The Second International Workshop on Dynamic Scheduling Problems
Adam Mickiewicz University in Poznań, June 26th – 28th, 2018

Scheduling non-preemptive data gathering affected by background communications

Joanna Berlińska *
Adam Mickiewicz University in Poznań, Poznań, Poland

Keywords: scheduling, data gathering networks, variable communication speed

1 Introduction

Scheduling in a data gathering network is usually analyzed under the assumption that the network performance is constant. For example, Choi and Robertazzi [4], Moges and Robertazzi [7] proposed algorithms for partitioning the total amount of gathered data between the network nodes, in order to minimize the makespan. Scheduling algorithms for gathering fixed amounts of data from the network nodes were proposed, e.g., by Berlińska [1, 2], Luo et al. [6]. However, real communication parameters of a network may change in time. Preemptive scheduling in data gathering networks with variable communication speed was studied by Berlińska [3]. This work considers non-preemptive scheduling in a data gathering network with performance affected by background communications.

2 Problem formulation

We study a star data gathering network that consists of m worker nodes $P_1, \ldots, P_m$ and a single base station P_0. Each worker P_i holds dataset D_i of size α_i, which has to be transferred to the base station in a single message. At most one node can communicate with the base station at a time. The communication rate, i.e. the inverse of speed, of the link between P_i and P_0 in an otherwise unloaded network is C_i. However, background communications required by other applications may degrade the link performance. We will be calling a link *loaded* if it is used by background communications, and *free* in the opposite case. We assume that the network implements QoS Percentage-Based Policing (Szigeti et al. [8]), hence the communication rate perceived by the analyzed data gathering application for a loaded link between P_i and P_0 is δC_i, for some fixed $\delta > 1$. Thus, the

* Speaker, email: `Joanna.Berlinska@amu.edu.pl`

maximum time that may be necessary to gather data from all worker nodes is $\overline{T} = \delta \sum_{i=1}^{m} C_i \alpha_i$. For each node P_i, we are given a set of n_i disjoint time intervals $[t'_{ij}, t''_{ij}]$, where $j = 1, \ldots, n_i$ and $t'_{ij} < \overline{T}$, in which the corresponding communication link is loaded. The total number of such intervals is $n_1 + \cdots + n_m = n$.

The scheduling problem is to choose the sequence of datasets sent consecutively to the base station, such that all data is transferred in the shortest possible time.

3 Complexity and algorithms

Let us first observe that transferring data of size x with communication rate C is equivalent to sending data of size Cx with communication rate 1. Thus, from now on we will assume without loss of generality that $C_i = 1$ for $i = 1, \ldots, m$.

We prove that the analyzed problem is strongly $\mathcal{NP}$-hard, using a pseudo-polynomial transformation from the 3-PARTITION problem (Garey and Johnson [5]). Then, we propose the following exponential-time dynamic programming algorithm. Let $\tau(D_i, t)$ be the time necessary to transfer dataset D_i, starting at moment t. For each subset $\mathcal{D} \subset \{D_1, \ldots, D_m\}$, we compute the shortest time $T(\mathcal{D})$ in which the datasets from $\mathcal{D}$ can be transferred to the base station, using the following formulas:

$$T(\mathcal{D}) = \begin{cases} 0 & \text{if } \mathcal{D} = \emptyset, \\ \min_{D_i \in \mathcal{D}} \{T(\mathcal{D} \setminus \{D_i\}) + \tau(D_i, T(\mathcal{D} \setminus \{D_i\}))\} & \text{if } \mathcal{D} \neq \emptyset. \end{cases}$$

The minimum schedule makespan is $T(\{D_1, D_2, \ldots, D_m\})$, and the optimum dataset sequence can be easily tracked. This algorithm runs in $O((m+n)2^m)$ time.

Furthermore, we propose the following three greedy heuristic algorithms, each of which has $O(m(m+n))$ complexity.

1. Algorithm *gTime* always chooses to send the dataset that will be transferred in the shortest time.

2. Algorithm *gRate* selects the dataset that will be sent with the best average communication rate.

3. Algorithm *gSlowtime* chooses the dataset for which the time when data is transferred over a loaded link will be the shortest.

In all the three heuristics ties are broken by selecting a larger dataset. We also implement algorithm *Rnd*, which constructs a random dataset sequence.

4 Experimental results

The quality of solutions delivered by our greedy heuristics and algorithm *Rnd* was tested in computational experiments. We generated two groups of tests. In the *periodic* instances, for a link between P_i and P_0 we selected randomly the common length of its free intervals $f_i \in [1, F]$, and the common length of its loaded intervals $l_i \in [1, L]$. The link was loaded periodically, in intervals $[t'_{ij}, t''_{ij}] = [jf_i + (j-1)l_i, j(f_i + l_i)]$, for $j = 1, 2, \dots n_i$. In the *random* tests, the lengths of all free intervals $f_{ij} \in [1, F]$, and the lengths of all loaded intervals $l_{ij} \in [1, L]$ for a link between P_i and P_0 were selected independently. The analyzed values of maximum lengths of free and loaded intervals were $F = 10, 30$, and $L = 5, 10, \dots, 50$. We used $\delta = 2$, $m = 20$, and dataset sizes α_i were chosen randomly from the interval $[1, 20]$. For each tested setting, 100 instances were generated. Solution quality was measured by the ratio of the makespan delivered by a given heuristic to the optimum computed by the exact algorithm.

Table 1: Average solution quality for random instances

L	$F = 10$ *gTime*	$F = 10$ *gRate*	$F = 10$ *gSlowtime*	$F = 30$ *gTime*	$F = 30$ *gRate*	$F = 30$ *gSlowtime*
10	1.124	1.056	1.073	1.064	1.019	1.013
20	1.153	1.074	1.098	1.100	1.043	1.034
30	1.168	1.079	1.109	1.137	1.056	1.052

A subset of the obtained results can be found in Table 1. The complete results can be summarized as follows.

1. As expected, the quality of solutions delivered by all heuristics deteriorates with increasing L, and tests with big F are easier than those with small F.
2. It is easier to find good schedules for periodic instances than for the random ones, although the problem remains strongly $\mathcal{NP}$-hard in the periodic case.
3. The greedy heuristics obtain much better results than algorithm *Rnd*.
4. Algorithm *gTime* is significantly outperformed by *gRate* and *gSlowtime*.
5. For $F = 10$, algorithm *gRate* obtains better results than *gSlowtime*.
6. For $F = 30$, the results delivered by algorithms *gRate* and *gSlowtime* are very similar for periodic instances, and algorithm *gSlowtime* slightly outperforms algorithm *gRate* on random tests.

5 Future research

In the future, we want to analyze in more detail the subproblem with periodic background communications, and design for it a dedicated polynomial-time heuristic.

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

References

[1] J. Berlińska, Communication scheduling in data gathering networks with limited memory, *Applied Mathematics and Computation*, **235** (2014), 530–537.

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

[3] J. Berlińska, Scheduling data gathering with variable communication speed, *Proceedings of the 1st International Workshop on Dynamic Scheduling Problems*, Poznań, June 30–July 1, 2016, 29–32.

[4] K. Choi, T. G. Robertazzi, Divisible load scheduling in wireless sensor networks with information utility, *Proceedings of the IEEE International Performance Computing and Communications Conference 2008*, Austin, December 7–9, 2008, 9–17.

[5] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W.H. Freeman, San Francisco, 1979.

[6] W. Luo, Y. Xu, B. Gu, W. Tong, R. Goebel, G. Lin, Algorithms for communication scheduling in data gathering network with data compression, *Algorithmica*, 2017, doi: 10.1007/s00453-017-0373-6, in press.

[7] M. Moges, T. G. Robertazzi, Wireless sensor networks: scheduling for measurement and data reporting, *IEEE Transactions on Aerospace and Electronic Systems*, **42** (2006), 327–340.

[8] T. Szigeti, C. Hattingh, R. Barton, K. Briley, *End-to-End QoS Network Design: Quality of Service for Rich-Media & Cloud Networks*, 2nd ed., Cisco Press, Indianapolis, 2013.