New-Normal Cinema Seating Planning

Algorithms for Decision Support 2020

August 21, 2020

1 This assignment

This is the description of the group project for the course *Algorithms for Decision Support* 2020/2021.

2 Dates

- September 8, 2020: Groups should be formed.
- September 18, 2020: Hand in a test instance for the offline problem and a test instance for the online problem
- October 9, 2020: Group presentation
- October 16, 2020: Hand in scientific paper and software

3 Groups

Each group consists of four or five participants of the course. We prefer that you form groups with five students. You must inform the teachers of the group formation via Blackboard (see the respective assignement) ultimately on September 8, 2020.

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 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.

5 Project description

5.1 Minimum requirements

Each group must fulfill the following deliverables:

1. Two *test instances*.
2. A *project presentation*. At October 9, 2020, each group gives a presentation of 12 – 15 minutes of the project.
3. A *scientific paper*. The scientific paper is written with a specific format (described below), and handed in as a pdf-file.
   - The paper must follow a format, described below.
   - In the paper, you give at least one proof: for at least one strategy for the online problem, you give a proof of a competitive ratio.
   - The paper is at least eight pages long, but can be longer. There is no real upper limit to the size (but try to keep it below 50 pages.)
4. Two *programs*. One program gives exact solutions for (relatively) small instances of the New-Normal Cinema Seating problem, and one gives a strategy for the online version.

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

5.2 The New-Normal Cinema Seating problem

Under the current restrictions, imposed by the Dutch government to cinema's because of the Covid-19 situations, cinema's are not allowed to have visitors of different households seated within a distance of 1.5 meter from each other. People from the same household however may set within such a distance, and would want to sit next to each other.

Cinema's would prefer to have as many as possible visitors seated in a cinema room. Question is: how can we place the visitors such that the 1.5 meter condition is fulfilled, groups are seated next to each other (in the same row), and as many as possible visitors are placed. We look at two variations: exact algorithms for the offline problem and a strategy for the online problem.

5.2.1 The offline problem

Given are: a cinema layout, and a multiset of group sizes. The problem is: what is the maximum number of visitors we can place in the cinema.

Cinema layouts In this project, a cinema layout is a subset of a grid. (This is a simplification of reality.)

The layout is given in the following format:

- One one line: A positive integer that is at most 1000 that gives the number of rows. Call this number n.
- One one line: A positive integer that is at most 1000 that gives an upper bound on the maximum number of locations of a row. Call this number m.
- n lines with m characters each; each character is a 1 or a 0. A 1 denotes a chair; a 0 denotes a position without a chair.

Here is a very small cinema, which has a horizontal and a vertical corridor, and one chair missing is the left front corner.

```
7
9
011101111
111101111
111101111
000000000
111101111
111101111
```

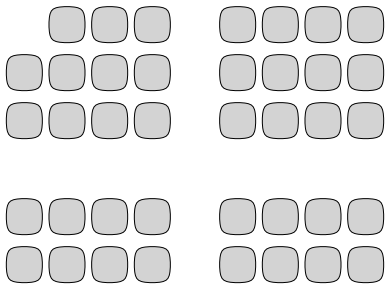

Figure 1: Chairs in a cinema room from the example

One and a half meter In this project, we make the following assumptions:

- Chairs next to each other on the same row, or same column are closer than 1.5 meter.
- Chairs diagonally next to each other are closer than 1.5 meter.
- Chairs on the same row with only one chair in between are closer than 1.5 meter.

- All other chairs are further away than 1.5 meter.

So, look at the following diagram. The `@` is close to all positions with a `+` but to no position with a `-`.

```
--------
--------
---+++--
--++@++-
---+++--
--------
--------
```

Note that we assume here that the distance between rows is larger than the distance between positions on the same row.

Here is an example of a cinema with a row in the middle. A single person is sitting on the (green) chair with the black circle; the (red) chairs with crosses may not be occupied.

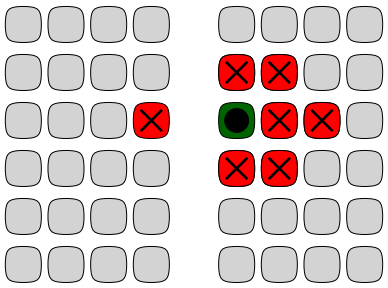

Figure 2: The 1.5 meter rule illustrated for our setting

Groups A group of cinema visitors has at least one and at most eight persons. Groups must be seated next to each other, on the same row. That is: the groups is seated on chairs (so, one positions with a 1 in the diagram), that are directly next to each other. (Having a group split by positions with a 0 is not allowed. In the cinema given above, it is impossible to have seats for a group with five people.)

In the exact problem, we have in the input, after the cinema layout, eight integers. The first gives the number of groups of size 1, the second the number of groups of size 2, etc. All these numbers are on the same line

E.g., if there are two singles, four couples, one family with four members, and one family with five members that want to go to the movie, then (this part of) the input is:

```
2 4 0 1 1 0 0 0
```

The output The output gives a placement of the maximum number of people in the cinema. The output is given by a diagram that gives a `0` for a position without a square, a `1` for an empty chair, and a `x` for a position with a person.

If there are multiple solutions, only one should be given.

Example If the input is:

```
7
9
011101111
111101111
111101111
000000000
111101111
111101111
4 4 0 1 1 0 0 0
```

then a possible solution could be

```
01xx01xx1
x11101111
11x10xxxx
000000000
xx110x111
111x011xx
```

5.3 The online problem

In the online problem, we get as input:

1. A cinema layout, as above.
2. Now, one by one, an integer that is 1, 2, 3, 4, 5, 6, 7 or 8.
3. At some point, the program receives a 0, and now we stop and show the movie.

As soon as the program receives an integer as in the third point above, then it must, if possible place a group of that size. Only, when this group does no longer fit, then it can be rejected.

The teachers will test your program with respect to the public test instances, and with a randomized testing procedure, where we have groups of size 1, 2, 3, 4, 5, 6, 7, 8 with probabilities 0.2, 0.2, 0.2, 0.1., 0.1, 0.1, 0.05 and 0.05. (I.e., with 20 percent chance there is a group of size 1, 20 percent a group of size 2, 20 percent a group of size 3, 10 percent a group of size 4, 10 percent a group of size 5, 10 percent a group of size 6, 5 percent a group of size 7, and 5 percent a group of size 8.

When a group arrives, we output:

- The coordinates of the seat left-most member of the group. The first row has number 1 (which is also the first row which was given in the cinema layout); the first position in the row has number 1.
- 0 0, when the group cannot be seated.

When a 0 is received, we output the number of people seated in the cinema room. (Note that a group with four persons, counts for four people, so the example above has a value of 16.)

6 Test instances

At September 18, 2020, you should hand in two test instances:

- An instance for the offline problem
- An instance for the online problem (a room, plus a sequence of groups that arrive)

Both 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.

Hand in two files, in normal asci text, via Blackboard, at or before October 16, 2020.

The programs will be evaluated with respect to the test instances of all groups (which will be made public) and some test instances made by the teachers.

7 The scientific paper

Each group writes a scientific paper on the project. The paper must have a specific format. This format is provided via Blackboard.

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 Blackboard 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 algorithms 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 at or before October 16, 2020via blackboard.

8 A competitive ratio

Your scientific paper should at least contain a description of one online strategy with an analysis of the competetive ratio. This analysis should in any case be valid when the input is a cinema with a chair on each location (i.e., the input contains n rows with m chairs). You still can assume that groups have size at most 8.

9 Programs

At or before October 16, 2020, the programs 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 programs should be well written, and commented.

The online program is *not allowed to cheat*: it should output the assignment of a group directly after the group is given, and not (try to) read sizes of future groups. If we find that a project has a cheating online strategy, then we divide the project score by 2.

10 Presentations

At October 9, 2020, 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.

11 Extensions

For a better grade, you can add additional features to the project. These can include:

- Additional theoretical results. Can you prove (maybe with additional conditions), the exact problem NP-complete? Can you give polynomial time approximation algorithms with a proven approximation ratio? Can you give polynomial time algorithms for special cases? Etc.
- Closer to real-life settings. Take an existing cinema, and look what the 1.5 meter actually means for the distances between chairs in this cinema. What if chairs are not in a grid, but can be diagonally next to each other?
- Tables of exact solutions if groups have size at most 2, for a specific layout. I.e., we assume we only have singles and couples going to this movie: list all the 'best' pairs of nb of singles and doubles that fit. (Describe the approach you used.)
- Very good algorithms that give better results than that of other groups on our test instances. (Better can be: closer to optimal, faster algorithms, program can handle larger inputs, etc.)
- But maybe you have other ideas for extensions that are interesting? When in doubt, ask the teachers via MS Teams.

12 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.