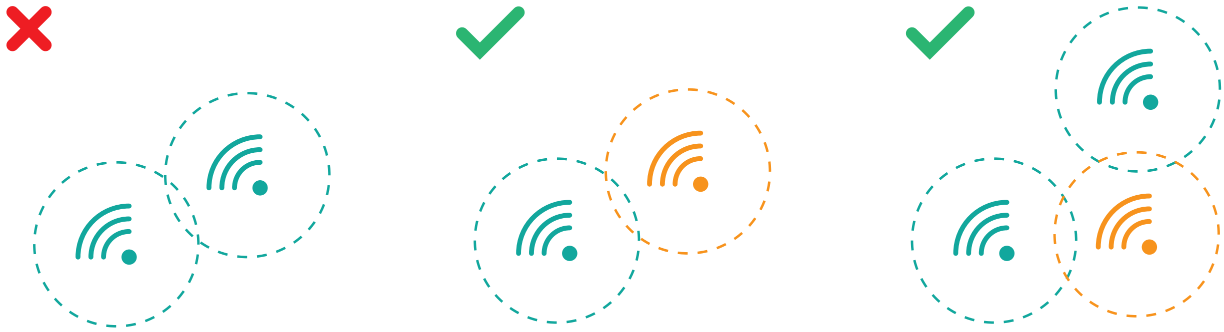
Quantum computing with neutral atoms: assignement

1 Introduction

A telecommunication company contacts you with the following situation. They have a certain number of antennas distributed over a geographical area, and each antenna uses a certain radio frequency to communicate. If two antennas are geographically close to each other, then they will interfere with each other if they use the same frequency (see Fig. 1). On the other hand, it's not possible to just assign a different frequency to each antenna, because the number of available frequencies is limited. Therefore, the client wants to assign a minimum number of different frequencies to their antennas, while minimizing interferences at the same time.



(a) (b) (c)

Figure 1: Schematic example of interference between antennas. (a) Bad frequency assignment. Two antennas are assigned the same frequency (represented by the color green) and they lie within the interference radius of each other (represented as a dashed circle around them). (b) The same antennas are assigned different frequencies (green and orange), so they will not interfere with each other even if they lie within the respective interference radius. (c) Two antennas can be assigned the same frequency (green) if they don't lie within each other's interference radius.

2 Materials

The client has 8 antennas. The physical (x, y) coordinates of the antennas are as follows:

```
antenna 1 (0, 0)
antenna 2 (3, 5.2)
antenna 3 (6, 0)
antenna 4 (9, -5.2)
antenna 5 (9, 0)
antenna 6 (9, 5.2)
antenna 7 (9, 10.4)
antenna 8 (12, 0)
```

The numbers are expressed in kilometers. Two antennas will interfere with each other if they are less than 8.7 km apart.

3 Assignment

Step 1
Model the problem of *antenna frequency assignment* as a *graph coloring* problem on unit-disk graphs (see 16/10 lecture)

Step 2
Design an algorithm to solve *graph coloring* by using repeated applications of *maximum independent set* (MIS) (see 16/10 lecture)

Step 3
Implement a MIS solver in Pulser (see 23/10 lecture)

Step 4
Implement your *graph coloring* algorithm in Pulser using the MIS solver of step 3

Step 5 (optional)
Improve the implementation by including realistic error sources and hardware requirements in the algorithm (see 30/10 lecture)

Step 6
Write a short report detailing the work

4 Practicalities

The report can be written with the document editor of your choice, although Latex or Typst are preferred over other options like Word. The code should be properly commented, and it has to be provided as a repository on GitHub or GitLab, with a README explaining how it works. Clarity of exposition and readability of the code will be taken into account in the assessment. Optimality of the solution found by the algorithm is not necessarily the most important aspect. For example an algorithm that gives the correct solution for the given graph instance, but that is not based on correct and solid ideas, will be assessed negatively.