Optimal Task Allocation Algorithms for Energy Constrained Multihop Wireless Networks

Wanli Yu, *Member, IEEE,* Yanqiu Huang, *Member, IEEE,* and Alberto Garcia-Ortiz, *Member, IEEE*

***Abstract*—In recent years, multihop wireless networks have been playing a key role in many Internet of things applications. Due to the limited resources of wireless nodes, extending the network lifetime is one of the most crucial issues that needs to be concerned. This work aims to maximize the network lifetime by appropriately distributing the tasks of the applications for each node in the network. Firstly, a centralized optimal task allocation algorithm for multihop wireless networks (COTAM) is proposed by modeling the problem of maximizing the network lifetime as a linear programming (LP) problem. As the centralized algorithm requires to know all the network parameters in advance, COTAM is mostly restricted to offline optimization in known environments. To extend the usability of the approach, this work further proposes a distributed optimal task allocation algorithm (DOTAM) based on Dantzig-Wolf decomposition. DOTAM divides the centralized large-sized LP problem into small-sized subproblems which are independently executed by each node. The proposed COTAM and DOTAM are tested by applying both the artificially generated applications and a realistic application. The extensive results demonstrate that DOTAM achieves the same performance as COTAM. Comparing with existing methods, they provide significant improvements on extending the network lifetime.**

***Index Terms*—Multihop wireless networks, energy efficiency, task allocation, network lifetime maximization, distributed optimization.**

I. INTRODUCTION

THE Internet of things (IoT) has been gaining vast attention from both academic and industrial communities. It enables the physical objects to communicate with one another to achieve specific objectives. One of the basic technologies to proliferate the IoT applications is multihop wireless networks, such as sensor networks or wireless ad-hoc networks [1], [2]. These networks consist of numbers of small-sized and low-cost sensor nodes with specific sensors. In many applications, like structural health monitoring [3] or video based surveillance [4], the sensor nodes are required to not only act as independent processing terminals but also collaborate with others to execute computationally intensive tasks [5]. Due to the fact that the sensor nodes are typically powered by battery energy, the maximization of the network lifetime is always one of the critical issues that need to be concerned.

As the sensor nodes may have different residual battery energy and computing capacities, the allocation of the processing tasks for them has a strong effect on the application performance in terms of energy consumption. Moreover, many real-time applications require the tasks to be completed within a limited period. Thus, the end-to-end execution time of the applications should also be considered when assigning tasks to the nodes. Consequently, it is crucial to design appropriate task allocation algorithms for extending the network lifetime while satisfying the execution time requirement.

Recent years have seen research progress on designing task allocation algorithms for resource constrained wireless networks. An optimal task allocation algorithm is proposed in [6], it provides a static partition of the tasks to extend the network lifetime. The authors in [7] develop a weighted centralized task allocation algorithm, which provides dynamic task allocation solutions for each node in the network. A centralized heuristic algorithm is proposed in [8] to distribute the tasks among the sensor nodes. In addition to the centralized algorithm, a distributed optimal online task allocation algorithm, DOOTA, is presented in [9] to maximize the network lifetime using dynamic partitions. Each node can calculate its own task allocation solutions with very light-weighted computation. However, these works focus on one-hop-cluster based wireless networks, which restricts their wide utilizations. For multihop wireless networks, heuristic algorithms adopting bionic intelligence are preferred, considering the simplicity and easy implementation. A modified binary particle swarm optimization (PSO) algorithm is designed in [10] to calculate the task allocation solutions for extending the network lifetime. The potential task allocation solutions are encoded as particles with binary matrix formats. Similarly, the authors in [11] present a task allocation algorithm based on discrete PSO. In addition to PSO, the authors in [12], [13] focus on the studies of genetic algorithm (GA) based methods to address the task allocation problem. They model the complete partition solutions of the application as the binary chromosomes. Nevertheless, these bio-inspired algorithms cannot guarantee the global optimal solutions. Even worse, they need numerous iterations to reach the better solutions, which could lead to a long processing time [14], [15].

In order to address the limitations of the existing task allocation algorithms, this paper proposes both a centralized and a distributed optimal task allocation algorithms to maximize the network lifetime while ensuring that the application execution time is within the predefined period. Comparing with the approaches in [6]–[9] which are limited to the one-hop-cluster based simple network structure, the algorithms in this paper are designed for more general multihop wireless networks. Comparing with the bio-inspired methods in [10]–[13], the

Wanli Yu, Yanqiu Huang and Alberto Garcia-Ortiz are with the Institute of Electrodynamics and Microelectronics, University of Bremen, 28359, Bremen, Germany (e-mail: wyu@uni-bremen.de, yanqiu@uni-bremen.de, agarcia@item.uni-bremen.de).

proposed task allocation algorithms need less computation while providing the optimal dynamic partition solutions for each node in the network. The main contributions of this paper are summarized as follows:

- It models the energy consumption of sensor nodes and the application execution time as linear functions of the partitions. Then, a centralized optimal task allocation algorithm (COTAM) is proposed by formulating the problem of maximizing the network lifetime as a linear programming (LP) problem. As a centralized algorithm, COTAM is mostly applied for offline optimization in known environment such as in industrial IoT.
- A distributed optimal task allocation algorithm (DOTAM) based on Dantzig-Wolf decomposition theory is further proposed to break through the limit of the centralized algorithm. It divides the large-scaled centralized LP problem into small-sized subproblems. Each sensor node independently executes one subproblem to calculate the optimal partition solutions.

The rest of this paper is organized as follows. Section II presents the task and network models, the cost functions and the definition of network lifetime. The proposed COTAM algorithm is presented in Section III. In the next section, we illustrate the detailed information of DOTAM algorithm. After that, extensive simulation results are reported in Section V. The last section summarizes this work.

For illustration purpose, this paper uses the bold notations to represent arrays or matrices and the normal notations to stand for the scalars, respectively. The frequently used symbols and abbreviations are illustrated in Table I.

TABLE I: Frequently used symbols and abbreviations.

Notation	Definition
$t_{pi,k}$	Execution time of node N_i for processing task v_k.
P_i	Average processing power of node N_i.
$E_{pi,k}$	Energy cost of node N_i for processing task v_k.
$e_{to,i}/e_{ro,i}$	Transmitting/receiving overhead energy cost of N_i.
e_{tx}/e_{rx}	Energy dissipated for transmitting/receiving 1 bit data.
$E_{tx,i}/E_{rx,i}$	Transmitting/receiving energy cost of node N_i.
$t_{to,i}/t_{ro,i}$	Transmitting/receiving overhead time of node N_i.
B_i	Bandwidth of node N_i.
$T_{tx,i}/T_{rx,i}$	Transmitting/receiving time of node N_i.
$\boldsymbol{C}_i$	The set of posterity nodes of node N_i.
$E^y_{i,i}/E^y_{i,j}$	Energy cost of node N_i for DAG i/j at y-th round.
E^y_i	Energy cost of node N_i at the y-th round.
$E^y_{s,i}$	Energy cost of sink node for node N_i at y-th round.
E^y_s	Energy cost of sink node at the y-th round.
Y/ψ	Network lifetime/The reciprocal of network lifetime.
$\boldsymbol{X}^y_i$	Complete partition solutions of DAG i at y-th round.
Bat_i/Bat_s	Battery energy of node N_i/sink node.
$\boldsymbol{X}^y_{i,i}/\boldsymbol{X}^y_{i,j}$	Partition: tasks of DAG i/j assigned to node N_i.
$\boldsymbol{E}_{pi,i}/\boldsymbol{E}_{pi,j}$	The set of processing cost of tasks in DAG i/j when they are executed by node N_i.
$\boldsymbol{L}_i$	The set of net generated data of tasks in DAG i.
$\boldsymbol{R}_{i,j}/\boldsymbol{R}_{s,j}$	The set of nodes on the path from N_j to N_i/sink.
T^y_i	Completion time of DAG i at the y-th round.
$\boldsymbol{T}_{pr,i}/\boldsymbol{T}_{ps,i}$	The set of processing time of tasks in DAG i when they are executed by node N_r/sink node.
K_i	The number of tasks in DAG i.
$\boldsymbol{D}^T_i$	Transpose of the incidence matrix of DAG i.
$\overline{E}_i/\overline{E}_s$	Average energy cost of N_i/sink node at each round.
$\boldsymbol{\chi}_{r,i}$	The set of probabilities of tasks in DAG i being executed by node N_r.
λ^j_i	Coefficient of j-th basic solution for i-th subproblem.
$\boldsymbol{\chi}^*_i/\rho^*_i$	Optimal solution/objective value of i-th subproblem.
$\boldsymbol{\Phi}$	The set of simplex multipliers for the subproblems.
C/DOTAM	Centralized/Distributed optimal task allocation for multihop wireless networks.
GA [12]	Genetic algorithm.
DOOTA [9]	Distributed optimal online task allocation.

II. System model

This section presents the system models: the task and network models, the cost functions of the wireless sensor nodes and the definition of the network lifetime.

A. Task and Network Models

Typically, the mission of a wireless network for an application consists of a set of dependent computation tasks [6], [9], [12], which can be modeled by a Directed Acyclic Graph (DAG). In a DAG, $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, each vertex $v \in \boldsymbol{V}$ represents one task of the application and connects with others by directed edges; and each edge $e \in \boldsymbol{E}$ stands for the communication from its source task to its direct destination task. Each task is executed until it receives the data from all of its predecessors, and then transmits the generated data to its successors.

The network is composed of n wireless sensor nodes, $N_1, \cdots, N_n$, and one sink node. Each sensor node is an individual source node with a specific sensor and has to periodically execute its own application[1]. In addition to the abilities of sensing, processing and transmitting, the sensor nodes can also operate as routing nodes. The sensed source data or processed data of the sender is propagated by multiple wireless hops to the destination (sink node). Besides forwarding the data, the routing nodes can also share the processing tasks of the sender. As the wireless transmission range is limited, only the neighbor nodes within the transmitting range of the sender are able to play the role of routing. The selection of routing nodes depends on the specific routing algorithms, e.g., minimum hop routing [16], [17], geographic location based routing [18], network experience based routing [19], etc. Note that the selected routing algorithm does not affect the core of our proposed methods. Once the routing path of each sensor node is built, the multihop wireless network can be formed as a tree structured network. Based on this, the proposed methods can be executed in both centralized and distributed ways to find the optimal task allocation solutions. The corresponding illustration of the task allocation for tree structured networks is shown in Fig. 1.

B. Cost Functions

In a multihop wireless network, each sensor node mainly spends energy on processing, transmitting, receiving and sleeping. As the sleeping power is typically very small [20], it is neglected in this work.

The execution time of node N_i for processing task v_k is $t_{pi,k} = w_k/f_i$, where w_k is the computation workload of task v_k (the required number of CPU clock cycles) and f_i

[1]This work can be directly extended for the scenarios that one node has multiple sensors and executes different applications.

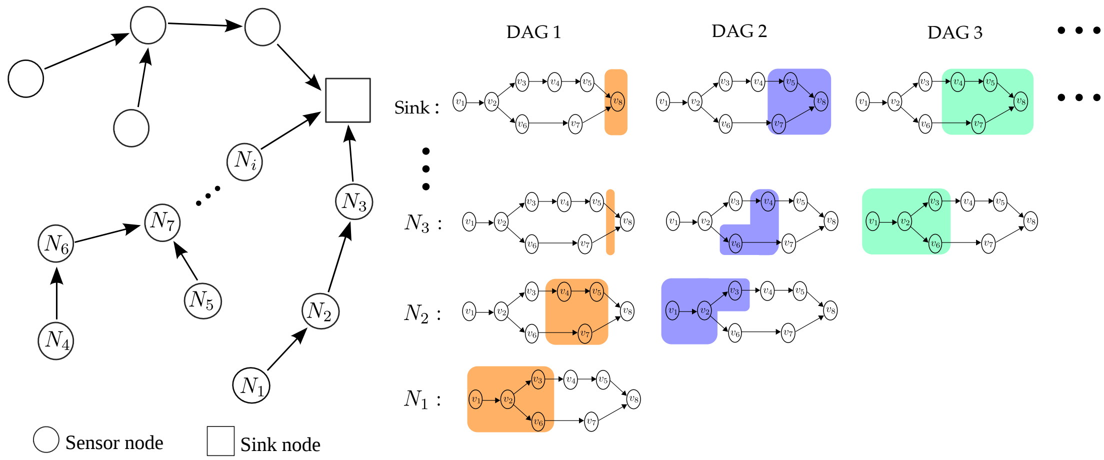


Fig. 1: The illustration of the task allocation for a tree structured multihop wireless network.

is the processing speed of node N_i. Let P_i denote the average processing power of node N_i, the corresponding processing energy cost can be formulated by:

$$E_{pi,k} = P_i\, t_{pi,k} \tag{1}$$

As the wireless communication process of a node includes not only the data packets communication but also overhead activities [6], the energy cost of node N_i for transmitting and receiving L bits of data, E_{tx} and E_{rx}, can be expressed as:

$$E_{tx,i} = e_{to,i} + e_{tx}L \tag{2}$$

$$E_{rx,i} = e_{ro,i} + e_{rx}L \tag{3}$$

where $e_{to,i}$ and $e_{ro,i}$ are the transmitting and receiving overhead cost of node N_i; e_{tx} and e_{rx} are the energy dissipated by transmitting and receiving 1 bit of data, respectively. Note that e_{tx} is typically a function associated with the transmission distance. The corresponding durations of node N_i for transmitting and receiving L bits of data are:

$$T_{tx,i} = t_{to,i} + L/B_i \tag{4}$$

$$T_{rx,i} = t_{ro,i} + L/B_i \tag{5}$$

where $t_{to,i}$ and $t_{ro,i}$ are the overhead time of node N_i for transmitting and receiving, respectively; B_i is the bandwidth of node N_i.

C. Network Lifetime

Extending the network lifetime is one of the most important issues in energy limited wireless networks. Network lifetime can be defined based on different metrics, such as the number of alive nodes, the coverage and connectivity of the network, the quality of service, etc. [15]. Moreover, the combination of these metrics has been used to define the network recently. For example, in [21], the network is firstly divided into multiple sleep-awake scheduled disjoint dominating sets, each of which provides full coverage of the network; a dominating set dies when the first node in it depletes the battery. The network lifetime is then the summation of the lifetime of all the dominating sets.

In many applications, e.g., mission critical applications, every sensor node is equally crucial to the operation of the network. Therefore, many existing references, e.g., [9], [12], [22], define the network lifetime as the duration from the network starts until the first node dies (1 out of n). In this work, we also consider such kind of applications and adopt this definition for the network lifetime.

III. Centralized Optimal Task Allocation

This section firstly presents the problem statement of maximizing the network lifetime. It then proposes a **C**entralized **O**ptimal **T**ask **A**llocation algorithm for **M**ultihop wireless networks, COTAM, to solve the network lifetime maximization problem based on linear programming (LP).

A. Problem formulation

As the sensor nodes periodically execute the applications (DAGs), their lifetimes can be considered as the total number of rounds of the complete execution of the DAGs. In order to maximize the network lifetime, we aim to balance the energy cost of each node using appropriate task allocation.

The energy cost of each node is associated with the partitions of the tasks of each application (DAG), i.e., the tasks assigned to each node. Let $\boldsymbol{C}_i$ denote the set of nodes whose paths to the sink node pass through node N_i, which means the nodes in $\boldsymbol{C}_i$ are the posterity of node N_i. Taking Fig. 1 for example, $\boldsymbol{C}_7 = \{N_4, N_5, N_6\}$. As mentioned in Section II-A, node N_i is able to play the role of routing and share the tasks of its posterity in addition to the tasks of its own application, DAG i. The energy cost of node N_i at the y-th round is:

$$E_i^y = E_{i,i}^y + \sum_{j \in \boldsymbol{C}_i} E_{i,j}^y \tag{6}$$

where $E_{i,i}^y$ is the energy cost of node N_i for executing the assigned tasks of its own DAG, which includes the cost of processing and transmitting; $E_{i,j}^y$ is the energy spent by node N_i for its posterity node N_j, which is made up of the cost of receiving, processing the assigned tasks of DAG j and transmitting. As the sink node is in charge of receiving the data from all of the n sensor nodes and finishing the rest tasks of all DAGs, its energy cost at the y-th round can be expressed by:

$$E_s^y = \sum_{i=1}^{n} E_{s,i}^y \tag{7}$$

where $E_{s,i}^y$ is the energy cost spent by the sink node for node N_i at the y-th round, which consists of the cost of receiving and processing the assigned tasks of DAG i.

According to the network lifetime definition, the problem of maximizing the network lifetime is to maximize the total number of rounds of the DAGs execution, Y, by appropriately assigning the tasks of each DAG to the sensor nodes and the sink node. Let $\boldsymbol{X}_i^y$ denote the complete map of the tasks in DAG i onto the sensor nodes and the sink node at the y-th round. In other words, $\boldsymbol{X}_i^y$ is a set of partitions of the tasks of DAG i. Consequently, the maximization problem is to find the optimal $\boldsymbol{X}_i^y$ at each round to maximize Y, which can be formulated by:

$$\begin{aligned} &\arg \max_{\boldsymbol{X}_i^y} \quad Y \\ &\text{subject to:} \quad \sum_{y=1}^{Y} E_i^y \leq Bat_i,\ i = 1, \cdots, n \\ &\qquad\qquad\quad \sum_{y=1}^{Y} E_s^y \leq Bat_s \\ &\qquad\qquad\quad \text{and constraints of each DAG} \end{aligned} \tag{8}$$

where Bat_i and Bat_s are the corresponding battery energy of node N_i and the sink node, respectively; the constraints of DAG i include: *Constraint a*), the first (sensing) and last tasks have to be executed by node N_i and the sink node, respectively; *Constraint b*), the complete execution time of DAG i has to be within the time limitation which is defined by the users.

B. COTAM Algorithm

The COTAM algorithm is described in this section. It formulates the problem Eq. (8) as a linear programming (LP) problem and provides the optimal task allocation solution for each node to maximize the network lifetime.

Since the battery energy of each node is constant, we focus on linearly formulating the energy cost of the nodes. Firstly, a binary variable $x_{i,k}$ is used to represent the relation between node N_i and task v_k: $x_{i,k} = 1$ indicates task v_k is assigned to node N_i; otherwise $x_{i,k} = 0$. Then, the tasks of DAG j assigned to node N_i can be expressed by the so-called partition $\boldsymbol{X}_{i,j} = [x_{i,1}, \cdots, x_{i,K_j}]^T$, where K_j is the number of tasks in DAG j. The energy cost of node N_i for executing the tasks of its own application DAG i at the y-th round, i.e., $E_{i,i}^y$ in Eq. (6), is made up of the cost for processing the assigned tasks and transmitting data to its direct routing node. According to Eqs. (1) and (2), $E_{i,i}^y$ can be formulated as:

$$E_{i,i}^y = \boldsymbol{E}_{pi,i}\boldsymbol{X}_{i,i}^y + e_{to,i} + e_{tx}\boldsymbol{L}_i\boldsymbol{X}_{i,i}^y \tag{9}$$

where $\boldsymbol{E}_{pi,i} = [E_{pi,1}, \cdots, E_{pi,K_i}]$ represents the processing cost of each task in DAG i when they are executed by node N_i; $\boldsymbol{L}_i = [l_i, \cdots, l_k, \cdots, l_{K_i}]$ stands for the net generated data of each task in DAG i, and l_k is the difference between the generated data and the input data of task v_k. The energy cost of node N_i for its posterity node N_j, i.e., $E_{i,j}^y$ in Eq. (6), consists of the cost of receiving the data, processing the assigned tasks and transmitting the generated data. Let $\boldsymbol{R}_{i,j}$ denote the set of nodes on the path from N_j to N_i (N_i is not included). Taking Fig. 1 for instance, $\boldsymbol{R}_{7,4} = \{N_4, N_6\}$. The amount of data that node N_i received and transmitted for DAG j are formulated by $\boldsymbol{L}_j \sum_{r\in\boldsymbol{R}_{i,j}} \boldsymbol{X}_{r,j}^y$ and $\boldsymbol{L}_j\big(\boldsymbol{X}_{i,j}^y + \sum_{r\in\boldsymbol{R}_{i,j}} \boldsymbol{X}_{r,j}^y\big)$, respectively. Consequently, $E_{i,j}^y$ can be given by:

$$\begin{aligned} E_{i,j}^y =& e_{ro,i} + e_{rx}\boldsymbol{L}_j \sum_{r\in\boldsymbol{R}_{i,j}} \boldsymbol{X}_{r,j}^y + \boldsymbol{E}_{pi,j}\boldsymbol{X}_{i,j}^y \\ &+ e_{to,i} + e_{tx}\boldsymbol{L}_j\big(\boldsymbol{X}_{i,j}^y + \sum_{r\in\boldsymbol{R}_{i,j}} \boldsymbol{X}_{r,j}^y\big) \end{aligned} \tag{10}$$

As the energy cost spent by the sink node for node N_i, i.e., $E_{s,i}^y$ in Eq. (7), consists of receiving the data of DAG i and executing the rest tasks, it can be calculated by:

$$E_{s,i}^y = e_{ro,s} + e_{rx}\boldsymbol{L}_i \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y + \boldsymbol{E}_{ps,i}\big(\boldsymbol{1} - \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y\big) \tag{11}$$

According to Eqs. (6), (7) and (9) to (11), the energy cost of node N_i and the sink node at the y-th round, E_i^y and E_s^y, are standard linear functions associated with the partitions $\boldsymbol{X}_{r,i}^y, r \in \boldsymbol{R}_{s,i}$.

DAG i is completed by the collaboration of nodes in $\boldsymbol{R}_{s,i}$, i.e., sensor node N_i and its routing nodes, and the sink node. Therefore, the completion time of DAG i at the y-th round, T_i^y, is expressed by:

$$\begin{aligned} T_i^y =& \sum_{r\in\boldsymbol{R}_{s,i}} \Big(\big(t_{ro,r} + \frac{\boldsymbol{L}_i}{B_r} \sum_{j\in\boldsymbol{R}_{r,i}} \boldsymbol{X}_{j,i}^y\big)\eta + t_{to,r} + \frac{\boldsymbol{L}_i}{B_r}\big(\boldsymbol{X}_{r,i}^y + \sum_{j\in\boldsymbol{R}_{r,i}} \boldsymbol{X}_{j,i}^y\big) \\ &+ \boldsymbol{T}_{pr,i}\boldsymbol{X}_{r,i}^y \Big) + t_{ro,s} + \frac{\boldsymbol{L}_i}{B_s} \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y + \boldsymbol{T}_{ps,i}\big(\boldsymbol{1} - \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y\big) \end{aligned} \tag{12}$$

where $\boldsymbol{T}_{pr,i} = [t_{pr,1}, \cdots, t_{pr,K_i}]$ and $\boldsymbol{T}_{ps,i} = [t_{ps,1}, \cdots, t_{ps,K_i}]$ represent the processing time of each task in DAG i when they are executed by node N_r and the sink node, respectively; $\eta = 0$ when $r = i$, otherwise $\eta = 1$, because sensor node N_i does not need to receive any data from DAG i. It is obvious that Eq. (12) is also a standard linear equation associated with the partitions $\boldsymbol{X}_{r,i}^y, r \in \boldsymbol{R}_{s,i}$.

Based on the linear Eqs. (6), (7) and (9) to (12), the original maximization problem Eq. (8) can be formulated by a binary integer LP (BILP) problem as follows:

$$\arg \max_{\boldsymbol{X}_{r,i}^y, r\in\boldsymbol{R}_{s,i}} \quad Y, \quad i = 1, \cdots, n \tag{13}$$

$$\text{subject to:} \quad \sum_{y=1}^{Y} E_i^y \leq Bat_i \tag{14}$$

$$\sum_{y=1}^{Y} E_s^y \leq Bat_s \tag{15}$$

$$1 \leq \boldsymbol{1} \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y \leq K_i - 1 \tag{16}$$

$$\boldsymbol{D}_i^T \sum_{r\in\boldsymbol{R}_{s,i}} \boldsymbol{X}_{r,i}^y \geqq \boldsymbol{0} \tag{17}$$

$$T_i^y \leq t_{li} \tag{18}$$

where Eqs. (16) and (17) are the linear formulation of *Constraint a* as mentioned in Section III-A; $\boldsymbol{D}_i^T$ is the transpose of the incidence matrix of DAG i, which has the row for each task and column for each edge ($\boldsymbol{D}_i(v,e) = 1$ if edge e leaves task v, $\boldsymbol{D}_i(v,e) = -1$ if edge e enters task v and $\boldsymbol{D}_i(v,e) = 0$ otherwise); Eq. (18) represents the linear formulation of *Constraint b*, in which t_{li} is the execution time limitation of DAG i defined by the users. By solving the above BILP problem, the network lifetime can be maximized with the partition $\boldsymbol{X}_{r,i}^y$ in every round. However, it is very hard to solve this problem, because: i) BILP is a non-convex optimization problem with high complexity; ii) The number of variables is $Y\sum_{i=1}^n K_i|\boldsymbol{R}_{s,i}|$, where the network lifetime Y is usually very large (can be hundreds of thousands) and $|\cdot|$ is the cardinality of the set.

Towards the above-mentioned problem, the COTAM algorithm is proposed by converting the non-convex BILP to a convex LP optimization problem with fixed number of variables. According to the definition of the network lifetime, the optimization problem is to maximize the minimum lifetime among the nodes. The lifetimes of the sink node and sensor node N_i can be expressed by $Bat_s/\overline{E}_s$ and $Bat_i/\overline{E}_i$, where $\overline{E}_s$ and $\overline{E}_i$ denote the average energy cost of the sink node and sensor node N_i at each round, respectively.

Based on Eqs. (7) and (11), $\overline{E}_s$ can be given by:

$$\begin{aligned}\overline{E}_s &= \frac{1}{Y}\sum_{y=1}^{Y}\sum_{i=1}^{n} E_{s,i}^y \\ &= \sum_{i=1}^{n}\Big(e_{ro,s} + e_{rx}\boldsymbol{L}_i\sum_{r\in\boldsymbol{R}_{s,i}}\boldsymbol{\chi}_{r,i} + \boldsymbol{E}_{ps,i}\big(\boldsymbol{1} - \sum_{r\in\boldsymbol{R}_{s,i}}\boldsymbol{\chi}_{r,i}\big)\Big) \end{aligned} \tag{19}$$

where

$$\boldsymbol{\chi}_{r,i} = \frac{1}{Y}\sum_{y=1}^{Y}\boldsymbol{X}_{r,i}^y. \tag{20}$$

As Y is very large and each element in $\boldsymbol{X}_{r,i}^y$ equals either 1 or 0, the element in $\boldsymbol{\chi}_{r,i}$ can be treated as a real number between 0 and 1, which indicates the probability of each task being executed by the sensor nodes. Accordingly, based on Eqs. (6), (9) and (10), $\overline{E}_i$ is given by:

$$\begin{aligned}\overline{E}_i = \boldsymbol{E}_{pi,i}\boldsymbol{\chi}_{i,i} + e_{to,i} + e_{tx}\boldsymbol{L}_i\boldsymbol{\chi}_{i,i} + \sum_{j\in\boldsymbol{C}_i}\Big(\boldsymbol{E}_{pi,j}\boldsymbol{\chi}_{i,j} \\ + e_{ro,i} + e_{rx}\boldsymbol{L}_j\sum_{r\in\boldsymbol{R}_{i,j}}\boldsymbol{\chi}_{r,j} + e_{to,i} + e_{tx,i}\boldsymbol{L}_j\big(\sum_{r\in\boldsymbol{R}_{i,j}}\boldsymbol{\chi}_{r,j} + \boldsymbol{\chi}_{i,j}\big)\Big)\end{aligned} \tag{21}$$

As each element in $\boldsymbol{\chi}_{r,i}$ is a real number, based on Eqs. (19) and (21), maximizing the network lifetime can be formulated by a LP problem as follows:

$$\begin{aligned}\arg\min_{\boldsymbol{\chi}_{r,i}, r\in\boldsymbol{R}_{s,i}} \quad & max\Big\{\frac{\overline{E}_s}{Bat_s}, \frac{\overline{E}_i}{Bat_i}\,\Big|\, i = 1,\cdots,n\Big\} \\ \text{subject to:} \quad & 1 \le \boldsymbol{1}\sum_{r\in\boldsymbol{R}_{s,i}}\boldsymbol{\chi}_{r,i} \le K_i - 1 \\ & \boldsymbol{D}_i^T\sum_{r\in\boldsymbol{R}_{s,i}}\boldsymbol{\chi}_{r,i} \geqq \boldsymbol{0}\end{aligned} \tag{22}$$

The optimal partition solution $\boldsymbol{\chi}_{r,i}$ for each node to maximize the network lifetime can be obtained by solving the above COTAM algorithm. Note that the obtained $\boldsymbol{\chi}_{r,i}$ consists of different partitions with the corresponding weights, which can be easily calculated. For example, $\boldsymbol{\chi}_{r,i} = [1, 0.2, 0, 0]$ corresponds to partition solutions $[1, 1, 0, 0]$ and $[1, 0, 0, 0]$ with weights 20% and 80%, respectively. Comparing with the BILP problem Eq. (13), COTAM significantly reduces the complexity because: i) It transforms a non-convex optimization into a convex LP optimization; ii) It reduces the number of variables from $Y\sum_{i=1}^n K_i|\boldsymbol{R}_{s,i}|$ to $\sum_{i=1}^n K_i|\boldsymbol{R}_{s,i}|$.

IV. Distributed Optimal Task Allocation

Although the centralized COTAM can provide the optimal task allocation solutions, it is frequently unrealistic to know all the parameters in advance; it also requires a large amount of battery and computation resources to execute the complex computation, i.e., Eq. (22). Therefore, COTAM is mostly applied for offline optimization in known environments as in partial cases of industrial IoT. To extend the usability of the approach, this section proposes a **D**istributed **O**ptimal **T**ask **A**llocation algorithm for **M**ultihop wireless networks, DOTAM, based on Dantzig-Wolf (D-W) decomposition. It solves the centralized problem in a distributed way and provides the same optimal solutions as COTAM. This section firstly reviews the background of D-W decomposition and then presents DOTAM algorithm.

A. Dantzig-Wolf Decomposition

The key underlying theory of DOTAM algorithm is D-W decomposition. For a large scale LP problems with special block-angular structured constraints, D-W can divide it into small sized subproblems connected with one master problem [23], [24]. Solving the transformed problem is equivalent to addressing the original centralized LP problem. The solutions are calculated iteratively by D-W, where the subproblems are solved and coordinated at each step, and ultimately the overall problem is solved.

Considering a centralized LP with the following format:

$$\begin{aligned}&\arg\min_{\boldsymbol{z}_i} \boldsymbol{f}_1^T\boldsymbol{z}_1 + \cdots, +\boldsymbol{f}_i^T\boldsymbol{z}_i + \cdots, +\boldsymbol{f}_n^T\boldsymbol{z}_n \\ &\text{subject to:} \begin{bmatrix}\boldsymbol{G}_1 & \boldsymbol{G}_2 & \cdots & \boldsymbol{G}_n \\ \boldsymbol{A}_1 & & & \\ & \boldsymbol{A}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{A}_n\end{bmatrix}\begin{bmatrix}\boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \vdots \\ \boldsymbol{z}_n\end{bmatrix} = \begin{bmatrix}\boldsymbol{b}_0 \\ \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \\ \vdots \\ \boldsymbol{b}_n\end{bmatrix} \\ &\boldsymbol{z}_i \geqq \boldsymbol{0}, \quad i = 1,\cdots,n\end{aligned} \tag{23}$$

where $\boldsymbol{z}_i$ represents the set of variables of the i-th subproblem; $\boldsymbol{G}_1,\cdots,\boldsymbol{G}_n$ are the coefficient matrices for the global equality constraints; $\boldsymbol{A}_1,\cdots,\boldsymbol{A}_n$ are the coefficient matrices for the n subproblems, which are independent with each other. Note that inequality equations can be converted into equalities by adding slack variables [24]. For the i-th subproblem, min $\boldsymbol{f}_i^T\boldsymbol{z}_i$, s.t. $\boldsymbol{A}_i\boldsymbol{z}_i = \boldsymbol{b}_i$, the constraint set is a polytope. According to the convex combination property that any point in a polytope can be expressed by the convex combination of the vertices (extreme points), $\boldsymbol{z}_i$ can be written as:

$$\boldsymbol{z}_i = \sum_{j=1}^{M_i} \lambda_i^j \boldsymbol{z}_i^j \tag{24}$$

where $\boldsymbol{z}_i^j, j = 1, \cdots, M_i$ are the M_i extreme points of the i-th polytope; $\lambda_i^j \geq 0$ are the corresponding weighting coefficients, which satisfy $\sum_{j=1}^{M_i} \lambda_i^j = 1$. By using Eq. (24), Eq. (23) is equivalent to the master problem:

$$\begin{aligned} &\arg\min_{\lambda_i^j} \sum_{i=1}^{n} \sum_{j=1}^{M_i} \boldsymbol{f}_i^T \boldsymbol{z}_i^j \lambda_i^j \\ &\text{subject to: } \sum_{i=1}^{n} \sum_{j=1}^{M_i} \boldsymbol{G}_i \boldsymbol{z}_i^j \lambda_i^j = \boldsymbol{b}_0 \\ &\sum_{j=1}^{M_i} \lambda_i^j = 1, \quad \lambda_i^j \geq 0 \end{aligned} \tag{25}$$

The above master problem can be written as matrix format:

$$\begin{aligned} &\arg\min_{\boldsymbol{\lambda}} \quad \boldsymbol{\Gamma}^T \boldsymbol{\lambda} \\ &\text{subject to: } \boldsymbol{Q}\boldsymbol{\lambda} = \boldsymbol{b}, \boldsymbol{\lambda} \geqq \boldsymbol{0} \end{aligned} \tag{26}$$

where $\boldsymbol{\Gamma}^T = [\boldsymbol{f}_1^T \boldsymbol{z}_1^1, \cdots, \boldsymbol{f}_1^T \boldsymbol{z}_1^{M_1}, \cdots, \boldsymbol{f}_n^T \boldsymbol{z}_n^{M_n}]$, $\boldsymbol{\lambda} = [\lambda_1^1, \cdots, \lambda_1^{M_1}, \cdots, \lambda_n^{M_n}]^T$, the column of $\boldsymbol{Q}$ associated with λ_i^j is $[\boldsymbol{G}_i \boldsymbol{z}_i^j, \boldsymbol{e}_i]^T$ in which $\boldsymbol{e}_i$ denotes the i-th unit vector in E^n, and $\boldsymbol{b} = [\boldsymbol{b}_0^T, 1, \cdots, 1]^T$.

Suppose an extreme point for the master problem, $\boldsymbol{z}_B$, is known, the corresponding basis matrix $\boldsymbol{Q}_B$ (a square matrix with the same number of rows as $\boldsymbol{Q}$) and $\boldsymbol{\Gamma}_B^T$ can be calculated. The master problem then generates simplex multipliers $\boldsymbol{\Phi}$ by:

$$\boldsymbol{\Phi} = \boldsymbol{\Gamma}_B^T \boldsymbol{Q}_B^{-1}, \tag{27}$$

and sends to each subproblem to solve:

$$\begin{aligned} &\rho_i^* = \arg\min_{\boldsymbol{z}_i} \left(\boldsymbol{f}_i^T - \boldsymbol{\Phi}_0 \boldsymbol{G}_i \right) \boldsymbol{z}_i - \phi_{m+i} \\ &\text{subject to: } \quad \boldsymbol{A}_i \boldsymbol{z}_i = \boldsymbol{b}_i, \ \boldsymbol{z}_i \geqq \boldsymbol{0} \end{aligned} \tag{28}$$

where $\boldsymbol{\Phi}_0$ and ϕ_{m+i} are the first m elements and the $m+i$ element of $\boldsymbol{\Phi}$, respectively, with m being the number of rows of $\boldsymbol{G}_i$. The optimal solution for Eq. (28), $\boldsymbol{z}_i^*$, is one of the extreme points, i.e., $\boldsymbol{z}_i^* \in \boldsymbol{z}_i^j | j = 1, \cdots, M_i$. If the objective value of Eq. (28) $\rho_i^* < 0$, applying $\boldsymbol{z}_i^*$ can reduce the objective value of the master problem Eq. (25). Thus, $\boldsymbol{z}_i^*$ will be added to the solution of the master problem. Meanwhile, $\boldsymbol{Q}_B$ and $\boldsymbol{\Gamma}_B^T$ of the master problem are updated by using the column in $\boldsymbol{Q}$ associated with $\boldsymbol{z}_i^*$ and $\boldsymbol{f}_i^T \boldsymbol{z}_i^*$ to replace one column of $\boldsymbol{Q}_B$ and one element in $\boldsymbol{\Gamma}_B^T$, respectively, through pivot operation [23], [24]. If all $\rho_i^* \geq 0, i = 1, \cdots, n$ are nonnegative, the objective value of the master problem cannot be further decreased and the current solutions of the subproblems are optimal for the master problem.

B. DOTAM Algorithm

Based on D-W decomposition, this section presents the DOTAM algorithm. Let $\psi = 1/Y$ represent the reciprocal of the network lifetime and $\boldsymbol{\chi}_i = [\boldsymbol{\chi}_{r,i}^T | r \in \boldsymbol{R}_{s,i}]^T$. Based on Eqs. (19) and (21), the network lifetime maximization problem Eq. (22) can be rewritten as:

$$\arg\min_{\psi, \boldsymbol{S}, \boldsymbol{\chi}_i} \quad \psi \tag{29}$$

$$\text{subject to: } \quad \sum_{i=1}^{n} \boldsymbol{G}_i \boldsymbol{\chi}_i - \boldsymbol{1}\psi + \boldsymbol{S} = \boldsymbol{b}_0 \tag{30}$$

$$\boldsymbol{A}_i \boldsymbol{\chi}_i \leqq \boldsymbol{b}_i \tag{31}$$

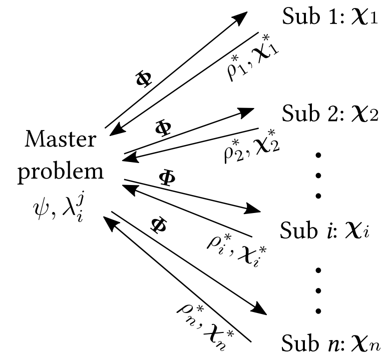


Fig. 2: The diagram of DOTAM algorithm based on D-W decomposition.

where $\boldsymbol{1}$ is a column vector with all $n+1$ elements being 1; $\boldsymbol{S}$ is a column vector with $n+1$ nonnegative slack variables $s_0, s_1, \cdots, s_n$; the derivations of $\boldsymbol{G}_i$, $\boldsymbol{b}_0$, $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ are illustrated in Appendix. It is obvious that Eq. (29) exactly fits the D-W format with the global equality constraints Eq. (30) and the independent inequality constraints Eq. (31) for each subproblem. Let us assume that we have already known the extreme points of each subproblem, according to the Eqs. (24) to (26), the above problem can be reformulated as a master problem:

$$\begin{aligned} &\arg\min_{\psi, \boldsymbol{S}, \boldsymbol{\lambda}} \quad \psi \\ &\text{subject to: } \quad \boldsymbol{Q}[\psi, s_0, \cdots, s_n, \lambda_1^1, \cdots, \lambda_1^{M_1}, \cdots, \lambda_n^{M_n}]^T = \boldsymbol{b} \end{aligned} \tag{32}$$

where $\boldsymbol{Q} = \begin{bmatrix} -\boldsymbol{1} & \boldsymbol{I} & \boldsymbol{G}_1 \boldsymbol{\chi}_1^1 & \cdots & \boldsymbol{G}_1 \boldsymbol{\chi}_1^{M_1} & \cdots & \boldsymbol{G}_n \boldsymbol{\chi}_n^{M_n} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_1 & \cdots & \boldsymbol{e}_n \end{bmatrix}$, $\boldsymbol{I}$ is an identity matrix; and $\boldsymbol{b} = [\boldsymbol{b}_0^T, \boldsymbol{1}]^T$.

When using DOTAM, the master problem is solved by the sink node and the subproblems are independently executed by each sensor node as shown in Fig. 2. Before DOTAM starts, a basis $\boldsymbol{Q}_B$ should be initialized: the sink node initializes one extreme point for each subproblem namely $\boldsymbol{\chi}_i^1$ with $\lambda_i^1 = 1$ by considering each sensor node just executes the first task of its own DAG and all the rest are done by the sink node, and sets $s_0 = 0$; the columns of $\boldsymbol{Q}$ associated with $\psi, s_1, \cdots, s_n, \lambda_1^1, \cdots, \lambda_n^1$ construct the initial $\boldsymbol{Q}_B$, and the corresponding weights are calculated by $\boldsymbol{W} = \boldsymbol{Q}_B^{-1} \boldsymbol{b}$. Note that sink node stores only $\boldsymbol{Q}_B^{-1}$ and $\boldsymbol{W}$.

After the initialization, the sink node generates the simplex multiplier $\boldsymbol{\Phi}$ and broadcasts to the sensor nodes. As only the coefficient of ψ is non-zero, based on Eq. (27), $\boldsymbol{\Phi}$ is actually the first row of $\boldsymbol{Q}_B^{-1}$. Based on $\boldsymbol{\Phi}$, sensor node N_i starts to solve its own subproblem to calculate ρ_i^*. Since the coefficient of $\boldsymbol{\chi}_i^j$ in Eq. (32) is $\boldsymbol{0}$, node N_i only needs to solve the i-th subproblem:

$$\rho_i^* = \arg\min_{\boldsymbol{\chi}_i} \quad -\boldsymbol{\Phi}_0 \boldsymbol{G}_i \boldsymbol{\chi}_i - \phi_{m+i} \qquad (33)$$
$$\text{subject to:} \quad \boldsymbol{A}_i \boldsymbol{\chi}_i \le \boldsymbol{b}_i, \; \boldsymbol{\chi}_i \geqq \boldsymbol{0}$$

The obtained ρ_i^* and the corresponding $\boldsymbol{\chi}_i^*$ are sent to the sink node. When the sink node receives these data, it selects the minimum ρ_i^*. For notational simplicity assume that ρ_i^* is the minimum. If $\rho_i^* < 0$, sink node generates a new column Nc to enter $\boldsymbol{Q}_B^{-1}$ using the associated $\boldsymbol{\chi}_i^*$ as follows:

$$Nc = \boldsymbol{Q}_B^{-1} \begin{bmatrix} \boldsymbol{G}_i \boldsymbol{\chi}_i^* \\ \boldsymbol{e}_i \end{bmatrix} \qquad (34)$$

Then, sink node updates $\boldsymbol{Q}_B^{-1}$ and $\boldsymbol{W}$ through pivot operation as introduced in [23], [24]. If $\rho_i^* \ge 0$, DOTAM stops and the sink node broadcasts a $confirm$ message to indicate the current extreme points with the corresponding weights are the optimal partition solutions.

The pseudo codes of DOTAM executed in sink node and sensor node N_i are shown in Algorithm 1 and Algorithm 2. The subproblems are solved by each sensor node independently and the sink node confirms the results by checking the master constraint. DOTAM provides the same solution as solving the original equivalent large-scaled LP problem using COTAM. DOTAM divides the original LP problem with $\sum_{i=1}^{n} K_i|\boldsymbol{R}_{s,i}|$ variables into n small-sized subproblems. Each subproblem has $K_i|\boldsymbol{R}_{s,i}|$ variables and can be solved individually by resource limited sensor node N_i to obtain the partition solution of DAG i.

Algorithm 1 Sink node algorithm

Initialize $\boldsymbol{Q}_B^{-1}$ and $\boldsymbol{W}$
Loop:
Calculate $\boldsymbol{\Phi}$ (first row of $\boldsymbol{Q}_B^{-1}$) and broadcast it
Receive ρ_i^* and $\boldsymbol{\chi}_i^*$ from each node
if $min\{\rho_i^* | i = 1, \cdots, n\} \ge 0$ **then**
  Broadcast $confirm$
  end loop
else
  Generate Nc using Eq. (34)
  Update $\boldsymbol{Q}_B^{-1}$ and $\boldsymbol{W}$ using Nc
end if

Algorithm 2 Node N_i algorithm

if receive $confirm$ **then**
  Break
else if receive $\boldsymbol{\Phi}$ **then**
  Calculate ρ_i^* and $\boldsymbol{\chi}_i^*$ using Eq. (33)
  Transmit to sink node
end if

V. Simulation Results

In this section, extensive simulations are employed to evaluate the performance of COTAM and DOTAM algorithms using both artificially generated applications and a realistic application. In order to illustrate their advantages, the simulation results are compared with the non-scheduling strategy, DOOTA algorithm proposed in [9] and a Genetic Algorithm (GA) based task allocation algorithm proposed in [12]:

- Non-scheduling strategy: each sensor node just executes the first task of its own DAG and directly forwards the data of its posterity nodes; all the rest tasks are done by the sink node.
- DOOTA [9]: as DOOTA provides the optimal solution for one-hop-cluster based networks, it is executed by considering that each node connects with the sink node by one wireless hop in this paper.
- GA [12]: a set of chromosomes (static task allocation solutions) iteratively generate the next generation by inheritance, crossover and mutation, until reaching the maximum number of iterations.

A. Simulation Setup

The network is generated with n sensor nodes randomly located in a two dimensional network area of 200 × 200 square meters and 1 sink node at the center. The maximum transmission range of each node is 40 meters. Each node transmits data to the sink node using the minimum hop routing algorithm [16], [17]. The battery energy of the wireless nodes and sink node are distributed within the ranges of $[1kJ, 5kJ]$ and $[10kJ, 20kJ]$, respectively. Besides, we consider the nodes execute the same application in the network.

The energy related parameters of the wireless nodes and the sink node in the networks are obtained from the datasheets of CC2538 system-on-chip [20] and the Texas Instruments TMS320C5509A [25], respectively. All of the nodes are considered to use the same RF module which works at the $2.4GHz$ ISM band with a bandwidth of 250 $Kbps$.

Each reported simulation result corresponds to the average values and the standard deviations of 500 test instances.

B. Simulation Results on Artificial Applications

This section evaluates the performance of COTAM and DOTAM using the artificial applications. The corresponding DAGs are randomly generated based on the number of tasks and edges. The computation workload of each task is distributed within the range of $[100, 1000]$ kilo clock cycles (KCCs). The amount of communicated data on each edge is distributed in the range of $[100, 500]$ bits. The increase of network lifetime with respect to the non-scheduling strategy and the algorithm runtime measured by executing them in Matlab are investigated by changing: a) The number of sensor nodes, n; b) The number of tasks, K. The configuration parameters are summarized in Table II, and only one parameter is changed in each experiment.

TABLE II: Configuration parameters.

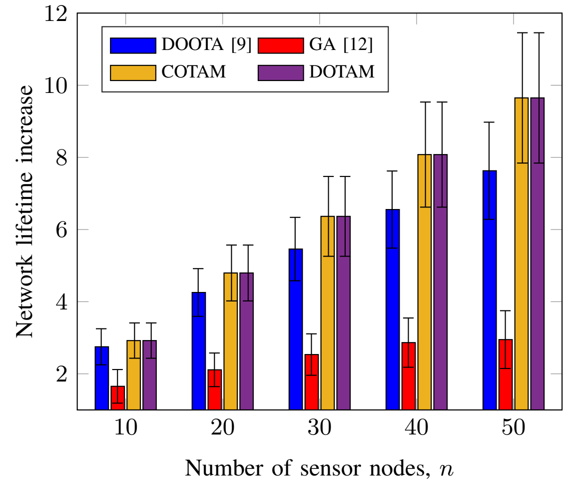
Parameters	Values	
	Default	Varied
Number of nodes, n	20	{10, 20, 30, 40, 50}
Number of tasks, K	10	{5, 10, 15, 20}

The first set of simulations is conducted to investigate the effect of the number of sensor nodes, n, on the algorithm performance. It can be easily seen from Fig. 3a that the improvements of the network lifetime by using the task allocation algorithms, with respect to the non-scheduling strategy, increase as n changes from 10 to 50. For instance, DOTAM extends the network lifetime on average by a factor of 2.92 and up to 9.65 times. This is because the task allocation algorithms can appropriately distribute the workload for the sensor nodes and the sink node, while the non-scheduling strategy leads the sink node exponentially overloaded and consequently causes a short network lifetime. As we expected, among the four algorithms, DOTAM extends the network lifetime as much as COTAM. Since both of them provide the optimal solutions, they increase the network lifetime the longest. Meanwhile, GA performs the worst, because it cannot fairly balance the workload distribution with one static partition for each node. Moreover, the superiority of COTAM and DOTAM over DOOTA is more significant as n increases, e.g., COTAM and DOTAM extend the network lifetime from 108.15% to 128.32% longer than DOOTA. The reason is that large number of nodes results in many wireless hops, while DOOTA only considers that the sensor nodes connect with the sink node by one hop. Consequently, the proposed algorithms perform better for network with more sensor nodes.

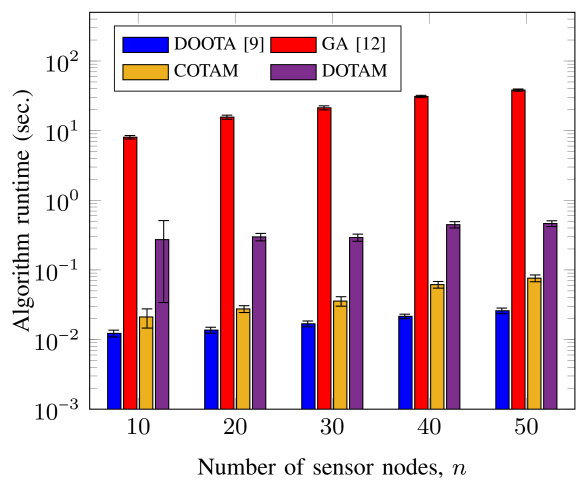
Further on, Fig. 3b shows that the time requirements for executing the task allocation algorithms. It is obvious that executing GA requires the longest time, which needs on average 38.24 seconds when there are 50 nodes. This is due to the fact that GA is an iterative algorithm and needs a large number of iterations for a good solution. An interesting observation is that executing the DOTAM algorithm needs more time than executing COTAM. This indicates that the centralized algorithm can be efficiently solved by some efficient solvers, e.g., Matlab. Thus, in some indutrial IoT scenarios with known environments, e.g., smart robotics, intelligent machine status monitoring, etc., it is better to use COTAM. In contrast, DOTAM can be applied in more general scenarios, where it is unrealistic to know all the network parameters in advance. Moreover, the runtime of COTAM and DOTAM slight increases as n changes from 10 to 50. The average runtime of DOTAM is within 0.5 seconds. Therefore, the overhead for executing the proposed algorithm is tolerant for small-to-medium sized wireless networks.

In addition to estimate the effect of the number of sensor nodes, another set of simulations is conducted to investigate the algorithm performance by changing the number of the tasks. As demonstrated in Fig. 4a, the average gains of the network lifetime improvement by using COTAM, DOTAM and DOOTA algorithms slightly increase while the gain of GA fluctuates, when the number of tasks, K, changes from 5 to 20. This is because that GA cannot guarantee the optimal partition solutions. Moreover, it can be seen that the superiorities of COTAM and DOTAM over DOOTA decrease. Specifically, they extend the network lifetime on average from 1.25 to 1.14 times longer than DOOTA, when K increases from 5 to 20. The reason is that the effect of multihop communication cost on DOOTA declines as the number of computation tasks increases.

Moreover, Fig. 4b depicts the execution time of the task



(a)



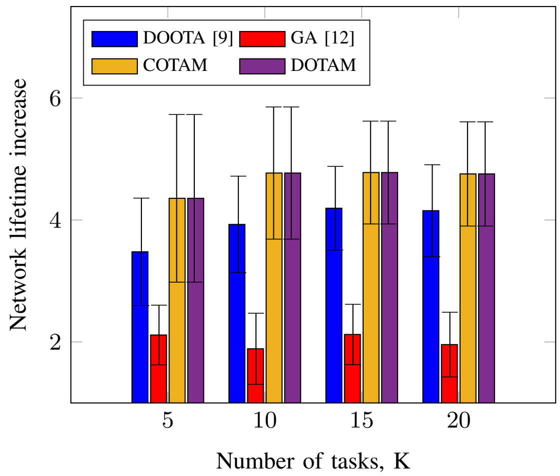
(b)

Fig. 3: Effect of the number of nodes on the performance of COTAM, DOTAM, GA [12] and DOOTA [9] algorithms (there are 10 tasks in each artificially generated application): (a) network lifetime increase w.r.t. non-scheduling strategy; (b) algorithm runtime.

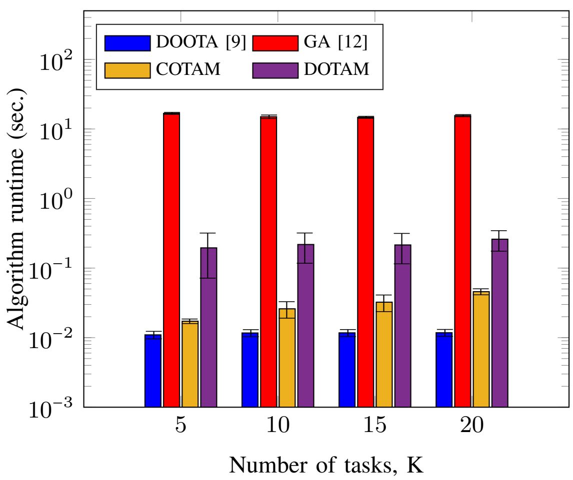
allocation algorithms with the increasing number of tasks, K. It is visible that the algorithm runtime of COTAM apparently grows while the runtime of the others slightly increase or even keep stable, when K changes from 5 to 20. This can be explained by the fact that the number of variables in COTAM algorithm is proportional to nK. As a result, the increase of execution time of COTAM is much more significant than the others.

C. A Case Study Using Realistic Applications

This section uses a realistic in-network processing application, maximum entropy power spectrum computation (MEPS) [26], [27], as a case study to further evaluate the algorithm performance. Fig. 5 depicts the modeled DAG of the MEPS application. The corresponding execution time of each task related to CC2538 system-on-chip and TMS320C5509A hardwares are shown in Table III. We consider that the network is



(a)



(b)

Fig. 4: Effect of the number of tasks in each artificially generated application on the performance of COTAM, DOTAM, GA [12] and DOOTA [9] algorithms (there are 20 sensor nodes in the network): (a) network lifetime increase w.r.t. non-scheduling strategy; (b) algorithm runtime.

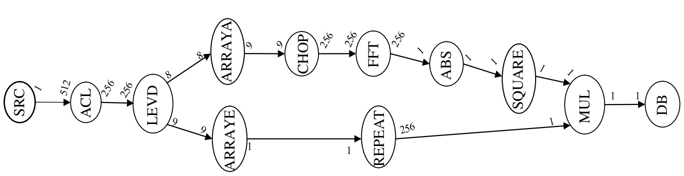


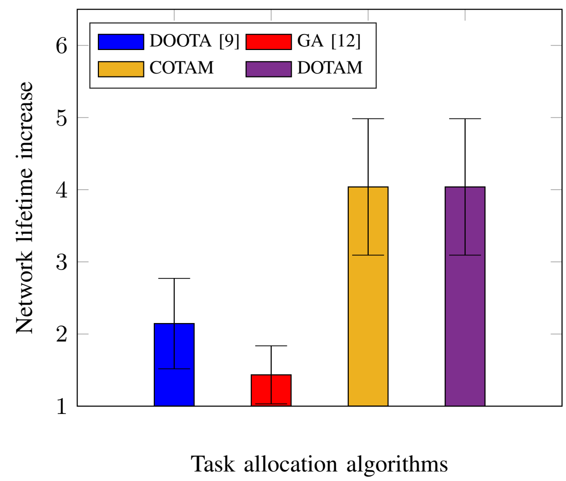
Fig. 5: DAG of MEPS application.

randomly generated with 20 sensor nodes and one sink node as presented in Section V-A.

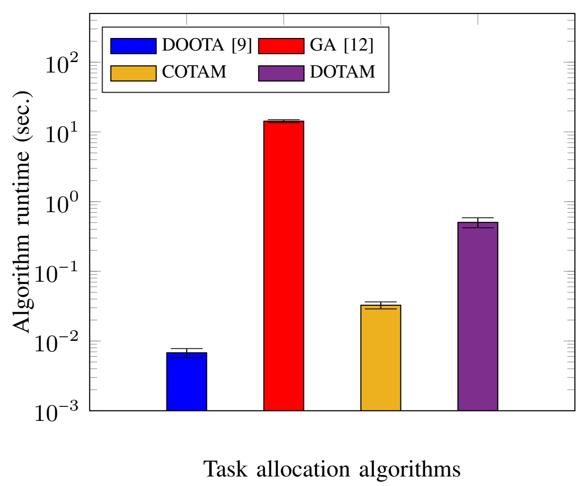
Fig. 6 illustrates the performance of the task allocation algorithms when each sensor node executes MEPS application. As shown in Fig. 6a, both COTAM and DOTAM perform the best as expected. They achieve on average 4.04 times longer network lifetime with respect to the non-scheduling strategy. Comparing with DOOTA algorithm, the gains of COTAM and DOTAM are almost doubled. The reason is that the effect of multihop communication using MEPS is larger than that using artificially generated applications, which makes DOOTA

TABLE III: Execution time of tasks in the DAG of MEPS application.

Tasks	Number of iterations	Ave. exe. time (sec.) on CC2430, 32MHz CLK	Ave. exe. time on TMS320C5509A, 200MHz CLK
SRC	512	7.92E-5	4.64E-6
ACL	1	1.65	1.24E-1
LEVD	1	0.56	5.91E-6
ARRAYA	1	6.83E-5	1.62E-6
ARRAYE	1	1.01E-5	7.0E-8
REPEAT	1	8.77E-4	2.95E-5
CHOP	1	1.24E-3	4.69E-5
FFT	1	0.23	4.81E-3
ABS	256	4.46E-5	2.6E-5
SQUARE	256	1.5E-5	9.05E-7
MUL	256	1.93E-5	2.9E-6
DB	256	2.26E-5	2.66E-5



(a)



(b)

Fig. 6: A case study using realistic MEPS application to estimate the performance of COTAM, DOTAM, GA [12] and DOOTA [9] algorithms (there are $K = 12$ tasks in MEPS application as shown in Fig. 5, and the network is generated with 20 sensor nodes): (a) network lifetime increase w.r.t. non-scheduling strategy; (b) algorithm runtime.

perform worse. Regarding the execution time of the algorithms when applying MEPS application, the results are similar to the artificially generated applications as shown in Fig. 6b. DOOTA still needs the least execution time. The runtime of DOTAM is a little larger than COTAM, and both of them are less than 0.5 seconds. The results of this case study are consistent with Figs. 3 and 4.

VI. Conclusion

This work aims to extend the network lifetime of multi-hop wireless networks by appropriately distributing the tasks for each node. Firstly, a centralized optimal task allocation algorithm, COTAM, is proposed by modeling the problem of maximizing the network lifetime as a linear programming (LP) problem. It provides the optimal task allocation solutions for each node. Moreover, this work further presents a distributed optimal task allocation algorithm, DOTAM. Based on Dantzig-Wolf (D-W) decomposition, DOTAM divides the centralized large-sized LP problem into small-sized subproblems which are executed by each node. The proposed COTAM and DOTAM are tested by applying both the artificially generated applications and a realistic MEPS application. The results demonstrate that both COTAM and DOTAM provide the same optimal task allocation solutions and significantly outperform the existing algorithms on extending the network lifetime.

As COTAM is a centralized algorithm, it needs to know all of the network parameters in advance. Therefore, COTAM is mostly applied for offline optimization in known environments as in partial cases of industrial IoT scenarios, e.g., smart robotics, intelligent machine status monitoring, etc. Moreover, the results provided by COTAM can be treated as a metric to evaluate the other approaches. For the general scenarios in which it is hard or even unrealistic to know all the network parameters in advance, it is better to apply DOTAM for the online optimization to achieve the optimal task allocation solutions.

Appendix

Derivation of $\boldsymbol{G}_i$, $\boldsymbol{A}_i$, $\boldsymbol{b}_0$ and $\boldsymbol{b}_i$ in Section IV-B:

$\boldsymbol{G}_i$ is a matrix with $n+1$ rows. The first row of $\boldsymbol{G}_i$ consists of $NR_{s,i}$ elements and can be represented by:

$$\boldsymbol{G}_i(1,:) = [\overbrace{e_{rx}\boldsymbol{L}_i - \boldsymbol{E}_{ps,i}, \cdots, e_{rx}\boldsymbol{L}_i - \boldsymbol{E}_{ps,i}}^{NR_{s,i}}]$$

where $NR_{s,i}$ is the number of nodes in $\boldsymbol{R}_{s,i}$, i.e., the number of sensor nodes on the path from node i to the sink node. The $i+1$ row of $\boldsymbol{G}_i$ is:

$$\boldsymbol{G}_i(i+1,:) = [\boldsymbol{E}_{pi,i} + e_{tx}\boldsymbol{L}_i, \overbrace{\boldsymbol{0}, \cdots, \boldsymbol{0}}^{NR_{s,i}-1}]$$

The $j+1$ row ($j \in 1, \cdots, n,$ and $j \neq i$) of $\boldsymbol{G}_i$ is:

$$\boldsymbol{G}_i(j+1,:) = \begin{cases} [\overbrace{(e_{rx}+e_{tx})\boldsymbol{L}_i, \cdots, (e_{rx}+e_{tx})\boldsymbol{L}_i}^{NR_{j,i}}, \boldsymbol{E}_{pj,i} + e_{tx}\boldsymbol{L}_i],\ j \in \boldsymbol{R}_{s,i} \\ [\overbrace{\boldsymbol{0}, \cdots, \boldsymbol{0}}^{NR_{s,i}}], \quad \text{otherwise} \end{cases}$$

Taking Fig. 1 for example, the corresponding $\boldsymbol{G}_1$ is a matrix with $n+1$ rows and can be presented as:

$$\boldsymbol{G}_1 = \begin{bmatrix} e_{rx}\boldsymbol{L}_1 - \boldsymbol{E}_{ps,1}, & e_{rx}\boldsymbol{L}_1 - \boldsymbol{E}_{ps,1}, & e_{rx}\boldsymbol{L}_1 - \boldsymbol{E}_{ps,1} \\ \boldsymbol{E}_{p1,1} + e_{tx}\boldsymbol{L}_1, & \boldsymbol{0}, & \boldsymbol{0} \\ (e_{rx}+e_{tx})\boldsymbol{L}_1, & \boldsymbol{E}_{p2,1} + e_{tx}\boldsymbol{L}_1, & \boldsymbol{0} \\ (e_{rx}+e_{tx})\boldsymbol{L}_1, & (e_{rx}+e_{tx})\boldsymbol{L}_1, & \boldsymbol{E}_{p3,1} + e_{tx}\boldsymbol{L}_1 \\ \boldsymbol{0}, & \boldsymbol{0}, & \boldsymbol{0} \\ \vdots & \vdots & \vdots \\ \boldsymbol{0}, & \boldsymbol{0}, & \boldsymbol{0} \end{bmatrix}$$

$\boldsymbol{b}_0$ is a column vector with $n+1$ elements. The first element of $\boldsymbol{b}_0$ is:

$$\boldsymbol{b}_0(1) = -\sum_{i=1}^{n}(e_{ro,s} + \boldsymbol{E}_{ps,i}\boldsymbol{1})$$

The $i+1$ element ($i = 1, \cdots, n$) of $\boldsymbol{b}_0$ is:

$$\boldsymbol{b}_0(i+1) = -e_{to,i} - NC_i(e_{ro,i} + e_{to,i})$$

where NC_i is the number of posterity nodes of node i, i.e., the length of the set $\boldsymbol{C}_i$. Let us still take Fig. 1 for instance, the corresponding $\boldsymbol{b}_0$ is:

$$\boldsymbol{b}_0 = \begin{bmatrix} -\sum_{i=1}^{n}(e_{ro,s} + \boldsymbol{E}_{ps,i}\boldsymbol{1}) \\ -e_{to,1} \\ -e_{to,2} - (e_{ro,2} + e_{to,2}) \\ -e_{to,3} - 2(e_{ro,3} + e_{to,3}) \\ \vdots \end{bmatrix}$$

As $\boldsymbol{A}_i$ and $\boldsymbol{b}_i$ can be easily derived from the constraints of Eq. (22), here we do not present the detailed information.

References

[1] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, “The Internet of things: The next technological revolution,” *Computer*, vol. 46, no. 2, pp. 24–25, Feb. 2013.

[2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013.

[3] M. Z. A. Bhuiyan, G. Wang, J. Wu, and etc., “Dependable structural health monitoring using wireless sensor networks,” *IEEE Trans. on Depe. and Secu. Comp.*, vol. 14, no. 4, pp. 363–376, July 2017.

[4] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. Gupta, “An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city frameworki,” *Future Generation Computer Systems*, vol. 83, pp. 619–628, June 2018.

[5] M. Z. Hasan, H. Al-Rizzo, and F. Al-Turjman, “A survey on multipath routing protocols for QoS assurances in real-time wireless multimedia sensor networks,” *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1424–1456, thirdquarter 2017.

[6] Y. Huang, W. Yu, and A. Garcia-Ortiz, “Accurate energy-aware workload distribution for wireless sensor networks using a detailed communication energy cost model,” *Journal of Low Power Electronics*, vol. 10, no. 2, pp. 183–193, June 2014.

[7] W. Yu, Y. Huang, and A. Garcia-Ortiz, “Modeling optimal dynamic scheduling for energy-aware workload distribution in wireless sensor networks,” in *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, May 2016, pp. 116–118.

[8] X. Yin, K. Zhang, B. Li, A. K. Sangaiah, and J. Wang, “A task allocation strategy for complex applications in heterogeneous cluster-based wireless sensor network,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, pp. 1–15, 2018.

[9] W. Yu, Y. Huang, and A. Garcia-Ortiz, “Distributed optimal on-line task allocation algorithm for wireless sensor networks,” *IEEE Sensors Journal*, vol. 18, no. 1, pp. 446–458, Jan. 2018.

[10] J. Yang, H. Zhang, Y. Ling, and etc., “Task allocation for wireless sensor network using modified binary particle swarm optimization,” *IEEE Sensors Journal*, vol. 14, no. 3, pp. 882–892, March 2014.

[11] W. Guo, J. Li, G. Chen, Y. Niu, and C. Chen, “A PSO-optimized real-time fault-tolerant task allocation algorithm in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3236–3249, Dec. 2015.

[12] Y. Jin, J. Jin, A. Gluhak, K. Moessner, and M. Palaniswami, “An intelligent task allocation scheme for multihop wireless networks,” *IEEE Trans. on Para. and Dist. Syst.*, vol. 23, no. 3, pp. 444–451, March 2012.

[13] A. G. Y. Jin, S. Vural and K. Moessner, “Dynamic task allocation in multi-hop multimedia wireless sensor networks with low mobility,” *Sensors*, vol. 13, no. 10, pp. 13 998–14 028, Oct. 2013.

[14] M. Mitchell, *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998.

[15] W. Yu, Y. Huang, and A. Garcia-Ortiz, “Energy aware task allocation in WSNs,” in *Mission-Oriented Sensor Networks and Systems: Art and Science*, H. M. Ammari, Ed. Springer International Publishing, 2019.

[16] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, “Energy-efficient routing protocols in wireless sensor networks: A survey,” *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 551–591, secondquarter 2013.

[17] S. S. Chiang, C. H. Huang, and K. C. Chang, “A minimum hop routing protocol for home security systems using wireless sensor networks,” *IEEE Transactions on Consumer Electronics*, vol. 53, no. 4, pp. 1483–1489, Nov. 2007.

[18] A. Foerster, *Introduction to wireless sensor networks*, 1st ed. Wiley-IEEE Press, 2016.

[19] F. Al-Turjman, L. Mostarda, E. Ever, A. Darwish, and N. Shekh Khalil, “Network experience scheduling and routing approach for big data transmission in the Internet of things,” *IEEE Access*, vol. 7, pp. 14 501–14 512, 2019.

[20] Texas-instruments, *CC2538 Datasheet*, Chipcon Products from Texas Instruments, USA: Texas Instruments, Copyright, 2013.

[21] T. Pino, S. Choudhury, and F. Al-Turjman, “Dominating set algorithms for wireless sensor networks survivability,” *IEEE Access*, vol. 6, pp. 17 527–17 532, 2018.

[22] W. Yu, Y. Huang, and A. Garcia-Ortiz, “An altruistic compression-scheduling scheme for cluster-based wireless sensor networks,” in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, June 2015, pp. 73–81.

[23] G. B. Dantzig and P. Wolfe, “Decomposition principle for linear programs,” *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

[24] L. David G., *Linear and Nonlinear Programming*, 2nd ed. Addison-Wesley Publishing Company, 1989.

[25] Texas-instruments, *TMS320C5509A Datasheet*, Chipcon Products from Texas Instruments, USA: Texas Instruments, Copyright, 2017.

[26] C.-C. Shen, W. L. Plishker, D.-I. Ko, S. S. Bhattacharyya, and N. Goldsman, “Energy-driven distribution of signal processing applications across wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 6, no. 3, pp. 24:1–24:32, June 2010.

[27] D.-I. Ko, C.-C. Shen, S. Bhattacharyya, and N. Goldsman, “Energy-driven partitioning of signal processing algorithms in sensor networks,” in *Embedded Computer Systems: Architectures, Modeling, and Simulation*, ser. Lecture Notes in Computer Science, S. Vassiliadis, S. Wong, and T. Haemaelaeinen, Eds. Springer Berlin Heidelberg, 2006, vol. 4017, pp. 142–154.