QUERY ALGORITHM OF AGROMETEOROLOGICAL BIG DATA BASED ON DISTRIBUTED MATCHING / 基于分布式匹配的农业气象大数据查询算法研究

Assoc. Professor Liu X.C.*[1], Ph.D. Chen S.[2], Ph.D. Wen L.Y.[3]
[1] Department of Information Engineering, Anhui Vocational and Technical College, Hefei/China; [2] Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei/China; [3]Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown/USA
Tel: +8613956016097; E-mail: liuxiaochuanmail@126.com

Keywords: *big data in agriculture, query system, distributed matching, storage cost, delay cost*

ABSTRACT

The query system established through integer linear programming (ILP) model can rapidly respond to query operation in agrometeorological data query service. However, related existing studies have only discussed about known network layer information, extremely high time complexity exists under complicated large-scale environment, and query result cannot be acquired. Therefore, a distributed matching-based heuristic ILP query algorithm was proposed in this study for large-scale network and unknown status of network layer information to improve distributed data query capability. The lower bound of space–time cost of the query system was certified under unknown status of network layer information, under which circumstance ILP model and algorithm design were proposed formally. Finally, disjoint path allocation and node replication allocation methods were used to design universal query system algorithms when network layer information was known and unknown, respectively. Experimental results showed that the distributed matching-based query system algorithm proposed in this study could obtain optimal solution in large-scale agrometeorological network and was not restricted to fixed network topology. When the cost ratio of storage to a unit delay cost was 10, the query system cost would present a linear growth; when the ratio was greater than 30, the storage cost would rapidly decrease, but the delay cost increased and the algorithm would run fast. Accordingly, the proposed algorithm could flexibly realize effective balance between storage cost and query delay. Conclusions of this study can provide theoretical and technical references for data query in intelligent agrometeorological service cloud system.

摘要

在农业气象大数据查询服务时，使用整数线性规划(Integer Linear Programming，ILP)模型构建查询系统能够快速响应查询操作。但现有相关研究仅讨论了已知网络层信息的情况，并且在大规模网络环境下，其时间复杂度过高甚至得不到查询结果。为了提升分布式数据查询能力，针对大规模网络以及网络层信息未知情况，本文提出了一种基于分布式匹配的启发式 ILP 查询算法。证明了在网络层信息未知时查询系统的时空开销下界，形式化地提出了在网络层信息未知时的整数线性规划模型和算法设计。最后，利用不相交路径分配和冗余结点分配方法设计了网络层信息已知和未知时通用的查询系统算法。研究结果表明：本文提出的基于分布式匹配的查询系统算法在农业气象大规模网络中能够得到优良的解，并且不受限于固定的网络拓扑结构，当单位存储开销与延迟开销的开销比为 10 时，查询系统开销会随之线性增长，大于 30 时，存储开销快速降低，延迟开销随之增大，算法运行时间会变快，这表明该算法可以灵活实现存储开销和查询延迟的有效平衡。本文的研究结论可以为智慧农业气象服务云系统中进行数据查询提供理论与技术参考。

INTRODUCTION

The Intergovernmental Panel on Climate Change once emphasized in the second and fourth evaluation reports that meteorological disaster would generate an enormous influence on the agriculture of China; if measures are not taken, the production capacity of the planting industry of China would drop by 5% to 10% up to 2030, and the sub-crop yield in 2050 would reduce by approximately 30% *(Yingjia et al., 2013)*. Therefore, China is vigorously developing precision agriculture and intelligence agriculture, which use modernized technologies and the use of agrometeorological big data is playing an increasingly significant role. The use of meteorological big data is significant to improving agricultural productivity and realizing high yield, high quality, high efficiency, safety and ecological and sustainable agricultural development *(Guangsheng, 2015)*. Agrometeorological big data feature mass distribution, timeliness and diversity. Agrometeorological data, which are acquired through ground automatic meteorological observation, radar remote sensing monitoring, and satellite meteorological monitoring, are usually stored in hundreds of host computers. Then, saved data under geographical distribution are formed. Distributed query system is an extensively applied method to simultaneously analyse data on these different host computers and acquire real-time agrometeorological information.

When distributed query system is used to analyse and query high-correlation datasets, even damage of one patch of data will result in failure of the entire data analysis process; therefore, compared with traditional query system, distributed query system lays more emphasis on data survivability *(Mckendrick, 2014)*. In terms of the present distributed query technologies, integer linear programming (ILP) is superior to existing algorithms, such as A-star, in establishing query system *(Dokeroglu et al., 2014)*. However, most studies have ineffectively processed the space–time cost problem of query system in large network. Yangming studied distributed matching-based query system and proposed an improved ILP query system model that could effectively balance query system cost and storage cost under a few constraints *(Yangming et al., 2011)*. Nevertheless, only the ILP model of distributed matching query when network layer information was known was given in this study while effectively realizing and implementing the proposed scheme in large network system were difficult. In a large network environment, the query system design problem when network layer information was unknown was formally expressed as ILP model in this study. A universal algorithm when network layer information was known and unknown was also designed, which effectively improved the performance of distributed query system in large-scale network, enhanced agrometeorological service accuracy and data-supporting capacity, timely implemented intelligent data analysis of agrometeorological situation, and provided excellent scientific service for agricultural scientific research, agricultural production, and meteorological disaster prewarning.

The present research on distributed query system mainly aims at algorithm improvement and optimization. Bruno used aggregate method to control query system and data statistics under system operation and proposed an optimization method for distributed query processing of big data ***(****Bruno et al., 2013****)***. Marin introduced a metadata-based indexing strategy to improve the efficiency of inquiring mobile object in large-scale system *(Marin and Rodríguez, 2010)*. Sharma proposed a random decision support system of query optimization based on heuristic method, and this system significantly improved the query performance ***(****Sharma et al., 2016****)***. However, these optimization algorithms have lacked research on rapid big data matching, and they cannot effectively solve the bottleneck problem that restricts the operation efficiency of query system.

Wittmann-Hohlbein discussed the entire solution of the multi-parameter mixed ILP problem of uncertain items in constraint matrix, right-hand vector, and objective function coefficients *(Wittmann-Hohlbein and Pistikopoulos, 2013)*; neither this algorithm, nor the scheme proposed by Yangming *(Yangming et al., 2011)* was restricted to fixed network topology and both could realize fault-tolerant backup and were thus of favourable extendibility. However, these schemes could barely be implemented and applied under large-scale network environment with the following deficiencies: (1) Only the circumstance when network layer information is known is considered, but that when information is unknown is disregarded; (2) When the delay is not so important and the storage constraint is very strict, the distributed match making cannot find the tradeoffs between the used storage capacity and the response delay.

In view of the preceding analysis, the lower bound of the space–time cost of the query system when network layer information was known was analysed in this study. The ILP model and algorithm design of the query system design problem when network layer information was unknown was formally proposed. For the excess complicacy problem of ILP in large-scale network, a universal heuristic algorithm when network layer information was known and unknown was designed based on disjoint path allocation (DPA) and node replication allocation (NRA) algorithms, and this algorithm was named DPA-NRA-ILP. The algorithm framework proposed in this study did not rely on specific topological structure and could effectively realize balance of storage cost and query delay.

MATERIAL AND METHODS

ILP Formulation for The System Without Network Layer Information

In distributed matching query system of agrometeorological big data, two node sets $P(v)$ and $Q(v)$ are defined for each node v: $P(v)$ is the redundant node set of node v, namely, data of node v should be backed up on all nodes in $P(v)$, and $Q(v)$ is the node set that should be queried when node v generates query request. For each node pair (v,v'), the intersection $|P(v)\cap Q(v')|$ is calculated. If the intersection is non-null, then some nodes in $P(v)$ can be queried by node v', and node v' can query node v.

The objectives of distributed matching are to (1) minimize the average of $|P(v)\cap Q(v')|$ taken over all node pairs (v,v') and (2) find the tradeoffs between the lower bound of objective (1) and the average number of elements (or worst case number of elements) in $S(v)$, where $S(v)=\{j:v\in P(j)\}$.

For quantifying the optimization efficiency of the query system, the query system cost is set as C, the storage cost is C_{memory}, the delay cost is C_{delay}, the number of storage equipment is N_{memory}, the algorithm parameter γ is the cost ratio of a storage to a unit delay cost, and the query system cost is formally defined as

$$\begin{aligned} C &= C_{memory} + C_{delay} \\ &= \gamma \times N_{memory} + C_{delay} \end{aligned} \quad (1)$$

p_{ik} is set as binary variable. When $p_{ik}=1$, data of node i are backup of node k, and $p_{ik}=\begin{cases}1 & k\in P(i)\\ 0 & k\notin P(i)\end{cases}$. q_{jk} is the binary variable, and $q_{jk}=\begin{cases}1 & k\in Q(j)\\ 0 & k\notin Q(j)\end{cases}$. m_{ij}^{k} is the binary variable, and $m_{ij}^{k}=\begin{cases}1 & k\in P(i)\cap Q(j)\\ 0 & k\notin P(i)\cap Q(j)\end{cases}$. S is the maximum storage space of each node. B is the maximum number of communication times in each query. When network layer information is unknown, the following (2)–(7) ILP problems can be used to describe the distributed matching-based query system.

Objective Minimize:
$$\sum_i\sum_k p_{ik} + \sum_j\sum_k q_{jk} \quad (2)$$

Subject to:
$$p_{ik} \ge m_{ij}^{k} \quad \forall i,j,k \quad (3)$$
$$q_{jk} \ge m_{ij}^{k} \quad \forall i,j,k \quad (4)$$
$$\sum_k m_{ij}^{k} \ge K \quad \forall i,j \quad (5)$$
$$\sum_i p_{ik} \le S \quad \forall k \quad (6)$$
$$\sum_k q_{jk} \le B \quad \forall j \quad (7)$$

Function (2) means to make minimum storage cost and communication cost, $\sum_i\sum_k p_{ik}$ expresses the storage cost, and $\sum_j\sum_k q_{jk}$ expresses the communication cost. Formula (3) expresses $m_{ij}^{k}=1 \Rightarrow p_{ik}=1$, namely, $k\in P(i)\cap Q(j)\Rightarrow k\in P(i)$. Formula (4) expresses $m_{ij}^{k}=1\Rightarrow q_{jk}=1$, namely, $k\in P(i)\cap Q(j)\Rightarrow k\in Q(i)$. Formula (5) means that the system can reliably operate even when $K-1$ nodes are failing in the system. Formulas (6) and (7) represent that the maximum number of storage units in each node is S and the maximum number of communication times in each query is B.

In this model, N is the node number in the network, and $2NK+KN^2$ variables and $2KN^2+N^2+K+N$ constraint conditions exist. This problem is difficult to directly solve in large-scale network. Therefore, the lower bound of objective function (2) will be discussed and the algorithm will be designed to achieve this lower bound.

Query System Design Algorithm Without Network Layer Information

Lower Bound of Objective Function

A graph with $2n+v$ nodes called $a(1)\ldots a(n)$, $b(1)\ldots b(n)$, $r(1)\ldots r(v)$ is constructed to study the lower bound of the objective. $a(i)$ stands for the arguments of P, $b(i)$ stands for the arguments of Q, and $r(i)$ stands for the rendezvous elements. Whenever j is the rendezvous element of $P(i)\cap Q(k)$, an edge is put from $a(i)$ to $r(j)$ and the edge is from $r(j)$ to $b(k)$.

For the triples (i,j,k), if the path $a(i)\to r(j)\to b(k)$ exists, then they are called good triples. At least Kn^2 good triples can then be obtained from constraint condition (5).

$d(j)$ is set as the total edge number from $a(i)$ to $r(j)$; $e(j)$ is the total edge number from $r(j)$ to $b(k)$; and for any j, at most $d(j)e(j)$ good triples include j ($r(j)$ is taken as intermediate node). $|P(i)+|Q(i)||$ is the size of the node set, $d(i)$ and $e(i)$ meet the following relational expression:

$$\sum_{i=1}^{v}[d(i)+e(i)]=\sum_{i=1}^{n}(|P(i)+|Q(i)||),$$

which is referred to as objective function (2), and

$$\sum_{i=1}^{v}d(i)\geq Kn.$$

The storage unit number of each node is S. At most, S i s satisfy $j\in P(i)$, that is, $d(j)\leq S$, and

$$s\sum_{i=1}^{v}e(i)\geq\sum_{i=1}^{v}e(i)d(i)\geq Kn^2.$$

Define $m=\sum_{i=1}^{v}[d(i)+e(i)]$, then

$$\begin{aligned}ms&=s\sum_{i=1}^{v}[d(i)+e(i)]=s\sum_{i=1}^{v}d(i)+s\sum_{i=1}^{v}e(i)\\&\geq s\sum_{i=1}^{v}d(i)+Kn^2\geq sKn+Kn^2\end{aligned}.$$

Thus, the lower bound of the objective is obtained as

$$m\geq Kn(1+\left\lceil\frac{n}{s}\right\rceil), \tag{8}$$

where n is the total node number, K is the number of nodes responding to each query, and s is the storage unit number of each node.

Construction of Query Algorithm

Definition: Convergent matrix R is an n×n matrix, and each of its matrix element r_{ij} is a set including intersection $P(i)\cap Q(j)$.

The definition implies that

$$\bigcup_{j=1}^{n}r_{ij}=P(i)\quad and\quad\bigcup_{i=1}^{n}r_{ij}=Q(j).$$

Convergent matrix R of the following form is constructed:

$$R=\begin{bmatrix}\{1..K\}&\{1..K\}&\cdots&\{1..K\}\\\vdots&\vdots&\vdots&\vdots\\\{1..K\}&\{1..K\}&\cdots&\{1..K\}\\\{K+1,..2K\}&\{K+1,..2K\}&\cdots&\{K+1,..2K\}\\\vdots&\vdots&\vdots&\vdots\\\{K+1,..2K\}&\{K+1,..2K\}&\cdots&\{K+1,..2K\}\\\cdots&\cdots&\cdots&\cdots\end{bmatrix}.$$

All elements of the first S rows in the matrix are $\{1,\ldots,K\}$, all elements from the $(S+1)$ row to $2S$ row are $\{K+1,\ldots,2K\}$, and so on; then,

$$P(i)=\{K\left\lceil\frac{i}{s}\right\rceil+1,\ldots,K(\left\lceil\frac{i}{s}\right\rceil+1)\},\ Q(j)=\{1,\ldots,K\left\lceil\frac{n}{s}\right\rceil\}.$$

In this case, $\sum_{i=1}^{n}(|P(i)|+|Q(i)|)=nK+nK\left\lceil\frac{n}{s}\right\rceil$,

namely, the lower bound of objective function (2) certified in the preceding section. Thus, the query system constructed through the above-mentioned method is optimal.

DPA-NRA-ILP Query System Design

Main Idea

The heuristic query system design method proposed in this study ensures that the query paths of any two nodes will not intersect, the redundant node position of each node is calculated, and a feasible solution can be certainly obtained only by finding disjoint paths in this method.

DPA and NRA are used in this method. In the DPA, the legal path needs to meet the constraints of node disjoint and shared risk link groups (SRLG). For each node pair, k groups of legal path should be determined to meet the reliability requirement (at the moment, $(k-1)$ mistakes can be allowed in the system) *(Dijkstra, 1959)*. The total delay cost of the k groups of paths should be as small as possible. In the NRA, paths used for query are constructed, and the query delay and storage costs of the algorithm should be adjusted. The redundant node location of each node should be determined until paths of the same group are at different nodes to ensure system survivability. If residual spaces exist to store a large amount of data, then an attempt can be made to use these spaces to reduce overall cost in Equation (1). When the overall cost cannot be further optimized or no residual data storage space exists, then the algorithm ends.

Algorithm Process

The process of DPA-NRA-ILP is shown in Figure 1. The overall query system cost in Equation (1) is used to determine the quality of feasible solution. The left half in the figure (steps 1–7) is the DPA process, and k groups of legal paths are found for each node pair. The right half in the figure (steps 8–13) is the NRA process, which is used to construct a feasible solution and reduce the system cost.

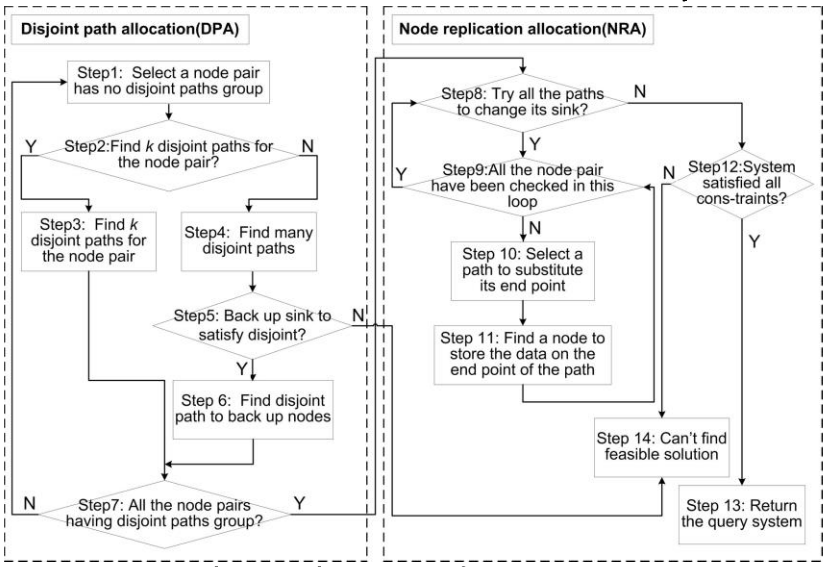

Disjoint path allocation(DPA)
Node replication allocation(NRA)
Step1: Select a node pair has no disjoint paths group
Step2:Find k disjoint paths for the node pair?
Y
N
Step3: Find k disjoint paths for the node pair
Step4: Find many disjoint paths
Step5: Back up sink to satisfy disjoint?
Y
N
Step 6: Find disjoint path to back up nodes
Step7: All the node pairs having disjoint paths group?
N
Y
Step8: Try all the paths to change its sink?
N
Y
Step9:All the node pair have been checked in this loop
Y
N
Step 10: Select a path to substitute its end point
Step 11: Find a node to store the data on the end point of the path
Step12:System satisfied all cons-traints?
N
Y
Step 14: Can't find feasible solution
Step 13: Return the query system

Fig. 1 - Main process of DPA-NRA-ILP

Description of detailed steps: steps 1–2 of the DPA select source node and sink node of the path, and an attempt is made to find k groups of legal paths. If yes, then step 3 will be executed; otherwise, steps 4–5 will be executed and under this circumstance, redundant sink nodes will be transferred onto other nodes to amplify legal paths. If legal paths are insufficient to constitute k groups yet, then the feasible solution cannot be found under the present reliability requirement, and the algorithm ends. The above-mentioned steps should be repeated until k groups of legal paths are found for all node pairs. Under abundant data operation, data should be replicated onto nodes nearby original node as far as possible.

If k groups of legal paths are found for all node pairs, then the NRA steps should be started (space substitution for time): step 8 judges whether a space is available to optimize time efficiency. Steps 10–11 modify the sink nodes of existing paths, as specifically described in the next paragraph. After redundancy is used to optimize the conversion of time efficiency, step 12 will be finally executed to test whether the present result meets all reliability requirements; if yes, then result will be output; otherwise, the algorithm ends.

Construction and Optimization of Feasible Solution

The following definitions are given:

Disjoint Path Set (DPS_{uv}): disjoint path set DPS_{uv} refers to the path set of query request from node u to node v. All paths must take u as starting node (source node) but not certainly take v as end node (can end at redundant node of v). p_{uv}^{k} is the k path between nodes u and v, and $DPS_{uv} = \{p_{uv}^{k}\}$.

Weighted Path Delay (WPD_{uv}^{k}): it is related to the overall delay cost of p_{uv}^{k} . Under most circumstances, WPD_{uv}^{k} is rightly the overall delay cost, but if p_{uv}^{k} is not the only path with end node being v , then a large number should be superposed onto WPD_{uv}^{k} .

In step 8 of the NRA process, the algorithm will attempt to modify sink nodes for all paths to construct a feasible solution or optimize solutions. Steps 9–11 constitute an internal loop. Step 9 calculates $l = \arg\max_{k}\{WPD_{uv}^{k}\}$ for each node pair (u,v) and prepares to modify path l to optimize existing solutions. Step 10 finds the path that has not been tried but with maximum delay. Step 11 backs up sink node data onto node v', which is most adjacent to it and meets the following constraint conditions:

- Node v' is not the backup node of node v .
- The path exists between nodes u and v' and it does not intersect with other path except l in the present DPS set (allowed to intersect with l because l will be substituted). Delayed input parameter γ is lower than WPD_{uv}^{k}. The path meeting the above-mentioned conditions is $p_{uv}'^{l}$.

If this node v' can be found, then $p_{uv}'^{l}$ is used to substitute p_{uv}^{l} in the DPS set: $DPS_{uv} / p_{uv}^{l} \cup p_{uv}'^{l} \rightarrow DPS_{uv}$, or we will return to step 10, and another path is used for further trial. The cost to back up onto other nodes will be great because the selected new backup node is a legal node most adjacent to the original end point. Consequently, end point substitution of each path can be implemented once only.

Performance Analysis

The key idea of the DPA-NRA-ILP is to optimize results by steps on the condition that feasible solution is guaranteed. As shown in Figure 2, three disjoint paths exist between nodes 1 and 2: 1→2, 1→3→2, and 1→4→2. The three paths cannot be directly used for query because all paths take node 2 as end point. When single-node fault occurs to node 2, the system cannot maintain reliability. If the data redundancy of node 2 is put onto nodes 3 and 4, then the three paths can be transformed into 1→2, 1→3, and 1→4 to solve the reliability problem. This transformation can be realized by substituting sink node (end point).

During the path optimization process through the NRA steps, if the path with end point being v is not unique, then a large number will be superposed onto the overall query system cost WPD_{uv}^{k}. In this way, paths can be dispersed to different nodes by the algorithm to constitute feasible solution as a priority. When searching for substituting nodes, this algorithm will judge whether delayed input parameter γ of new path is lower than present delayed WPD_{uv}^{k}. The delay cost and storage cost of the algorithm can be controlled by inputting parameter γ. As γ increases, the consumption of storage units will be reduced and the delay cost will correspondingly increase.

$O(N^2)$ time complexity is needed to calculate the shortest path. Therefore, for each node pair, $O(KN^2)$ time complexity is needed to calculate K disjoint paths. During the NRA process, the end point of each path should be substituted, $O(N^2)$ is needed by the Dijkstra method *(Dijkstra, 1959)* to calculate all of the shortest paths from one node, and $O(N\log N)$ is needed for sorting. Substitution is tried successively for each node, and the substitution needs to be repeated for $O(N)$ times under the worst circumstance. Given N^2 node pairs in the system, the upper bound of overall time complexity of the algorithm is $O(K^2N^5\log N + 2KN^4 + KN^3)$, namely, $O(K^2N^5\log N)$. The shortest path is also calculated when the main time is used for substitution.

RESULTS

Experimental Scheme

The experiment time was selected in June in 2016, and the place was the Chaohu Lake in Anhui Province, China. This period comprised great meteorological change, mass meteorological data, and high requirement of agricultural production for real-time meteorological information. Data were acquired through field sampling, ground automatic meteorological observation, and radar remote sensing monitoring, as shown in Figure 3.

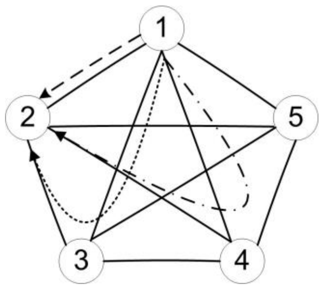


Fig. 2 - Disjoint paths from nodes 1 to 2

Fig.3 - Meteorological Data Acquisition

The algorithm in this study was implemented based on intelligent agrometeorological service cloud system and its system framework is shown in Figure 4 *(Xiang et al., 2015)*.

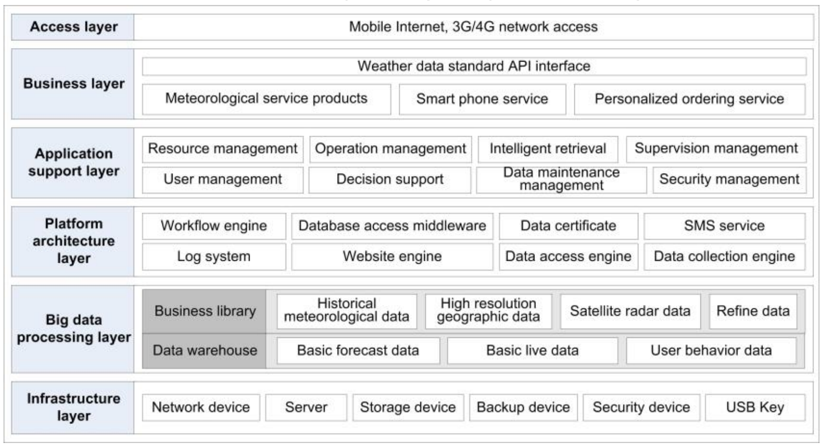


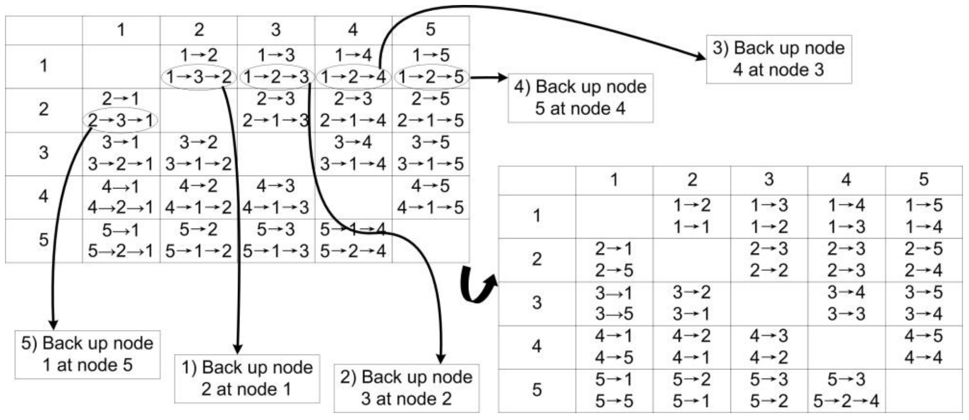
Fig. 4 - System Framework of Intelligent Agrometeorological Service Cloud

The experimental objective was to explore the influences of network scale and algorithm parameter, namely, cost ratio γ of storage to unit delay cost, on the algorithm execution. Section 2 proved that the algorithm in this study could still obtain optimal solution when network layer information was unknown. The experimental emphasis here was laid on the testing algorithm performance in simulation environment of known network layer information.

Experimental Result Analysis

Influence of Network Scale

Five network nodes were selected from agrometeorological network to construct a small-scale network test DPA-NRA-ILP algorithm, as shown in Figure 2. The execution process of the algorithm is shown in Figure 5.



	1	2	3	4	5
1		1→2 1→3→2	1→3 1→2→3	1→4 1→2→4	1→5 1→2→5
2	2→1 2→3→1		2→3 2→1→3	2→3 2→1→4	2→5 2→1→5
3	3→1 3→2→1	3→2 3→1→2		3→4 3→1→4	3→5 3→1→5
4	4→1 4→2→1	4→2 4→1→2	4→3 4→1→3		4→5 4→1→5
5	5→1 5→2→1	5→2 5→1→2	5→3 5→1→3	5→1→4 5→2→4	

	1	2	3	4	5
1		1→2 1→1	1→3 1→2	1→4 1→3	1→5 1→4
2	2→1 2→5		2→3 2→2	2→3 2→3	2→5 2→4
3	3→1 3→5	3→2 3→1		3→4 3→3	3→5 3→4
4	4→1 4→5	4→2 4→1	4→3 4→2		4→5 4→4
5	5→1 5→5	5→2 5→1	5→3 5→2	5→3 5→2→4	

3) Back up node 4 at node 3
4) Back up node 5 at node 4
5) Back up node 1 at node 5
1) Back up node 2 at node 1
2) Back up node 3 at node 2

Fig. 5 - Example of DPA-NRA-ILP execution based on the network in Fig. 2

The delay cost between any two nodes was set 1, except the delay cost on (4, 5) was set 100, and the algorithm parameter $\gamma = 2$. The left table in Figure 5 shows the DPA results. After disjoint paths were found for each node pair, NRA was started for path adjustment. For example, in the first loop, the algorithm adjusted the second path from nodes 1 to 2, and the data on node 2 were backed up on node 1. Each path was queried through sequential substitution, and redundancy was created. The obtained results are shown in the right table in Figure 5. The algorithm generated the optimal solution.

The influence of network scale on the algorithm was analysed. A large number of network nodes were selected from agrometeorological network, a large-scale analog network was constructed through the method of Waxman *(Cheng et al., 2000)* and X and Y coordinate values of nodes were within 0 and 100. The shortest paths of all node pairs were calculated, and these lengths were taken as the delay cost between two nodes.

The influences of network scale on algorithm running time and query system cost are shown in Figures 6 and 7. Each node in the figure is the average cost of hundreds of query systems designed through DPA-NRA-ILP. In all simulations, the algorithm parameter γ was set as 10 and each node could store five copies of data of other nodes (namely, the storage unit number was 6). The storage spaces of all nodes in the result were used, and Figure 6 reflects the influence of network scale on algorithm running time. Figure 7 shows that the cost of query system largely presented a linear growth and the algorithm running time increased rapidly because the time complexity of DPA-NRA-ILP under the worst circumstance was $O(K^2N^5 \log N + 2KN^4 + KN^3)$, which was much faster than the linear growth.

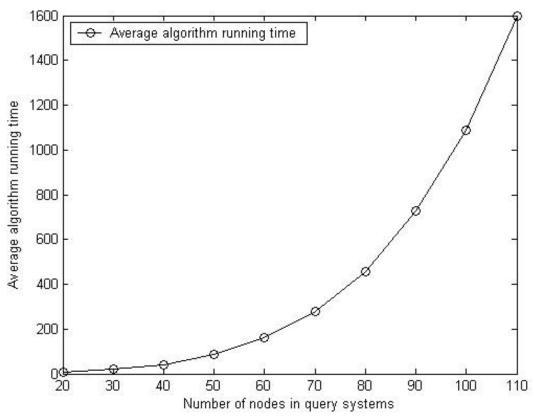


Fig. 6 - Influence of Network Scale on Algorithm Time

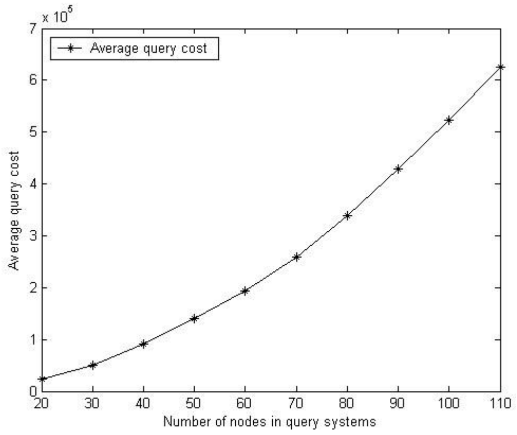


Fig. 7 - Influence of Network Scale on Query System Cost

Impact of Cost Ratio γ

An important feature of algorithm framework in this study was that it could conveniently adjust the time and space cost of the algorithm, which is significant in practical application. The influence of algorithm parameter γ on the space–time cost of the algorithm is shown in Figures 8–11.

Figure 8 reflects the relationship between γ and algorithm cost under system scales. The results showed that, when γ increased, the algorithm cost kept increasing. This finding was unrelated to system scale because, when γ increased, much space was needed to store redundant data.

Figure 9 shows the relationship between γ and storage cost. The results indicated that, if γ was small, all usable space would be used by the algorithm to reduce delay cost. When γ increased to a large degree (approximately 30), the storage cost would rapidly decline because using space to store redundant data was cost effective.

Figure 10 indicates that the change in delay cost was unobvious because all storage space was used, and the location of redundant data approximated to the original node considerably. As γ increased, the change in location of redundant data would result in slight reduction in delay cost. When γ increased to above 30, the delay cost increased with γ. The reason was that time was used to substitute space by the algorithm at the beginning, which indicated that the algorithm could realize conversion between time and space costs.

Figure 11 reflects the relationship between algorithm running time and γ. At the beginning, the algorithm running time would increase with γ because when γ increased. A large number of nodes should be tried by the algorithm to obtain optimal node with small cost. When γ was large, the algorithm operated fast because the number of nodes needing calculation reduced at the moment.

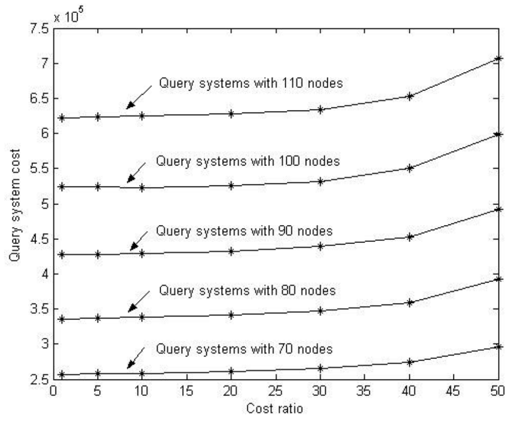


Fig. 8 - Influence of γ on Algorithm Cost

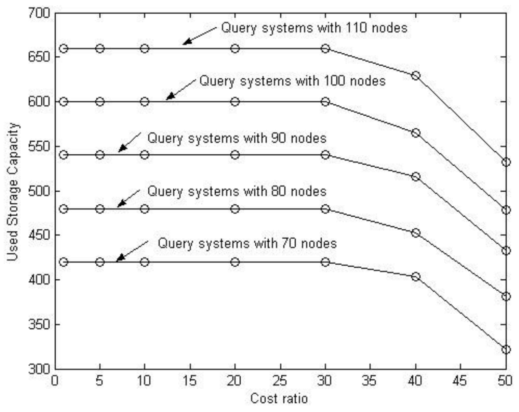


Fig. 9 - Influence of γ on Storage Cost

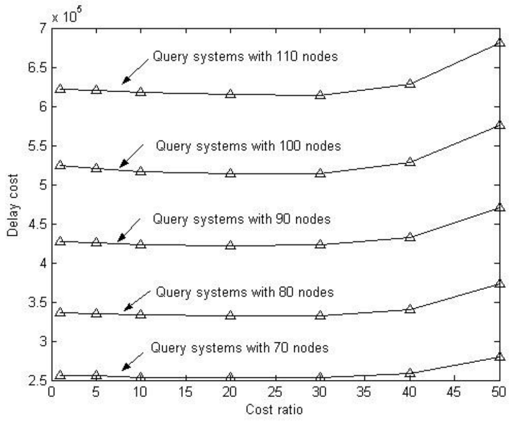


Fig.10 - Influence of γ on Delay Cost

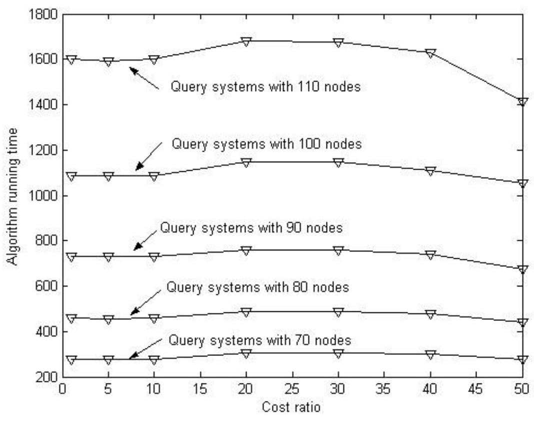


Fig.11 - Influence of γ on Algorithm Running Time

Performance Comparison Between the Algorithm in this Study and the ILP Algorithm

The performance comparison between the DPA-NRA-ILP algorithm proposed in this study and the ILP algorithm of Yangming *(Yangming et al., 2011)* is shown in Table 1. Large-scale network test data were the result based on network, as shown in Figure 2. As under large-scale network environment and when no network layer information was available, the ILP algorithm was not applicable. An emphasis was laid on comparing the performance of the two in small-scale network.

Table 1

Comparison between the algorithm in this study and ILP

Algorithm	**Small-scale network**		**Large-scale network**	**Without network layer information**
	Delay cost	**Storage cost**		
ILP	36	10	Not applicable	Not applicable
DPA-NRA-ILP	36	10	Applicable	Applicable

The test analysis in the previous paragraph implied that the solution obtained by DPA-NRA-ILP was different from that obtained by the ILP algorithm. However, Table 1 indicates that the two had the same cost. Thus, both were optimal solutions of the problem. However, the algorithm in this study could still obtain an excellent solution when network layer information was unknown in large-scale network.

CONCLUSIONS

Distributed matching-based query algorithm was investigated in this study under large-scale network environment based on theoretical analysis to improve the performance of data query system in intelligent agrometeorological service cloud system. The obtained main conclusions were as follows:

(1) When network layer information was unknown, using the ILP problem to describe the distributed matching-based query system model could obtain optimal query system at minimum system cost.

(2) The lower bound of objective function, which was related to the total number of nodes, the number of storage units in each node, and the number of nodes responding to each query, should be defined to realize optimization of query system in large-scale network. When the objective function achieved lower bound, the space–time cost of the query system was the minimum and the algorithm was optimal.

(3) The constraint conditions in distributed matching-based query system model were separated in different algorithm processes and a flexible algorithm framework could be obtained. Thus, adjusting the time and space costs of the algorithm would be convenient.

An important supplementation was made for distributed query system design of agrometeorological big data. However, when an attempt was made for node substitution, the calculated shortest path had certain impact on the time cost of the system. Therefore, reasonable experimental design and accurate numerical simulation should be developed, which will be used for in-depth analysis of the above-mentioned problem. These influencing factors should be considered in future studies.

ACKNOWLEDGMENTS

This study was supported by the National Natural Science Foundation of China (No. 61273329), and Natural Science Research Project of Anhui Vocational and Technical College (No. AZY2016KJ04).

REFERENCES

[1] Bruno N., Jain S., Jingren Z., (2013), Recurring job optimization for massively distributed query processing, *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, Vol. 36, Issue 1, IEEE, USA, pp.46-55;

[2] Cheng J., Qian C., Sugih J., (2000), Inet: internet topology generator, *IEEE Transactions on Magnetics*, Vol. 19, Issue 3, University of Michigan, USA, pp.960-963;

[3] Dijkstra E. W., (1959), A note on two problems in connexion with graphs, *Numerische mathematik*, Vol. 1, Issue 1, Springer, Heidelberg/German, pp.269-271;

[4] Dokeroglu T., Bayır M. A., Cosar A., (2014), Integer linear programming solution for the multiple query optimization problem, *Information Sciences and Systems 2014*, Springer, Switzerland, pp. 51-60;

[5] Guangsheng Z., (2015), Research prospect on impact of climate change on agricultural production in China, *Meteorological and Environmental Sciences*, Vol. 38, Issue 1, Henan Meteorological Bureau, Zhengzhou/China, pp.80-94;

[6] Marin M., Rodríguez M. A., (2010), A meta-index for querying distributed moving object database servers, *Information Systems,* Vol. 35, Issue 6, Elsevier, Oxford/England, pp.637-661;

[7] Mckendrick J., (2014), Data integration in the era of big data, *Database Trends and Applications*, Vol. 28, Issue 1, Information Today Inc., New Jersey /USA, pp.4-6;

[8] Sharma M., Singh G., Singh R., (2016), Design and analysis of stochastic DSS query optimizers in a distributed database system, *Egyptian Informatics Journal*, Vol. 17, Issue 2, Elsevier, Oxford/England, pp.161-173;

[9] Wittmann-Hohlbein M., Pistikopoulos E. N., (2013), On the global solution of multi-parametric mixed integer linear programming problems, *Journal of Global Optimization*, Vol. 57, Issue 1, Springer, Netherlands, pp. 51-73;

[10] Xiang S., Huarui W., Qingxue L., (2015), Design and implementation of massive agricultural information resource management platform, *Computer Applications and Software*, Vol. 32, Issue 3, Shanghai Institute of Computing Technology, Shanghai/China, pp.75-78;

[11] Yangming Z., Sheng W., Huacheng C., et al., (2011), A new framework to design distributed query system, *International Conference on Advanced Intelligence and Awareness,* IET, Shenzhen/China, pp.270-274;

[12] Yingjia Z., Yaohui L., Yuhua C., (2013), Major progress of global and China regional climate change projection, *Journal of Arid Meteorology*, Vol. 31, Issue 4, Chinese Meteorological Society, Lanzhou/China, pp.803-813.