文章编号:1671-4598(2021)11-0183-06 DOI:10.16526/j.cnki.11—4762/tp.2021.11.033 中图分类号:TP3 文献标识码:A

基于改进蚁群算法的遥感信息处理负载均衡任务调度算法研究

赵 斐,陈 昊,白建东,刘 铁
(北京跟踪与通信技术研究所,北京 100094)

摘要:遥感信息服务链动态构建技术是根据用户提出的航天信息需求,以及用户0终端行为感知后形成的主动推送需求,将遥感信息获取与处理作为一种服务对待,利用服务组合与优化,动态构建服务链,实现网络环境下的信息资源按需聚合与高效协同,以满足对“端”的遥感信息支援应用需求;文章首先研究了蚁群算法和模拟退火算法在遥感信息处理计算节点任务上的调度原理,并分析了上述传统算法在得出最优解之前会出现的问题;基于蚁群算法并结合其他启发式算法的优点,提出了一种基于改进蚁群算法的负载均衡任务调度算法,完成了遥感信息多任务处理服务链的计算任务分配,提升了天基信息处理系统整体的计算效率;最后通过仿真实验验证了算法的有效性。

关键词:蚁群算法;模拟退火算法;负载均衡;遥感信息处理

One Remote Sensing Information Processing Task Allocation Algorithm for Collaborative Planning Based on Knowledge Model and Genetic Algorithm

ZHAO Fei, CHEN Hao, BAI Jiandong, LIU Tie
(Beijing Institute of Tracking and Communication Technology, Beijing 100094, China)

Abstract: The dynamic construction technology of remote sensing information service chain is based on space information needs put forward by users and the active push needs formed by user terminal behavior perception. The remote sensing information acquisition and processing is treated as a kind of service and uses service composition and optimization to dynamically construct the service chain, so as to realize the on-demand aggregation and efficient collaboration of information resources in the network environment, meeting the remote sensing needs of the opposite and information support application requirements. The scheduling principle of ant colony algorithm and simulated annealing algorithm in remote sensing information processing computing node tasks is studied and analyzed problem that will occur before traditional algorithm get optimal solution by this paper. Based on ant colony algorithm and advantage of other heuristic algorithm, a load balancing task scheduling algorithm is proposed, which completes the calculation task allocation of remote sensing information multi-task processing service chain and improves the overall processing system. Finally, the effectiveness of algorithm is verified by simulation experiments.

Keywords: ant colony algorithm; simulated annealing algorithm; load leveling; remote sensing information processing

0 引言

遥感信息处理服务组合面临的关键问题在于对用户需求、遥感信息及其处理服务语义关联的充分准确理解,并在组合过程中利用这些语义控制处理服务的选择和构成复杂结构的服务链。遥感信息多任务服务链在线整合是一个并发执行调度服务与分布处理服务的过程,在多源异构与时空分布不均衡、网络环境动态多变、处理节点负载不均衡的服务执行环境中,遥感信息处理调度策略对于系统执行的整体性能最优至关重要。

自适应负载均衡是指无论系统处于空闲、稳定还是繁忙状态,负载均衡算法都会自动评估系统的服务能力,进行合理的资源分配,使整个系统始终保持较好的性能,不产生饥饿或者过载、宕机。自适应负载均衡有两个主要目标:保持较短的请求响应时间和较小的请求阻塞概率;负载均衡算法在可控级别,不占用过多的CPU、网络等资源。自适应负载均衡建立在现有网络结构之上,它提供了一种廉价有效透明的方法扩展网络设备和服务器的带宽、增加吞吐量、加强网络数据处理能力、提高网络的灵活性和可用性,使用自适应负载均衡能够更合理的利用资源,提高运行性能[1-4]。

现有的自适应负载均衡算法主要分为静态和动态两类。静态负载均衡算法以固定的概率分配任务,不考虑服务器的状态信息,如轮转算法、加权轮转算法等,所以静态负载均衡算法不适合遥感信息服务链的优化与服务资源配置。

收稿日期:2021-04-06; **修回日期**:2021-05-12。

作者简介:赵 斐(1974-),男,宁夏固远人,博士,主要从事信号与信息处理方向的研究。

引用格式:赵 斐,陈 昊,白建东,等.基于改进蚁群算法的遥感信息处理负载均衡任务调度算法研究[J].计算机测量与控制,2021,29(11):183-188.

动态负载均衡算法以服务器的实时负载状态信息来决定任务的分配，如最小连接法[5-7]、加权最小连接法[8-10]等，但是上述算法考虑的影响输入因子比较少，而且权值无法自动选择确定，所以对遥感信息服务链的优化也不适合。

本文提出了一种改进蚁群算法（EXACO）的整合负载自适应分配算法进行遥感信息服务链的优化与服务资源配置。蚁群算法（ACO，ant colony optimization）是一种用来在图中寻找优化路径的随机搜索寻优技术，它由意大利学者 Marco Dorigo 于 1992 年在他的博士论文中引入，蚁群算法在求解多种组合优化问题中获得了广泛的应用，如在计算机技术应用中，它可以作为网络路由控制的工具；在交通控制中，它成功地解决了车辆调度问题；在图表制作中，它被用来解决颜色填充问题。而为了使蚁群算法在进行任务调度时在不陷入局部最优解的前提下尽可能加速收敛，本文提出了一种改进的蚁群算法，引入模拟退火算法的思想对原有的蚁群算法做出一定修改：当蚂蚁 k 发现了一条最优路径并形成最优解 X^* 后，随机交换两个资源所执行的任务作为扰动，产生一个新解 X，通过模拟退火算法的方式决定保留最优解 X^* 还是用新解 X 替换最优解 X^*；同时将改进蚁群算法应用于遥感信息服务链处理系统中去，从而实现遥感信息服务在线协同调度优化执行。

1 蚁群算法

蚁群算法的诞生主要来源于自然界蚂蚁觅食行为。蚂蚁的视觉十分有限，但却能依靠蚁群中每只蚂蚁在路径上释放的“信息素”这种物质在“黑暗”的外界环境中寻找到从洞穴通向食物的最短路径。蚂蚁在行走过程中不断地释放信息素来标识自己的行径，在寻找食物的过程中不断根据信息素的浓度选择行走的方向，最终到达食物所在的地方。由于每一只蚂蚁都有着各自不确定的行动路径，这导致较短路径上爬行的蚂蚁数量会比较长路径上爬行的蚂蚁数量多，从而使得较短路径上被蚂蚁释放的信息素浓度越高，后出发的蚂蚁总是趋向于往信息素浓度高的地方爬行。随着时间的推移，越来越多的蚂蚁聚集到最短的路径上去，这种动态的正反馈机制使得蚁群算法具有很好的鲁棒性。

蚁群算法最早用于解决著名的旅行商问题[11-12]，该问题可被描述为：某商人从一座城市出发，途径 n 座城市，最后再回到出发的城市，并且每座城市必须且只能经过一次，求一种旅行方案使得所需路径最短。该问题的数学描述为：设 n 座城市组成的集合为 $C=\{c_1,c_2,\ldots,c_n\}$，每两座城市之间的道路为 $R=\{r_{ij}\mid c_i,c_j\in C\}$，$G=(C,R)$ 是一个有向图，求有向图 G 的一条最短汉密尔顿回路[13]。

当蚂蚁 $k(k=1,2,\ldots,m)$ 出发寻找城市之间的最短路径时，会根据每条邻近路径上的信息素浓度来确定下一步通过哪条路径前往下一座城市。为了防止蚂蚁重复经过已到达过的城市，引入禁忌表 $tabu_k(k=1,2,\ldots,m)$ 来表示蚂蚁 k 已经经过的城市，集合 $allowed_k=\{0,1,\ldots,n-1\}-tabu_k$ 表示蚂蚁 k 下一步允许选择的城市，这一约束在自然界中的蚂蚁身上并不存在。蚂蚁 k 根据以下公式选择下一个到达的城市：

$$p_{ij}^k=\begin{cases}\dfrac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{is}^{\alpha}\eta_{is}^{\beta}(t)}, & j\in allowed_k\\ 0, & \text{else}\end{cases}$$

其中：$\tau_{ij}(t)$ 表示 t 时刻在 ij 连接线上残留的信息素的量，η_{ij} 表示由城市 i 转移到 j 的期望程度：$\eta_{ij}=\dfrac{1}{d_{ij}}$。其中，$d_{ij}$ 表示城市 i 和 j 之间的距离。

为了防止信息素在路径上的过度积累导致算法提前达到终止条件而结束，需要不断更新信息素。在 $\tau+n$ 时刻，路径 r_{ij} 上的信息素更新依据以下公式：

$$\tau_{ij}(t+n)=\rho*\tau_{ij}(t)+\Delta\tau_{ij}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^k$$

其中：$(1-\rho)$ 表示信息素消散的程度。$\Delta\tau_{ij}^k$ 表示蚂蚁 k 在本次循环中残留在路径 r_{ij} 上的信息素含量，$\Delta\tau_{ij}$ 表示本次循环中残留在路径 r_{ij} 上的信息素含量。

$$\Delta\tau_{ij}^k=\begin{cases}\dfrac{Q}{L_k} & \text{蚂蚁 } k \text{ 在本次循环中途径 } r_{ij}\\ 0 & \text{else}\end{cases}$$

其中：Q 为常数，L_k 表示蚂蚁 k 在本次循环中途经的总路程，参数 α、β 表示蚂蚁在运动中积累的信息和启发式因子在蚂蚁选择路径时所起的不同作用。初始时刻 $\tau_{ij}(0)$ 为常数 C，并且 $\Delta\tau_{ij}=0$。参数 Q、C、α、β、ρ 均可以用实验的方式确定最优的组合。至于算法的停止条件，则可以设定为超过固定的循环次数或进化趋势不明显。

相较于遗传算法等其他启发式算法，蚁群算法具有以下特点[14-16]：

1）鲁棒性强。在算法的运算过程中，如果系统的状态发生了改变，蚁群算法仍然可以进行自我调整。例如蚁群当前的路径上出现了不可逾越的障碍物，蚁群仍然能够在一段时间的调整后重新找到新的路径。

2）自组织。蚁群中的蚂蚁在刚开始觅食时随机并且无序地寻找各自的路径，但随着时间的推移，每只蚂蚁个体之间通过信息素的作用开始互相影响，最后趋向于最优化的路径。

3）可并行。蚁群中每个蚂蚁个体的行为互不干扰，仅仅通过路径上的信息素进行通信，因此蚁群算法从本质上具备并行性，易于并行实现以改善算法性能。

4）扩展性强。蚁群算法可以与模拟退火算法等多种启发式算法结合使用，因此具备多种拓展后的变种蚁群算法。

蚁群算法的基本流程如图 1 所示。

2 基于改进蚁群算法的遥感信息处理服务负载均衡任务调度算法

用户通过终端发出任务指令以后，需要对用户的任务需求进行快速地分析处理，来帮助用户高效快速地获得所

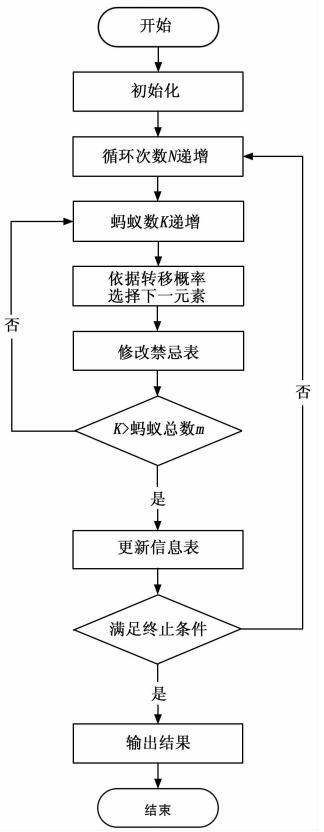

图 1　蚁群算法流程图

需要的遥感信息产品，而遥感信息服务链动态构建技术即是来解决这一问题。遥感信息处理服务链是指从终端提出需求、卫星获取、处理、传输、信息分发到终端的信息节点构成的一条动态链路。

服务链动态构建包含两方面内容：依据任务的重要性对单个节点的执行序列动态调整，根据各节点的能力状况对链路动态调整。满足遥感信息“按需保障、随时随地保障”的需求，需要动态构建服务链，包括对卫星、星群的调度完成数据获取，动态调度链路上计算资源对获取到的（光学、SAR）数据进行处理生成数据产品等。服务链动态构建技术难点在于遥感信息处理涉及的服务链中星上处理能力，中间节点计算、存储及处理能力，端群的计算处理能力，网络节点多，可用能力动态变化，如何高效、快速完成信息处理。

负载均衡主要是进行合理的资源分配，使整个系统始终保持较好的性能，不产生饥饿或者过载、宕机。负载均衡任务调度问题可以被描述为：求一种最优的任务分配策略，将 n 个长度不等的任务 $T=\{T_1,T_2,\ldots,T_n\}$ 按照某一策略分配给 m 个处理能力不同的服务节点 $S=\{S_1,S_2,\ldots,S_m\}$，并且使 n 个任务的总完成时间 $C=\max\{c_1,c_2,\ldots c_m\}$ 达到最短。每一种分配策略都对应着该问题的一个可行解 X，而具有最小完成时间的分配策略就是该问题的最优解。

蚁群算法适合解决各种离散问题，但是当问题规模较大时，蚁群算法在实际运用中容易遇到收敛速度慢、易陷入局部最优解的情况。使用基本的蚁群算法能够较好地解决任务调度问题，但是在实际应用过程中，可以针对特定的任务调度问题进行一定的策略和算法优化。

本文研究了一种基于改进蚁群算法进行负载调度算法设计。调度问题除了考虑将总完成时间最低之外，还应顾及节点之间的负载均衡问题。设 X 为一组任务与节点的对应规则，则将集群的整体负载均衡度定义为 $LB(X)=\sqrt{\sum_{i=1}^{n}\left(1-Y\frac{A_i}{B_i}\right)}$。其中，$A_i$ 表示节点上的任务总量，B_i 表示节点的综合处理能力，Y 为归一化参数，使得 $0\leqslant Y\frac{A_i}{B_i}<1$。节点的综合处理能力可有节点 CPU 能力 p、内存容量 r 和磁盘容量 d 表示，通过为每一项指标设置不同的影响参数，进行加权运算后可得节点的综合处理能力为 $B_i=W_1*p(S_i)+W_2*r(S_i)+W_3*d(S_i)$。

因此，将完成任务的时间跨度和节点负载均衡度综合起来作为衡量指标的目标函数为 $F(X)=LB(X)*C(X)$。

蚁群算法中的信息素代表了对某一种资源分配方式的期待。在初始化信息素矩阵时，利用贪心算法的思想找出处理能力最好的节点，赋予其更多的信息素。使用各节点的 CPU 能力 p、内存容量 r 和磁盘容量 d 作为初始信息素。节点 i 的初始信息素可表示为 $\tau_i(0)=a*\frac{p_i}{p_0}+b*\frac{r_i}{r_0}+c*\frac{d_i}{d_0}$。其中，$(a+b+c)=1$，$p_0$、$r_0$、$d_0$ 分别为集群中各节点的 CPU 能力、内存容量、磁盘容量的均值。a、b、c 分别代表上述不同指标在初始信息素中所占的比重。

为了使蚁群算法在进行任务调度时在不陷入局部最优解的前提下尽可能加速收敛，引入模拟退火算法的思想对原有的蚁群算法做出一定修改。

模拟退火算法（simulated annealing algorithm）最早由 S. Kirkpatrick 等人于 1983 年提出，是一种通用概率演算法。模拟退火算法的思想来源于固体退火的场景：当一个固体被加热到很高的温度后，便进入缓慢的冷却。在其升温阶段，固体内部的粒子随着温度的不断升高而变得无序，同时内能也在不断增大；在降温阶段则恰恰相反，固体内部的粒子会逐渐趋于有序。当冷却时间足够长，冷却过程中的任一温度下固体都处于热平衡状态。最终固体冷却到最低温度时其内能也达到最小[17-19]。自然界总是趋于能量最低，而分子热运动则趋于破坏这种能量最低的状态。基于这一理论，通过热平衡状态把内能最小化作为优化目标的算法就是模拟退火算法[20]。

在当前状态 i 生成状态 j 后，如果新的状态 j 的内能小于原状态 i 的内能，即 $E_j<E_i$，则将新状态 j 作为当前的状态；否则，依概率 $e^{-(E_j-E_i)/kT}$。

接受新状态 j，其中 k 为玻尔兹曼常数，这一准则称为 Metropolis 准则。

若将温度 T 作为控制参数，待优化的目标函数 f 作为内能 E，固体处于某一温度 T_x 下的状态对应一个解 x，则可

将固体退火的思想应用于求解优化问题中。通过控制参数 T 的逐渐降低，目标函数 f 也随之降低，直至趋于全局最小值[21]。

模拟退火算法的初始参数包括：

1）控制参数 T 的初始值 T_0，即冷却开始时的温度；

2）控制参数 T 的衰减函数，最常见的一种衰减函数为 $T_{k+1}=<\lambda T_k, k=0,1,2,\cdots$。其中 λ 为常数，也称为退火系数，通常其取值范围为 $[0.5, 0.99]$；

3）控制参数 T 的终止值；

4）马尔科夫链的长度 L，即任一温度 T 下的迭代参数。

模拟退火算法的基本流程如图 2 所示。

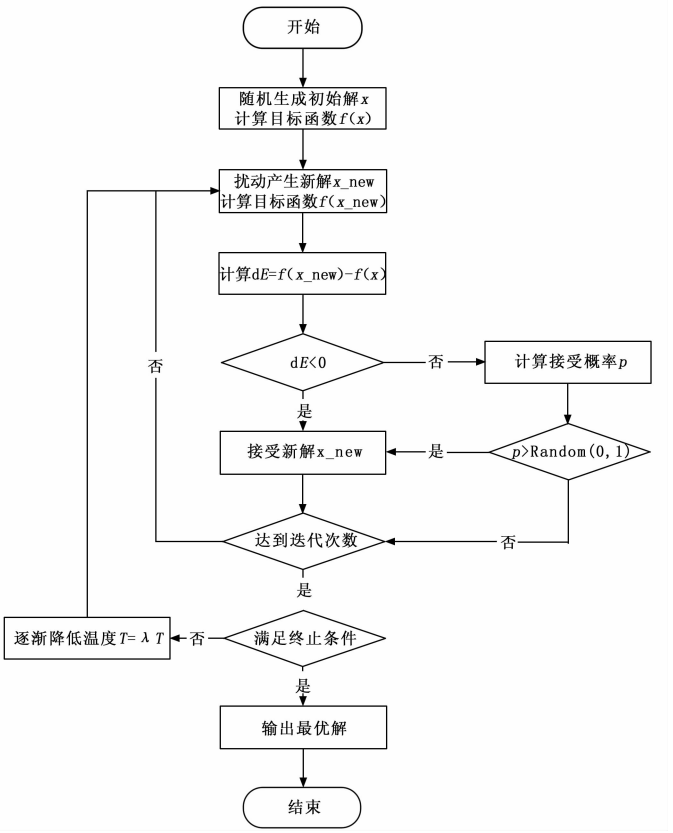

图 2 模拟退火算法流程图

引入模拟退火算法思想当蚂蚁 k 发现了一条最优路径并形成最优解 X^* 后，随机交换两个资源所执行的任务作为扰动，产生一个新的解 X，通过模拟退火算法的方式决定保留最优解 X^* 还是用新解 X 替换最优解 X^*：若 $\Delta F = F(X) - F(X^*) < 0$，则用新解 X 替换最优解 X^*；否则若 $\Delta F = F(X) - F(X^*) > 0$，则在 0～1 分布上依概率 $e^{-\Delta F/T}$，决定是否用新解 X 替换最优解 X^*。其中，T 为退火温度，初始值为 T_0，终止值为 T_{end}。退火温度 T 按照 $T(t+1)=\alpha T(t)$ 进行迭代，其中 α 为退火系数。退火温度衰减到终止值时结束模拟退火过程，并依照蚁群算法信息素更新公式利用模拟退火过程得出的最优解 X^* 更新路径上信息素，结束本次循环。当循环执行次数达到蚁群算法的最大迭代次数上限时，结束整个循环并输出结果。

基于改进蚁群算法的任务调度算法流程如图 3 所示。

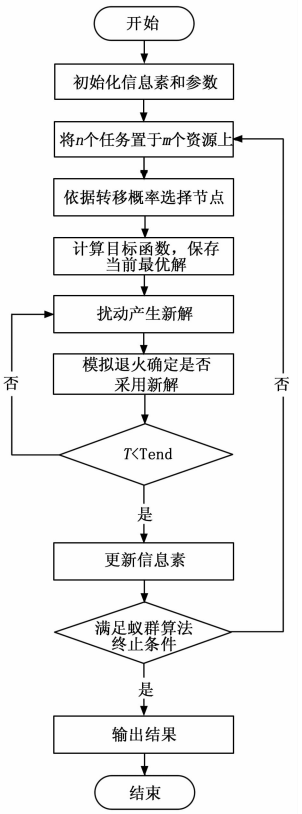

图 3 基于改进蚁群算法的负载调度算法流程图

3 仿真实验

为验证基于蚁群算法的整合负载自适应分配算法在遥感信息多任务服务链在线优化场景中的应用效果，通过云计算仿真平台，设定贴近遥感信息多任务服务链处理场景的任务负载、系统资源和调度算法，验证本文提出的基于蚁群算法的整合负载自适应分配算法的优化效果，并与传统遥感信息多任务服务调度方法的效果进行比对。

3.1 仿真平台介绍

仿真实验是在云计算仿真平台 CloudSim 环境下进行的。云计算仿真平台 CloudSim 是一个基于离散事件的云计算仿真工具箱。CloudSim 是在网格仿真平台 GridSim 的基础上开发的，提供各种异构云计算资源、用户、调度算法等的模拟与仿真。

基于 CloudSim 平台仿真的主要步骤如下：

1）仿真初始化；

2）创建云计算资源和任务；

3）编写任务调度算法；

4）仿真启动；

5）仿真结束，输出任务完成情况，对结果进行分析与评价。

3.2 实验场景设计

基于改进蚁群算法的任务调度算法中的 a，b，c 分别代表

节点的计算能力、内存和磁盘容量的重要性。一般来说，任务的执行受 CPU 的处理能力的影响较大，因此将 a,b,c 三个参数的值分别设为 0.6，0.2，0.2。算法中蚁群算法部分的信息素挥发系数影响着算法的全局搜索能力和收敛速度两项指标，综合考虑后将其确定为 0.5。参数 α 与 β 分别决定了信息素和匹配因子的重要程度，蚂蚁数量 m 决定迭代的次数，退火系数 λ 决定退火的速度。根据经验取参数 $\alpha=1$,$\beta=2$,$m=10$,$\lambda=0.95$,$T_0=10\ 000$,$T_{end}=1\ 000$。

表 1　算法各项参数设置

参数	取值
α	1
β	2
m	10
λ	0.95
T_0	10 000
T_{end}	1 000

为对比不同算法的任务调度效果，选择轮询算法、遗传算法和改进蚁群算法进行比较。在仿真开始前，利用 CloudSim Plus 创建 20 个节点。为模拟真实环境，每个节点具有的计算能力采用随机生成，分布于 1 000 至 6 000 MIPS 之间。根据任务数的不同，随机生成任务长度同样采用随机的方式生成，分布于 50 M 至 200 M 之间。根据任务数量的不同将实验分成五组，五组实验的任务数分别为 100、200、300、400、500。为尽量降低随机数据的偶然性对实验结果的影响，多次进行实验并取平均值作为每组实验的结果。

表 2　CloudSim 属性设置

实体类型	参数	值
虚拟机	VM 数量	20
	MIPS	1 000－6 000
	VM RAM(GB)	256－2 048
	带宽(Mbps)	50－200
	PE 数量	1－4
数据中心	数据中心数量	10
	主机数量	2－6

3.3　实验结果及分析

图 4 为使用不同种算法调度各组任务的总完成时间。

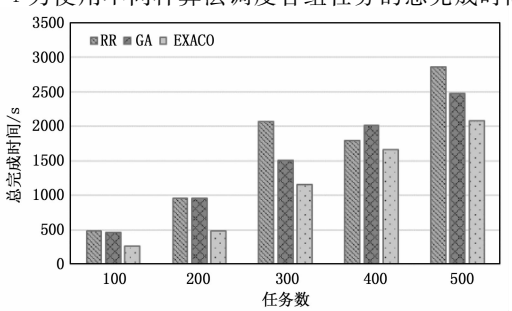

图 4　各算法执行不同任务数的总完成时间

图 4 中横坐标表示每组实验的任务数，纵坐标表示各组任务在分别不同调度算法下的总完成时间。由图可知，基于改进遗传算法的调度结果较其他对比算法具有更低的总完成时间。同时，随着任务数的增多，该算法的优势越来越显著。

为比较各节点在执行任务中的负载情况，在实验过程中记录下每个节点运行任务的总长度，针对每个节点计算其负载的任务总长度与其计算能力的比值。以执行 100 个任务时产生的实验数据为例，将得出的结果绘制如下统计图：

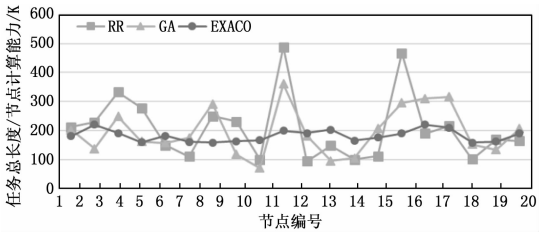

图 5　任务数为 100 个时各算法的节点负载均衡情况

图 5 中，以“■”标记的折线反映了使用轮询算法调度任务后各节点的负载情况，以“●”标记的折线反映了使用遗传算法调度任务后各节点的负载情况，以“▲”标记的折线反映了使用改进后的遗传算法调度任务后各节点的负载情况。由上图可知，轮询算法与蚁群算法在进行任务调度时由于没有充分考虑节点的负载均衡问题，各节点的负载水平相差很大，这一点在使用遗传算法后得到了些许改善，但我们可以很明显地看到使用改进后的蚁群算法时，各节点的负载程度相对均衡。为定量对比上述算法在负载均衡方面的表现，可以使用标准差来分析上述数据，如下公式：

$$STD=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i-\overline{x})^2}$$

其中：x_i 是图 5 中纵坐标的值，$\overline{x}$ 为 x_i 的平均值，n 为节点个数。使用该公式计算各算法的负载均衡程度 $STD_{RR}=113.66$,$STD_{GA}=82.49$,$STD_{EXACO}=20.56$。

分析以上实验结果可以得出，遗传算法和改进的蚁群算法相较于轮询算法而言，在任务总完成时间和节点负载均衡性上都有较好的表现。轮询算法由于其简单的策略，不可避免地会出现将较长的任务分配给计算能力较弱的节点的情况，造成任务的堆积，而延长了全部任务的总完成时间，这样还会造成系统中各节点负载不均衡。遗传算法相较于轮询算法能够在一定程度上优化任务的分配，使得全部任务的总完成时间减少。基于改进的蚁群算法较好地避免了蚁群算法容易陷入局部最优解的问题，并且进一步减少了任务总完成时间。同时，在任务分配时考虑了节点实际负载情况，使得系统中各节点的负载趋于平衡。实验结果表明，基于改进蚁群算法的负载调度算法在针对遥感信息服务链处理调度的场景下具备更优的效果。

4　结束语

本文提出了一种改进蚁群算法（EXACO）的整合负载

自适应分配算法进行遥感信息服务链的优化与服务资源配置，该算法完成遥感信息多任务处理服务链的计算任务分配，提升数据处理系统整体的处理效率和负载均衡度，能够支撑面向遥感信息支援保障的处理服务链动态构建技术的研究。

参考文献：

[1] 郭成城. 一种异构 Web 服务器集群动态均衡算法 [J]. 计算机学报，2005，28 (2)：179－184.

[2] 邓成玉. 动态负载均衡策略及相关模型研究 [J]. 计算机工程与应用，2001，47 (8)：131－134.

[3] 任国庆. 基于内容的 Web 服务器动态负载算法 [J]. 计算机工程，2010，36 (13)：82－83.

[4] 李　国，曲文丽. 一种改进的周期自适应动态负载均衡算法 [J]. 小型微型计算机系统，2015，36 (7)：1476－1480.

[5] 王　玥，蔡皖东. 一种自适应动态负载均衡算法 [J]. 计算机工程与应用，2006 (21)：125－127.

[6] 李文中. 服务组合中的一种自适应负载均衡算法 [J]. 软件学报，2006 (5)：1068－1077.

[7] 刘亚秋，孙新越，景维鹏. 一种异构云环境下的负载均衡算法 [J]. 计算机应用研究，2018，35 (12)：259－262.

[8] WANG Yonghui. The improvement and implement of performance optimation and load balancing in high performance Nginx Web server [D]. Chengdu：University of Electronic Science and Technology of China，2015 (in Chinese).

[9] 黄伟华，马　中，戴新发，等. 一种特征加权模糊聚类的负载均衡算法 [J]. 西安电子科技大学学报 (自然科学版)，2017，44 (2)：127－132.

[10] LAKHINA U，SINGH N，JANGRA A. An efficient load balancing algorithm for cloud computing using dynamc cluster mechanism [C] //International Conference on Computing for Sustainable Global Development. IEEE，2016.

[11] WANG H，ZHANG H，WANG K，et al. Off－road Path Planning Based on Improved Ant Colony Algorithm [J]. Wireless Personal Communications，2018 (3)：1－17.

[12] LI G，CHOU W. Path planning for mobile robot using self－adaptive learning particle swarm optimization [J]. Science China Information Sciences，2018，61 (5)：052204.

[13] ZHANG Z，HU F，ZHANG N. Ant colony algorithm for satellite control scheduling problem [J]. Applied Intelligence，2018：1－11.

[14] LIAO Q，GUO Y，TU Y，et al. Fidelity－Based Ant Colony Algorithm with Q－learning of Quantum System [J]. International Journal of Theoretical Physics，2018，57 (3)：862－876.

[15] ZENG Y，LIU D，HOU X. Complex Vehicle Scheduling Optimization Problem Based on Improved Ant Colony Algorithm [J]. Lecture Notes in Electrical Engineering，2013，211：805－812.

[16] SHAN B M，ZHANG D X. Path planning of robot based on ant colony optimization algorithm [J]. Applied Mechanics and Materials，2014，614：199－202.

[17] CHUANG Menghsuan. Optimal Chiller. By Genetic Algorithm For Reducing Energy Consumption [J]. Energy and Buildings，2005 (37)：147－155.

[18] 庞　峰. 模拟退火算法的原理及算法在优化问题上的应用 [D]. 长春：吉林大学，2005.

[19] Whitley D. A genetic algorithm tutorial [J]. Statistics and computing. 1994，4 (2)：65－85.

[20] KONAK A，COIT D W，SMITH A E. Multi－objective optimization using genetic algorithms：A tutorial [J]. Reliability Engineering&System Safety. 2006，91 (9)：992－1007.

[21] ROTHLAUF F. Representations for Genetic and Evolutionary Algorithms [M]. Springer Science&Business Media，2006.

(上接第 182 页)

[7] 张丽军，潘家伟，曹志勇. 基于机载激光雷达技术的山区高速公路勘测设计方案探究 [J]. 公路工程，2018，43 (6)：169－173.

[8] 刘晓文，徐　工，杨晓琳. 地面三维激光扫描与近景摄影测量技术集成应用 [J]. 山东理工大学学报 (自然科学版)，2018，32 (5)：53－57.

[9] 张舜德，卢秉恒，丁玉成. 光学三维形面分区域测量数据的拼接研究 [J]. 中国激光，2001 (6)：533－536.

[10] 黄　潜，王泽勇，李金龙，等. 三维点云快速拼接方法研究 [J]. 信息技术，2018，42 (7)：77－80.

[11] 杜科林. 基于 BBF 改进的 kd－tree 算法及其在点云配准中的应用研究 [D]. 南昌：东华理工大学，2018.

[12] 黄　兴，应群伟. 应用激光雷达与相机信息融合的障碍物识别 [J]. 计算机测量与控制，2020，28 (1)：184－188.

[13] MATURANA D，SCHERER S. VoxNet：A 3D Convolutional Neural Network for real－time object recognition [C] // 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE，2015.

[14] QI C R，SU H，MO K，et al. PointNet：Deep Learning on Point Sets for 3D Classification and Segmentation [C] // 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE，2017.

[15] 孙大林. 路面点云建模并行算法及缺陷检测方法研究 [D]. 哈尔滨：哈尔滨工业大学，2018.

[16] 徐中源. 基于实测路面信息的三维虚拟路面重构研究 [D]. 青岛：青岛大学，2018.

[17] 吴启芳. 基于点云的路面建模技术研究 [D]. 哈尔滨：哈尔滨工业大学，2015.

[18] 韩　超. 基于无人机航拍图像的三维重建 SfM 算法研究 [D]. 呼和浩特：内蒙古工业大学，2019.

[19] 冯亦东，孙　跃. 基于 SURF 特征提取和 FLANN 搜索的图像匹配算法 [J]. 图学学报，2015，36 (4)：650－654.

[20] ENGELMANN F，KONTOGIANNI T，HERMANS A，et al. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds [C] // 2017 IEEE International Conference on Computer Vision Workshop (ICCVW). IEEE，2017.

[21] ROYNARD X. Paris－Lille－3D：a large and high－quality ground truth urban point cloud dataset for automatic segmentation and classification [J]. The International journal of robotics research，2017，37 (6)：545－557.