

文章编号:1001-9081(2012)04-1003-06

doi:10.3724/SP.J.1087.2012.01003

# 基于云计算平台的 CO<sub>2</sub> 空间数据融合算法

胡军国<sup>\*</sup>, 邱亨年

(浙江农林大学 信息工程学院, 浙江 临安 311300)

(\*通信作者电子邮箱 hawkhj@163.com)

**摘要:**为了对移动传感器网络采集到的时间、空间不确定的海量 CO<sub>2</sub> 浓度数据进行融合,首先对采集的 CO<sub>2</sub> 数据进行分析,把测试区域分成  $m \times n$  个网格,分析从每个网格取一个有效值来表示 CO<sub>2</sub> 浓度分布。然后根据云计算强大的计算能力,提出组合云模型,设计普通云、繁殖云、视觉云和邻接云,以云内相对独立运行和云间相互作用形成分布式并行计算机制。接着改造蚁群家族,设计普通蚂蚁、繁殖蚂蚁、视觉蚂蚁和邻接蚂蚁。各类蚂蚁分配到不同的云朵中,并按自身的规则运行,各类蚂蚁彼此配合工作,实现信息素和最优解在云内部局部交换和通过云服务器在云朵之间全局交换相结合。最后模拟生成有关临安的 11080 个数据,利用 Clounding V2 模拟平台进行大量实验,实验表明算法在 105 次寻优后基本趋于稳定,寻优能力是单机算法的 60 倍左右,并且普通云、繁殖云、视觉云和邻接云中的蚂蚁数量比设为 2:2:1:1 性能表现出最佳。

**关键词:**云计算;蚁群优化;CO<sub>2</sub> 分析;数据融合;并行计算

**中图分类号:** TP391    **文献标志码:**A

## Spatial data fusion algorithm of CO<sub>2</sub> based on cloud computing platform

HU Jun-guo<sup>\*</sup>, QI Heng-nian

(College of Information Engineering, Zhejiang Agriculture and Forestry University, Lin'an Zhejiang 311300, China)

**Abstract:** In order to fuse the massive CO<sub>2</sub> data, which were collected by the mobile sensor network from uncertain time and space, the paper analyzed the collected data. First, the test area was divided into  $m \times n$  grids, and CO<sub>2</sub> concentration was analyzed from each valid data of every grid. Second, according to the strong computing power of cloud computing, the paper put forward combined cloud model and design common clouds, breeding clouds, visual clouds and adjacent clouds. They ran relatively independently in the cloud and interacted with each cloud, forming distributed parallel computing system. Third, the paper modified the ants family, and designed common ants, breeding ants, visual ants and adjacent ants. All kinds of ants, which walked by their own rules, were assigned to different clouds and worked together harmoniously, with the pheromones and the optimal solution exchanging in local cloud and between global clouds by the cloud server. Finally, in Ling'an of Zhejiang province the authors sampled 11080 data, and used Clounding V2 simulation platform to do a lot of experiments. The result shows that after searching 105 times the algorithm reaches stabilization, the optimization capability is 60 times as strong as the single algorithm, and that the ants in common clouds, breeding clouds, visual clouds and adjacent clouds are set 2:2:1:1 can get the best performance.

**Key words:** cloud computing; ant colony algorithm; Ant Colony Optimization (ACO); CO<sub>2</sub> analyzing; data fusion; parallel computing

## 0 引言

CO<sub>2</sub> 是大气中主要的温室效应气体,其大量排放已造成全球气候变暖。据统计,从 1906 年起全球平均气温已增加 0.74°C,若温度再升高 2°C,将对人类造成毁灭性的灾难<sup>[1]</sup>。控制 CO<sub>2</sub> 排放直接关系到人类的生存和发展,已引起全球的极大关注。2009 年《哥本哈根协议》中就发达国家实行强制减排,到 2020 年要求减排率达到 1990 年的 25%~40%,对发展中国家采取自主减排要求。中国作为负责任的大国,积极主动减排,承诺到 2020 年单位国内生产总值 CO<sub>2</sub> 排放比 2005 年下降 40%~45%。2009 年底,国务院总理温家宝在国务院常务会议中,进一步强调应对气候变化工作的重要性,并提出相应的政策措施和具体行动<sup>[2]</sup>。但是,如何监测和控制 CO<sub>2</sub> 排放,各地没有统一标准,执行难度非常大。

近年来,在国家政策的大力引导下,不少学者提出了监测

大气 CO<sub>2</sub> 浓度的方案,主要有三种措施:

- 1) 在指定点建立观察站,进行长年累月的监测,并记录监测数据<sup>[3~5]</sup>,但只能在某个点上监测 CO<sub>2</sub> 浓度;
- 2) 采用网格化测量方法<sup>[6~8]</sup>,对监测区划分成 N 个网格,测出每个网格的 CO<sub>2</sub> 浓度,但网格监测需人工进行,无法实行长期实时监测;
- 3) 以我校为首的低碳与物联网联合实验室提出采用 Ad-Hoc 无线传感器网络监测 CO<sub>2</sub> 浓度<sup>[9]</sup>,虽能实现长期、实时监测,但现在还只能应用于非常小的区域。

针对此,本文提出基于移动车载的 CO<sub>2</sub> 浓度监测系统。由于移动系统的采集点会不断变动,因此获得的是时间上不同步、空间上不固定的海量的 CO<sub>2</sub> 浓度数据。如何对这些大规模、时空不确定数据进行融合,是本文要解决的关键问题。

收稿日期:2011-10-31;修回日期:2011-12-15。    基金项目:国家 973 计划项目(2011CB302705);浙江省重点项目(2011C21002)。

作者简介:胡军国(1978-),男,浙江临安人,副教授,硕士,CCF 会员,主要研究方向:智能计算、分布式计算; 邱亨年(1975-),男,浙江临安人,教授,博士,主要研究方向:模式识别、图像处理。

## 1 问题描述与模型建立

为了能在不同时间和空间获取 CO<sub>2</sub> 空间浓度数据,可以把移动车载 CO<sub>2</sub> 浓度监测系统架构在城市出租车、公交车等移动平台上。随着移动平台的运动,能采集到不同点的 CO<sub>2</sub> 浓度。设有 M 个 CO<sub>2</sub> 浓度监测终端架构在移动平台上,在 T 时间内每个终端采集 N 次数据,由于 M 个终端同时在工作,在 T 时间内共采集到 M × N 个不同时间、不同空间的 CO<sub>2</sub> 浓度数据。但是,如何求解这个城市 CO<sub>2</sub> 浓度分布情况呢?或者说,如何从这 M × N 个点中取出有效点来表示城市 CO<sub>2</sub> 浓度分布情况呢?

众所周知,随着日照强度的提高,植物的光合作用增强,能提高吸收 CO<sub>2</sub> 的能力;汽车排放尾气、工厂燃烧煤等都释放 CO<sub>2</sub>,并且 CO<sub>2</sub> 分子有扩散能力,会从排放源向四周扩散。也就是说,CO<sub>2</sub> 浓度会随时间和空间而变化。本文以 CO<sub>2</sub> 浓度时空平滑度因子表示 CO<sub>2</sub> 变化程度,并作如下定义。

**定义 1** 设 T<sub>1</sub> 时刻获得 A(x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>) 点的 CO<sub>2</sub> 浓度为 C<sub>A</sub>,设 T<sub>2</sub> 时刻获得 B(x<sub>2</sub>, y<sub>2</sub>, z<sub>2</sub>) 点的 CO<sub>2</sub> 浓度为 C<sub>B</sub>,则 A、B 点 CO<sub>2</sub> 浓度时空平滑度记为 S<sub>AB</sub>:

$$S_{AB} = \sqrt{\left(\frac{|\Delta c|}{E}\right)^2 + \left(\frac{F}{|\Delta d|}\right)^2 + \left(\frac{|\Delta t|}{G}\right)^2}$$

其中:Δc 表示 C<sub>B</sub> 与 C<sub>A</sub> 的 CO<sub>2</sub> 浓度差值,Δd 表示 AB 两点距离,Δt 表示 T<sub>1</sub> 和 T<sub>2</sub> 是时间间隔,E、F、G 是一个给定的常量。

如何从 M × N 个 CO<sub>2</sub> 浓度值中取出部分有效点,作为城市的 CO<sub>2</sub> 的分布的基础数据,使得 CO<sub>2</sub> 浓度空间分布比较平滑。把城市划分为 m × n(m × n < M × N) 个区域,每个区域记为 R<sub>i</sub>(1 ≤ i ≤ m × n),那么在 T 时间内的 M × N 个 CO<sub>2</sub> 浓度点落在不同的区域内,设区域 R<sub>i</sub> 内有 C<sub>i</sub> 个点,则有  $\sum_{i=1}^{m \times n} C_i = M \times N$ 。要使所取得的 CO<sub>2</sub> 浓度空间分布比较平滑,可以理解成从任意区域 R<sub>i</sub> 中取出一个点 R<sub>i</sub><sup>k</sup>(1 ≤ i ≤ m × n, 1 ≤ k ≤ C<sub>i</sub>),并组成 m × n 个点,使得  $\sum S_{R_i^k R_j^l}(1 \leq i, j \leq m \times n, i \neq j, 1 \leq k \leq C_i, 1 \leq l \leq C_j)$  最小,即所取的 m × n 个 CO<sub>2</sub> 点组成的平滑度最小。

## 2 研究方案确定与算法实现

### 2.1 研究方案

从第 2 章的分析中可以看出:从 m × n 个区域中各取出一个 CO<sub>2</sub> 浓度值,使得这些点的平滑度最小,是一个 NP 难问题。解决 NP 难问题的方法很多,如模拟退火算法、遗传算法、蚁群算法、神经网络算法等。但这类算法运算时间比较长,很难保证在较短时间内融合海量的 CO<sub>2</sub> 数据,确保分布情况的实时表现。近年来,随着云计算的兴起,通过云计算平台实现密集型数据处理已经成为可能。郑海等<sup>[10]</sup>提出了兼顾时间开销、数据依赖性和负载均衡三方面指标的数据布局策略与方法,在云计算平台中有效降低跨数据中心数据传输导致的时间开销;赵俊华等<sup>[11]</sup>提出建立基于云计算的电力系统计算平台,从云计算的物理组成、系统架构、软件技术等方面详细讨论分析电力系统云计算平台的实现;王鹏等<sup>[12]</sup>分析了在云计算平台中数据密集型计算的特点,并指出云计算编程模型要解决的基本问题,分析国际上代表性的编程模型,并对这些编程模型的特点进行了比较,为数据密集型计算提

供强有力的技术支撑;Hanawa 等<sup>[13]</sup>提出 D-Cloud 的云计算模式,并实现测试并行软件和分布式软件的性能,此模式不仅可对云计算系统进行自动配置还可同时测试多种软件实例;Google 公司构造 MapReduce 编程规范<sup>[14]</sup>来简化云计算平台的编程,用 Map(映射)和 Reduce(化简)两个简单的函数来构成运算基本单元,用户只需提供自己的 Map 函数以及 Reduce 函数即可并行处理海量数据。

综上所述,云计算平台已经能实现海量密集性数据运算。但针对 NP 难问题,若没有合理的算法,云计算平台也无能为力。而蚂蚁优化算法是非常优秀的智能算法,尤其有很强的并行能力,非常适合在云计算环境中,以解决复杂的智能计算问题。不少学者已经对蚂蚁算法的分布处理和并行技术进行研究,为其在云计算平台中应用提供基础。Chu 等<sup>[15]</sup>提出并设计实现了蚁群系统(Ant Colony System, ACS)的并行处理,它将一个蚁群平均分为多个小的蚁群,每个蚁群对应一个处理器,并行求解每个蚁群有各自任务,然后用三种通信方法更新信息素。并在 TSP 问题中应用,取得比 ACS 和蚂蚁系统(Ant System, AS)更好的效果。陈峻等<sup>[16]</sup>提出了并行蚁群算法中处理机间信息交流的两种策略,使得各处理机能够自适应地选择其他处理机以进行信息交换和相应信息素的全局更新。并且还提出了一种确定处理机之间进行信息交流的时间的策略,可以根据解的分布情况自适应地确定信息交流的时间,以取得全局收敛速度和解的多样性之间的平衡。吕强等<sup>[17]</sup>提出通过多个蚁群的并行协作来处理任务,蚂蚁在解的构造过程和信息素更新过程中共享同一个信息素矩阵。Wang 等<sup>[18]</sup>提出基于交叉熵的分布式蚁群算法来解决网络中的多限制路由问题,并在模拟仿真中有很快速的求解功能。Taškova 等<sup>[19]</sup>提出蚁群优化算法的分布式处理应用,并用此解决网络分割问题,取得比较好的性能和作用。Sameh 等<sup>[20]</sup>提出结合主从处理器和信息交换方法,主处理器产生从处理器进程,并获取从处理器的最优解。局部搜索和信息素更新都依赖于从处理器。从处理器独立交換解,并降低主处理器通信处理瓶颈。

借鉴众多学者的研究成果,本研究在云计算环境下,利用蚁群算法的并行寻优能力,以融合海量的 CO<sub>2</sub> 数据。论文根据蚁群优化算法的特点,提出组合云模型:设计普通云、繁殖云、视觉云和邻接云,并建立分布式并行蚁群算法,相应地改造蚂蚁家族,设计普通蚂蚁、繁殖蚂蚁、视觉蚂蚁和邻接蚂蚁,蚂蚁在各自云朵中的主机上运行,并以自身规则行走。蚂蚁从 R<sub>i</sub><sup>k</sup>(表示第 R<sub>i</sub> 个区域中的第 k 个点) 行走到 R<sub>j</sub><sup>m</sup>,经过所有区域完成一轮循环,并在云朵内部更新信息素,且对优越的个体进行云朵之间信息素更新,使局部更新和全局更新相协调,最终获取最优解,实现 CO<sub>2</sub> 空间浓度平滑分布。

### 2.2 设计组合云计算模型

云计算环境是在整个 Internet 中进行,云计算控制中心根据用户的请求,将计算任务分成若干子任务,再动态地将各子任务分配给云计算平台的计算设备。各子任务完成后,其计算结果通过 Internet 重新汇总到控制中心,最后再反馈给用户。但对于一个完整的计算任务,如何对其进行任务分解,如何把子任务分配到云计算平台的各个计算终端,是一个巨大的挑战。本文针对计算 CO<sub>2</sub> 分布问题的特殊性,提出组合云模型,设计普通云、繁殖云、邻接云和视觉云,每朵云有自己的计算模式。并且,普通蚂蚁位于普通云,繁殖蚂蚁位于繁殖

云,邻接蚂蚁位于邻接云,视觉蚂蚁位于视觉云。每朵云自主形成计算群,信息素在云内部局部交换和通过云服务器在云朵之间交换相结合。如图1,云终端提交的CO<sub>2</sub>分布计算任务被分配到普通云、繁殖云、视觉云和邻接云中计算。

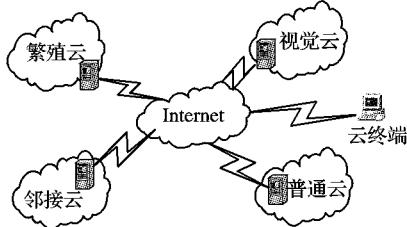


图1 云计算模型

为了描述方便,对文中出现的词汇和表达作个统一,全局指整个云平台,局部指某朵云,个体指独立主机。以下分别对普通云、繁殖云、视觉云和邻接云作定义。

**定义2 普通云。**普通蚂蚁在普通云中运行。云朵内每台主机存在n只普通蚂蚁,其以信息素为主要指导因素计算,每只蚂蚁完成一轮运算更新个体信息素;在局部周期内以云朵内部信息素为主,局部计算周期结束后,云内主机都需向普通云服务器汇集信息素和最优解,并更新云内局部信息素和最优解;在全局周期内以各云朵产生的全局信息素(来自全局最优解)为指导。并且,普通云服务器可获得全局云服务器的信息素,云内各主机又能从局部云服务器获取共享信息素,以有效更新个体私有信息素。

**定义3 繁殖云。**繁殖蚂蚁在繁殖云朵中运行,云中最优解可与全局最优解相互更新。云朵内的每台主机存在一只繁殖蚂蚁,蚂蚁按自身规则边运行边繁殖,一轮结束后产生的优秀解向繁殖云服务器汇报,同时获取局部最优解作为繁殖蚂蚁下轮行走路径。

**定义4 视觉云。**视觉蚂蚁在视觉云中运行。视觉云相对比较独立,不受其他云影响。云朵内每台主机运行n只视觉蚂蚁,其以自身规则运算,求出的优秀解在云内汇集,并向普通云和繁殖云提供局部最优解。

**定义5 邻接云。**邻接蚂蚁在邻接云中运行,其计算模式与视觉云类似,差异在于云内存在的是邻接蚂蚁,云内主机是按邻接蚂蚁规则进行。

**定义6** 每朵云内部的服务器称为局部云服务器,它管理云朵内的各主机;整个云计算平台的服务器称为全局云服务器,它管理每朵云的局部服务器。

每朵云在其内部独立计算,云内主机每轮计算结束就向局部云服务器提供最优解,同时各主机又会从局部云服务器获得云内最优解;局部云服务器根据解的质量向全局云服务器汇报最优解,并且全局云服务器也会提醒局部云服务器更新当前的最优解。普通云中以信息素指导工作,各个主机内存有个体私有信息素,也可访问局部共享信息素,并且信息素更新与最优解同步。因此,云朵内部相对独立运行与云朵之间相互作用,实现局部最优与全局最优并进。

### 2.3 设计分组蚁群优化算法

为了使蚁群优化算法能和云计算平台相适应,需对蚂蚁进行分布式并行处理,把其分配到不同的云朵中,同时要让各云朵内的蚂蚁并行运算。本文对基本蚁群算法进行改进,在蚂蚁家族中设立四种不同类型的蚂蚁,四种蚂蚁分别分布于不同的云,并行运算,共同协作,最优解共享,最终完成任务。

其中:第一类蚂蚁具有繁殖功能,称为繁殖蚂蚁;第二类蚂蚁有视觉功能,称为视觉蚂蚁;第三类蚂蚁能感知邻接区域能力,称为邻接蚂蚁;最后一类是普通蚂蚁。

#### 2.3.1 问题说明

根据第1章所描述的,在T时刻内获取M×N个CO<sub>2</sub>浓度值,这M×N个数据被分成m×n个区域,如图2所示。每个区域有C<sub>i</sub>个数据,即R<sub>1</sub>区域有C<sub>1</sub>个点,R<sub>2</sub>有C<sub>2</sub>个点,...,R<sub>mn</sub>有C<sub>mn</sub>个点。现要从R<sub>1</sub>,R<sub>2</sub>,...,R<sub>i</sub>,...,R<sub>mn</sub>区域中各取一个数据组成m×n个数据,使其所组成的CO<sub>2</sub>分布最平滑。在本文中假设蚂蚁在寻优中是从第R<sub>i</sub>个区域走向与其相邻接的区域,如图2,将M×N个数据划分为m×n个区域,区域按R<sub>1</sub>,R<sub>2</sub>,...,R<sub>mn</sub>编号。

**定义7** 蚂蚁从R<sub>i</sub>区域选择点i,从R<sub>j</sub>区域选择点j,以此类推,直到走完R<sub>mn</sub>区域的k点,这样所构成的行程称为蚂蚁路径或蚂蚁行程。

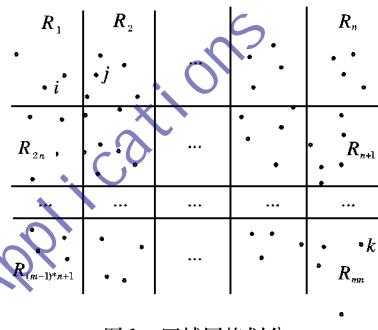


图2 区域网格划分

#### 2.3.2 普通蚂蚁

##### 1) 普通蚂蚁定义。

在普通云中运行的蚂蚁称为普通蚂蚁,它能感知信息素,并根据各条路径上的信息量决定其转移方向。普通蚂蚁的信息素来自于本主机中的蚂蚁、普通云内的蚂蚁和其他云中表现优秀的蚂蚁,而对于蚂蚁自身不用考虑信息素的来源,其只需根据个体私有信息素表行走。

##### 2) 普通蚂蚁行走规则。

s<sub>ij</sub><sup>kl</sup>表示R<sub>i</sub>区域中的第k个点(设记为点R<sub>i</sub><sup>k</sup>)到R<sub>j</sub>区域中的第l个点(设记为点R<sub>j</sub><sup>l</sup>)的平滑度,则s<sub>ij</sub><sup>kl</sup>= $\sqrt{(\frac{|\Delta c|}{E})^2 + (\frac{F}{|\Delta d|})^2 + (\frac{|\Delta t|}{G})^2}$ ,其中:Δc表示R<sub>i</sub><sup>k</sup>与R<sub>j</sub><sup>l</sup>两点的CO<sub>2</sub>浓度差值,Δd表示R<sub>i</sub><sup>k</sup>R<sub>j</sub><sup>l</sup>两点距离,Δt表示R<sub>i</sub><sup>k</sup>和R<sub>j</sub><sup>l</sup>点采集CO<sub>2</sub>数据的时间间隔,E,F,G是一个给定的常量。由于普通云中各主机蚂蚁并行运行,现考虑在主机H上,有M只蚂蚁,在t时刻点R<sub>i</sub><sup>k</sup>和点R<sub>j</sub><sup>l</sup>之间的路径上的信息素浓度为τ<sub>ij</sub><sup>kl</sup>(t);在0时刻,各条路径上信息量相等,设τ<sub>ij</sub><sup>kl</sup>(0)=C(C为常数)。蚂蚁a(a=1,2,...,M)在运动过程中,根据各条路径上的信息量决定转移方向。p<sub>ij</sub><sup>kl</sup>(a,t)表示在t时刻蚂蚁a由点R<sub>i</sub><sup>k</sup>到点R<sub>j</sub><sup>l</sup>的概率:

$$p_{ij}^{kl}(a,t) = \begin{cases} \frac{[\tau_{ij}^{kl}(t)]^\alpha [\eta_{ij}^{kl}]^\beta}{\sum_{m \in R_j} [\tau_{ij}^{km}(t)]^\alpha [\eta_{ij}^{km}]^\beta}, & m \in R_j \\ 0, & m \notin R_j \end{cases} \quad (1)$$

其中:η<sub>ij</sub><sup>km</sup>为点R<sub>i</sub><sup>k</sup>和点R<sub>j</sub><sup>l</sup>之间启发信息,一般取η<sub>ij</sub><sup>km</sup>=1/s<sub>ij</sub><sup>km</sup>;α为在R<sub>i</sub><sup>k</sup>R<sub>j</sub><sup>l</sup>两点上残留信息的重要程度;β为启发信息的重要程度;m表示R<sub>j</sub>区域中的点。

经过n个时刻,M只蚂蚁在主机H上走完所有的区域,完

成一次循环。每只蚂蚁所走过的点组成一个解。随着时间的推移,路径上原有的信息素会逐步挥发掉,同时,有蚂蚁经过的路径会留下相应的信息素,因此,要对各路径上的信息素进行更新:

$$\tau_{ij}^{kl}(t+1) = (1 - \rho) * \tau_{ij}^{kl}(t) + \Delta\tau_{ij}^{kl} \quad (2)$$

$$\Delta\tau_{ij}^{kl} = \sum_{a=1}^M \Delta\tau_{ij}^{kl}(a) \quad (3)$$

其中: $\rho$  为信息素挥发系数,取值范围为  $\rho [0,1]$ ;  $\Delta\tau_{ij}^{kl}$  表示本次循环中  $R_i^k R_j^l$  两点上的信息量的增量;  $\Delta\tau_{ij}^{kl}(a)$  表示蚂蚁  $a$  在本次循环中在  $R_i^k R_j^l$  两点之间的路径上留下的信息素,其值常按以下模型计算:

$$\Delta\tau_{ij}^{kl}(a) = \begin{cases} Q/L_a, & \text{蚂蚁 } a \text{ 在本次循环中经过点 } R_i^k R_j^l \\ 0, & \text{其他} \end{cases} \quad (4)$$

其中: $Q$  为常数,  $L_a$  为蚂蚁  $a$  在本次循环中所行走路径的总  $\text{CO}_2$  平滑度和。

### 2.3.3 繁殖蚂蚁

#### 1) 繁殖蚂蚁定义。

繁殖蚂蚁分为局部繁殖蚂蚁和全局繁殖蚂蚁,它能按前一轮最优解方向行走,繁殖出的新蚂蚁能进一步优化前一轮的最优解。记繁殖云中的蚂蚁集合为  $BC$ , 存在蚂蚁  $k$ , 在主机  $H$  上运行。它从起点出发,访问了所有区域,其行程解集记为  $\{k, P_k, L_k | 1 \leq k \leq M\}$ , 其中: $k$  表示蚂蚁  $k$ ,  $P_k$  表示蚂蚁  $k$  的路径,  $L_k$  表示蚂蚁  $k$  行程的  $\text{CO}_2$  平滑度总和,  $M$  表示  $BC$  集合中蚂蚁数量。并记  $\min = \text{MIN}(k, P_k, L_k), 1 \leq k \leq M$ , 则  $\min$  表示行程中  $\text{CO}_2$  平滑度最小的蚂蚁,  $P_{\min}$  表  $BC$  集合中  $\text{CO}_2$  平滑度最小的路径,  $L_{\min}$  表示  $\text{CO}_2$  最小平滑度和,本文把  $\min$  作为局部繁殖蚂蚁。另记视觉云中的蚂蚁集合为  $VC$ , 邻接云中的蚂蚁集合为  $AC$ , 普通云中的蚂蚁集合为  $CC$ , 设  $\min_c = \text{MIN}(BC, VC, AC, CC)$  为  $BC, VC, AC, CC$  集合中平滑度最小的蚂蚁,则在云全局计算周期内,将  $\min_c$  作为繁殖蚂蚁。在云局部周期内,繁殖蚂蚁按  $P_{\min}$  路径行走;在云全局周期内,繁殖蚂蚁按  $P_{\min_c}$  路径行走。繁殖蚂蚁每经过一点就能繁殖出一只新蚂蚁,新蚂蚁按自身规则向未走过的点行走。

#### 2) 繁殖蚂蚁行走规则。

在云计算平台中,设  $\min$  为繁殖蚂蚁,其不受云内外的其他蚂蚁影响,按  $P_{\min}$  路径行走,经过点  $R_i^k (1 \leq i \leq m \times n, 1 \leq k \leq C_i)$ , 以一定的概率繁殖出邻接蚂蚁、视觉蚂蚁、普通蚂蚁中的一只蚂蚁  $A_i$ ,  $A_i$  选择不属于  $P_{\min}$  路径的点并以自身规则行走。 $A_i$  走完所有区域也作为这轮解集中的一个解。而  $\min$  继续向前走,继续繁殖,直至完成路径  $P_{\min}$ 。

### 2.3.4 视觉蚂蚁

#### 1) 视觉蚂蚁定义。

在视觉云中计算的蚂蚁称为视觉蚂蚁,可感知平滑度。它能感知当前节点下两个区域的平滑度,并以较大的概率往平滑度最小的方向行走。设视觉蚂蚁  $a$  位于当前点  $R_i^p$ ,  $\exists R_j^q, R_k^r$ , 使得  $s_{ij}^{op} + s_{jk}^{pq}$  最小,则蚂蚁  $a$  以较大的概率往  $R_j^q$  行走(其中  $R_i$  与  $R_j$  相邻,  $R_j$  与  $R_k$  相邻)。

#### 2) 视觉蚂蚁行走规则。

在视觉云中的主机  $H$  上,设蚂蚁  $a$  在  $t$  时刻位于点  $R_i^p$  时,按概率  $P_{ij}^{op}(a, t)$  选择下一个区域  $R_j$  的第  $p$  点。

$$P_{ij}^{op}(a, t) = \begin{cases} C_i / (C_i + C_j), & \text{使得 } s_{ij}^{op} + s_{jk}^{pq} \text{ 最小} \\ C_j / (C_i + C_j)(C_i - 1), & \text{使得 } s_{ij}^{op} + s_{jk}^{pq} \text{ 非最小} \end{cases} \quad (5)$$

其中: $P_{ij}^{op}(a, t)$  表示蚂蚁  $a$  在  $t$  时刻从  $R_i$  区域的第  $o$  个点走向  $R_j$  区域的第  $p$  个点的概率,  $C_i$  和  $C_j$  表示  $R_i$  和  $R_j$  区域的点的个数。

### 2.3.5 邻接蚂蚁

#### 1) 邻接蚂蚁定义。

邻接蚂蚁存在于邻接云中,它可感知与下个区域相邻接的、已走过区域的  $\text{CO}_2$  浓度,能综合考虑邻接区域情况,并以此指导选择下一个区域的点,以较大概率选择与邻接区域比较光滑的点。

#### 2) 邻接蚂蚁行走规则。

邻接蚂蚁  $a$  当前位于  $R_i$  区域中的  $k$  点,见图 3,下一步要选择  $R_j$  区域中的一点。假设与  $R_i$  相邻接的区域有  $R_d, R_i, R_e, R_f$ , 其中  $R_d$  中的  $p$  点和  $R_i$  中的  $k$  点已走过,现以较大概率选择与  $R_d^p, R_i^k$  所形成的平滑度最小的点,记为:

$$p_{ij}^{ky}(a, t) = \frac{s_{ji}^{yk} + s_{jd}^{yp}}{\sum_{x=1}^4 s_{ji}^{xk} + s_{jd}^{xp}} \quad (6)$$

其中: $y \in C_j$ ,  $p_{ij}^{ky}(a, t)$  表示蚂蚁  $a$  在  $t$  时刻从点  $R_i^k$  到  $R_j$  区域的任意一点  $y$  的概率,  $s_{ji}^{yk}$  表示点  $R_j^y$  到点  $R_i^k$  的平滑度值。

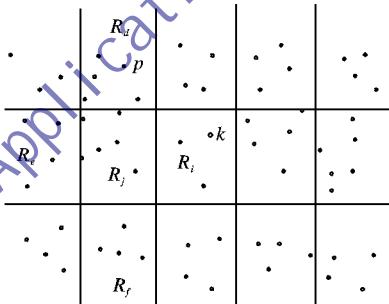


图 3 邻接蚂蚁行走规则

### 2.4 算法实现步骤

步骤 1 云终端提交计算  $\text{CO}_2$  浓度空间分布请求。

步骤 2 云服务器形成四朵云:普通云、繁殖云、视觉云和邻接云,并把蚁群算法的计算任务分配到相应的朵云中。

步骤 3 任务在云朵中的主机上并行进行。

1) 普通云中运行的是普通蚂蚁,其按私有信息素表指导行走,云中各主机蚂蚁完成一轮后,从共享信息素表中获取新的信息素,继续下一轮行走,并以当前这轮中的优秀蚂蚁更新共享信息素表。云中各主机并行,以同样方式行走。当解达到给定要求时,终止。

2) 繁殖云中运行的是繁殖蚂蚁,云中各主机蚂蚁取得云内局部最优解或全局最优解作为下一轮的繁殖蚂蚁,繁殖蚂蚁按前轮最优路径边走边繁殖,形成新的解集,再求出新的繁殖蚂蚁,继续求下轮的解集,并把每轮生成的优秀解提供给普通云更新共享信息素表。云中各主机并行,以同样方式行走。当解达到给定要求时,终止。

3) 视觉云中运行的是视觉蚂蚁,云中各主机蚂蚁能感知当前节点下两个区域的平滑度,并以较大的概率往平滑度最小的方向行走。各主机以  $n$  只蚂蚁完成所有区域为一轮,将当前轮中优秀的解向上汇报,为普通云和繁殖云提供解集。当解未达到给定要求,继续下一轮;否则就终止。

4) 邻接云中运行的是邻接蚂蚁,云中各主机蚂蚁也以自身规则行走,将产生的优秀解向上汇报,也为普通云和繁殖云提供解集,当解达到要求时终止;否则就继续循环。

步骤 4 求出满足条件的解,寻优结束,解散云集。

步骤5 通过云服务器把最优解返回给云终端,云计算结束。

### 3 实验与仿真

为了验证算法的有效性,本文以浙江省临安市为测试地点,模拟100辆公交车装载CO<sub>2</sub>采集终端,在运行中实时采集CO<sub>2</sub>数据,本次实验共采集11 080个数据,分成25×25的网格。并开发Clouding V2模拟软件,以信息技术实验室的60台主机组成云计算平台,并对实验数据进行大量测试,得出三个结论。

1)普通云、繁殖云、视觉云和邻接云中的蚂蚁数量比设为2:2:1:1性能最佳。①仅取普通云和繁殖云,云内蚂蚁数量比设3:1、2:1、1:1、1:2、1:3、1:4,每种比例都做了8次实验,每次最优解在4 821左右,计算出平均轮数,发现普通蚂蚁比繁殖蚂蚁为1:1时,在取得4 821解所需运行轮数最少,见图4。②仅取普通云和视觉云,云内蚂蚁数量比设4:1、3:1、2:1、1:1、1:2、1:3,每种比例也做了8次实验,在取得相同最优解时,普通蚂蚁比视觉蚂蚁为2:1时,所需运行轮数最少,见图5。③仅取普通云和邻接云,云内蚂蚁数量比设4:1、3:1、2:1、1:1、1:2、1:3,每种比例也做了8次实验,发现普通蚂蚁比邻接蚂蚁为2:1时,所需运行轮数最少,见图6所示。

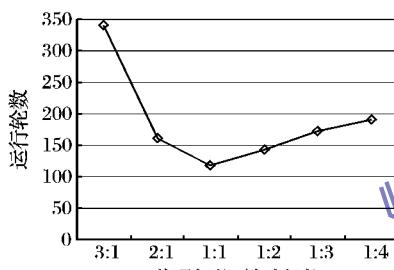


图4 普通蚂蚁与繁殖蚂蚁比例关系

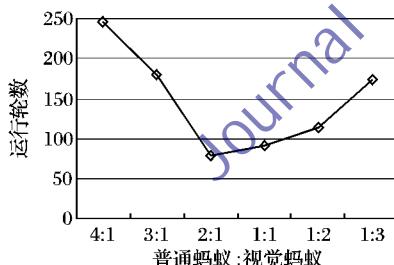


图5 普通蚂蚁与视觉蚂蚁比例关系

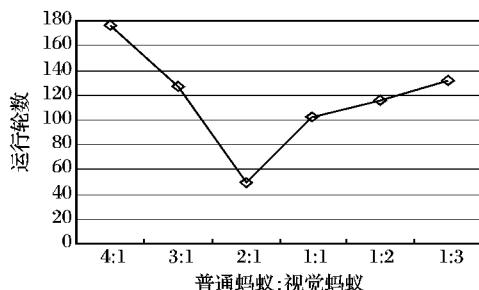


图6 普通蚂蚁与邻接蚂蚁比例关系

因为蚁群优化算法以蚂蚁信息素更新为基础,所以普通蚂蚁对算法贡献最大;而繁殖蚂蚁能在前一轮最优解基础上强化搜索,能较好地找出更优的解,其贡献也非常大;视觉蚂蚁以较大概率向与当前节点最平滑的点行走,邻接蚂蚁以较

大概率选择与附近点所构成的平滑度最小,这行走规则与实际行走相符,故对算法也有一定贡献,而它们又往往会以一定的概率突破此规则,同时也避免算法停滞。最后选择普通云、繁殖云、方向云和视觉云之间比值2:2:1:1性能最佳。

2)云计算平台的蚁群优化算法收敛速度快、稳定性高、鲁棒性好。对模拟数据进行了8次实验,结果见图7。在8次实验中,前40轮中都快速收敛,平滑度从5 600左右立即降至4 860左右,寻优能力强。因为云平台有60台主机在并行工作,求解速度快,并且繁殖云、视觉云和邻接云为普通云作巨大贡献,使得蚂蚁信息素更新更趋向于有效解。40轮之后,8次实验基本都缓慢收敛,并在105轮左右基本趋向稳定,这时主要依赖繁殖云、视觉云和邻接云中以较优解为基础所生成的随机性蚂蚁在起作用,因此寻优速度降低;但因依赖之前的较优解,寻优不至于停滞。在105轮后基本结束,算法表现出稳定性高、鲁棒性好。并随机取了一次计算结果,通过ArcGIS软件的Kriging算法实现CO<sub>2</sub>分布描图,结果见图8,颜色越深说明CO<sub>2</sub>浓度越高。

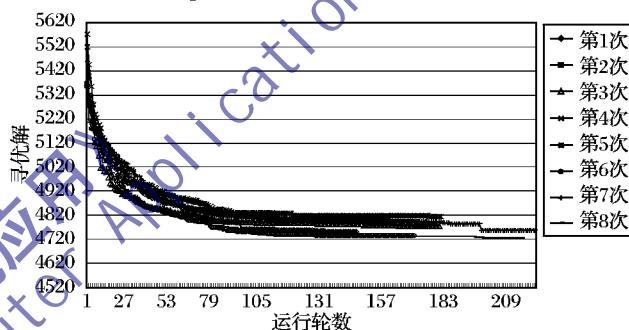


图7 云计算平台8次实验结果对比

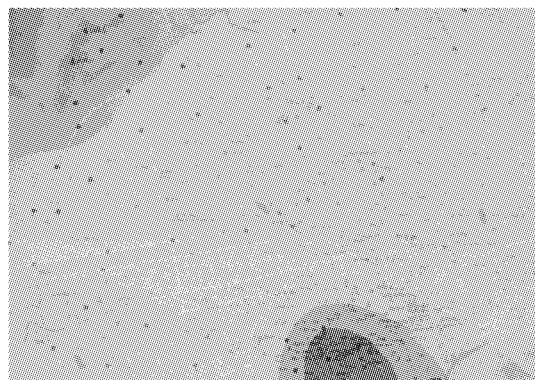


图8 CO<sub>2</sub>浓度分布

3)对Clouding V2模拟软件进行改造,使其在单机上运行。在单机上运行8次所取平均解与云平台平均解进行比较,发现单机在780轮左右取到5 200解(见图9),云计算中只需要在13轮左右就能达到(见图10);单机在10 860轮达到5 000左右并基本达到稳定,而云计算在205转达到4 750左右,可见云计算平台寻优能力远远高于单机。

### 4 结语

随着无线传感器网络的快速发展,对其采集到的海量数据进行存储与分析具有巨大的挑战。论文所提出的采用云计算平台对移动车载系统所采集的CO<sub>2</sub>数据进行融合有一定的创新性,尤其提出组合云模型,能给不同的云安排不同的任务,充分发挥每朵云的作用;同时也改造蚁群算法,使其能在云平台中运行,进一步提高组合云模型计算能力。

本算法虽然已经在60台主机上运行,但云的规模还比较小,下一步要在以下两方面所有突破:一要考虑是否能在整个

校网中运行,使其存储、计算能力更大;二要考虑算法是否可在Hadoop云计算平台运行。

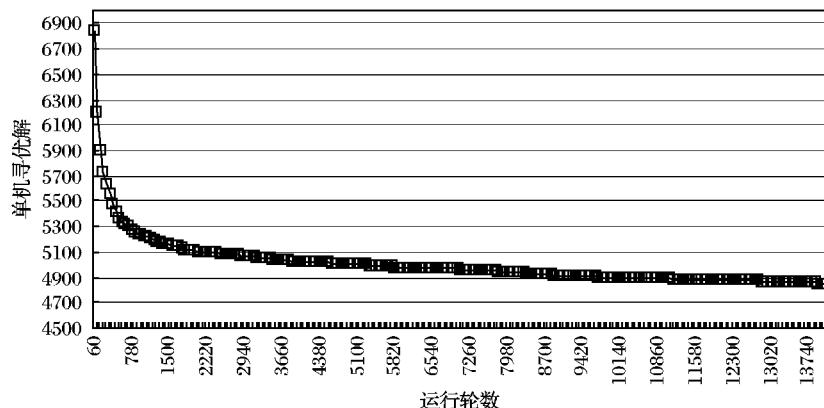


图9 单机平台运行8次平均解

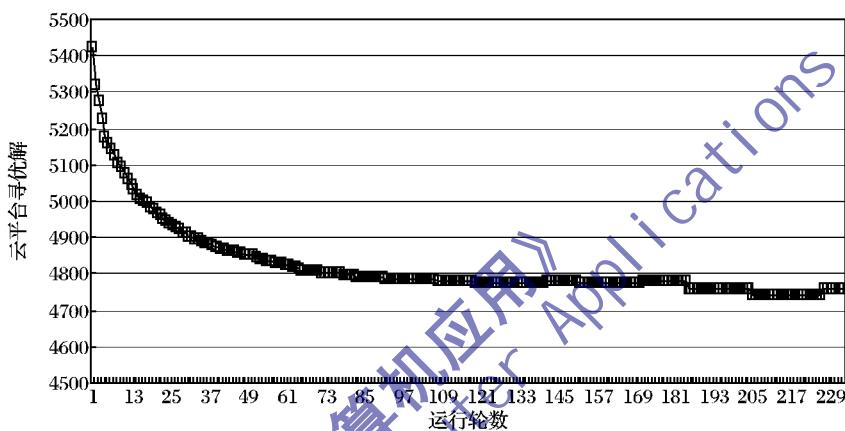


图10 云计算平台运行8次平均解

## 参考文献:

- [1] Intergovernmental Panel on Climate Change (IPCC). Climate change 2007: The physical science basis [M]. New York: Cambridge University Press, 2007: 1 - 223.
- [2] 中国承诺大幅降低单位GDP碳排放. 中国日报网 [EB/OL]. [2011-05-01]. [http://www.chinadaily.com.cn/hqzx/tjqhdh/2009-12/04/content\\_11364678.htm](http://www.chinadaily.com.cn/hqzx/tjqhdh/2009-12/04/content_11364678.htm).
- [3] PU YI-FEN. Sensitivity test of the distribution of Indian ocean surface CO<sub>2</sub> partial pressure on the physical and biochemical processes by a two-dimensional ocean carbon cycle model [J]. Climatic and Environmental Research, 2009, 14(3): 301 - 308.
- [4] 张弥, 温学发, 于贵瑞, 等. 二氧化碳储存通量对森林生态系统碳收支的影响 [J]. 应用生态学报, 2010, 21(5): 1201 - 1209.
- [5] BERBIGIER P, BONNEFOND J M, MELLIMANN P. CO<sub>2</sub> and water vapor fluxes for 2 years above Euroflux forest site [J]. Agricultural and Forest Meteorology, 2001, 10(8): 183 - 197.
- [6] BALDOCCHI D. "Breathing" of the terrestrial biosphere: Lessons learned from a global network of carbon dioxide flux measurement systems [J]. Australian Journal of Botany, 2008, 5(6): 1 - 26.
- [7] 马晓明, 王东海, 易志斌, 等. 城市大气污染物允许排放总量计算与分配方法研究 [J]. 北京大学学报: 自然科学版, 2006, 42(2): 271 - 275.
- [8] IDSO S B, IDSO C D, BALLING C R. Seasonal and diurnal variations of near-surface atmospheric CO<sub>2</sub> concentration within a residential sector of the urban CO<sub>2</sub> dome of Phoenix, AZ, USA [J]. Atmospheric Environment, 2002, 36(10): 1655 - 1660.
- [9] GreenOrbs. A long-term kilo-scale wireless sensor network system in the forest [EB/OL]. [2011-06-01]. <http://www.greenorbs.org>.
- [10] 郑海, 崔立真, 王海洋, 等. 云计算环境下面向数据密集型应用的数据布局策略与方法 [J]. 计算机学报, 2010, 33(8): 14720 - 1480.
- [11] 赵俊华, 文福拴, 薛禹胜, 等. 云计算: 构建未来电力系统的核心计算平台 [J]. 电力系统电动化, 2010, 34(15): 1 - 8.
- [12] 王鹏, 孟丹, 詹剑锋, 等. 数据密集型计算编程模型研究进展 [J]. 计算机研究与发展, 2010, 47(11): 1993 - 2010.
- [13] HANAWA T, BANZAI T, KOIZUMI H, et al. Large-scale software testing environment using cloud computing technology for dependable parallel and distributed systems [C]// 2010 The Third International Conference on Software Testing, Verification, and Validation Workshops. [S. l.]: IEEE, 2010: 6 - 10.
- [14] Wikipedia. MapReduce framework [EB/OL]. [2011-04-04]. <http://en.wikipedia.org/wiki/MapReduce>.
- [15] CHU S C, RODDICK J F, PAN J S, et al. Parallel ant colony systems [J]. International Symposium on Methodologies for Intelligent Systems, LNCS 2871. Berlin: Springer, 2003: 279 - 284.
- [16] 陈峻, 章春芳. 并行蚁群算法中的自适应交流策略 [J]. 软件学报, 2007, 18(3): 617 - 624.
- [17] 吕强, 高彦明, 钱培德. 共享信息素矩阵: 一种新的并行ACO方法 [J]. 自动化学报, 2007, 33(4): 418 - 421.
- [18] WANG HUA, WANG GANG, MA JUN. A distributed ant colony algorithm based on cross-entropy for multi-constraints QoS routing [C]// The 9th International Conference on Advanced Communication Technology. [S. l.]: IEEE, 2007: 1809 - 1814.
- [19] TAŠKOVA K, KOROŠEC P, ŠILC J. A distributed multilevel ant colonies approach for graph partitioning [J]. Informatika, 2008, 32(3): 307 - 317.
- [20] SAMEH A, AYMAN A, HASAN N. Parallel ant colony optimization [J]. International Journal of Research and Reviews in Computer Science, 2010, 1(2): 77 - 82.