Makespan minimization in tree data gathering networks with memory limits

Joanna Berlińska *

1 Introduction

A data gathering network is a computer system comprising a set of worker nodes and a base station. The workers obtain or produce datasets which have to be transferred to the base station. Data gathering wireless sensor networks find applications in military surveillance, environment monitoring and healthcare. A distributed system, where the computation results obtained by many workers have to be passed to a single server, is also a data gathering network.

In a star data gathering network, the workers communicate directly with the base station. Contrarily, in a tree network, data are relayed through a set of intermediate nodes. Scheduling algorithms were designed for optimizing data gathering in several types of star networks, such as, e g., networks with data compression [1, 5, 6], with variable communication speed [3] and with limited base station memory [2]. Tree data gathering networks have attracted less attention so far. Minimizing the maximum lateness in a tree network with dataset release times and due dates was studied in [4].

This work considers scheduling in a tree data gathering network with limited memory. Each of the worker nodes holds a dataset that has to be sent to an appropriate intermediate node. The intermediate node processes the dataset and then sends it to the base station. A dataset occupies the intermediate node's memory buffer from the moment when it starts being received until the time when its transfer to the base station completes. The total size of datasets coexisting in the memory of an intermediate node can never exceed its buffer size. Our goal is to organize dataset transfers and processing so as to minimize the total time required to gather the data.

2 Problem formulation

The data gathering network consists of n worker nodes, m intermediate nodes and a single base station. An intermediate node P_i, where $1 \leq i \leq m$, gathers data

*Joanna.Berlinska@amu.edu.pl. Faculty of Mathematics and Computer Science, Adam Mickiewicz University, Poznań, Uniwersytetu Poznańskiego 4, 61-614 Poznań, Poland

from n_i workers P_{ij}, where $1 \leq j \leq n_i$. Thus, $n_1 + \cdots + n_m = n$. A worker P_{ij} holds a dataset D_{ij} of size α_{ij}, which has to be sent to the intermediate node P_i for processing. The processed dataset is then passed by P_i to the base station. Sending dataset D_{ij} from P_{ij} to P_i takes time $c_1\alpha_{ij}$, and processing this dataset by P_i requires time $a\alpha_{ij}$. The processed dataset D_{ij} is sent to the base station in time $c_2\alpha_{ij}$. Preemptions in communication or computation are not allowed. Each node, including the base station, can receive at most one dataset at a time. An intermediate node can process at most one dataset at a time, but it can receive one dataset, process another dataset, and send yet another dataset simultaneously. Each intermediate node processes and sends datasets in the order in which it receives them. The size of the memory buffer of node P_i is $B_i \geq \max_{j=1}^{n_i} \alpha_{ij}$. At the moment when a dataset D_{ij} starts being sent to P_i, a memory block of size α_{ij} is allocated at P_i. The block is released when the transfer of the processed dataset D_{ij} to the base station is completed. The total size of memory blocks allocated at P_i cannot at any time exceed B_i. The scheduling problem is to minimize the makespan, i.e., the time by which all processed datasets arrive at the base station.

3 Results

The analyzed problem generalizes makespan minimization in a star data gathering network with limited base station memory [2], and hence, it is strongly NP-hard. We first formulate it as an integer linear program (ILP). The program contains $O(n)$ rational variables, $O(n^3)$ binary variables and $O(n^3)$ constraints, and in consequence, it cannot be used in practice for solving even moderate size instances. Therefore, we also propose heuristic algorithms.

First, we design simple heuristics running in $O(n^2)$ time. Each of these algorithms is defined by two rules. The first one determines the order in which the datasets are transferred to each intermediate node P_i, and the second one defines the order in which the datasets are passed to the base station. We consider the following rules for sending datasets to intermediate nodes.

- Inc: send the datasets in the order of non-decreasing sizes α_{ij}.
- LF: always choose the largest dataset which fits in the memory currently available at the intermediate node.
- Rnd: send the datasets in a random order.

Since the datasets are sent to the base station by an intermediate node P_i in the order in which they were received and processed, a rule designed for sending the datasets to the base station only has to choose the order of the datasets received from different intermediate nodes. We analyze two scheduling methods for this stage.

- FIFO: transfer the datasets in the order in which their processing at the intermediate nodes completed.

- B: choose the dataset from the intermediate node which has the smallest currently available memory.

An algorithm that uses Rule1 for sending the datasets to the intermediate nodes, and Rule2 for transferring them to the base station, is denoted by Rule1-Rule2.

Furthermore, we design a variable neighborhood search algorithm VNS. In this algorithm, a schedule is represented by an array $x[1..n]$ of dataset priorities, where a smaller number means a higher priority. The priority of dataset D_{ij} is $x[\sum_{k=1}^{i-1} n_k + j]$. In order to compute the makespan, the schedule is constructed using the defined priorities both for sending the datasets to the intermediate nodes, and for transferring them to the base station. Variable neighborhood search consists in systematically changing the neighborhoods used during local search. Suppose a sequence of neighborhoods $N_1, \ldots, N_{k_{max}}$ are defined. The algorithm starts with the current neighborhood number $k = 1$ and a given initial solution x. In each step, the best solution x' in neighborhood $N_k(x)$ is found. If x' is better than x, then x is changed to x' and k is set to 1. If x' is not better than x, then k is increased by 1. The search continues until k exceeds k_{max}. In our algorithm, the initial solution is $x = [1, \ldots, n]$, and the following three neighborhoods are used.

- $N_1(x)$ contains all arrays obtained from x by reversing any subarray $x[i..j]$.
- $N_2(x)$ contains all arrays obtained from x by swapping a pair of values $x[i]$ and $x[j]$.
- $N_3(x)$ contains all arrays obtained from x by moving a value $x[i]$ to an arbitrary position $j \neq i$ in the array.

The proposed algorithms were implemented in C++ and tested in a series of computational experiments on randomly generated instances. The obtained results lead us to the following conclusions.

- The computational cost of ILP is very high even for small instances.
- The B rule for scheduling the transfers to the base station is counterproductive. Much better results are obtained by the FIFO rule.
- Among the simple heuristics, the best solutions are usually produced by LF-FIFO. Still, if c_2, δ_B or m is very large, or c_1 or a is very small, then Inc-FIFO obtains better schedules.
- VNS significantly outperforms the simple heuristics. It delivers very good results in a short time.

Acknowledgement. This research was partially supported by the National Science Centre, Poland, grant 2016/23/D/ST6/00410.

References

[1] J. Berlińska (2015). Scheduling for data gathering networks with data compression. *European Journal of Operational Research* 246, 744-749.

[2] J. Berlińska (2020). Heuristics for scheduling data gathering with limited base station memory. *Annals of Operations Research* 285, 149-159.

[3] J. Berlińska (2020). Scheduling in data gathering networks with background communications. *Journal of Scheduling* 23, 681-691.

[4] J. Berlińska (2022). A comparison of priority rules for minimizing the maximum lateness in tree data gathering networks. *Engineering Optimization* 54, 218-231.

[5] W. Luo, B. Gu, and G. Lin (2018). Communication scheduling in data gathering networks of heterogeneous sensors with data compression: Algorithms and empirical experiments. *European Journal of Operational Research* 271, 462-473.

[6] W. Luo, Y. Xu, B. Gu, W. Tong, R. Goebel and G. Lin (2018). Algorithms for Communication Scheduling in Data Gathering Network with Data Compression. *Algorithmica* 80, 3158-3176.