

Rapport de stage

Nathan Boyer

July 9, 2024

Abstract

It's possible to distribute the Internet to users via drones. However, this raises the question of how to place the drones around the users, and how to distribute the bandwidth between the different users. A reinforcement AI has already been designed to address this problem. However, in this article, we will see how learning and optimization can be combined to further improve performance.

1 Probleme presentation

We have m users, classified into 3 categories, each class having its own bandwidth demand for a drone. We then want to place n drones, and for each drone, decide how much of its bandwidth it gives to each class of drone, so that as many drones as possible are satisfied, i.e. the bandwidth available to them is greater than or equal to their demand.

2 Problem solving by constrained optimization

2.1 Rigorous problem definition

This problem can be solved by constrained optimization.

First, we have to define the equation that computes how well a user will receive a base station's connection depending on where it is.

We have the following Equations for the drones's SINRs, which is a factor that represents how much data is effectively received by the user, in relation to the data sent by the base station to the user:

$$\text{SINR}_{i,j}^t = \frac{pc\mu(y_j, u_i^t) \left((\|y_j - u_i^t\|)^2 + (h)^2 \right)^{-\alpha/2}}{\sum_{k \in \mathcal{U} \setminus i} pc\mu(y_j, u_k^t) \left((\|y_j - u_k^t\|)^2 + (h)^2 \right)^{-\alpha/2} + \sigma^2}$$

To evaluate a configuration we will thus proceed as follows:

We first need to associate each user with its nearest base station.

We then share the bandwidth of a base station dedicated to a given class between all users of this class associated to this base station.

Then, we compute each user SINR and decide for the user is satisfied or not.

The percentage of satisfied user is thus what we want to maximize.

2.2 Equations

2.3 Optimisations

3 Apprentissage pour résoudre le problème

3.1 Apprentissage par réenforcement

3.2 Apprentissage par rapport à la solution optimum

3.3 Mixer les 2

3.4 Les graph neural network

4 Ouverture: Utiliser la distance à l'optimum comme erreur pour l'apprentissage

5 Conclusion