Jam Web Space Telescope Scheduling

Algorithms for Decision Support 2022/2023

September 14, 2022

Contents

1 Introduction

This is the description of the group project for the course *Algorithms for Decision Support* 2022/2023.

1.1 Important dates

- September 22 (Thursday): Groups should be formed.
- October 6 (Thursday): Hand in a test instance for the offline problem
- October 27 (Thursday): Hand in scientific paper and program
- November 1 (Tuesday) or November 3 (Thursday): Group presentation

2 Telescope scheduling problem

Human beings are curious about the universe. Recently, a small brilliant country launched a cutting-edge space telescope named "Jam Web". The telescope stays 2 million kilometers from Earth, takes pictures of the deep universe that has never been explored by human beings, and sends the images back to Earth. However, because the distance between the telescope and Earth and the mysterious space, there are lots of uncertain situations such that the channel is blocked and cannot transmit the data during some time intervals. The challenge of the data transmission is that the transmission is blocked from time to time due to the uncertainty of space environment. In this assignment, the scheduling for transmitting the images in an uncertain space environment is to be made.

2.1 Formal problem definition

There are n images $\{f_1, f_2, f_3, \cdots, f_n\}$, where image f_i is of real-number size s_i. There is a channel for transmitting the images from the telescope to Earth. At any time, this channel can transmit at most one image. An image with size s_i needs s_i units of time to transmit. Moreover, the image transmission is non-preemptive. That is, to successfully transmit an image f_i, the channel has to be assigned to transmit the image for a contiguous time interval of length s_i. There are m unavailable time intervals $I_1, I_2, \cdot, I_m$, where the connection of the channel is unavailable, and the images cannot be transmitted during these intervals. More formally, time interval $I_j = [t_j, t_j + \ell_j)$, where t_j and ℓ_j are the start time and the length of the j-th unavailable interval, respectively. The goal of the algorithms is to transmit all the n images to Earth as soon as possible.

In the *offline setting*, all the parameters are known to the algorithm from the very beginning. Contrastingly, in the *online setting*, the algorithm knows all the n images and their sizes. However, the unavailable time intervals information, both the start time and the length, are unknown. The online algorithm learns about the start time of an unavailable time interval only when it starts, and learns about that the channel is available again when the unavailable time interval finishes. If the online algorithm is transmitting an image at the time when an unavailable time interval starts, the image transmission is failed and needs to start over again. Note that the online algorithm does not need to resend this image when the channel is available again but can schedule some other images and reschedule this failed image later.

2.2 Input and output formats for the experiment

The general problem parameters are:

- the n images and their sizes, and
- the start times and lengths of the m unavailable time intervals.

The problem is: when is the time when all images are transmitted to Earth. You should also provide a feasible schedule for the images transmission such that all the images can be transmitted by the claimed time.

For the offline problem, the algorithm knows the information about the images and the information about the unavailable time intervals. The output should provide the time of transmission complete time and the time of starting transmission of each image.

Example of input. For the case where there are 5 images (of sizes 0.618, 2.5, 10, 1, and 3.14) and 2 unavailable time intervals ($[4, 4.25)$ and $[5, 16)$), the input will be:

```
5
0.618
2.5
10
6
3.14
2
4, 0.25
5, 11
```

Here the first line is the number (5) of images. The following 5 lines consist 5 real numbers, which are the sizes of the images. After that, the 7-th line is the number (2) of unavailable time intervals. Each of the last two lines consists two real numbers, where the first one is the start time of the first/second unavailable time interval, and the second one is the length of the first/second unavailable time interval. (See Figure 1 for the illustration of this instance.)

For this instance, the output can be:

```
41.618
35
29.6
```

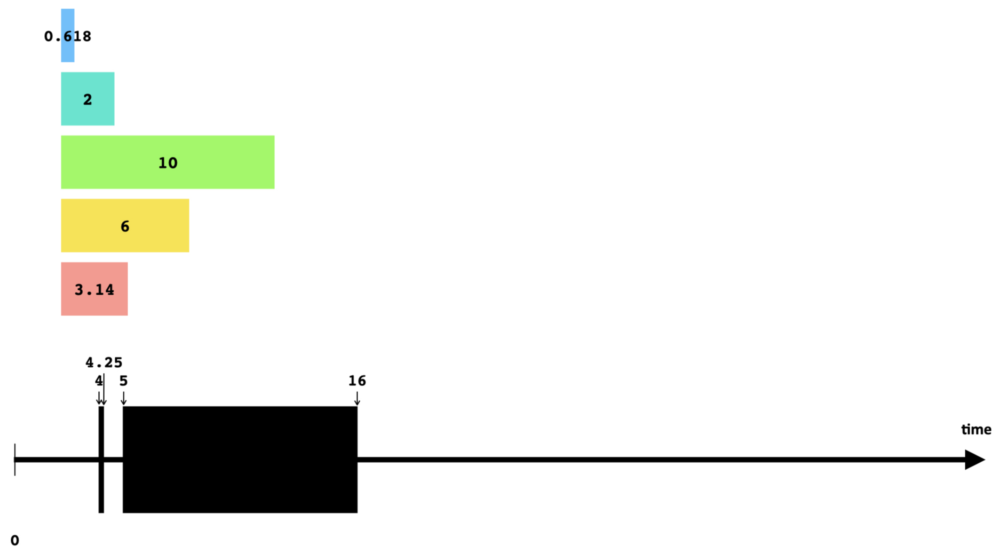

Figure 1: An instance consists of 5 images (the colored rectangles) and 2 unavailable time intervals (the tall black rectangles).

```
18
35.618
0.86
```

In the output, the first lines is the time when all images are transmitted. The next n lines are the times when the 1-st to the n-th images start transmitting, respectively. (See Figure 2 for the illustration corresponding to this schedule.)

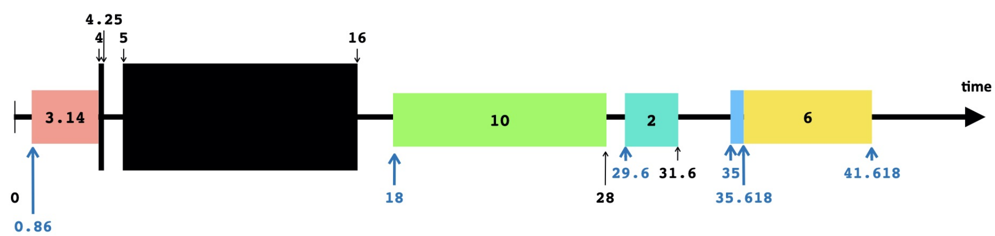

Figure 2: A schedule for the instance with the 5 images and the 2 unavailable time intervals. Note that the blue numbers at the bottom are the outputs.

3 Requirements

3.1 Groups

Each group consists of four, five, or six participants of the course. We prefer that you form groups with five students. From groups with six participants, a better project is expected for the same grade. Groups with four participants will not be given a benefit.

You must inform the teachers of the group formation via Blackboard (see the respective assignment) ultimately on September 22.

3.2 Test instances

By October 6, you should hand in at least one test instance for the offline problem, together with the correct optimal answer.

The instance(s) should be correct, in the sense that they fulfill the description given above. You are allowed to hand in instances where your own programs work well.

By October 6, the instance(s) and the corresponding answers should be handed in via Blackboard in one normal ASCII-text file. You are allowed to hand in more test instances.

3.3 The scientific paper

Each group writes a scientific paper on the project. The paper must have a specific format. This format is provided via the MS Teams site and Blackboard of the course.

The format includes the following:

- The paper is written in LaTeX, following the Lipics style file as provided. The paper has author names, an abstract, keywords, main text (see below), proper formatted reference list, and optionally an appendix. (More specifically, the lipics-v2019 style is used. You can download the necessary files from the Teams site, or from `https://submission.dagstuhl.de/documentation/authors#lipics`
- The main text must have the following sections:
  - *Introduction.* In the first section, you describe the problem, possibly describe related scientific literature, and describe in brief sentences (or as formatted theorems) your results.
  - *Preliminaries / Definitions.* In the second section, you give definitions of mathematical notions that you use or need. If you use well known scientific results / lemmas / theorems / algorithms as subroutine, you can also mention these here.

- – One or more chapters that: describe the *algorithm(s)* you use, prove a competitive ratio for at least one case of an online strategy, and optionally give more theoretical results.
  - – One or more chapters that: give the results from the *experiments*. Describe the setting (what computer(s) did you use to test, what programming language did you use, what OS, etc.), and give the results of the experiments. Try to draw some conclusions from your experiments. These should report on all or some of the provided test instances. You can use more test instances that you made or found yourself, if you want.
  - – *Conclusions*. In a (possibly short) last chapter, you repeat the main findings of the project, and possibly give ideas for further research, open problems, etc.

- In the (optional) appendix, you could give additional information, e.g., additional long tables with results from experiments, etc.
- The reference list must be proper formatted, following standard style of scientific writing.
- You should follow rules of scientific integrity. This includes that you cannot use images that are copyrighted by others, and give a reference for non-copyrighted images you use. It is always better to make images yourself. Of course, you should never commit fraud (e.g., changing data) or plagiarism; we follow the rules of Utrecht University here.

You hand in the paper by October 27 via blackboard.

3.4 Experiments

By October 27, the program should be made visible to the teachers. You can do this by uploading in the specific Blackboard assignment the URL of a public visible code.

The program should be well written, and commented.

3.5 Presentations

On November 1 or November 3, each group gives a presentation of at least 12 and at most 15 minutes of the project. These will be given on the campus. There are four parallel sessions.

The presentation can be given by one, some or all group members.

Presence at the presentation of your groups, and the other groups in your session is obligatory, except when you have a good reason not to come. Fear of corona-infection is regarded as a good reason. If you cannot come to the presentation session, you should inform the teachers BEFORE the presentation session, when possible, and otherwise you must inform the teachers after the session, with the reason of absence. We can have a deduction from a grade if the reason for absence is not considered to be valid by the teachers, or came too late. Failing to contact the teachers in case of absence is regarded as an uncompleted course.

4 Grading

The project has a number of different deliverables. The grade for a project depends on the following:

- The project has a number of minimum requirements. If these all are done "reasonably well", then the grade will be 6.
- You can get a better grade by doing something additional, or performing in the minimum requirement tasks in a (very) good / excellent manner. More information is given below in subsection 4.2.
- In case the distribution of work among group members was very uneven, you can contact the teachers, and we can have a different grade for different members of the same project.

4.1 Minimum requirements

Each group must fulfill the following deliverables:

1. At least ONE *test instance* for the offline version.
2. A *project presentation*. On November 1 or November 3, each group gives a presentation of $12 - 15$ minutes of the project. The exact date will be decided in the week October 18.
3. A *scientific paper*. The scientific paper contains specific theoretical elements (described below), and handed in as a pdf-file.
    - In the paper, you give at least ONE algorithm, which can be an offline or an online algorithm.
    - You have to give at least TWO proofs. A valid proof can be a time-complexity analysis, a correctness proof, a competitive analysis of an online algorithm, a lower bound of the competitive ratio of an online algorithm, a lower bound of the competitive ratio of the online problem, an NP-hardness proof, etc..
    - The paper is at least six pages long, but can be longer. There is no real upper limit to the size (but try to keep it concise and below 50 pages.)
4. ONE *program* for the offline version, which should give exact solutions for (relatively) small instances of the problem. Note that your program should be able to handle any feasible instance.

It is possible to *extend* the project, for a better grade (see subsection 4.2).

4.2 Price list of theoretical results variations

There are various directions of theoretical results. The following is a list of examples of the directions you can try. The number of stars indicates the difficulty of the variant. You get higher grade if you correctly provide theoretical results.

- Prove that the problem is NP-complete for general instances ★★
- Offline optimal algorithm and proof of correctness ★☆
- Prove that the problem is polynomial-time solvable for some special instances. ☆ ~ ★★
- Give an online algorithm and show that it is at most $\Omega(t_m + \ell_m)$-competitive ☆
- Give an online algorithm and show that it is at least c-competitive for some constant $c > 1$ ★☆
- A 2-competitive online algorithm for $m = 1$ case ★★
- No online algorithm can be $O(1)$-competitive when $m \geq 2$ ★ ★ ★

You can have some combo attacks to get a higher grade:

- Show that there exists an online algorithm that is exactly c-competitive for some $c > 1$ (for some special cases) ★ ★ ★ ★ ★
- Show that some online algorithm is optimal (for some special cases) ★★★★★

Note that although your program should deal with general input instances, you can tackle special cases for the problem. In the special instances, you can impose restrictions on the sizes of the images, the number of unavailable intervals, or the start times/lengths of the unavailable intervals. The special instances include but not limit to:

- Unit-size images: $s_j = 1$ for all j
- Bounded-size images: $s_j \in [s_{\min}, s_{\max}]$ for some constants $s_{\min}$ and $s_{\max}$
- Only one unavailable interval: $m = 1$
- Periodic unavailable intervals: for all i, the i-th unavailable interval has $t_i = i \cdot p$ for some p
- Periodic and regular unavailable intervals: for all i, the i-th unavailable interval has $t_i = i \cdot p$ for some p and $\ell_i = \ell$
- Bounded-length unavailable intervals: for all i, the i-th unavailable interval has $\ell_i \in [\ell_{\min}, \ell_{\max}]$ for some constants $\ell_{\min}$ and $\ell_{\max}$

4.3 Rules of conduct

- Do not commit fraud or plagiarism.
- Give all your sources. It is allowed to use free software from the internet, but you should mention this in the report.
- Do the project in your group. Do not use the help of others, except the staff of this course. If in doubt, consult the teachers. Do not cooperate with other groups.
- Have a fair division of work within the group. If at the end of the project, you find that the division of work was not fair, you should report this to the teachers.
- Do not start this course and this project if there is a large chance you cannot complete it. Not completing the course makes the life for the other students much harder, and is unkind, impolite and unfair towards them. Do a fair share of the work of the team, start in time, and keep your promises to the other team members.
- Start in time; as a group, do a fair division of work, and make a good planning.
- If you have questions, you can contact the teachers/staff of this course via the MS Teams group of this course. Handing in materials is via Blackboard; other communication to the teachers via MS Teams team of the project.