EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

LCBPA: two-stage task allocation algorithm for high-dimension data collecting in mobile crowd sensing network

Ning Zhou[1,2*], Jianhui Zhang[1], Binqiang Wang[1] and Jia Xiao[3]

Abstract

Mobile crowd sensing (MCS) is a novel emerging paradigm that leverages sensor-equipped smart mobile terminals (e.g., smartphones, tablets, and intelligent wearable devices) to collect information. Compared with traditional data collection methods, such as construct wireless sensor network infrastructures, MCS has advantages of lower data collection costs, easier system maintenance, and better scalability. However, the limited capabilities make a mobile crowd terminal only support limited data types, which may result in a failure of supporting high-dimension data collection tasks. This paper proposed a task allocation algorithm to solve the problem of high-dimensional data collection in mobile crowd sensing network. The low-cost and balance-participating algorithm (*LCBPA*) aims to reduce the data collection cost and improve the equality of node participation by trading-off between them. The *LCBPA* performs in two stages: in the first stage, it divides the high-dimensional data into fine-grained and smaller dimensional data, that is, dividing an *m*-dimension data collection task into *k* sub-task by *K*-means, where ($k < m$). In the second stage, it assigns different nodes with different sensing capability to perform sub-tasks. Simulation results show that the proposed method can improve the task completion ratio, minimizing the cost of data collection.

Keywords: Task allocation, Mobile crowd sensing, High-dimensional data collection

1 Introduction

1.1 Background

The term mobile crowd sensing (MCS) has been coined by Ganti et al. [1] in 2011, which introduced a new data collection method by leveraging mobile terminals such as smart phones. Compared with traditional data collection technologies, MSC has some unique characteristics. First, the mobile devices have more computing, communication, and storage capability than mote-class sensors. Second, by leveraging the mobility of the mobile terminal users, the deployment cost of specialized sensing infrastructure for large-scale data collection applications would be largely reduced. Currently, MSC has been widely used in many applications, including environmental monitoring [2], transportation [3], social behavior analysis [4], healthcare [5], and others [6–9], which demonstrates that MCS is a useful solution for large-scale data collection applications. In general, the MCS process always consists of four steps: assigning sensing tasks to mobile terminals, executing the task on the mobile terminals, collecting, and processing sensed results from the crowd [10–14]. Obviously, assigning sensing tasks to mobile terminals is the primary issue to deal with the following steps, which is also the main issue in this paper.

1.2 Related work

Recently, a lot of efforts have been focused on task allocation [15–17] and generally can be divided into two categories: rule-based task allocation method [18–20] and map-based task allocation method [21–24]. The rule-based task allocation mainly allocates task according to each node's sensing capability, such as position and power of perception. By dividing node characteristics

* Correspondence: 214998408@qq.com
[1] National Digital Switching System Engineering and Technology Research Center, Zhengzhou 450002, People's Republic of China
[2] Xinxiang University, NO.191, Jinsui Avenue, Xinxiang 453000, China
Full list of author information is available at the end of the article

into different task groups, the system assigns the corresponding task to each task group. In Ref [18], a task assignment algorithm dual task assigner (DTA) has been designed. The DTA has leveraged learn weights to evaluate the sensing capability of participations for each task, and the server allocates tasks according to the sensing capability level to maximize the benefit. Angelopoulos et al. [19] have selected appropriate users by selecting the optimal characteristics of nodes (quotation and quality) to allocate tasks, achieving the equalization between cost and task completion ratio. Shibo et al. [20] have considered the number of mobile nodes, the number of tasks and the task completion time, and proposed the optimal scheduling algorithm for each user in the dispatching area to reduce the sensing cost. Secondly, map-based task allocation method has been used to combine the geographical locations to task types to build a task map, and participations can obtain the content and location of a task by downloading the task map. When participations arrive in a specific task location, they can form a task group by self-organizing, and coordinate with each other to accomplish the sensing task. Dang et al. [21] have proposed the map-based mobile sensing task allocation framework named Zoom for the first time. Based on Zoom, Huy et al. [22] have studied the assignment of pixel values in the task map, and proposed a scalable reuse method for task map pixel values. In Ref [23], a raster-vector mixed task distribution method for mobile crowd sensing system has been proposed, which raster the sensing area first, and encodes the task information to improve the information utilization and reduce the data redundancy. A vector task map has also been proposed in Ref [24], which can substantially reduce the amount of data in the task map by gradually distributing the sensing tasks.

In summary, the above methods for task allocation always believe that the mobile terminals can handle all types of required data. They seldom take into consideration the fact that a sensing node needs to carry out a data sensing task that beyond to its sensing capability. For example, to construct a radio environment map to monitor the wireless resource, the application requires more than twenty types of data: data collection time, GPS information, and the device identification, and some 2G/3G/4G/Wi-Fi network data. Usually, few nodes can collect all types of the data because of their limited sensing capability, and most of the nodes can perform a fraction of types of the sensing data. The problem is that the sensing node with lower sensing capability compared with high-dimensional data is difficult, and we defined it as high-dimension data sensing problem. In this situation, the first efficient step is to divide the high-dimension data sensing task into multiple lower dimensional tasks. The second step is to assign different sub-tasks to nodes with different sensing capability, and the optimal goal is to minimize the total cost, as well as improving the task completion ratio.

1.3 Methods and contributions

This paper proposes an efficient data collection mechanism based on two-stage task allocation named low-cost and balance-participating algorithm (*LCBPA*), and the contributions are as follows:

- We design a two-stage task allocation algorithm *LCBPA* for high-dimensional data collection in MCS network. In the first stage, in order to divide a high m-dimension data collection task into k-multiple sub-tasks with lower dimensional data, we leverage the K-means method to make partition based on the similarity of sub-tasks. In the second stage, we allocate multiple nodes with different sensing capability with one or more sub-tasks based on certain optimal conditions.
- To minimize the total sensing cost and avoid some nodes to be allocated with too many sub-tasks while some nodes have only few sub-tasks, we introduce the equality parameter λi to adjust the node participation probability to prevent the inequality problem stated above. We also make our node selection policy by trading-off between minimizing the total cost and maximizing the equality λi through the weight parameter α.
- We also analyze the influence of the variation of sub-task number k and trade-off weight parameter α to the task completion ratio, total sensing cost, and node sub-task degree distribution in different scale of networks. Simulation results show that, compared with non-task-division methods, our LCBPA can reduce the total cost, and it can also make a more even sub-task degree distribution among sensing nodes.

The rest of this paper is organized as follows: System model and task allocation problem are discussed in Section 2. The detailed design of our proposed LCBPA algorithms is discussed in Section 3. The performance of our algorithm is evaluated in Section 4. Section 5 illustrates the conclusions and our future work.

2 Problem description and challenges

2.1 Problem description

As shown in Fig. 1, there is a sensing task which refers to m-dimension data collection, and there are N sensing nodes in a minimum sensing unit. It would be difficult for a node with limited sensing capability to finish m-dimension data collection in a particular place, for the

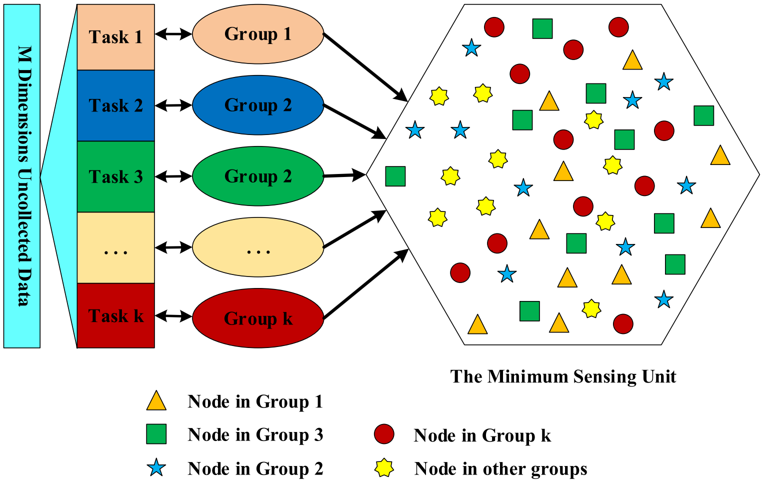

Fig. 1 Groups cooperatively complete the dimensions data sensing

node can only handle partial types of data sensing. In order to deal with the high-dimensional data collection, it needs to divide the high-dimensional data collection task into k sub-task with lower data dimension to reduce the load of a sensing node. For each sub-task, it corresponds to a sensing group with number C sensing nodes. The problem of task allocation is transformed into dividing the m-dimension data sensing task into k sub-tasks, and then for each sub-task, choosing enough sensing nodes to form a sensing group to finish the data collection.

For simplicity, we make assumptions as follows: first, the data types contained in each sub-task will not overlap and the sum of all sub-tasks' dimension is m; second, each sub-task requires the same minimum number of sensing nodes as C; third, in each group, the node number C can cover the whole sensing unit, and we will not consider the location of a node. As is shown in Fig. 1, we assume that all the sensing nodes in each group will cover the minimum sensing area.

2.2 Challenges in task allocation

We propose a task partition model to solve the problem of the high-dimensional data when we collecting data from a single node. Through this method, the data dimension is reduced, and multiple sensing nodes that work together can accomplish high-dimensional data collection tasks. Challenges for this high-dimensional data collection are as follows:

(1) How to improve the task completion ratio. In conventional methods, one node has to sense all types of required data, and this may beyond its sensing capability, which will reduce the task completion ratio. In the MCS, by dividing a high-dimensional data into lower dimensions, a node only needs to collect partial types in all of the data types.

(2) How to minimize the total sensing cost as much as possible. For a large-scale complex sensing task, it requires a significant number of nodes to collect data, and the cost of data collection is enormous for large-scale tasks. It needs to select nodes for a suitable sub-task to minimize the cost of data collection under the premise of guaranteeing the task completion.

(3) How to equalize the participation probability of each node, that is to say, avoiding that some nodes join in a significant number of sub-tasks while others only join in very few sub-tasks. The current task allocation methods are always aiming to select the node that can minimize the total cost, which may result in the fact that some nodes may be overloaded to deal with a great number of sub-tasks, while some nodes only perform small number of sub-tasks. It is difficult to minimize the whole cost of the system while making sub-tasks distributed among participants evenly.

3 The proposed algorithm of LCBPA

3.1 Model definition

Assuming that a system requires mobile nodes (mobile terminals) to finish an m-dimension data collection task, and for simplicity, one dimension corresponds to one data type. The required data type set is A, which is a matrix with m rows and N columns. Let $A = \{G_1, G_2, \cdots, G_m\}$ $A = \{G_1, G_2, \ldots, G_m\}$, and each G_i denotes a data type in the m-dimension data task collection. For each vector G_i, the element g_{ij} represents that the data type G_i can be collected by node j, and j is the node number, ($j \in (1,$

N)). If $g_{ij} = 1$, it means that the ith dimensional data can be collected by the node j, else $g_{ij} = 0$.

$$A = \begin{bmatrix} g_{11} & \cdots & g_{1N} \\ \vdots & g_{ji} & \vdots \\ g_{m1} & \cdots & g_{mN} \end{bmatrix} \tag{1}$$

The divided sub-task set is S, and $S = \{s_1, s_2, \ldots, s_k\}$, where k is the number of sub-task. For $\forall s_i$, ($1 \le i \le k$), we assume that the data types contained in each sub-task are not overlapped and the sum of all sub-tasks' dimension is m.

Each node corresponds to a triple $\psi_i = \{i, T_i, V_i\}$, where i is the node number, and $i \in (1, N)$. T_i is the task which can be sensed by user i, which is a k-dimension vector. For each element t_{ij} in T_i, if $t_{ij} = 0$, it means that the node i does not have the capability to sense the data type required in the sub-task s_j, while, if $t_{ij} = 1$, it means that node i can perform the sub-task s_j. V_i is the cost of the sensing task, which is also a k dimensional vector. For each element v_{ij}, the value is the cost of node i for sensing sub-task s_j.

The total cost for a mobile crowd sensing system to finish the data collection task is

$$W = \sum_{i=1}^{N} \sum_{j=1}^{k} t_{ij} v_{ij} \tag{2}$$

The goal of our system is to find a matrix T^*, which will minimize W, that is

$$T^* = \begin{bmatrix} t_{11}^* & \cdots & t_{1k}^* \\ \vdots & t_{ij}^* & \vdots \\ t_{N1}^* & \cdots & t_{Nk}^* \end{bmatrix} \tag{3}$$

$$\min W = \sum_{i=1}^{k} \sum_{j=1}^{N} t_{ij}^* v_{ij} \tag{4}$$

The trade-off between minimizing the total cost and λi is performed as follows:

$$p_{ij} = t_{ij} \cdot \begin{cases} \dfrac{w_j - v_{ij}}{w_j \cdot (r_j - 1)} + \alpha \cdot \dfrac{1}{r_j}, & \text{if } j = 1. \\ \dfrac{w_J - v_{ij}}{w_j \cdot (r_j - 1)} + \alpha \cdot \dfrac{1}{r_j} \cdot \lambda_i, & \text{if } j \ge 2. \end{cases} \tag{5}$$

Here, α denotes weight factor ($0 \le \alpha \le 1$), which can be used to make the trade-off between cost constraints and node participation on the probability of node selection [25]. Let w_j denote the total quoted price that all the nodes can sense G_j, then $w_j = \sum_{i=1}^{N} v_{ij}$. Let r_j denote the number of nodes that can perform sub-task, and $r_j = \sum_{i=1}^{N} t_{ij}$.

As stated above, in order to avoid some nodes joining in too many sub-tasks while others only join in few sub-tasks, we introduce the adjustment coefficient λi [26]. When a node has already been selected in the previous sub-task allocation rounds, this parameter may reduce the probability that a node being selected. When we select a node to participate in a sub-task, the system will make a trade-off between the total cost and the adjustment coefficient λi. We compute λi by Eq. (6):

$$\lambda_i = \begin{cases} 1, & \text{if } j = 1. \\ \prod_{h=1}^{j-1} \left(1 - \dfrac{\sum_{h=1}^{j-1} t_{ih} \cdot r_h}{\sum_{i=1}^{N} \sum_{j=1}^{N} t_{ij}} \right), & \text{if } j \ge 2. \end{cases} \tag{6}$$

where t_{ih} represents whether the node i was assigned with the sub-task s_h, and $t_{ih} = 1$ means that the node i is assigned to sub-task h, else $t_{ih} = 0$. r_h denotes the node number that can perform the sub-task s_h, which can be computed by Eq. (7):

$$r_h = \sum_{i=1}^{N} t_{ih} \tag{7}$$

3.2 The two-stage task allocation algorithm of LCBPA

The detail steps for the system to select sensing nodes are divided into two stages:

Stage one: Dividing an m-dimensional data collection task into k sub-tasks by K-means method. According to the sensing capability attributes (the data type a node can sense) of each node, we need to divided the whole sensing task into k sub-tasks. Here, we choose the K-means algorithm, which is a classical algorithm to solve the clustering problem, and it is simple and fast. When processing large data sets, the k-means algorithm maintains scalability and efficiency. The general processes are as follows: First, chose k data types from data type set A as the initial clusters center randomly. Second, for the remaining data types in A, assign them to the nearest k initial clusters according to the distance D. When calculating the cluster center of each new cluster, keep repeating the process until the standard measure function D is converged.

The standard measure function D for converging clusters is

$$D = \min \sum_{i=1}^{k} \sum_{X \in s_i} dist(G_i, G_j) \tag{8}$$

where the distance D is computed by the Euclidean distance d_{ij} from G_i and G_j. The shorter the distance between them, the higher similarity between those two data types, and the higher probability those two data types will be divided into the same sub-task.

$$dist(G_i, G_j) = \sqrt{\sum_{i=1}^{N} (G_i - G_j)^2} \tag{9}$$

The correspondent algorithm for the first stage is as follows:

```
Algorithm1 Dividing $m$ Dimensions of Data to $k$ Sub-Tasks
Input: $G_i\ \forall i$ , $k$
Output: $s_i\ \forall i$
Init: $S = \{s_1, \ldots, s_k\}$ , $s_i = \forall G_i \in G$
for $i = 1$ to $m$ do
    for $j = 1$ to $k$ do
        if $G_i \notin S$ then
            Compute $d_{ij}$ using (8)
            if $d_{ij} = \min_{1 \le j \le k} d_{ij}$ then
                add $G_i$ to $s_j$
            end if
        end if
        Compute the mean vector of all nodes in $s_i$
        Update $S$
        while $D = \min \sum_{i=1}^{k} \sum_{X \in s_i} dist(G_i, G_j)$
    end for
end for
```

However, dividing the whole sensing task into *k* sub-tasks is only a basic step in our proposal, we need further to adjust the mapping between participating nodes and sub-tasks by other considerations, such as minimizing the total cost and avoiding a node being assigned too many sub-tasks in the following step.

Stage two: Assigning different nodes with different sensing capability to a suitable sub-task by the trade-off between minimizing the total cost and node participation equality. After the first data type clustering stage, the system will allocate node to each sub-task. For each sub-task $s_j \in S$, it will judge whether a node can perform the sub-task and calculate the adjustment factor λi according to formula 6. It will also calculate the probability p_{ij} to decide whether a node can be selected in a sub-task by Eq. 5. The system will sort the probabilities p_{ij} of all the nodes from high to low, and select the top-*C* nodes for sub-task s_j. The correspondent algorithm for the second stage is stated in algorithm 2:

```
Algorithm 2 Selecting Node for Each Sub-Task by Trade-off Between Total Cost and Node Participation Equality
Input: S, $\psi_i = \{i, T_i, V_i\}\ \forall i, C, \alpha$
Output: $T_i^*\ \forall s_i$ .
Init: $T_i^* = [0]\ \forall s_i$
for $j = 1$ to $k$ do
    for $i = 1$ to $N$ do
        $r_j = \sum_{i=1}^{n} t_{ij}$
        if $r_j > C$ do
            Compute $\lambda_i$ using (6) $\forall i$
            Compute $p_{ij}$ using (5) $\forall i$
            Sort $p_{ij}$ from high to low and find the
            $M$-th $p_{ij}$ as a threshed value $pp$
            if $p_{ij} \ge pp$ then
                $t_{ij}^* = 1$
            end if
        else
            $t_{ij}^* = t_{ji}$
        end if
    end for
end for
```

4 Results and discussion

4.1 Simulation setting

In this section, we will evaluate the performance of our proposed LCBPA scheme by simulation. Our simulation environment is Ubuntu 14.04. We compare the performance of our LCBPA with non-task-division (*NTD*) method, which is the method that nodes participate high-dimension data collection directly. The simulation parameters are set as follows: the sensing area is 600 × 600 m^2, and the sensing radius of each node is 25 m. The network size (sensing node number) varies from 50 to 500, the data dimension is 50, and the sub-task number ranges from 10 to 50, and the trade-off weight parameter α varies from 0.1 to 0.9, the minimum node number required in a minimum sensing unit is 50.

We analyze the performance of the LCBPA algorithm in the following parameters: (1) Task completion ratio η; (2) The total cost of data collection *W*; (3) The equality of node participation u_i. The parameters η and u_i are defined as follows:

(1) The task completion ratio η is defined as the number of the whole m-dimension tasks that can be completed, that is

$$\eta = \frac{B}{C} \tag{10}$$

where

$$B = \begin{cases} \min\limits_{1\le j\le k} \sum\limits_{i=1}^{N} t_{ij}, & \text{if } \min\limits_{1\le j\le k} \sum\limits_{i=1}^{n} t_{ij} < C; \\ C, & \text{if } \min\limits_{1\le j\le k} \sum\limits_{i=1}^{N} t_{ij} \ge C; \end{cases} \tag{11}$$

(2) μ_i denotes the participation equality of node i, which is the percentage of the total number of sub-tasks assigned to each node:

$$\mu_i = \frac{1}{k} \cdot \sum\nolimits_{j=1}^{k} t_{ij} \tag{12}$$

4.2 Simulation results

4.2.1 Comparison of task completion ratio

Figure 2 shows the comparison of the task completion ratio of our *LCBPA* with *NTD* method under different network scale. In this scenario, the number of sub-task is 20, and $\alpha = 0.5$. The simulation shows our *LCBPA* has a higher task completion ratio, and the ratio increases as the network node number increases, which is because the node number for participating each sub-task is increased.

Figure 3 shows the comparison of the task completion ratio of our *LCBPA* with various sub-task number k varies from 10 to 50 under different network scale. The result in Fig. 3 shows that the task completion rate is increasing as a high-dimensional task is divided into more sub-tasks. Under the same network scale condition, the task completion ratio is increased when k is increased, which may be because the more detailed the division of the high-dimension task, the higher probability a node is assigned to the right sub-task, improving task completion ratio.

Figure 4 shows the comparison of the task completion ratio of our *LCBPA* with the weight parameter α varying from 0.1 to 0.9 under a different network scale. The result in Fig. 4 shows that the five lines are overlapped, which means that the task completion rate remains unchanged with different value of α. It may because that the total number of nodes that need to perform a sub-task is more than C, which makes no difference on the task completion ratio.

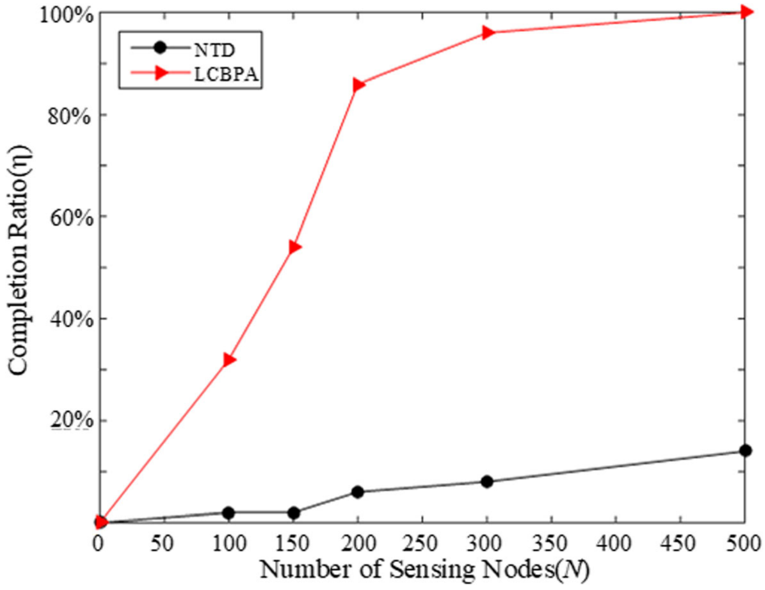

Fig. 2 Task completion ratio comparison of *LCBPA* and *NTD* under different network scale when $k = 20$, $\alpha = 0.5$

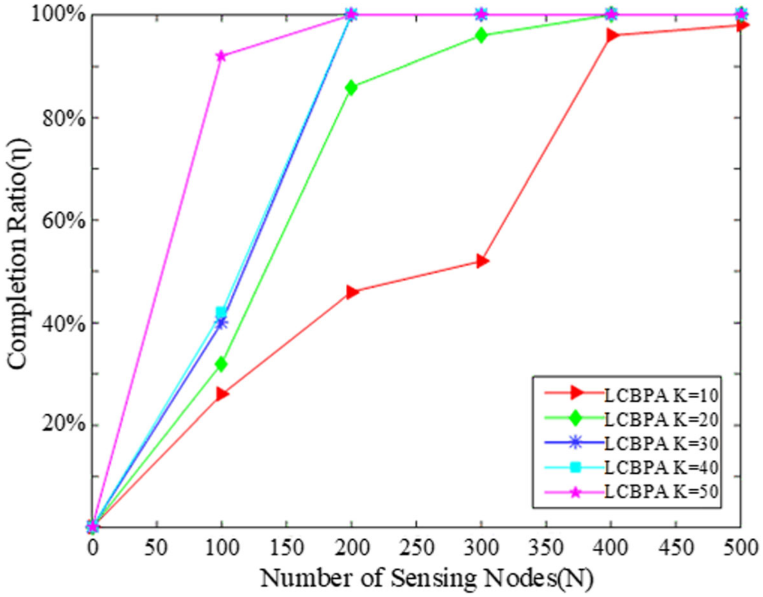

Fig. 3 Task completion ratio comparison of *LCBPA* with different sub-task number *k*, when α = 0.5 under different network scale

4.2.2 Comparison of the total cost

Figure 5 shows the comparison of the total cost of data sensing in our *LCBPA* with *NTD* method under different network scale. In this scenario, the number of sub-task is 20, and $\alpha = 0.5$. The simulation result shows that our *LCBPA* has a lower total cost, and the total cost increases as the network scale is extended, which may be because the participating node number for each sub-task is increased.

Figure 6 shows the comparison of the total cost of our *LCBPA* with different sub-task number *k* varying from 10 to 50 under different network scale. The result in Fig. 6 shows that as the size of the network increases, the total cost also increases. Under the same network scale, the larger the number of the sub-task *k*, the lower the total cost of the system, which may be because more sub-task *k* makes different nodes being assigned to more different suitable sub-tasks, saving some unnecessary cost.

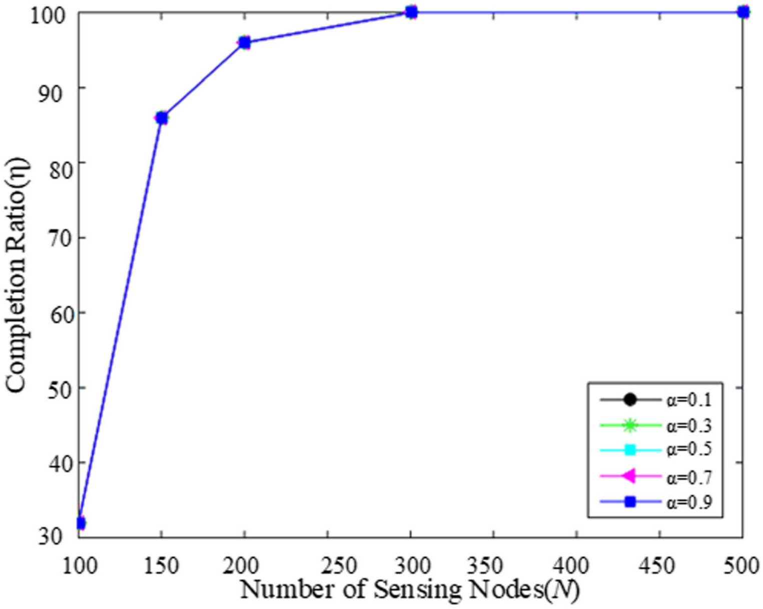

Fig. 4 Task completion ratio comparison of *LCBPA* with different *α* varying from 0.1 to 0.9 when *k = 20* under different network scale

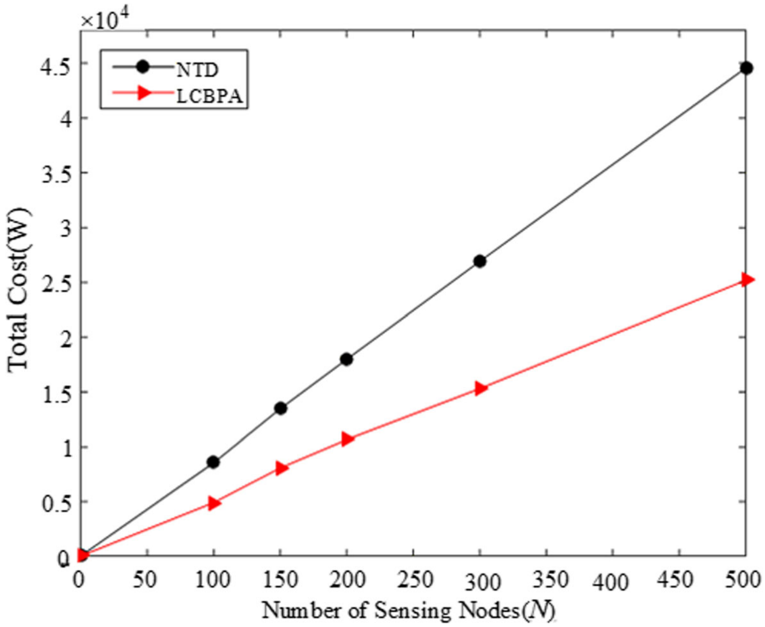

Fig. 5 Total cost comparison of *LCBPA and NTD* when $k = 20$, α = 0.5 under different network scale

Figure 7 demonstrations the comparison of the total cost of our *LCBPA* at different network with trade-off weight parameter α varying from 0.1 to 0.9 and at $k = 20$. The result in Fig. 7 shows that the total cost is increased as the network scale is higher. Under the same network scale, the larger the number of the sub-task, the higher the total cost of the system because a large α value may take less consideration of minimizing the total cost, improving the total cost.

4.2.3 Comparison of the node participation equality under a different network scale

Figure 8 shows the comparison of the node participation equality of our *LCBPA* with *NTD* method under the

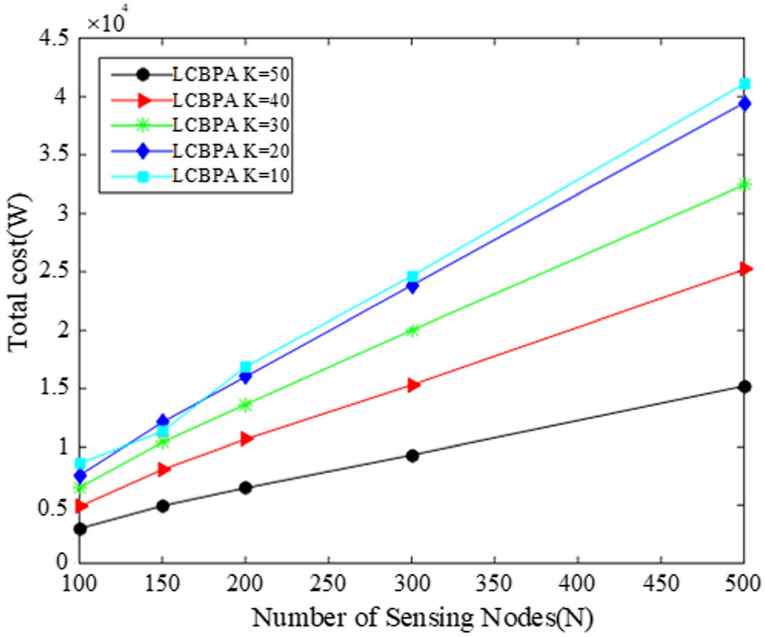

Fig. 6 Total cost comparison of *LCBPA* with different sub-task number *k*, when $\alpha = 0.5$under different network scale

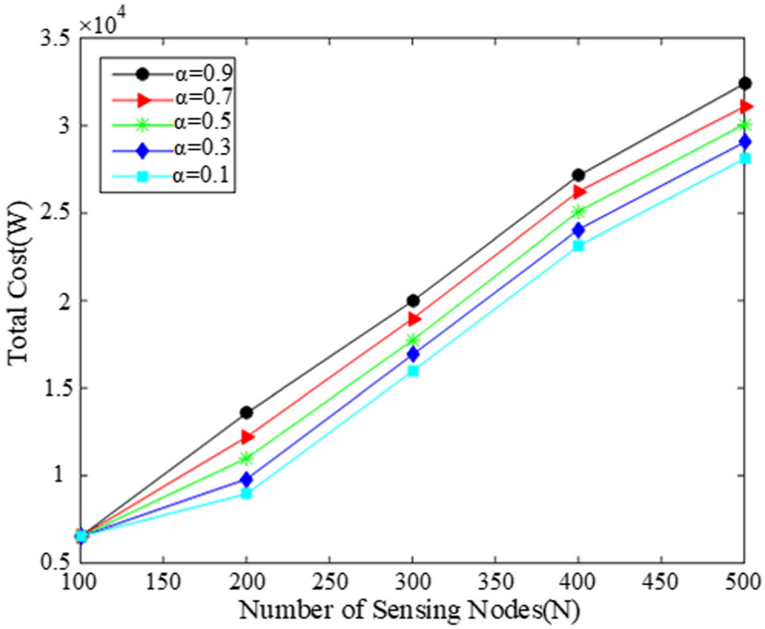

Fig. 7 Total cost of *LCBPA* with different trade-off parameter *α* when *k* = 20 under different network scale

same network scale. In this scenario, the sub-task number is 20, and α = 0.5. The simulation shows that in our *LCBPA*, a larger proportion of nodes are allocated with a smaller proportion of sub-tasks, and the proportion of nodes to perform more sub-tasks are significantly reduced.

Figure 9 shows the comparison of the node participation equality of *LCBPA* with the sub-task number *k* varying from 10 to 50 under same network scale. The result in Fig. 9 shows that the node participation becomes more concentrated when a high-dimensional task is divided into more sub-tasks. It means that the proportion of nodes performing too many tasks will be reduced, and most of the nodes are assigned to an average amount of sub-tasks.

Figure 10 shows the comparison of the node participation equality of our *LCBPA* with different trade-off weight parameter α which varies from 0.1 to 0.9.

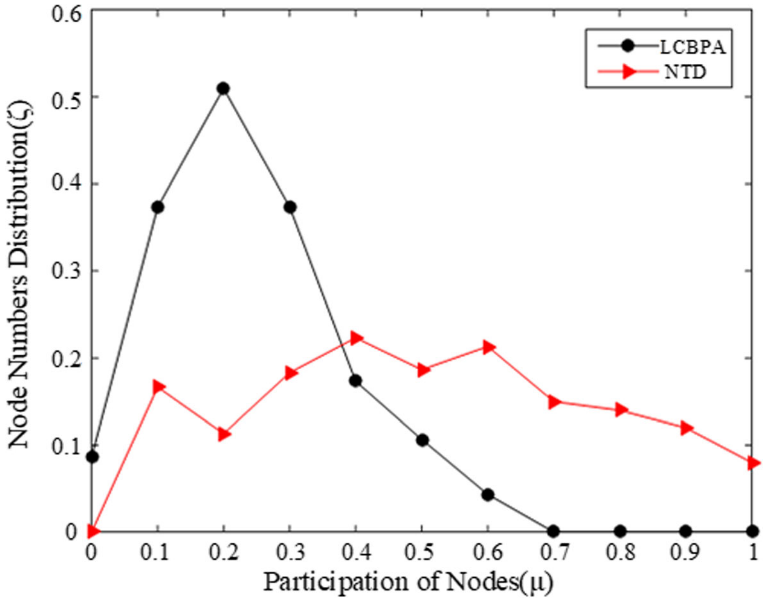

Fig. 8 Comparison of node participation equality of *LCBPA* and *NTD* when *k* = 20, the nodes number *α* = 0.5, *N* = 300

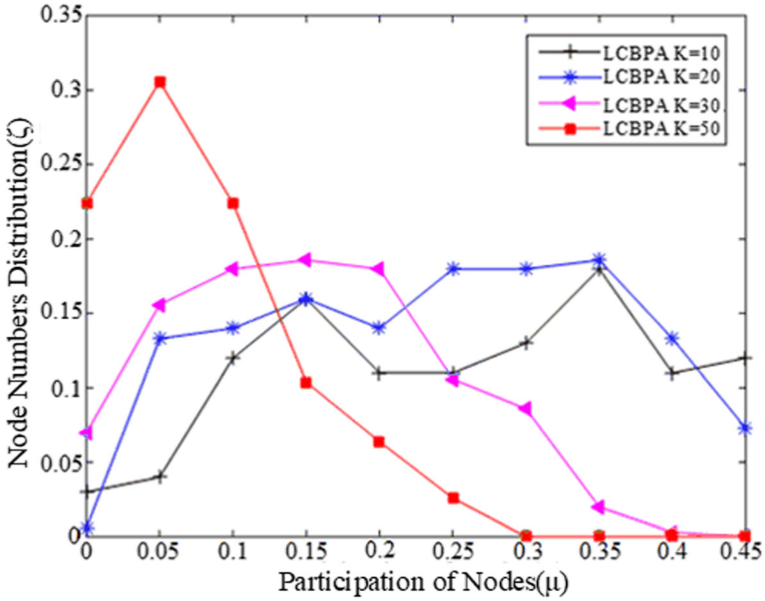

Fig. 9 Comparison of node participation equality of *LCBPA* with different sub-task number k when $\alpha = 0.5$, $N = 300$

Figure 10 shows that the participation of nodes become more concentrated as α increased, which means that the proportion of nodes performing too many tasks will be reduced as α is increased, and most of the nodes are assigned to an average amount of sub-tasks.

5 Conclusion

This paper proposes a high-dimension data collection algorithm LCBPA for mobile crowd sensing network. In particular, we introduce the two-stage operation scheme to deal with the problem that a node with lower sensing capability confronted with the higher dimensional collection data. To evaluate our proposed scheme, we formulate the evaluation parameters, and we also calculate the task completion ratio, the total cost of the data, and the node participation equality. As a result, our scheme can effectively work when the network scale varies from 50 to 500, and the sub-task number k varies from 10 to

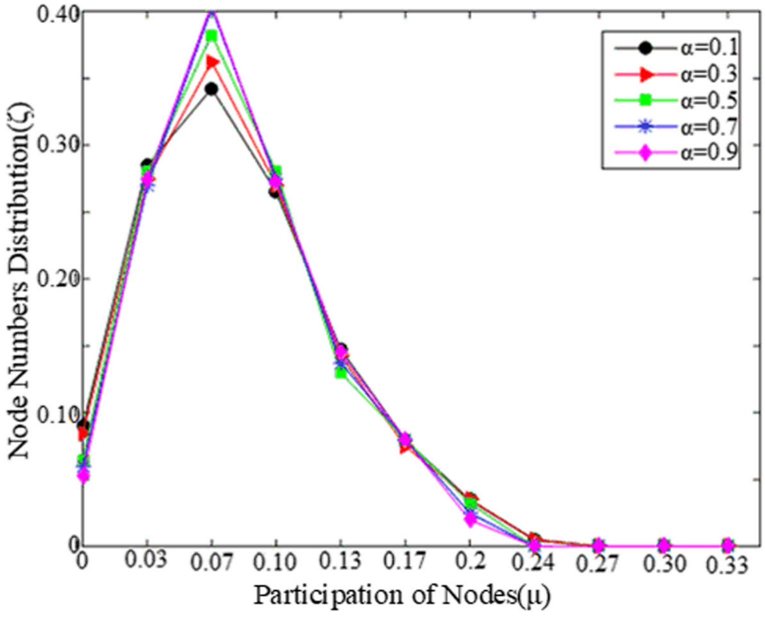

Fig. 10 Comparison of node participation equality of LCBPA with different trade-off parameter α when $k = 20$, $N = 500$

50, with trade-off weight parameter α varying from 0.1 to 0.9. However, there are also some limitations of our proposed schemes: (1) we only simulated our proposed method by simulations, not in the real sensing activities; (2) Our method can reduce the sensing cost and difficulty, while it is only works based on the assumption that each dimension of the data can be collected independently, which means that there is no correlation between different dimensions of the data, which may be not always the case. (3) When assigning each sensing task to different nodes, we do not consider the location and mobility of a node. (4) In practical, the value of the data will be elapsed as the time passed, which is also not considered in our proposed method. In the future work, we will evaluate hardware-based experiments, and take the node mobility and location into consideration, and we will also consider the data value as one of the factors in task allocation.

Abbreviations

DTA: Dual task assigner; LCBPA: Low-cost and balance-participating algorithm; MCS: Mobile crowd sensing; NTD: Non-task-division

Acknowledgements

Not applicable.

Authors' contributions

NZ proposed an idea and provided the guidance for deriving these expressions. JZ verified the correctness of the simulations cooperatively. BW improved the presentation of the draft and provided valuable suggestions. JX has great and distinct contribution to the writing and editing on the new revision version of this manuscript, and all the author agree to add him as the forth author. All authors read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

The datasets generated and analyzed during the current study are not publicly available, but are available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]National Digital Switching System Engineering and Technology Research Center, Zhengzhou 450002, People's Republic of China. [2]Xinxiang University, NO.191, Jinsui Avenue, Xinxiang 453000, China. [3]Beijing University of Posts and Telecommunications, Beijing 100086, China.

Received: 10 June 2019 Accepted: 29 November 2019
Published online: 23 December 2019

References

1. R.K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: Current state and future challenges. IEEE Commun. Mag. **49**(11), 32–39 (2011)
2. R. K. Rana, C. T. Chou , S. S. Kanhere, et al. Ear-Phone: An End-to-End Participatory Urban Noise Mapping System. ACM/IEEE International Conference on Information Processing in Sensor Networks. (ACM, Stockholm, 2010) pp.105–116
3. S. Liu, Y. Liu, L. Ni, et al., Detecting crowdedness spot in city transportation. IEEE Trans. Veh. Technol. **62**(4), 1527–1539 (2013)
4. Y. Altshuler, M. Fire, N. Aharony, et al., Trade-offs in social and behavioral modeling in mobile networks. Int. Conf. Soc. Comput. **78**(12), 412–423 (2013)
5. R. Pryss, M. Reichert, J. Herrmann, et al., *Mobile Crowd Sensing in Clinical and Psychological Trials -- A Case Study*, IEEE International Symposium on Computer-Based Medical Systems (IEEE, Sao CArlos, 2015), pp. 23–24
6. M. Ra, B. Liu, T.L. Porta, et al., *Medusa: A Programming Framework for Crowd-Sensing Applications*, International Conference on Mobile Systems (2012), pp. 337–350
7. L.Y. Qi, W.C. Dou, W.P. Wang, et al., Dynamic mobile crowdsourcing selection for electricity load forecasting. IEEE ACCESS **6**, 46926–46937 (2018)
8. Z. Gao, H.Z. Xuan, H. Zhang, et al., Adaptive fusion and category-level dictionary learning model for multi-view human action recognition. IEEE. Intern. Things. vol. 6 J., 1–14 (2019)
9. L.Y. Qi, R.L. Wang, S.C. Li, et al., Time-aware distributed service recommendation with privacy-preservation. Inf. Sci. **480**, 354–364 (2019)
10. D. Zhang, L. Wang, H. Xiong, et al., 4W1H in mobile crowd sensing. IEEE Commun. Mag. **52**(8), 42–48 (2014)
11. C. Yan, X.C. Cui, L.Y. Qi, et al., Privacy-aware data publishing and integration for collaborative service recommendation. IEEE ACCESS **6**, 43021–43028 (2018)
12. L.Y. Qi, Y. Chen, Y. Yuan, et al., A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems. World Wide Web. vol. 22 J., 1–23 (2019)
13. W. Li, X. Liu, J. Liu, et al., On Improving the accuracy with auto-encoder on conjunctivitis. Appl. Soft Comput., vol 81, 1–11 (2019)
14. W.W. Gong, L.Y. Qi, Y.W. Xu, Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment. Wirel. Commun. Mob. Comput. **2018**, 1–8 (2018)
15. S. He, Q. Yang, R.W.H. Lau, et al., *Visual Tracking Via Locality Sensitive Histograms*, Proceedings of the IEEE conference on computer vision and pattern recognition (IEEE, Portland, 2013), pp. 2427–2434
16. L.Y. Qi, S.M. Meng, X.Y. Zhang, et al., An exception handling approach for privacy-preserving service recommendation failure in a cloud environment. Sensors **18**(7), 1–11 (2018)
17. L.Y. Qi, W.C. Dou, J.J. Chen, et al., Weighted principal component analysis-based service selection method for multimedia Services in Cloud. Computing **98**(1), 195–214 (2016)
18. C.J. Ho, J.W. Vaughan, *Online Task Assignment in Crowdsourcing Markets*, AAAI Conference on Artificial Intelligence (2012), pp. 45–51
19. C.M. Angelopoulos, S. Nikoletseas, T.P. Raptis, et al., *Characteristic Utilities, Join Policies and Efficient Incentives in Mobile Crowd Sensing Systems*, Proceedings of the IFIP Wireless Days (2014), pp. 1–6
20. S. He, D.H. Shin, J. Zhang, et al., Toward optimal allocation of location dependent tasks in crowdsensing. Proce. IEEE. Int. Conf. Comput. Commun **5**(11), 745–753 (2014)
21. T. Dang, W.C. Feng, N. Bulusu, *Zoom: A Multi-Resolution Tasking Framework for Crowdsourced Geo-spatial Sensing*, Proceedings - IEEE INFOCOM (2011), pp. 501–505
22. T. Huy, N. Bulusu, D. Thanh, et al., *Scalable Map-Based Tasking for Urban Scale Multi-Purpose Sensor Networks*, Proceedings of the IEEE Topical Conference on Wireless Sensors and Sensor Networks (2013), pp. 142–144
23. H. Chen, Z. Siwang. *Raster-vector mixed task distribution method for mobile crowd sensing system*, J Computer Engineering and Applications. (23), 130-134 (2016)
24. J.T. Zhang, Z.H. Zhao, S.W. Zhou, Vector task map:progressive task allocation in crowd-sensing. Chinese J. Comput. **39**(3), 1–14 (2016)
25. C.C. Yin, C.W. Wang, Optimality of the barrier strategy in de Finetti's dividend problem for spectrally negative Lévy processes: an alternative approach. J. Comput. Appl. Math. **233**, 482–491 (2009)
26. P. Li, S.L. Zhao, R.C. Zhang, A cluster analysis selection strategy for supersaturated designs. Comput. Stat. Data. Anal. **54**(6), 1605–1612 (2010)