网格环境中一种改进的蚁群任务调度算法 *

黄 漾[1a], 李肯立[2], 曾 文[1b]

(1. 湖南铁路科技职业技术学院 a. 信息技术系; b. 机车车辆系, 湖南 株洲 412000; 2. 湖南大学 信息科学与工程学院, 长沙 410082)

摘 要: 针对在蚁群算法中初始参数设置对算法收敛性能的影响较大,提出了一种新的改进蚁群算法 NACA (new ant colony algorithm),针对蚁群算法中的四个关键参数随机编码,得到初始的染色体,从而获得一组较优解;再利用遗传算法的优点对上一步的结果单点顺序交叉、对换变异、选择操作以产生更好的解;然后以这组数据为蚁群算法下一次的工作备选值,并进行最大次数的循环迭代直至停止,即求得参数组合的近似最优解。将它应用于网格系统任务调度中,系统的性能得到了明显的改善。仿真模拟结果表明,所提出的算法具有更短的调度长度和更宽的适应性,当任务已知时,执行时间约缩短了 21.7%,且负载变化时对网格中各处理器资源的影响大大减小。

关键词: 网格; 任务调度; 蚁群算法; GridSim

中图分类号: TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2012)02-0455-04

doi:10.3969/j.issn.1001-3695.2012.02.013

Improved ant colony algorithm for task scheduling in grid

HUANG Yang[1a], LI Ken-li[2], ZENG Wen[1b]

(1. *a. Dept. of Information Technology, b. Dept. of Locomotive & Vehicle, Hunan Vocational College of Railway Technology, Zhuzhou Hunan* 412000, *China*; 2. *School of Information Science & Engineering, Hunan University, Changsha* 410082, *China*)

Abstract: It has greater impact on the algorithm convergence that setting the initial parameters in ant colony algorithm. This paper presented an improved ant colony algorithm NACA. Firstly, it made the four parameters of the ant colony algorithm coding randomly and got the chromosomes, a set of optimum solutions could be gained by using the ant colony algorithm. Then they crossover, mutate and select by using the advantages of genetic algorithms. Finally, took the value of this group to explore the next round as the ant colony's original value, ran the maximum number of loop iterations until it stopping. The performance of the system had been significantly improved when it was applied to the grid task scheduling systems. The result of algorithm analysis shows the proposed scheduling algorithm has a shorter length and wider adaptability. When the task is known, execution time can be reduced about 21.7%. The execution time of the task is shorten greatly.

Key words: grid; task scheduling; ant colony algorithm; GridSim

0 引言

所谓网格[1],即将空间分散,集存储、通信、计算等资源于一体而形成的一个功能丰富、虚拟的巨型计算机。网格从广义上说是一个集成的计算和资源环境[27],从而方便人们使用网格大型计算等能力的要求,不需要知道整个资源具体的组织形式和配置方法。作为一个分布式计算平台,网格能够解决科学、工程和经济中的大规模计算问题和数据密集型问题,能够实现地理上广泛分布的高性能计算资源、信息资源、应用系统、服务系统、组织、人员等各种资源的共享与聚合。

网格计算[8](grid computing)是伴随着互联网技术而迅速发展起来的、专门针对复杂科学计算的新型计算模式。在这种计算模式下,利用互联网把空间上分散的计算机组织成一个虚拟的超级计算机,但是其结构异构、资源动态分配等特点使得原本不容易的网格任务高度问题变得更加复杂。它既要求用户能最大跨度地提交任务,从而达到系统的负载平衡以提高吞吐量;又期望这样一种调度策略低碳高效,提升客户的满意度,从而加快网格计算的发展。显而易见,设计一种快速高效的网格任务调度算法有着深远的意义。

众所周知,传统的静态调度算法有 Suffrage[2]、Max-Min[4]、Max-Max、Min-Min[3],而传统的动态高度策略则有遗传算法、粒子群算法、蚁群算法[5]、免疫遗传算法[6]、细菌觅食算法、模拟退火算法[7]等。其中蚁群算法等群智能优化算法成功应用于网格任务调度,并取得了较好的效果,更是使其成为研究的热点。文献[9]通过采用动态因子更新信息素,较好地解决了快速收敛与停滞现象之间的矛盾;文献[10]通过加入一个方向启发因子,使算法在路径搜索的初始阶段摆脱了盲目性。

本文充分基于蚁群和遗传这两种群智能算法,根据各自的特点,设计出蚁群遗传算法 NACA,成功解决了基于网络环境下的任务调度问题,较大地缩短了交互时间,提升了计算效率。该算法针对蚁群算法的调度结果对初始参数设置敏感的特点,通过蚁群算法参数的优化来对算法进行改进,首先对蚁群算法

收稿日期: 2011-07-26; **修回日期**: 2011-09-23 **基金项目**: 国家自然科学基金资助项目(90715029);湖南省科技计划项目(2011FJ3067)

作者简介: 黄漾(1974-),女,湖南株洲人,讲师,硕士,主要研究方向为分布式系统、并行计算(hy@hntky.com);李肯立(1971-),男,教授,主要研究方向为并行处理、DNA 计算机;曾文(1975-),男,经济师,本科,主要研究方向为分布式系统、高教管理.

中的四个参数进行随机编码,产生染色体,利用蚁群算法得到一组较优解;再利用遗传算法的优点对这组值进行单点顺序交叉、对换变异,选择得到更优的解;然后以这组数据为蚁群算法下一次的工作备选值,并进行最大次数的循环迭代直至停止,即求得参数组合的近似最优解。

本文中算法的仿真实验在 GridSim 网格模拟器下进行。实验结果表明,该算法具有更短的调度长度和更宽的适应性,当任务已知时,执行时间约缩短了 21.7%,能有效缩短任务的执行时间,且负载变化时对网格中各处理器资源的影响大大减小。然而当任务未知和系统规模很大时,算法需要进一步完善。

1　蚁群算法基本原理

1.1　蚁群算法特点

蚁群算法是继神经网络、遗传算法、免疫算法之后又一种新兴的启发式群智能优化算法。它无须先验知识,开始以概率形式随机选择搜索路径,随着各路径信息素差异的增大,搜索路径的规律性逐步显现,步步逼近以至达到全局的最优。蚁群算法在解决组合优化问题方面显示出了较好的效果,它有以下特点:

a)蚁群算法是一种全局智能优化仿生算法。

它具有较强的发现近似最优解的能力。这是一种正反馈或称为增强型的机制,算法中通过信息素的不断更新最终收敛于最优的路径上。换句话说,这也是一种基于蒙特卡罗方法的学习方法。并且算法中还隐含有负反馈,能使算法稳定在一个合适的搜索范围,从而使结果尽快地收敛于近似的最优解。不仅如此,它不但可用于求解单目标优化问题,同样也适合多目标问题的求解。

b)它是一种分布式的优化算法。

它不仅适应于目前通用的串行计算机,同样也适合于方兴未艾的并行计算机。蚁群个体很快达到局部最优,多个个体之间通过合作能很快收敛于解空间的某一子集,有利于对解空间的进一步探索,从而发现较好解。

c)蚁群算法具有较好的健壮性。

相对于其他算法,蚁群算法对初始路线要求不高。

d)基本蚁群算法具有自组织性。

蚁群算法是一种通用型的随机优化算法,但人工蚂蚁绝不是对实际蚂蚁的简单模拟,它融入了人工智能。系统中个体的增加而带来的系统通信开销将非常小。自组织性是相对其组织性而言的,蚁群算法的搜索通过人工智能的调整,不需要十分清楚每个过程,同样可以用这种算法来求解问题。

1.2　蚁群算法的数学模型

基本蚁群算法吸收了生物界中蚂蚁群体行为的特征,因其能感知小范围内区域的状况,判断出食物的位置和同伴的信息素轨迹,并随时释放自己的信息素,因此在没有任何先验知识的情况下,蚂蚁仍然有能力找到巢穴与食物源之间的最佳路径。即使在有障碍物的情况下,它们依然能很快重新找回新的最短路径。

蚁群算法首先成功应用于旅行商问题。为了方便,作以下约定:

l 代表分配到 n 个城市的蚂蚁数;

$b_i(t)$ 代表 t 时刻城市 i 中蚂蚁的数量;

d_{ij} 代表两城市 i、j 之间的距离;

η_{ij} 代表边城市 i、j 之间的可见度;

$\tau_{ij}(t)$ 代表城市 i、j 之间信息素的强度。

令 $m=\sum_{i=1}^{n} b_i(t)$ 为总的蚂蚁数量。单个蚂蚁具有以下特点:

a)蚂蚁有记忆功能,能记住曾经访问过的城市,因此不会重复选择访问过的城市;

b)蚂蚁的行动不是简单盲目的,它们知道两个城市间的距离,并且它们倾向选择访问邻近它们位置的城市;

c)蚂蚁遍历每一地方的同时,会在经过的路径中释放一种挥发性的物质——信息素。

开始时,各条路径上的信息量相等,设 $\tau_{ij}(t)=C$(C 为常数),蚂蚁 k 访问完城市 i 之后选择访问城市 j 的概率为

$$P_{ij}^{k}=\begin{cases}\dfrac{\tau_{ij}^{\alpha}(t)\times\eta_{ij}^{\beta}(t)}{\sum_{s\in \text{allowed}_k}\tau_{is}^{\alpha}(t)\times\eta_{is}^{\beta}(t)} & s\in \text{allowed}_k\\ 0 & \text{otherwise}\end{cases}\tag{1}$$

其中:η_{ij} 是城市 i 到 j 之间的可见度,定义为 $\eta_{ij}=1/d_{ij}$(d_{ij} 为两城市 i、j 之间的距离);β 是衡量 η_{ij} 的参数,即可见度的重要程度;$\tau_{ij}^{\alpha}(t)$ 表示 t 时刻城市 i 与 j 之间信息素的强度,α 是表征 τ_{ij} 的参数;allowed_k 是没有访问的城市的集合。

若开始有 l 只蚂蚁随机地分配到 n 个城市中,每只蚂蚁根据式(1)提供的概率选择下一个将要访问的城市。访问完 n 个城市后,每只蚂蚁完成一次遍历。很容易得知,遍历路径较短的蚂蚁比那些遍历路径长的蚂蚁留下更多的信息素。于是,可定义更新信息素的规则如下:每只蚂蚁在其经过的路径上留下的信息素的量为 $\frac{Q}{L_k}$(Q 为常数,L_k 是路径的长度)。由于随着时间的过去,信息素会逐渐消失,因此,信息素的更新公式可定义如下:

$$\tau_{ij}(t+1)=\rho\times\tau_{ij}(t)+\Delta\tau_{ij}\tag{2}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{l}\Delta\tau_{ij}^{k}\tag{3}$$

式中:$\rho\in[0,1]$,是控制 τ_{ij} 减少的参数;$\Delta\tau_{ij}$ 是第 i 个城市到第 j 个城市路径上信息素的总增长;$\Delta\tau_{ij}^{k}$ 是第 k 只蚂蚁所引起的城市 i 到 j 路径上信息素的增长;t 则是一个迭代的计数器。

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁在本次循环中经过 } ij\\ 0 & \text{否则}\end{cases}\tag{4}$$

$\Delta\tau_{ij}(t)$、$\Delta\tau_{ij}(t)$、$P_{ij}^{k}(t)$ 的表达形式可以不同,要根据具体问题而定。Dorigo 曾给出三种不同模型,分别称为 ant-cycle system、ant-quantity system、ant-density system,它们的差别在于式(4)的不同。Ant-cycle system 模型如式(4)。

在 ant-quantity system 模型中:

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{d_{ij}} & \text{若第 } k \text{ 只蚂蚁在时刻 } t \text{ 与 } t+1 \text{ 之间经过 } ij\\ 0 & \text{否则}\end{cases}\tag{5}$$

根据基本蚁群算法[11]的步骤,可以用如下所示的算法说明:

```
for t←1 to iteration number
  for k←1 to l do
    Repeat until ant k has accomplished a travel
    Select the next city j to be visited with probability P_ij given by Eq. (1)
```

Calculate L_k

Update the trail given by Eqs. (2,3,4)

end for

if n is great than maximum iterative number or satisfying the condition of stoping

then endfor

2　算法描述

蚁群算法原理是一种正反馈机制,它通过信息素的积累和更新而收敛于最优路径,具有分布、并行、全局收敛能力。由于初期信息素匮乏,所以求解速度慢。为了改进蚁群算法的性能,研究人员作了许多研究,其中一种有效的方法就是与遗传算法结合,利用遗传算法的随机搜索、快速性、全局收敛性产生有关问题的初始信息素分布。然后充分利用蚂蚁算法的并行性、正反馈机制以及求解效率高等特性来求精确解,这样融合后的算法优势互补,在时间效率上优于蚁群算法,在求解效率上优于遗传算法,期望获得性能优化和时间高效的双赢[10,12]。

NACA 算法就建立在这种思想之上,充分利用遗传算法和蚁群算法的优势,设计出用于求取基于网格下任务调度的优化算法。该算法运用遗传算法对蚁群算法的四个关键参数进行编码,得到初始的染色体,从而获得一组较优解,再将上一步的结果单点顺序交叉、对换变异、选择操作以产生更好的解,并将此结果传递给蚁群算法的参数。通过信息素的修正,可选路径概率变化,再循环利用遗传算法和蚁群算法迭代,直到达到最大的迭代次数或者满足事先设定的算法停止条件才停止。充分利用遗传算法的随机性、快速性,以及蚁群算法的正反馈加速搜索的收敛性,其负反馈均衡系统的负载平衡性,从而求得问题的近似最优解。

2.1　算法模型与初始条件

网格系统中,资源、计算能力和通信能力都是动态变化的。系统每启动一次任务集,资源管理系统下的作业调度器都会监测到节点的变化,在备选的调度策略中,寻找一个优秀策略来执行任务。因此网格资源的计算、通信、存储等能力的随时变化等基本信息便可构成蚁群算法中的信息素。

大型的任务集可分解成若干个有某种相关性的子任务,为描述方便,本文作如下约定:

定义 1　任务集 $T=\{T_1,T_2,\cdots,T_n\}$,其中 T_i 为其中的一个子任务。

定义 2　算法开始时,蚂蚁随机分布在系统中的各个节点上,之后每一步路径的选择,均按照概率转移公式进行。在时刻 t,蚂蚁 k 在节点 i 转移到未访问的节点 j 的概率为

$$P_{ij}^{k}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{e\in U}([\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta})} & u\in \text{有效的资源 } U\\ 0 & \text{其他}\end{cases}\tag{6}$$

式中:$\tau_{ij}(t)$ 表征在时刻 t,蚂蚁 k 在资源 i、j 上的信息素浓度;α 是衡量 $\tau_{ij}(t)$ 的参数;η_{ij} 是资源 i、j 之间的可见度(即计算、存储等能力);而 β 表征 η_{ij} 指标的参数。因为搜索不能停滞不前,所以约定 $\eta_{ij}(t)=\tau_{ij}(0)$。

定义 3　在算法 NACA 中,结合信息素局部更新和全局调整的策略,每结束一次循环都将进行奖惩。顺利完成遍历的节点,增加信息素浓度,如式(7)所示,其中 C_e、K 为奖励系数。没有成功完成任务的节点,减少其信息素浓度,如式(8)所示,其中 C_p 为处罚系数。

$$\Delta\tau_j=C_e\times K\tag{7}$$

$$\Delta\tau_j=C_p\times K\tag{8}$$

当蚂蚁成功完成所有节点的遍历后,选择其中花费时间最短的,即经过路径最短的线路进行全局信息素的调整,调整幅度为

$$\Delta\tau_j^k=\begin{cases}\dfrac{Q}{L_k} & \text{若第 } k \text{ 只蚂蚁经过资源节点 } j\\ 0 & \text{否则}\end{cases}\tag{9}$$

其中:Q 表示信息素的强弱,L_k 表示第 k 只蚂蚁在整个循环中经历的总路径长度。

综上,信息素更新公式为

$$\tau_{ij}^{\text{new}}=\begin{cases}(1-\rho)\times\tau_{ij}^{\text{old}}\cdot\Delta\tau_{ij}^{\text{best}} & \tau_{\min}\leqslant\tau_{ij}^{\text{new}}\leqslant\tau_{\max}\\ \tau_{\min} & \tau_{ij}^{\text{new}}\leqslant\tau_{\min}\\ \tau_{\max} & \tau_{ij}^{\text{new}}\geqslant\tau_{\max}\end{cases}\tag{10}$$

2.2　网格计算环境中 NACA 的设计

根据 NACA 的基本思想及假设,NACA 描述如下:

a)编码,初始化算法

确定问题的目标函数和变量,便可以对变量进行编码。问题的解是用数字串来表示的,并且后续的遗传算子直接操作的对象也是串。

本文将采用常用的二进制表示个体。任务量的总数反映在染色体的长度上,每一位均为正整数。因此问题空间即反映在执行时的遗传空间上,并通过染色体直接表现出来。在本算法中,蚁群算法的四个参数取值区间为 $0\leqslant\beta\leqslant5$;$0\leqslant\alpha\leqslant3$;$0.01\leqslant\rho\leqslant0.99$;$10\leqslant Q<10000$。

b)建立搜索表

禁忌表 $\text{tabu}_k(s)$ 中将记录第 k 只蚂蚁所处的位置,并根据式(1)的概率转移公式记录下一步要遍历的节点。如此重复,直至执行完 n 个需要遍历的节点。遍历的过程中,信息素的更新按照式(2)和(3)进行。当某只蚂蚁完成 n 个节点的遍历后,可以很容易地根据 $\text{tabu}_k(s)$ 算出它所耗费的时间,从而选出其中时间最短的,并将式(10)更新整体的信息素。

c)禁忌表的清空

禁忌表的清空以及全局信息素的更新均按照以上的规则进行,选择其中最短时间的个体,进行单点顺序交叉和对换变异操作。

(a)选择操作[11]。即从父代中以一定比例选择两个父本进行交叉,在得到的两个子代中,选择一个适应度高的替代父代群体中适应度最低的个体。如果交叉后的子代适应度比父代中的适应度最差的还低,那么本次操作作废,重新选择父代进行交叉。

(b)单点顺序交叉[11]。本文选用单点顺序交叉。其原理是:如果两父 A、B,随机选取交叉点,定义交叉点后面为匹配区域。先将 B 和 A 的匹配区域分别加到 A 和 B 的前面,从而得到 A'、B',最后可以得到在匹配区域相同的码,得到最终子串 A''、B''。

(c)对换变异操作。对换变异,即随机选择串中的两个点,交换其值。

d)算法收敛性的判定

定义 4　如果最终的种群由个体 $x_1,x_2,\cdots,x_n$ 构成,大小为 N,$f_1,f_2,\cdots,f_n$ 为其对应的适应度大小,按升序排列得到

$f'_1, f'_2, \cdots, f'_n$，取前 $N/2$ 个适应度值较大的个体，其平均适应度值为 $\bar{f'} = \frac{2}{n}\sum_{i=0}^{n/2} f_i'$，则定义 $\Delta = \sqrt{\frac{2}{n}\sum_{i=0}^{n/2}(f'_i - \bar{f'})^2}$ 来表征种群收敛的产生。

这个定义之所以只计算前 $N/2$ 个适应度值较大的个体的差异，原因有：a）排除了适应度值低的个体对种群分布的影响，更直接地反映较优个体的重复率以及趋同率；b）遗传进化中的交叉操作总是两两配对的，而且较优个体的适应度值较大，被选中进入下一代的概率也就较大，若个体越优，其进行交叉和变异的概率也就越小，这样种群中 $N/2$ 个较优个体重复或者趋同进化，就会因为失去种群多样性而停滞不前，从而新算法发生收敛。

2.3　NACA 总体描述

首先对产生的四个群体，选择路径短即适应度最大的两个个体，经过单体顺利交叉和对换变异得到两个新个体；再用这两个新个体替换两个最差的个体，从而构成一个适应度较高的新群体，进入下一步的循环换代。其基本框图如图 1 所示。

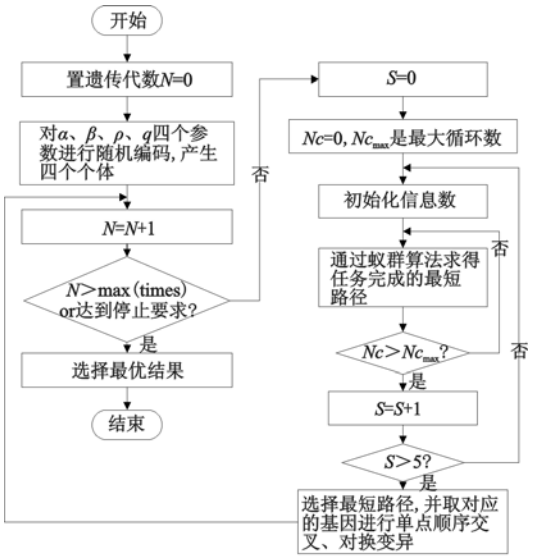

图1　NACA算法的基本框图

3　仿真设计与分析

本文的实验基于 GridSim[13,14]，利用 Java 的与平台无关性，调用其中的 simjava 函数库，在 Windows 下编写代码进行实验。另外，由于 Gridsim 代码的开源，用户只需修改其中的部分便可用来实现各自的算法，十分方便。

首先创建了一个虚拟网格，包括七个资源和三个用户，性能参数如表 1 所示。

表 1　资源性能参数

ID	PE	MIPS of PE	RATE
R0	1	387	21
R1	1	315	16
R2	1	312	24
R3	2	355	11
R4	2	360	15
R5	3	380	14
R6	3	695	1

在此基础上进行了 NACA 和 ACA、文献[9]提出的 AACA 和文献[10]提出的 GCACA 算法的对比，求出同组任务所对应的完成时间，如表 2 所示。

表 2　ACA、AACA、GCACA 与 NACA 执行情况对比

Tast node	ACA	AACA	GCACA	NACA
20	345	223	210	170
30	870	521	498	301
40	1058	692	653	485
50	2254	1465	1328	1201
60	3242	2459	2278	2012
80	4785	3886	3674	3172

由表 2 不难得出，算法 NACA 经过一段时间，关键路径上的信息素浓度达到一定值，便有了很好的收敛效果。从表 2 所示的任务节点和时间图可以得出，如果有相同的节点数，NACA 的任务总执行时间显著减少，从而提高了调度效率，执行时间得到了有效的减少。

由图 2 可以得知，在相同的任务节点条件下，NACA 算法的收敛时间明显优于禁忌搜索以及蚁群算法。蚁群算法正反馈的特点加快了初始解的构造，扩大可行解的选取范围；并且降低了前期结果对后期的影响，避免算法过早在局部最优处停滞，从而得到全局最优解，性能能够得到明显的提高，克服蚁群算法易陷入局部最优的缺点。

再以 80 个用户为例比较三种算法负载平衡的特点，具体情况如表 3 所示。

表 3　资源负载平衡的比较

资源编号	ACA	Tabu	NACA
01	0.54	0.58	0.41
02	0.31	0.28	0.38
03	0.20	0.18	0.36
04	0.45	0.52	0.42
05	0.16	0.13	0.28
06	0.70	0.55	0.53
07	0.82	0.85	0.61
08	0.12	0.14	0.31
09	0.35	0.42	0.45
10	0.25	0.23	0.38

表 3 所示的资源负载平衡结果显示，算法 NACA 与 Tabu、ACA 算法相比，少数资源的负载量有所上升，部分资源的负载量有所回落，整体趋势较平缓。将表格中的数据绘制成图表格式，如图 3 所示，能更直观地了解资源负载情况。

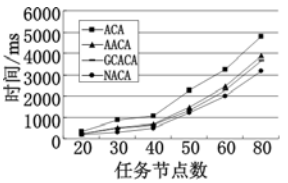

图2　时间与任务节点

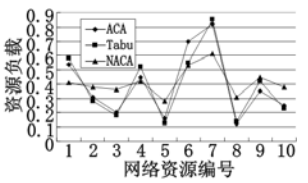

图3　负载平衡比较

在图 3 中，Tabu 和 ACA 算法在资源点 1、6、7 的负载过高，而这两种算法在资源点 3、5、8 是比较低的。很明显，从图中可以看出，NACA 算法很好地改善了资源的负载情况。其原因是系统中蚂蚁产生的信息素，通过算法潜在的负反馈作用，有效地调节了资源节点的负荷，不至于波动较大，从而使系统的负荷趋于平衡。

4　结束语

本文针对蚁群算法对参数敏感的特点，提出适合于网格计算环境下的任务调度策略 NACA。本文针对蚁群算法中的四个关键参数随机编码，得到初始的染色体，从而获得一组较优解；再利用遗传算法的优点对前一步的结果进 （下转第 462 页）

方法有最好的位置追踪性能。其他实验结果如表 2 所示。从表中的实验数据可以看出,所提改进策略能有效改进粒子滤波的性能。

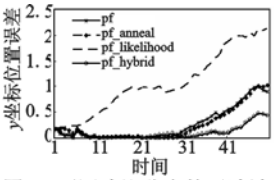

图1 不同建议分布粒子滤波的位置追踪性能评估

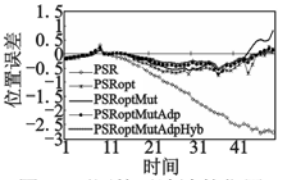

图2 不同粒子滤波的位置追踪性能评估

表 2 不同粒子滤波的追踪性能评估

粒子滤波算法 ($T_h = 2/N_p$; $T_l = 1/2N_p$)	时间 (50 step)	RMSE	
		距离/m	角度/rad
基于 PSR	0.109 38	2.275 73	0.015 88
基于 PSRopt	0.984 38	0.407 41	0.012 39
基于 PSRoptMut	1.015 60	0.355 67	0.010 64
基于 PSRoptMutAdp	0.906 25	0.290 72	0.009 99
基于 PSRoptMutadpHyb	0.908 16	0.206 27	0.009 13

4 结束语

近年来,粒子滤波引起了机器人学、目标追踪等诸多应用领域学者的关注。针对粒子滤波算法的样本退化问题,目前主流的改进策略为选择好的建议分布和执行重采样算法。在此研究基础上,本文提出了一种粒子滤波算法改进策略。针对建议分布,考虑采用混合建议分布对状态转移和观测似然概率来综合处理,并结合退火参数来控制这两种概率的比例。针对重采样算法,基于有效样本大小实现自适应重采样,结合权重优化思想,提出了一种改进的部分系统重采样算法,并在重采样后执行粒子变异操作来保证样本的多样性。通过单机动目标跟踪的仿真实验,验证了所提粒子滤波算法改进策略的性能和有效性。

参考文献:

[1] GORDON N J, SALMOND D J, SMITH A F M. Novel approach to nonlinear/non-Gaussian Bayesian state estimation[J]. IEEE Proceedings on Radar and Signal Processing,1993,140(2):107-113.

[2] ARULAMPALAM M S, MASKELL S, GORDON N, *et al*. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking [J]. IEEE Trans on Signal Processing, 2002, 50 (20): 174-188.

[3] 于金霞, 蔡自兴, 段琢华. 基于粒子滤波的移动机器人定位关键技术研究综述[J]. 计算机应用研究,2007,24(11): 9-14.

[4] DOUC R, CAPPE O. Comparison of resampling schemes for particle filtering[C]//Proc of Image and Signal Processing and Analysis. Zagreb, Croatia: IEEE Press, 2005: 64-69.

[5] BOLIC M, DJURIC P, HONG Sang-jin. New resampling algorithms for particle filters[C]// Proc of International Conference on Acoustics, Speech, and Signal Processing. [S.l.]: IEEE Press, 2003: 589-592.

[6] 谌剑, 严平, 张静远. 权值优化组合粒子滤波算法研究[J]. 计算机工程与应用, 2009, 45(24): 33-35, 39.

[7] 胡士强, 敬忠良. 粒子滤波原理及其应用[M]. 北京: 科学出版社, 2010.

[8] YU Jin-xia, TANG Yong-li, LIU Wen-jing. Research on the adaptive mechanisms in particle filter[C]//Proc of Chinese Control and Decision Conference. [S.l.]: IEEE Press, 2010:1316-1321.

(上接第 458 页)行单点顺序交叉、对换变异、选择操作以产生更好的解;然后以这组数据为蚁群算法下一次工作的备选值进行最大次数的循环迭代直至停止,即求得参数组合的近似最优解。本文最后的仿真模拟结果表明,所提出的算法具有更短的调度长度和更宽的适应性,当任务已知时,执行时间约缩短了 21.7%,且有效地均衡了网格中各处理器负载,提升了系统效率。本方案具有较深的理论价值和现实意义。

参考文献:

[1] FOSTER I. What is the grid? A three point checklist[J]. Grid Today,2002,1(6):22-25.

[2] MAHESWARAN M, ALI S, SIEGEL H J, *et al*. Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems[C]//Proc of the 8th Heterogeneous Computing Workshop. Washington DC: IEEE Computer Society,1999:30-33.

[3] BRAUN T D, SIEGEL H J, BÖLÖNI L L, *et al*. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing system[J]. Joumal of Parallel and Distributed Computing,2001,61(6):810-837.

[4] FUJIMOTO N, HAGIHATA K. A comparison among grid scheduling algorithms for independent coarse-grained tasks[C]//Proc of Symposium on Applications and the Internet-Workshops. Washington: IEEE Computer Society,2004:674-680.

[5] 赵扬帆. 基于遗传算法和蚁群算法的网格任务调度策略[D]. 青岛:中国海洋大学,2006.

[6] 陈廷伟, 张斌, 郝宪文. 基于免疫遗传算法的网格任务调度[J]. 东北大学学报:自然科学版,2007,28(7): 229-332.

[7] ABRAHAM A, BUYYA R, NATH B. Nature's heuristics for scheduling jobs on computational grids[C]//Proc of the 8th International Conference on Advanced Computing and Communications. New Delhi:Tata McGraw-Hill Publishing, 2000:45-52.

[8] FRAN B, FOX G, HEY T. Grid computing: making the global infrastructure a reality[M]. [S.l.]: Wiley,2003:652-657.

[9] 张金标,陈科. 并行设计任务调度的自适应蚁群算法[J]. 计算机辅助设计与图形学学报,2010,22(6):1070-1074.

[10] WANG Hua, SHI Zhao, GE An-fen, *et al*. An optimized ant colony algorithm based on the gradual changing orientation factor for multi-constraint QoS routing[J]. Computer Communications,2009,32(4):586-593.

[11] 杨金辉. 若干组合优化的智能计算方法与应用研究[D]. 长春:吉林大学,2008.

[12] ZHANG Qiang, ZHANG Qiu-wen. An improved ant colony algorithm for the logistics vehicle scheduling problem[C]//Proc of the 2nd International Symposium on Intelligent Information Technology Application. Washington DC: IEEE Computer Society,2008:55-59.

[13] SONG H J, LIU Xin, JAKOBSEN D, *et al*. The MicroGrid: a scientific tool for modeling computational grids[J]. Scientific Programming,2000, 8(3):4-10.

[14] LAMEHAMEDI H, SHENTU Z J, SZYMANSKI B, *et al*. Simulation of dynamic data replication strategies in data grids[C]//Proc of the 17th International Symposium on Parallel and Distributed. Washington DC: IEEE Computer Society,2003:175-183.

[15] 吴庆洪, 张纪会, 徐心和. 具有变异特征的蚁群算法[J]. 计算机研究与发展,1999,36(10):1240-1245.