只考虑保密性、完整性和真实性等安全性能,则其身份可靠性安全水平值为

$$\left. \begin{aligned} SL_{r_u} &= \omega_1 SL_{r_u}^e + \omega_2 SL_{r_u}^g + \omega_3 SL_{r_u}^a \\ \text{s.t. } \sum_{i=1}^3 \omega_i &= 1, \omega_i \geq 0 \\ 0 \leq SL^k &\leq 1, k \in \{e, g, a\} \end{aligned} \right\} \quad (2)$$

在公式(2)中,  $\omega_i$  表示不同安全服务的权重, 权重越大, 表示其相应的安全服务越重要.

### 3.2 资源节点的信誉度评估策略

**定义 2(节点信誉度).** 节点信誉度主要表征资源节点的历史行为表现特征, 定义为一个随时间变化的函数, 除了与网格资源节点累计使用率有关以外, 还与节点在累计执行任务获得的信任评价相关. 资源节点  $r_u$  在网格系统中的信誉度评估策略为

$$\left. \begin{aligned} RL_{r_u} &= \alpha \times e^{-TB_{r_u}} + \beta \times RS_{r_u} + \gamma \times RU_{r_u} \\ \text{s.t. } 0 < \alpha, \beta, \gamma < 1 \\ \alpha + \beta + \gamma &= 1 \end{aligned} \right\} \quad (3)$$

在公式(3)中,  $\alpha$  为时间衰减权重系数,  $\beta$  为网格资源节点  $r_u$  累计执行调度任务获得的信任评价权重系数,  $\gamma$  为资源累计使用率权重系数;  $TB_{r_u}$  为网格节点  $r_u$  最近一次成功执行任务到目前为止的时间间隔,  $TB_{r_u} > 0$ ;  $RU_{r_u}$  为节点  $r_u$  累计使用率, 可用节点  $r_u$  累计执行任务时间与累计活跃时间的比值来表示;  $RS_{r_u}$  为节点累计执行任务获得的信任评价, 包含 3 个因子, 即时间、身份可靠性安全值和行为信誉度, 其计算方法为

$$RS_{r_u} = \chi \times \frac{\sum_i (c_{io} - c_i)}{\sum_i c_{io}} + \delta \times \frac{\sum_i (s_i - s_{io})}{\sum_i s_{io}} + \eta \times \frac{\sum_i (r_{io} - r_i)}{\sum_i r_{io}} \quad (4)$$

在公式(4)中, 系数  $\chi, \delta$  和  $\eta$  分别为时间、身份可靠性和信誉度对节点  $r_u$  信任评价值的贡献率,  $0 < \chi, \delta, \eta < 1$  且  $\chi + \delta + \eta = 1$ ;  $c_{io}$  为预估计完成时间,  $c_i$  为实际花费时间;  $s_{io}$  为资源节点  $r_u$  声明的安全值,  $s_i$  为实际获得的安全值; 同样,  $r_{io}$  为资源节点  $r_u$  调度时声称的信誉度值,  $r_i$  为实际获得的信誉度.

### 3.3 安全融合模型

**定义 3(身份可靠性隶属度).** 身份可靠性隶属度表征网格任务对资源节点的保密性、完整性及真实性等固有安全水平的满意程度. 网格任务  $t_i$  对资源节点  $r_u$  的身份可靠性满意度可由下式定义的隶属度函数获得:

$$SV(t_i, r_u) = \begin{cases} 1, & \text{if } TS_{t_i} \leq SL_{r_u} \\ 1 - \frac{TS_{t_i} - SL_{r_u}}{\max\_level - SL_{r_u}}, & \text{else} \end{cases} \quad (5)$$

在公式(5)中,  $TS_{t_i}$  为网格任务  $t_i$  的可靠性安全需求,  $SL_{r_u}$  为网格资源节点  $r_u$  的身份可靠性安全水平值.

**定义 4(行为可信性隶属度).** 行为可信性隶属度表征网格任务对资源节点的历史行为表现特征的满意程度. 网格任务  $t_i$  对资源节点  $r_u$  的行为可信性满意度隶属度函数可参照文献[19]定义如下:

$$RV(t_i, r_u) = \begin{cases} 1, & \text{if } TR_{t_i} \leq RS_{r_u} \\ 1 - \frac{\exp(-(RS_{r_u} + 1 - TR_{t_i}))}{\exp(-(1 - TR_{t_i}))}, & \text{else} \\ \frac{1 - \frac{\exp(-1)}{\exp(-(1 - TR_{t_i}))}}{1 - \frac{\exp(-1)}{\exp(-(1 - TR_{t_i}))}}, & \end{cases} \quad (6)$$

在公式(6)中,  $TR_{t_i}$  为网格任务  $t_i$  的对资源节点行为可信性的安全需求,  $RS_{r_u}$  为网格资源节点  $r_u$  的外部行为信誉度安全水平值.

**定义 5(安全满意度).** 网格任务调度安全满意度是指网格任务对向其提供服务资源节点的安全属性满意程度, 其值由隶属度函数获得的综合安全效益值表征. 网格任务  $t_i$  被分配到资源节点  $r_u$  上执行获得的综合安全

效益值定义为

$$\left. \begin{aligned} Sec(t_i, r_u) &= \omega_1 SV(t_i, r_u) + \omega_2 RV(t_i, r_u) \\ \text{s.t. } 0 &\leq \omega_1, \omega_2 \leq 1 \\ \omega_1 + \omega_2 &= 1 \end{aligned} \right\} \quad (7)$$

在公式(7)中,  $\omega_1, \omega_2$  分别代表任务  $t_i$  在节点  $r_u$  上获得的身份可靠性和行为可信性满意度的权重,  $SV(t_i, r_u)$  和  $RV(t_i, r_u)$  的值可分别由公式(5)和公式(6)获得。

**定义 6(安全融合模型).** 安全融合模型定义为一个四元组  $(R, T, A, S)$ , 其中  $R$  代表网格资源节点集合即服务提供者,  $T$  代表应用任务集合即服务需求者,  $A$  代表安全属性集合即包括任务调度的身份可靠性和行为可信性,  $S$  代表安全效益函数集合(任务需求满意度)即包括身份可靠性隶属度函数和行为可信性隶属度函数. 在网络安全融合模型中, 综合考虑了网格资源节点的固有安全性和行为安全性, 采用安全效益函数使任务安全需求和资源节点安全属性间关系得以确立, 将调度机制和安全机制较好地融合起来。

#### 4 双目标优化的网格任务调度新模型

网格任务分配调度策略可由任务-资源映射图表示, 如图 2 所示. 其中, 图的顶点表示任务-资源的映射对, 顶点的权值为任务的计算量、安全需求以及相应资源节点的计算能力和安全水平, 边表示任务之间的约束及通信关系, 边的权值为具有约束关系相邻任务间的通信量、通信安全需求及对应资源节点间的单位通信能力和链路安全性。

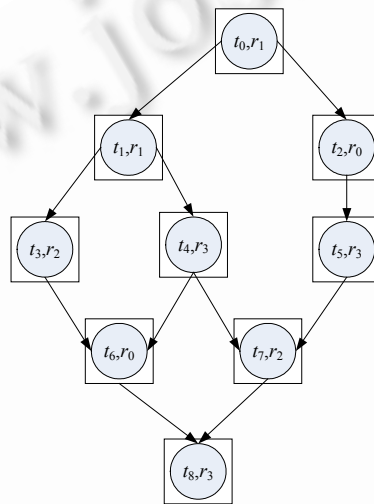


Fig.2 Model of grid task-resource distribution

图 2 网格任务-资源分配模型

**定义 7(网格任务-资源分配图表示模型).** 网格任务资源分配映射图可以用一个四元组  $T-RAG$  表示:  $T-RAG=(V_L, E_L, W_L, U_L)$ . 其中,

- $V_L$  为任务-资源分配图的顶点集合  $V_L = \{l_0, l_1, \dots, l_{n-1}\}, l_x = \{(t_i, r_u) | 0 \leq i < n, 0 \leq u < m, r_0, r_1, \dots, r_{m-1}\}, (t_i, r_u)$  表示任务  $t_i$  映射到网格资源节点  $r_u$  上执行.
- $E_L$  为任务-资源分配图边的集合  $E_L = \{e_{l_x, l_y} | l_x = (t_i, r_u), l_y = (t_j, r_v) \text{ and } x \neq y\}$ , 表示资源之间具有通信关系的集合,  $e_{l_x, l_y}$  表示网格资源  $r_u$  上的任务  $t_i$  和资源  $r_v$  上的任务  $t_j$  之间的通信关系.
- $W_L$  为任务-资源分配图顶点权值集合,  $W_L = (W_T, W_R)$  为一个二元组,  $W_T$  和  $W_R$  分别为任务图和资源拓扑结构图中相应顶点权值.

- $U_L$  为任务-资源分配图边权值集合,  $U_L=(U_T, U_R)$  也是一个二元组, 表示图中顶点  $l_x=(t_i, r_u)$  和顶点  $l_y=(t_j, r_v)$  连接边上的权值集合.

**定义 8(任务调度系统安全效益值).** 网络任务调度系统获得的安全效益值表征网格应用在资源节点上调度的安全满意程度, 定义为任务资源分配图中每个任务获得安全效益值的平均值即系统安全效益值, 可由下式获得:

$$S(G_K) = \frac{\sum_{i=0}^{n-1} \sum_{u=0}^{m-1} x_{iu} Sec(t_i, r_u)}{N} \quad (8)$$

其中,  $Sec(t_i, r_u)$  可由公式(7)得到, 为任务  $t_i$  在网格节点  $r_u$  上获得的安全效益值;  $x_{iu}$  为决策变量, 其定义如下:

$$x_{iu} = \begin{cases} 1, & \text{任务 } t_i \text{ 分配到节点 } r_u \text{ 上执行} \\ 0, & \text{否则} \end{cases} \quad (9)$$

**定义 9(任务调度长度).** 网络任务调度长度为任务-资源分配图模型中从入口节点到出口节点最大路径的长度:

$$L(G_K) = \max \{L(P_S)\} = \max \left\{ \sum_{V_S \in V_L} L(V_S) + \sum_{E_S \in E_L} L(E_S) \right\} \quad (10)$$

在公式(10)中:

- $L(P_S)$  表示图  $T-RAG$  (简称图  $G$ ) 中任意某路径  $P_S$  从入口节点  $S_e$  到出口节点  $S_o$  的时间花费.
- $L(E_S)$  为路径  $P_S$  中间的数据通信时间, 若任务  $t_i, t_j$  分别分配到资源  $r_u$  和  $r_v$ , 则有  $L(E_S) = c(t_i, t_j) / d(r_u, r_v)$ ,  $c(t_i, t_j)$  为任务  $t_i$  和  $t_j$  之间数据传输量,  $d(r_u, r_v)$  为资源  $r_u$  和  $r_v$  之间的通信能力.
- $L(V_S)$  为路径  $P_S$  中某一节点的时间花费, 若任务  $t_i$  分配网格资源节点  $r_u$  上, 则有  $L(V_S) = a(t_i) / b(r_u) + ST(t_i, r_u)$ ,  $ST(t_i, r_u)$  为任务  $t_i$  在网格节点  $r_u$  上获得安全效益值的时间花费, 其可由公式(11)得到:

$$ST(t_i, r_u) = \sum_{k \in \{a, e, g\}} ST(SV^k(t_i, r_u)) \quad (11)$$

其中, 公式(11)的计算方法可查阅我们在文献[30]中所做的工作.

**定义 10(双目标优化调度模型).** 网络任务的双目标优化调度模型是由需求模型、网格资源拓扑结构模型、安全模型及分配调度方案组成, 可表示为一个四元组  $(T, R, S, MAP)$ , 其中,  $MAP$  为分配调度方案集合, 即  $MAP = \{map_1, map_2, \dots, map_n\}$ . 分配调度方案为二元组  $map = (a, \varphi)$ , 其中,  $a$  表示将网格应用任务  $T$  中  $n$  个任务映射到资源节点  $R$  上的分配方案(可由网格-任务资源分配图表示),  $\varphi$  表示单个网格节点  $r_u$  上的调度方案. 那么, 异构网格系统下任务调度的目标就是寻找任务需求模型与资源拓扑结构模型之间的映射调度方案, 在满足任务间依赖关系的约束下, 使网格任务的调度长度和获得的安全效益达到最优.

其数学模型表示如下:

$$\left. \begin{aligned} \min : & \{L(G_K), 1 - S(G_K)\} \\ \text{s.t. } & \sum_{u=0}^{m-1} x_{iu} = 1, i = 0, 1, \dots, n-1 \\ & f(t_j, r_v) - f(t_i, r_u) \geq \frac{c(t_i, t_j)}{d(r_u, r_v)} + \frac{a(t_j)}{b(r_v)} + ST(t_j, r_v) \end{aligned} \right\} \quad (12)$$

其中,  $L(G_K)$  和  $S(G_K)$  为网格任务的调度长度和安全满意度, 可分别由公式(8)和公式(10)获得; 第 1 个约束条件保证每个网格任务有且仅分配到 1 个资源节点上完成, 第 2 个约束条件满足任务执行时的依赖关系.

## 5 双目标离散粒子群算法

### 5.1 PSO的进化方程

标准粒子群算法作为一种基于群智能的全局优化进化算法, 已成功应用到不同领域, 其进化方程为



$$V_i^{t+1} = \omega V_i^t + c_1 \text{rand}(pbest^t - X_i^t) + c_2 \text{rand}(gbest^t - X_i^t) \quad (13)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (14)$$

其中,  $V$  为粒子的速度,  $\omega$  为惯性权重,  $c_1$  和  $c_2$  为学习因子,  $\text{rand}$  为  $[0,1]$  上的随机数. 粒子在第  $t$  代(即一个候选解)的位置为一个  $n$  维向量  $X_i^t = (x_{i1}, x_{i2}, \dots, x_{im})$ . 标准 PSO 通过公式(13)和公式(14)不断迭代, 希望  $X_i^t$  能够趋向最优解. 但公式(13)和公式(14)中的各量均为实数运算, 而调度问题的解应为整数, 因此标准 PSO 方法不能直接用于调度问题. 由于 PSO 方法的具有操作简单、易于实现及收敛速度快和分布式运算等优点, 为了利用 PSO 求解调度问题, 必须对其进行修改. 本文针对网格任务调度问题的具体特点, 结合 PSO 的基本原理, 对算法部分做了如下工作: (1) 基于调度实际问题的离散空间特征进行了粒子位置编码与解码; (2) 对粒子的速度和运算规则重新进行了定义; (3) 基于均匀分布和浓度设计了新的多目标选择算子.

## 5.2 编 码

任务调度问题的染色体编码形式有很多种, 本文基于调度实际问题的离散空间特征使用间接编码方式, 即对每个子任务占用网格资源编码. 如图3所示为  $n$  个网格任务在  $m$  个网格资源节点上的一个粒子位置编码表示.

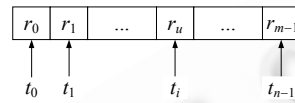


Fig.3 Sketch diagram of grid task-resource coding

图3 网格任务-资源编码示意图

在图3中,  $t_i$  表示任务的编号,  $r_u$  表示任务  $t_i$  被分配到的资源编号. 在产生初始粒子群时, 每一个粒子中的资源编号是随机产生的, 经过粒子方程进化迭代后, 任务  $t_i$  可能占用任何一个可用的网格资源, 所以最优解一定对应一个粒子编码. 此编码表示任务调度方案简单清晰, 但是已有 PSO 的进化方程不能用于上述的编码方式. 为了利用此编码方式, 必须构造新的粒子进化方程.

## 5.3 解 码

在进化产生一个粒子编码后, 还必须对其解码. 由于任务之间依赖约束关系的存在, 对于映射到同一网格资源上的任务必须考虑子任务之间的逻辑关系, 否则可能会形成任务之间的长时间等待, 甚至发生死锁现象. 因此, 解码时必须满足任务 DAG 图中规定的逻辑关系. 对于分配到了同一个网格资源上的子任务, 遵循深度值优先顺序, 深度值越小, 越优先执行; 深度值相同时, 关联耦合强度大的优先执行, 若关联耦合度相同则随机选择执行顺序. 深度值排序和关联度排序的描述如下:

### 5.3.1 深度值排序

在解码生成的任务-资源分配图中, 若某些任务的执行顺序不能满足任务图中任务的依赖关系, 因而形成死锁, 这样的图称为非法图.

基于深度值的排序<sup>[31]</sup>可以避免映射到同一网格资源上的任务之间出现死锁现象, 深度值计算公式如下:

$$\text{level}(t_i) = \begin{cases} 0, & t_i \text{ 无父节点} \\ 1 + \max(\text{level}(\text{parent}(t_i))), & \text{其他} \end{cases} \quad (15)$$

其中,  $\text{parent}(t_i)$  返回的是子任务  $t_i$  父节点集合,  $\max(\text{level}(\text{parent}(t_i)))$  返回的是  $t_i$  父节点集合中具有最大深度值的父节点深度值. 获取层次深度信息后, 对在相同网格资源上运行的任务序列, 就可以按层次深度值由小到大排序, 这样优先级越高的任务优先获得资源.

### 5.3.2 关联度排序

对于深度值相同的子任务, 它们之间不存在依赖关系, 可以并行执行. 如果以任意次序执行, 则可能造成堵塞现象, 将达不到最优效果. 基于关联耦合强度排序<sup>[32]</sup>可以较好地解决堵塞问题, 缩短关键路径长度. 任务图节点耦合强度定义如下:

**定义 11(任务关联耦合强度).** 在任务图 DAG 中,任务  $t_i$  的关联耦合强度  $CS(t_i)$  为其所在路径的向前关联耦合强度和向后关联耦合强度之和,计算公式为

$$CS(t_i) = FCS(t_i) + BCS(t_i) = |Pre(t_i)| + |Sub(t_i)| \quad (16)$$

其中,  $FCS(t_i)$  和  $BCS(t_i)$  分别表示  $t_i$  的向前关联耦合强度和向后关联耦合强度,  $Pre(t_i)$  和  $Sub(t_i)$  分别为  $t_i$  的前驱任务集合和后继集合.

#### 5.4 粒子进化方程的重新定义

粒子的位置用  $n$  维向量  $X=(x_1, \dots, x_i, \dots, x_n)$  定义表示,其中,  $1 \leq x_i \leq m$  表示第  $i$  个任务分配到了节点  $x_i$  上.由于当前粒子与最优粒子位置难以用定量的方式表示,导致粒子的速度和位置不能参照标准粒子群进化方程更新.

**定义 12.** 两个位置相减的结果为速度  $V$ ,使用符号“ $\ominus$ ”表示新的减法运算.个体极值  $pbest_i$  和全局极值  $gbest$  对当前粒子位置的影响程度用速度表示,其定义为

$$V_1 = pbest_i \ominus X_i = c_1 \times h(pbest_i, X_i) \quad (17)$$

$$V_2 = gbest \ominus X_i = c_2 \times h(gbest, X_i) \quad (18)$$

其中,  $c_1, c_2$  为影响因子(也称学习因子);函数  $h(X_i, pbest_i)$  为  $X_i$  对  $pbest_i$  学习操作,其  $X_i$  向  $pbest_i$  第  $j$  维分量学习过程为

$$h(x_{i,j}, pbest_{i,j}) = \begin{cases} pbest_{i,j}, & \text{if } p_1 \leq rand(\cdot) \\ x_{i,j}, & \text{else} \end{cases} \quad (19)$$

其中,  $p_1$  为个体极值  $pbest_i$  对当前粒子影响程度概率参数,  $rand(\cdot)$  为  $[0, 1]$  间的随机数.  $X_i$  对全局极值  $gbest$  的学习操作类似.

**定义 13.** 位置与速度的加法运算实现了粒子位置的移动,使粒子进入了一个新的位置,使用符号  $\oplus$  表示新的加法运算,其定义为

$$X' = X \oplus V_j = (x_{i1} \oplus v_{j1}, \dots, x_{ij} \oplus v_{jj}, \dots, x_{in} \oplus v_{jn}) \quad (20)$$

新位置的任意一维  $x'_{ij}$  的值由公式(21)获得:

$$x'_{ij} = x_{ij} \oplus v_{jj} = swap(x_{ij}, v_{jj}) \quad (21)$$

由于离散量运算的特殊性,惯性的主要作用是产生扰动,旨在增加粒子种群的多样性,避免算法早熟及局部收敛.本文引入了一个均匀扰动速度,即依次以等概率从  $[1, m]$  中随机产生一个  $v'_i \in [1, m]$ ,连续执行  $n$  次,得到一个新的速度  $V_{even} = (v'_1, \dots, v'_i, \dots, v'_n)$  作为扰动速度.该扰动速度有利于保持种群的多样性,可以有效防止算法陷入局部最优,而且该扰动速度具有简单、易执行的特点.

**定义 14.** 为了使速度之间的作用不相互干扰,采用分步计算速度和修改粒子位置的方式,更新后的粒子进化方程定义为

$$\begin{cases} X = X \oplus V_{even} \\ X = X \oplus V_1 = X \oplus c_1 \times h(pbest, X) \\ X = X \oplus V_2 = X \oplus c_2 \times h(gbest, X) \end{cases} \quad (22)$$

#### 5.5 基于均匀分布和浓度的选择算子

由于双目标优化问题并无单个最优解,不能像单目标优化问题那样直接确定  $Gbest$ ,而  $Gbest$  在引导整个种群朝着 Pareto 前沿的进化过程中发挥着重要作用.本节对群体的全局最优位置  $Gbest$  提出了一种基于均匀分布向量的选取方法.

**算法 1.** 基于均匀分布向量的  $Gbest$  选择算子.

步骤 1. 按照文献[33]中的方法产生  $q$  个在目标空间中均匀分布的向量  $W_1, W_2, \dots, W_q$ ,其中,  $W_i = (w_{i1}, w_{i2}, \dots, w_{im})$ .具体步骤如下:首先产生一个  $q \times m$  矩阵:

$$U(q, m, b) = (a_{ij}) = \begin{bmatrix} 1 & b & \dots & b^{m-1} \\ 2 & 2b & \dots & 2b^{m-1} \\ \vdots & \vdots & & \vdots \\ q & qb & \dots & qb^{m-1} \end{bmatrix} \pmod{q} \quad (23)$$

其中,  $q \geq m+1$ ,  $q$  为素数. 公式(23)右端表示对矩阵每个元素求以  $q$  为模的同余数所得的矩阵,  $1 < b < q$ . 在本节中令  $q=11$ , 文献[34]中给出了对应的  $b$  值, 则可以在  $C^m$  上产生均匀分布的点集  $P^* = \{X_1, \dots, X_q\}$ , 其中,  $X_i = \{x_{i1}, \dots, x_{im}\}$ ,  $x_{ij} = \frac{2a_{ij} + 1}{2q}$ ,  $i=1 \sim q, j=1 \sim m$ . 每个点可定义一个权向量  $W_i = \frac{X_i}{\|X_i\|_1}$ ,  $\|X_i\|_1 = \sum_{j=1}^m x_{ij}$ , 于是得权向量集合:

$$W^* = \{W_1, \dots, W_q\}.$$

步骤 2. 求出每一个向量  $W_i \in W^*$  所对应的区域  $R(W_i) = \{f(X) \in FI | \text{ang}(f(X), W_i) = \min\{\text{ang}(f(X), W_k) | W_k \in W^*\}\}$ ,  $FI$  为当前所求得 Pareto 前沿.

步骤 3. 计算每一个区域元素的个数, 找出包含元素个数最少的那个区域, 在其中任意挑选一个向量, 记为  $f(Gbest)$ , 则  $Gbest = \text{arg}f(Gbest)$ .

对于每个粒子的个体极值  $Pbest$  采取如下比较准则进行选择确定: 1) 运用公式(8)和公式(10)计算每个粒子适应度, 若粒子  $X_i$  优于  $X_j$  则选择  $X_i$ ; 2) 若两个粒子无支配关系, 即粒子  $X_i$  与  $X_j$  之间通过 Pareto 优先关系无法比较, 则基于粒子浓度运用反赌轮选择算子进行选择, 浓度小的即稀疏区域的粒子被选择的概率较高. 粒子浓度的计算方法如下:

**算法 2.** 粒子浓度计算方法.

步骤 1. 令  $I = \{X_1, X_2, \dots, X_N\}$  为到第  $t$  代为止所发现的非劣解集, 将集合  $I$  中的每个个体初始浓度值设为  $\text{crowd}(X_i) = 0$ ;

步骤 2. for 每一个目标  $j \in \{1, 2\}$

$I = \text{sort}(I, j)$  //对  $I$  中所有个体按照第  $j$  个目标函数值由小到大排序//

for  $i=2$  to  $N-1$

$$\text{crowd}(X_i) = \text{crowd}(X_i) + \frac{I[i+1].j - I[i-1].j}{f_j^{\max} - f_j^{\min}} \quad //I[i].j \text{ 代表集合 } I \text{ 中第 } i \text{ 个个体的第 } j \text{ 个目标函数值,}$$

$$f_j^{\max}, f_j^{\min} \text{ 分别是第 } j \text{ 个目标的最大和最小函数值//}$$

end

$$\bar{d} = \frac{\sum_{i=1}^{N-1} I[i+1].j - I[i].j}{N-1}$$

$$\text{crowd}(X_1) = \text{crowd}(X_1) + \frac{I[2].j - I[1].j + \bar{d}}{f_j^{\max} - f_j^{\min}}$$

$$\text{crowd}(X_N) = \text{crowd}(X_N) + \frac{I[N].j - I[N-1].j + \bar{d}}{f_j^{\max} - f_j^{\min}}$$

End

## 5.6 离散粒子群算法流程

**算法 3.** 双目标优化的网格任务调度离散粒子群算法 DPSO.

步骤 1. 选择适当的参数, 给定粒子群的种群规模  $pop$  及最大进化代数  $\text{Maxgen}$ , 随机初始化粒子群中粒子的位置与速度, 产生初始群体  $pop(t), t=0$ , 运用公式(8)和公式(10)作为适应度函数找出初始群体的 Pareto 最优解, 将其存入外部存储器  $I$  中.

步骤 2. 将粒子的  $Pbest$  设置为当前位置, 运用算法 1 在外部存储器  $I$  中选取  $Gbest$ .

步骤 3. 对离散粒子群中的所有粒子, 执行如下操作:

(a) 产生一个均匀扰动速度  $V_{\text{even}}$ , 将其作用于当前粒子;

(b) 运用粒子进化方程式(20)更新粒子的速度和位置,得到进化后群体  $pop(t+1)$ .

步骤 4. 用  $pop(t+1)$ 更新外部存储器  $I$ .

步骤 5. 对  $pop(t+1)$ 中所有粒子,按照第 5.5 节中个体极值  $Pbest$  比较准则更新每个粒子的个体最优值;

步骤 6. 判断算法终止条件是否满足,如果满足,则转向步骤 7;否则,转向步骤 3.

步骤 7. 输出外部集合  $I$  中的所有粒子作为双目标优化网格任务调度问题的 Pareto 最优解,算法停止.

### 5.7 算法收敛性分析

为了方便 DPSO 算法的收敛性分析,首先引入如下重要概念:

**定义 15.** 称粒子  $b$  是从粒子  $a$  通过进化可达的,如果

$$P\{EG(a)=b\}>0 \quad (24)$$

其中,  $EG(a)$ 表示由粒子  $a$  通过进化所产生的粒子,  $P\{\cdot\}$ 表示随机事件  $\{\cdot\}$ 发生的概率.

**引理 1.** 若一种双目标进化算法满足如下两个条件<sup>[35]</sup>,则该双目标算法以概率 1 收敛到双目标优化问题的全局最优解集  $I_{true}$ :

(a) 对可行域  $S$  中任两个粒子  $a$  与  $b$ ,  $b$  是由  $a$  通过进化过程可达的;

(b) 种群序列  $I(t)$  是单调的,即对  $\forall t, I(t+1)$  中的任意解优于  $I(t)$  中的任意解或者至少不差于  $I(t)$  中的解.

**定理 1(全局收敛性).** DPSO 算法以概率 1 收敛到双目标优化问题的全局最优解集.

证明:(1) DPSO 算法中,任意两个粒子  $a$  与  $b$ ,  $b$  是由  $a$  通过粒子进化可达的.

设粒子  $c=(c_1, c_2, \dots, c_n)$  是由  $a=(a_1, a_2, \dots, a_n)$  通过均匀扰动速度  $V_{even}$  作用后产生的任一后代,由均匀扰动的过程可知,从  $a_i$  产生  $c_i$  的概率为  $\frac{1}{m-1}$ , 则

$$Prob\{E(a)=c\} = \frac{1}{m-1} \cdot \frac{1}{m-1} \cdot \dots \cdot \frac{1}{m-1} = \frac{1}{(m-1)^n} > 0 \quad (25)$$

同时,由于粒子参与进化的概率  $p_g$  为 1, 则粒子  $a$  通过粒子位置更新产生粒子  $b$  的概率为

$$Prob\{EG(a)=b\} \geq Prob\{E(a)=c\} \times Prob\{G(c)=b\} = \frac{p_g}{(m-1)^n} > 0 \quad (26)$$

即可行域  $S$  中任意两粒子  $a$  与  $b$ ,  $b$  是由  $a$  通过进化可达的.

(2) 再证  $I(0), I(1), \dots, I(t), \dots$  是单调的.事实上,由 DPSO 算法的选择策略和算法步骤 4 的更新过程可知,算法产生的序列  $\forall t, I(t+1)$  中任意解不劣于  $I(t)$  中的解,或改进了  $I(t)$  中的解(至少不劣于  $I(t)$  中的解),于是可得  $I(0), I(1), \dots, I(t), \dots$  是单调的.

综合(1)、(2)可知,DPSO 算法以概率 1 收敛到双目标网格任务调度问题的全局最优解集.  $\square$

## 6 仿真实验与性能分析

为了验证本文提出的离散粒子群进化算法 DPSO 的有效性,在不同的仿真参数和性能指标下将其与同样基于进化理论的遗传算法(genetic algorithm,简称 GA)<sup>[30,36]</sup>以及蚁群算法(ant colony optimization,简称 ACO)<sup>[37]</sup>进行了综合对比分析.

### 6.1 实验环境和参数设置

为了便于对比实验,在 Windows 环境下,基于 c++语言设计开发了一个由仿真实验环境产生模块、调度算法库模块及性能评测模块等部分组成网格任务调度模拟仿真平台.基于该平台,首先对网格任务及资源环境进行模拟,然后调用算法库中的算法生成调度方案,最后通过性能评测模块从算法执行效率(收敛性)和任务调度的性能指标(调度长度和安全性)进行衡量.

仿真平台中的实验环境产生模块由组件 GridGenerator 和组件 TaskGenerator 构成,其中,组件 GridGenerator 负责模拟网格环境,而组件 TaskGenerator 能够产生属性不同的异构任务.

网格资源的差异主要体现在可用资源的个数、资源本身处理能力及安全级别、资源间链路带宽及安全可靠性的差异上,因此,组件 GridGenerator 主要设定了如下参数:

(1)  $m$ :网格中可用资源的个数.

(2)  $\alpha$ :资源节点异构性收缩/膨胀因子( $\alpha \geq 1$ ).资源节点异构性体现的是资源节点单位时间处理数据量和安全级别的变化范围:对于一个随机取定的值  $a_1$ ,资源的处理能力取值为 $[a_1/\alpha, a_1\alpha]$ 之间的一个随机数.由于网格环境中大部分资源节点单位时间内处理数据量的大小各异,但也在一定的变化范围之内,因此可以随机取网格中任意资源节点的单位时间内所处理数据量大小  $a_1$  来进行度量其他节点处理能力;同时,收缩/膨胀因子  $\alpha$  保证了网格环境中所有节点的处理数据量能力在一个确定的范围内.类似地,对于网格资源节点异构性中的安全属性来说,随机取定的一个值  $a_2$ ,资源安全级别的取值为 $[a_2/\alpha, a_2\alpha]$ 之间的一个随机数.这种取值方法不仅充分考虑到资源节点间异构性会根据收缩/膨胀因子不同而变化,同时又保证了任一网格资源节点单位时间内处理数据量及资源节点安全级别在一个合理的范围内.

(3)  $\beta$ :网格链路异构性收缩/膨胀因子( $\beta \geq 1$ ).网格链路异构性体现的是网格中资源连接带宽和链路安全性的变化范围:对于一个随机取定的值  $b_1$ ,资源之间的网络传输带宽的取值为 $[b_1/\beta, b_1\beta]$ 之间的一个随机数;对于一个随机取定的值  $b_2$ ,资源间链路的安全级别的取值为 $[b_2/\beta, b_2\beta]$ 之间的一个随机数.与资源节点异构性类似,网格环境中任意资源节点间的连接带宽大部分不太一样,这样可以随机取两个节点间的连接带宽即单位时间传输的数据量  $b_1$  来度量其他链路带宽;同时,收缩/膨胀因子  $\beta$  保证了网格环境中所有链路单位时间内的传输处理数据量在一个确定的范围内.此外,资源节点间的链路安全级别中的随机数  $b_2$  的合理性分析类似,这里不再赘述.

组件 TaskGenerator 根据网格应用的任务数、任务的大小及通信计算比等特性可得到属性各异的任务图,其主要参数设定如下:

(1)  $n$ :任务图中需要完成的任务个数.

(2)  $\chi$ :任务图形状控制参数.任务图的高度  $H$  由  $\sqrt{n}/\chi$  的正态分布随机生成并取整运算获得,图中每一层中任务的个数由  $\chi \times \sqrt{n}$  正态分布随机生成并取整运算获得.在实际网格任务应用中,对具有  $n$  个子任务的任务图大部分高度适中,处于扁平化和线性形状的任务图较少;而且一般来讲,其形状越扁平或线性特征越明显,其个数就越少.其分布特征基本服从以平均高度为中心、左边若为扁平分布则其越扁平越少,右边若为线性分布则其线性特征越明显其个数越少,符合正态分布的基本特征.因此,采用正态随机生成是合理的.另外,为保证生成的任务图仅具有 1 个入口节点和 1 个出口节点,对图进行如下处理:若入度为 0 的节点数目多于 1 个,则添加一个节点与所有入度为 0 的节点相连,该节点记为开始节点;若出度为 0 的节点数目多于 1 个,则添加一个节点与所有出度为 0 的节点相连,该节点记为结束节点.

(3)  $\delta$ :通信计算比.通信计算比定义为整个任务图中数据传输量与计算量的比值, $\delta$ 越大,说明任务间交互通信量越大;反之,说明任务计算量越大,即任务类型更接近计算密集型.

(4)  $\eta$ :任务异构性收缩/膨胀因子( $\eta \geq 1$ ).任务异构性表征的是任务计算量变化范围:对于一个随机取定的值  $c$ ,任务的计算量取值为 $[c/\eta, c\eta]$ 之间的一个随机数.在网格复杂的应用中,可能面临不同类型的应用,应用类型决定了要处理应用任务数量级,这样可以依据其类型特征随机定义基准任务大小  $c$  的取值.对同构的任务来讲,收缩/膨胀因子  $\eta$  取为 1 即可;对于异构任务,其异构性的调整主要由收缩/膨胀因子  $\eta$  进行控制.

对本文中所进行的实验设定如表 2 所示的参数,通过变换这些参数可以模拟不同的网格环境及网格任务.

Table 2 Parameter setting of simulation experiments

表 2 仿真实验参数设定

Parameter type	Parameter value
$m$	100,150,200,...,500
$\alpha$	1,2,3,4,5,6,7,8,9,10
$\beta$	1,2,3,4,5,6,7,8,9,10
$n$	1 000,1 500,2 000,...,5 000
$\chi$	0.5,1.0,2.0
$\delta$	0.1,0.5,1,5,10
$\eta$	1,2,3,4,5,6,7,8,9,10

## 6.2 实验结果与分析

本节首先从算法收敛性、执行效率、任务调度长度、安全性等性能指标分别对本文提出的 DPSO 算法与遗传算法 GA 及蚁群算法 ACO 进行了实验分析比较,然后从调度长度和安全性两个目标同时与 GA 及 ACO 进行了仿真实验及综合分析.

### 实验 1. 算法收敛性能及执行计算时间的比较

实验的目的是为了验证算法的收敛性能和执行效率,对实验参数  $m=200, \alpha=\beta=5, n=3000, \chi=1, \delta=1$  和  $\eta=5$  时的实验结果如图 4 所示.

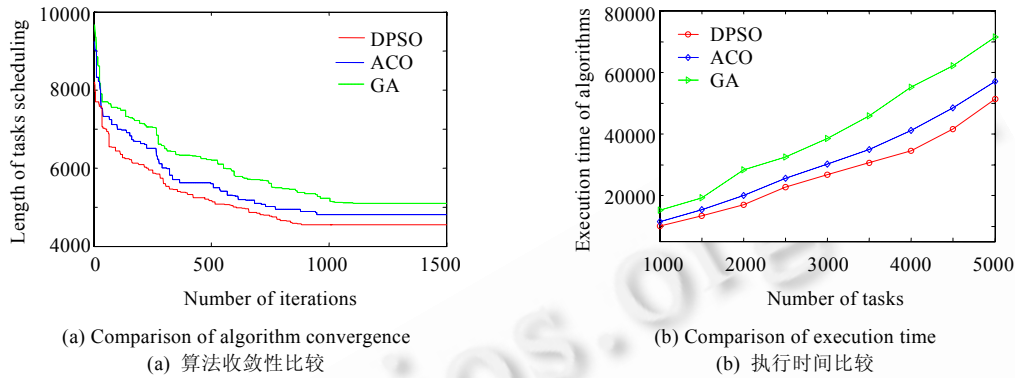


Fig.4 Comparison results of algorithm convergence and execution time

图 4 算法收敛性及执行时间结果比较图

从图 4(a)可以看出,在 1 500 次迭代过程中的调度长度变化曲线趋势中,粒子群 DPSO 算法和 ACO 算法比遗传算法 GA 具有更好的收敛性,而本文 DPSO 在收敛性方面要优于 ACO;从图 4(b)可以看出,随着问题规模的扩大,每个算法在找到最优解时所花的时间都呈线性增长.但在同等情况下,DPSO 和 ACO 算法明显优于 GA,并且随着问题规模的扩大,其优势更加明显.因此,无论是在收敛性还是在算法执行效率方面,本文的 DPSO 算法都明显占优.这主要是因为本文提出的算法是根据调度问题离散空间的实际特征进行设计的,解决了过去基于连续空间映射方式求解调度离散问题带来的冗余问题;同时,粒子具有记忆特性,通过“自我”学习和向“他人”学习,下一代可以从“先辈”那里继承更多信息,从而能够在较短时间内找到最优解.

### 实验 2. 网格环境不变,调度任务数、通信/计算比及任务异构性的变化情况下性能比较

本实验的目的是为了比较在网格资源的节点性能(节点计算能力和安全级别)和网络性能(网络带宽和链路安全可靠)相同的情况下,网格应用任务数、通信计算比和任务异构性变化分别对调度结果性能的影响.网格环境实验参数  $m=200, \alpha=\beta=5$ ;任务图实验参数变化范围为  $n=1000, 1500, 2000, \dots, 5000, \delta=0.1, 0.5, 1, 5, 10, \eta=1, 2, \dots, 10$ ,结果如图 5 所示.

在网格环境稳定情况下,任务图通信计算比  $\delta=1$ ,任务异构性  $\eta=5$ .从图 5(a)可以看出,随着任务数的增加,任务调度长度都趋于增加,其中,DPSO 和 ACO 优于遗传算法 GA;特别是随着问题规模的变大,其优势更明显,这说明 DPSO 更适合大规模的任务调度情形.从图 5(b)可以看出,在任务数较少时,DPSO 获得安全满意度稍占优,三者相差不多;但随着任务的增多,其优势进一步得到体现.同时,GA 获得的安全满意度对任务数改变比较敏感,其安全值下降较为明显.在其他实验仿真参数不变的情况下改变任务图的通信计算比,从图 5(c)可以看出,随着通信计算比的由小增大,每个算法的调度长度都有所增加.GA 在调度通信密集型任务时调度长度增加相当明显;而 DPSO 和 ACO 随着通信计算比的增加,其调度长度相对较缓增大,说明二者更适合通信密集型的调度任务,但在同等条件下 DPSO 性能占优.在其他实验仿真参数不变的情况下改变任务图的任务异构性,从图 5(d)可以看出,随着任务异构性的增大,3 种算法调度长度都呈线性增大.主要由于依赖任务之间关系的存在,使得等待前序任务完成时间更长;但在同等条件下,DPSO 优于 GA 和 ACO.主要因为本文算法在处理任务间约束关系时引

入了深度值和关联度处理方法,从而减少了等待时间.另外,通信计算比和任务异构性变化对安全效益值基本没有影响.

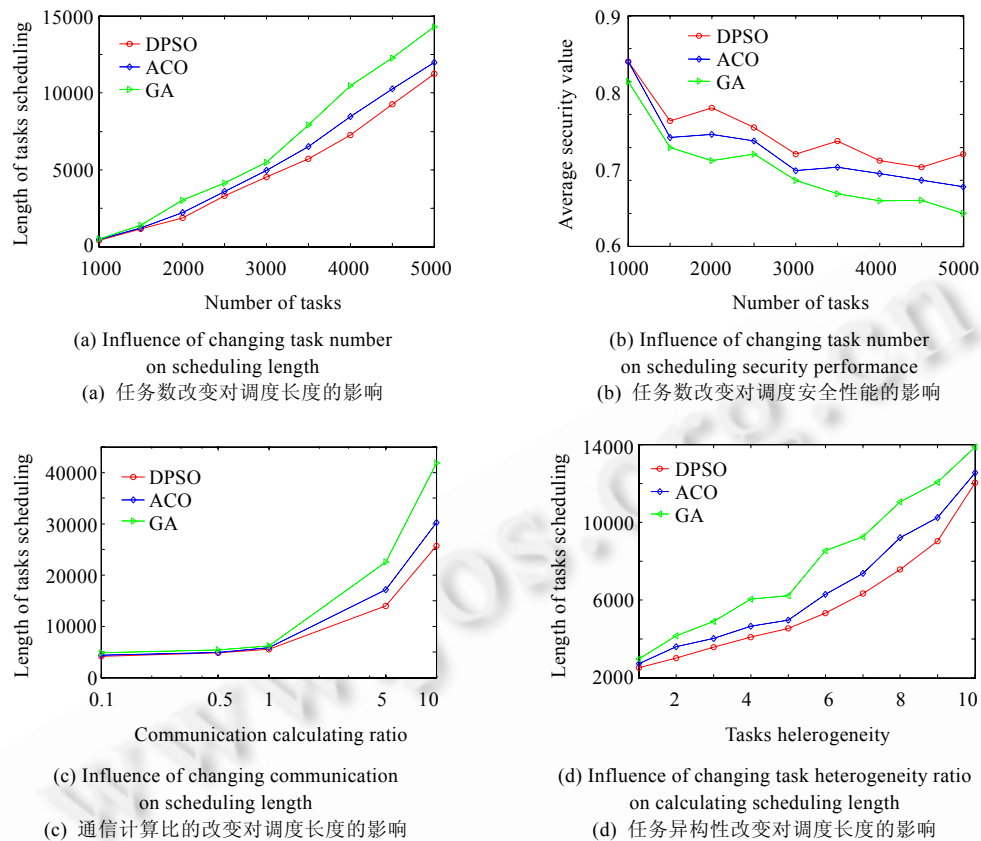


Fig.5 Grid environment fixed, different performance results under the situation of changing tasks number, communication/calculation ratio and task heterogeneity

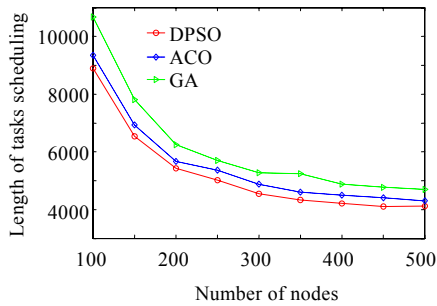
图 5 网格环境不变,改变任务数、通信计算比及任务异构性,不同的性能结果

**实验 3.** 任务图固定,网格环境变化情况下的性能影响比较.

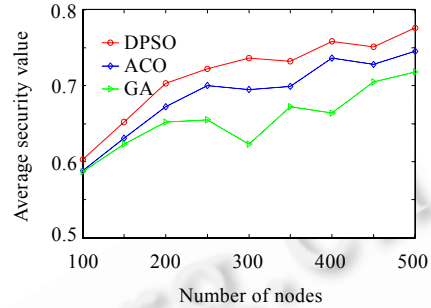
与实验 2 相对应,本实验是为了比较在网格应用任务数、通信计算比和任务异构性不变情况下,改变网格可用资源数及网格资源和链路异构性分别对调度结果的影响.网格应用任务图的实验参数  $n=3000, \chi=1, \delta=1, \eta=5$ ; 网格环境的实验参数变化范围  $m=100, 150, 200, \dots, 500, \alpha=1, 2, \dots, 10, \beta=1, 2, \dots, 10$ , 实验结果如图 6 所示.

在网格应用任务图固定情况下,网格资源节点异构性和链路异构性的收缩/膨胀因子  $\alpha=\beta=5$ .从图 6(a)可以看出,随着节点数的扩充,算法任务调度长度都有所减少.这主要是因为随着节点数的扩充,应用任务可以选择性能更高的节点运行从而减少时间长度;在网格节点数量达到一定程度后,调度长度都趋于稳定.但在同等情况下,DPSO 明显占优.从图 6(b)可以看出,随着节点数的增加,每种算法安全效益值都有所提高.这主要是因为对每个任务来说,可选择的节点数增加,以便其可以调度到安全级别更高的节点上执行;但在同等情况下,DPSO 和 ACO 明显优于 GA.同时,GA 获得的安全满意值对网格节点数改变比较敏感,其稳定性也不如 DPSO 及 ACO.从图 6(c)可以看出,随着网络异构性的增强,对每个网格应用任务来说可选择网格节点性能的范围变大,都尽可能地选择性能高的网格节点.这样使最终总的调度长度缩短,但在相同的网格环境中 DPSO 稍微占优.同样,从图 6(d)可以看出,节点异构性增强使每种算法获得的安全效益值都有所增大.在同等情况下,DPSO 和 ACO 明显优于 GA.特别值得一提的是,GA 获得安全效益值对节点性能比较敏感.随着网络链路异构性的增强,从图 6(e)可以

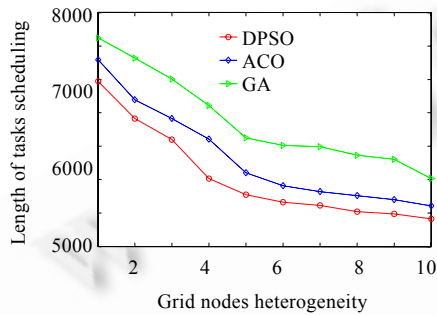
看出,每种算法调度的任务调度长度都有所减少.这主要是因为链路异构性增强后,对那些具有依赖关系的网格应用任务来说,将尽可能选择通信量大的相邻网格节点上执行,整体上缩短了调度长度.同样,由图 6(f)可以看出,随着网络链路异构性的增强,每种算法调度的安全效益值都有所增大;由于 GA 对网络特性比较敏感,其变化幅度比较大.而 DPSO 也有缓慢增大,说明其网络稳定性能更好.这主要因为链路异构性增强后,对每个关联任务数据来讲,其可选择范围变大,可以尽可能地选择安全可靠链路传输,从而提高了整体的安全性能.



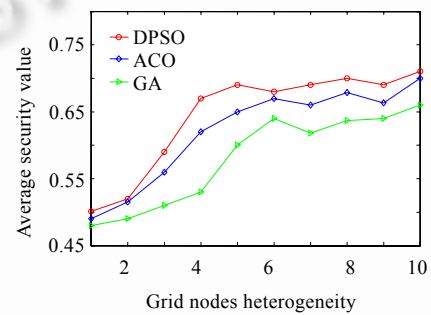
(a) Influence of changing grid node number on scheduling length  
(a) 网格节点数改变对调度长度的影响



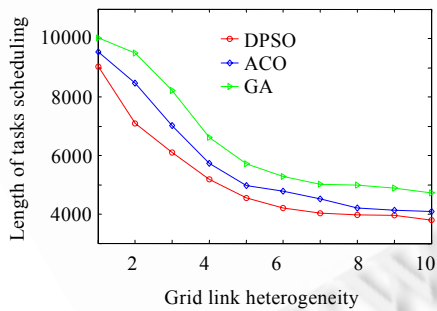
(b) Influence of changing grid node number on scheduling security performance  
(b) 网格节点数改变对安全性能的影响



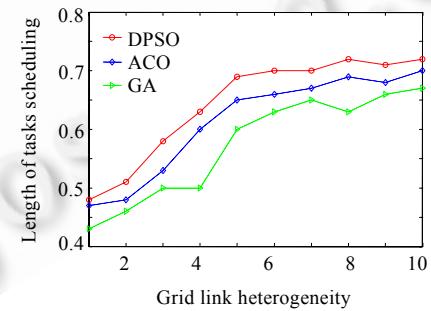
(c) Influence of changing grid node heterogeneity on scheduling length  
(c) 网格节点异构性改变对调度长度的影响



(d) Influence of grid node heterogeneity on security performance  
(d) 网格节点异构性改变对安全性能的影响



(e) Influence of grid link heterogeneity on scheduling length  
(e) 网格链路异构性改变对调度长度的影响



(f) Influence of grid link heterogeneity on security performance  
(f) 网格链路异构性改变对安全性能的影响

Fig.6 Grid task diagram fixed, influence of changing grid environment on different performances

图 6 网格任务图不变,改变网格环境对不同性能的影响



实验 4. 双目标优化的综合性能比较

与前面 3 个实验不同,本实验目的首先是考察算法运行过程中双目标 Pareto 非劣解的变化趋势情况.然后,着重考察在改变不同仿真实验参数下,双目标优化性能非劣解的变化趋势及解的分布情况.本文 DPSO 与 GA 和 ACO 比较的双目标优化综合性能实验结果如图 7 所示.

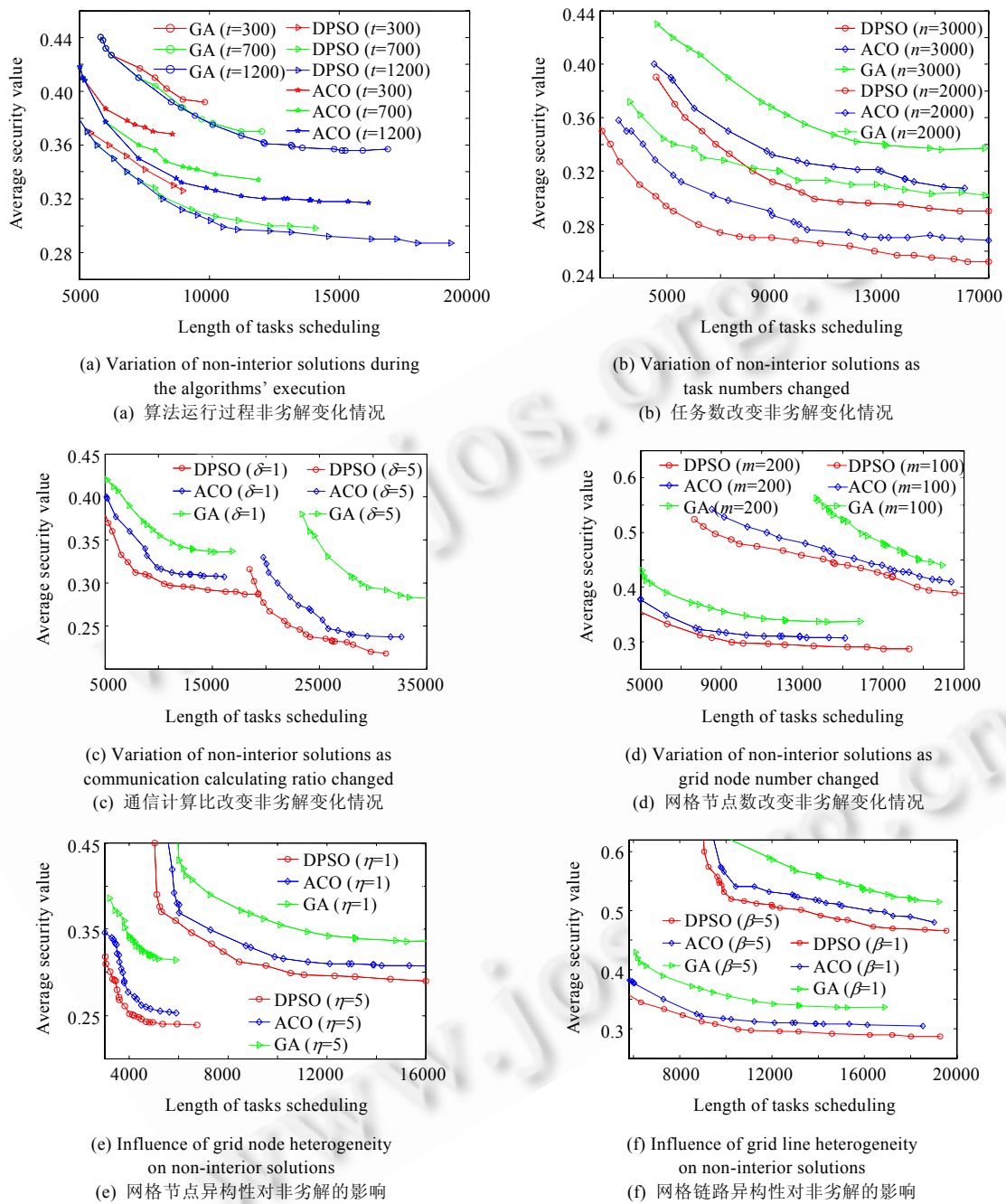


Fig.7 With different parameters changed, variation tendency of double-objective non-inferior solution set

图 7 不同参数改变情况下,双目标非劣解的变化趋势

图 7(a)分别给出了每种算法分别在 300 代、700 代及 1 200 代时的多目标非劣解分布情况.可以看出,随着进化代数的增大,每种算法的 Pareto 界面逐步向最优解方向移动,而且非劣解个数增多且其宽广性也都在增大;在进化代数超过 700 代之后算法趋于收敛,特别是在 1 200 代时,算法基本上已经收敛.这时明显可以看到,本文提出的 DPSO 算法无论是在最优解集质量方面还是分布均匀性方面均明显优于 ACO 和 GA.这主要是因为本文算法一方面基于调度问题的实际特征进行编码和进化方程重新设计,另一方面基于均匀分布向量和粒子浓度定义了新的选择策略.在网格环境及任务异构性等其他仿真实验参数不变的情况下,在将任务数  $n$  由 3 000 降低到 2 000 时,算法 Pareto 最优解集界面变化如图 7(b)所示.可以看出,随着任务数的减少,每种算法的 Pareto 最优解集界面都向原点移动,说明算法获得了更短的调度长度和更大的安全满意度,这与实验 2 单目标考察的结论完全吻合;但在不同的任务数下,DPSO 最优解集占有一定的优势.从图 7(c)可以看出,在其他参数不变的情况下,当网格任务通信计算比 CCR 由 1 变化到 5 时,算法的 Pareto 最优解集质量都有较大降低.其中,GA 降低特别明显,说明在通信密集型任务下其最优解集质量明显低于 ACO 和 DPSO,反之则说明 DPSO 更适合通信密集型网格应用.从图 7(d)可以看出,其他参数不变,当网格节点数  $m$  由 100 增大到 200 时,算法的 Pareto 最优解集界面都向原点移动,且在同等条件下,DPSO 算法由于采用了基于分布向量和粒子浓度的选择策略,其最优解集界面更宽广,非劣解分布更均匀.从图 7(e)和图 7(f)可以看出,在其他仿真实验参数不变的情况下,随着网格节点异构性和链路异构性的增大,算法的 Pareto 最优解集界面都向原点移动,说明调度算法获得的调度长度都有所降低,且安全满意度有所增大,特别是网格节点的异构性对算法 Pareto 最优解集的影响更加明显.但在同等情况下,本文提出的 DPSO 明显优于 GA.总之,从实验 4 的结果可以看出,在改变不同仿真实验参数的情况下,本文提出的 DPSO 算法无论是在最优解集质量方面还是均匀分布性方面都有一定的优势.这主要是因为 DPSO 算法基于调度问题的实际特征进行编码和进化方程的重新设计,充分发挥了粒子群算法在求解组合优化问题的优点;另一方面,因为提出的 DPSO 基于均匀分布向量和粒子浓度的新选择策略可以在进化过程中引导非劣解的进化方向,在进化选择时除了选择较优解集外,还尽可能地选择那些稀疏区域的非劣解,从而保证了收敛时最优解集的宽广性和分布均匀性.

## 7 结束语

本文通过对异构网格任务调度问题进行综合分析,考虑了网格资源节点的固有安全性和行为安全性,分别构建了一个网格资源节点身份可靠性度量函数和行为表现信誉度评估策略;通过定义安全效益函数确立了任务安全需求与资源节点安全属性之间的隶属关系,从而建立了一个网格任务调度的安全融合模型.以此为基础,提出了双目标优化的网格任务调度模型.为了求解该模型,结合网格任务调度问题的具体特点重新定义和设计了新的粒子进化方程;同时,基于均匀分布向量和粒子浓度定义了新的选择策略,从而提出一种双目标优化的网格任务调度粒子群进化算法,并运用概率论的有关知识证明了算法的收敛性.为验证本文 DPSO 算法的优化性能及证明其新颖性,在不同的仿真环境和实验参数下进行了大量的仿真实验,其结果表明,与同类算法相比,该算法在相同条件下具有较快的收敛速度和执行效率,并具有更短的调度长度和获得更高的安全可信值.特别是在多目标优化性能方面,除了最优解集的性能指标质量较好以外,解集的分布更加均匀、宽广.下一步工作将进一步考虑网格应用任务调度的费用花费等多性能指标,以及在网格环境下的动态多目标任务调度模型及算法研究工作.

## References:

- [1] Foster I, Kesselman C, Tuecke S. The anatomy of the grid: Enabling scalable virtual organization. *Int'l Journal of High Performance Computing Applications*, 2001,15(3):200-222. [doi: 10.1177/109434200101500302]
- [2] Chakrabarti A, Damodaran A, Sengupta S. Grid computing security: A taxonomy. *IEEE Security & Privacy*, 2008,6(1):44-51. [doi: 10.1109/MSP.2008.12]
- [3] Xie T, Qin X. Scheduling security-critical real-time applications on clusters. *IEEE Trans. on Computers*, 2006,55(7):864-879. [doi: 10.1109/TC.2006.110]

- [4] Song SS, Hwang K, Kwok YK. Risk-Resilient heuristics and genetic algorithms for security-assured grid job scheduling. *IEEE Trans. on Computers*, 2006,55(6):703–719. [doi: 10.1109/TC.2006.89]
- [5] Vadhiyar SS, Dongarra JJ. A metascheduler for the grid. In: *Proc. of the 11th IEEE Int'l Symp. on High Performance Distributed Computing (HPDC 2002)*. Los Alamitos: IEEE Computer Society Press, 2002. 343–351. [doi: 10.1109/HPDC.2002.1029934]
- [6] Braun TD, Slegel HJ, Becj N, Bölöni LL, Maheswaran M, Reuther AI, Robertson JP, Theys MD, Yao B, Hensgen D, Freun RF. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *Journal of Parallel and Distributed Computing*, 2001,61(6):810–837. [doi: 10.1006/jpdc.2000.1714]
- [7] Yu J, Buyya R. A taxonomy of scientific workflow systems for grid computing. *Sigmod Record*, 2005,34(3):44–49. [doi: 10.1145/1084805.1084814]
- [8] Sih GC, Lee EA. A compile-time scheduling heuristic for interconnection constrained heterogeneous processor architectures. *IEEE Trans. on Parallel and Distributed Systems*, 1993,4(2):175–187. [doi: 10.1109/71.207593]
- [9] Du XL, Jiang CJ, Xu GR. A grid DAG scheduling algorithm based on fuzzy clustering. *Journal of Software*, 2006,17(11):2277–2288 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/17/2277.htm> [doi: 10.1360/jos172277]
- [10] Wu XZ, Srikant R, Perkins JR. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Trans. on Mobile Computing*, 2007,6(6):595–605. [doi: 10.1109/TMC.2007.1061]
- [11] Hou ESH, Ansari N, Hong R. A genetic algorithm for multiprocessor scheduling. *IEEE Trans. on Parallel and Distributed Systems*, 1994,5(2):113–120. [doi: 10.1109/71.265940]
- [12] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proc. of the 4th IEEE Int'l Conf. on Neural Networks*. Perth: IEEE Press, 1995. [doi: 10.1109/ICNN.1995.488968]
- [13] Wei JX, Wang YP. Fuzzy particle swarm optimization for constrained optimization problems. *Journal of Electronics and Information Technology*, 2008,30(5):1218–1221. [doi: 10.3724/SP.J.1146.2007.00689]
- [14] Ji YM, Wang RC. Study on PSO algorithm in solving grid task scheduling. *Journal on Communications*, 2007,28(10):60–66 (in Chinese with English abstract).
- [15] Sih GC, Lee EA. A compile-time scheduling heuristic for interconnection-constrained heterogeneous processor architectures. *IEEE Trans. on Parallel and Distributed Systems*, 1993,4(2):175–187. [doi: 10.1109/71.207593]
- [16] Rinaldo N, Zimeo E. Time and cost-driven scheduling of data parallel tasks in grid workflows. *IEEE Systems Journal*, 2009,3(1):104–120. [doi: 10.1109/JSYST.2008.2011299]
- [17] Xie T, Qin X. Security-Aware resource allocation for real-time parallel jobs on homogeneous and heterogeneous clusters. *IEEE Trans. on Parallel and Distributed Systems*, 2008,19(5):682–697. [doi: 10.1109/TPDS.2007.70776]
- [18] Yuan LL, Zeng GS, Jiang LL, Jiang CJ. Dynamic level scheduling based on trust model in grid computing. *Chinese Journal of Computers*, 2006,29(7):1217–1224 (in Chinese with English abstract).
- [19] Zhang WZ, Liu XR, Yun XC, Zhang HL, Hu MZ, Liu KP. Trust-Driven job scheduling heuristics for computing grid. *Journal on Communication*, 2006,27(2):73–79 (in Chinese with English abstract).
- [20] Topcuoglu H, Harir S, Wu MY. Performance-Effective and low-complexity task scheduling for heterogeneous computing. *IEEE Trans. on Parallel and Distributed Systems*, 2002,13(3):260–274. [doi: 10.1109/71.993206]
- [21] Beaumont O, Boudet V, Robert Y. The iso-level scheduling heuristic for heterogeneous processors. In: *Proc. of the PDP 2002*. Los Alamitos: IEEE Computer Society Press, 2002. 335–350.
- [22] Chen TW, Zhan B, Hao XW. Dependent tasks scheduling in grid based on T-RAG optimization selection. *Journal of Computer Research and Development*, 2007,44(10):1741–1750 (in Chinese with English abstract). [doi: 10.1360/crad20071016]
- [23] He XS, Sun XH, von Laszewski G. QoS guided min-min heuristic for grid task scheduling. *Journal of Computer Science and Technology*, 2003,18(4):442–451. [doi: 10.1007/BF02948918]
- [24] Yi K, Wang RC. Nash equilibrium based task scheduling algorithm of multi-schedulers in grid computing. *Acta Electronica Sinica*, 2009,37(2):329–333 (in Chinese with English abstract).
- [25] Chen ZG, Yang B. Task scheduling based on multidimensional performance clustering of grid service resources. *Journal of Software*, 2009,20(10):2766–2775 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3506.htm> [doi: 10.3724/SP.J.1001.2009.03506]
- [26] Golconda KS, Ozguner F, Dogan A. A comparison of static QoS-based scheduling heuristics for a meta-task with multiple QoS dimensions. In: *Proc. of the Int'l Parallel and Distributed Processing Symp.* Los Alamitos: IEEE Computer Society, 2004. [doi:

- 10.1109/IPDPS.2004.1303054]
- [27] Ting TO, Rao MVC, Loo CK. A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. IEEE Trans. on Power Systems, 2006,21(1):411–418. [doi: 10.1109/TPWRS.2005.860907]
- [28] Wang KP, Huang L, Zhou CG, Pang W. Particle swarm optimization for traveling salesman problem. In: Proc. of the 2nd Int'l Conf. on Machine Learning and Cybernetics. IEEE Press, 2003. [doi: 10.1109/ICMLC.2003.1259748]
- [29] Zhong YW, Cai RY. Discrete particle swarm optimization algorithm for QAP. Acta Automatica Sinica, 2007,33(8):871–874 (in Chinese with English abstract).
- [30] Zhu H, Wang YP. Security-Driven task scheduling based on evolutionary algorithm. In: Proc. of the Computational Intelligence and Security 2008. IEEE Press, 2008. 451–456. [doi: 10.1109/CIS.2008.184]
- [31] Lin JN, Wu HZ. Scheduling in grid computing environment based on genetic algorithm. Journal of Computer Research and Development, 2004,14(12):2195–2199 (in Chinese with English abstract).
- [32] Yuan YC, Li XP, Wang Q, Zhang XD. Grid workflows schedule based on priority rules. Acta Electronica Sinica, 2009,37(7):1457–1464 (in Chinese with English abstract).
- [33] Wang YP, Jiao YC, Zhang FS. Uniform and orthogonal genetic algorithm for multiobjective optimization. Journal of Systems Engineering, 2003,18(6):481–486 (in Chinese with English abstract).
- [34] Leung YW, Wang YP. Multiobjective programming using uniform design and genetic algorithm. IEEE Trans. Systems, Man, and Cybernetics—Part C: Application and Reviews, 2000,30(3):293–304. [doi: 10.1109/5326.885111]
- [35] Bäck T. Evolutionary Algorithms in Theory and Practice. New York: Oxford University Press, 1996. 21–28.
- [36] Wu AS, Han Y, Jin SY, Lin KC, Schiavone G. An incremental genetic algorithm Approach to multiprocessor scheduling. IEEE Trans. on Parallel and Distributed Systems, 2004,15(9):824–834. [doi: 10.1109/TPDS.2004.38]
- [37] Chen WN, Zhang J. An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. IEEE Trans. on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 2009,39(1):29–43. [doi: 10.1109/TSMCC.2008.2001722]

#### 附中文参考文献:

- [9] 杜晓丽,蒋昌俊,徐国荣.一种基于模糊聚类的网格 DAG 任务图调度算法.软件学报,2006,17(11):2277–2288. <http://www.jos.org.cn/1000-9825/17/2277.htm> [doi: 10.1360/jos172277]
- [14] 季一木,王汝传.基于粒子群的网格任务调度算法研究.通信学报,2007,28(10):60–66.
- [18] 袁禄来,曾国荪,姜黎立,蒋昌俊.网格环境下基于信任模型的动态级调度.计算机学报,2006,29(7):1217–1224.
- [19] 张伟哲,刘欣然,云晓春,张宏莉,胡铭曾,刘凯鹏.信任驱动的网络作业调度算法.通信学报,2006,27(2):73–79.
- [22] 陈廷伟,张斌,郝宪文.基于任务-资源分配图优化选取的网格依赖任务调度.计算机研究与发展,2007,44(10):1741–1750. [doi: 10.1360/crad20071016]
- [24] 易侃,王汝传.基于 Nash 均衡的网格多调度节点的任务调度算法.电子学报,2009,37(2):329–333.
- [25] 陈志刚,杨博.网格服务资源多维性能聚类任务调度.软件学报,2009,20(10):2766–2775. <http://www.jos.org.cn/1000-9825/3506.htm> [doi: 10.3724/SP.J.1001.2009.03506]
- [29] 钟一文,蔡荣英.求解二次分配问题的离散粒子群优化算法.自动化学报,2007,33(8):871–874.
- [31] 林剑柠,吴慧中.基于遗传算法的网格资源调度算法.计算机研究与发展,2004,14(12):2195–2199.
- [32] 宛迎春,李小平,王茜,张晓东.基于优先级规则的网格工作流调度.电子学报,2009,37(7):1457–1464.
- [33] 王宇平,焦永昌,张福顺.解多目标优化的均匀正交遗传算法.系统工程学报,2003,18(6):481–486.



朱海(1978—),男,河南西峡人,博士,讲师,主要研究领域为网格计算,任务调度,进化算法及人工智能.



王宇平(1961—),男,博士,教授,博士生导师,主要研究领域为人工智能,网络和工程设计中的优化方法最优化理论,数据挖掘.