Multi-agent Project

# 1. Introduction

This project is a constrained optimization problem. The aim is to solve a problem posed by the Centre Electronique de l'Armement (CELAR): the distribution of radio frequencies, also known as Radio Link Frequency Assignment benchmark problems (RLFAP). The real case involves a resolution of 5000 binary constraints between 1000 variables. In order to scale the training, there are simplified cases with 200, 400 or 800 variables and more or less constraints.

The variables are frequency names, each of them is assigned to an allowed frequency range.

There are several domains (7 in the real case), and each domain has a unique list of frequencies that can be taken by the variables associated with that domain.

The constraints are binary, i.e. they only concern 2 variables at the same time. There are 2 types: equal constraints (|F1 and F2|=K12, the 2 frequencies must be separated by exactly K12 Hz) and unequal constraints (|F1 - F2| > K12, the 2 frequencies must be separated by at least K12 Hz). The origins of the stresses are multiple (frequency overlapping, accidental terrain...). We are not interested in this physical approach of the problem that generated the constraints, but more in the algorithmic solutions that can be found to optimize a solution).

Among all the scenarios proposed in training, only the first 3 can opt for a solution that checks all the constraints. The others (including the real case) cannot satisfy all the constraints at once. We thus move from a DCP (Distributed Constraints Processing) problem to a DCOP (Distributed Constraints Optimization Processing) problem, the goal being to reduce the penalties obtained when constraints are not respected.

The optimization is done on 2 criteria, the main goal is to facilitate the implementation of new radio links later:

* Minimization of the maximum frequency used: this way we keep space for higher future frequencies.
* Minimization of the number of frequencies used: so future radio links can be inserted between 2 used frequencies that has been spread as much as possible considering the 1st criteria.

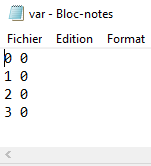
This problem will be solved with multi-agent algorithms (MGM, DSA…) from FRODO. Our goal is to provide a solution and especially to compare the behaviour of different algorithms in terms of efficiency for different scenarios of change in size and difficulty.

# 2. Presentation of the problem files and formats

### Initials problems files

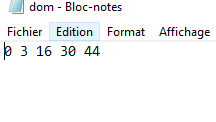
As described in the RLFAP website, each scenario is divided into 3 texts documents. We will illustrate them with a very simple problem made by ourselves. We called it scen00.

* Variables

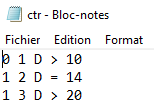


Here we have 4 variables (1,2,3,4) all belonging to the domain 0 (there is only 1 define domain here but there can me more). For more advanced scenario, as scen04, scen09 and scen10, we have a 3rd columns with the initial value of the variables and a 4th with the penalty if the constraint is not fulfilled.

* Domains

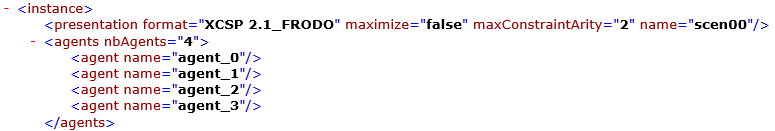
The 1st column is the domain number (referring to the domain indicate in the variable file), then the number of frequencies of the domain.

* Constraints

The 1st and 2nd columns are the frequencies (F1 & F2) targeted by the constraint. The 3rd is the origin of the constraint (for example D = difference) but not useful for the solving. Then the type of the constraint (equality or inequality) and finally the K12 presented in the introduction which is the (at least) difference expected between F1 and F2.

### Parsing to XML

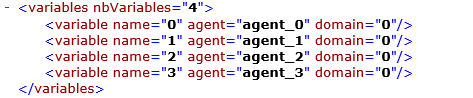
Frodo in using XML, so we had to parse the .txt files into the valid XML format. We used the FRODO\_User\_Manual to understand this format. It is quite simple, so we are going to present you the small XML associated with the scen00:



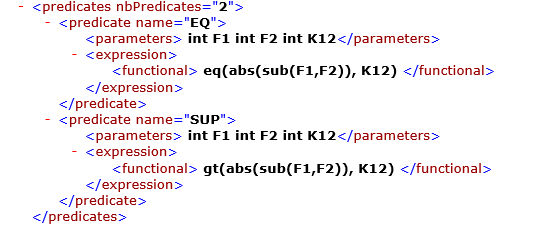
First, we define the agents, each one has a name, we precise the number of agents (generally one agent per variables)



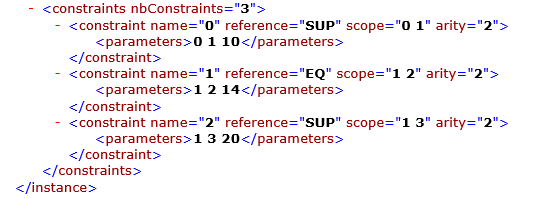
Then the domains, we precise the number of domains and give a name to each one, with their size and values



Then the variables: name, the agent taking care of this variables and the domain defining which value can take the variables



Then we have the predicates, it is the only tricky part. Each predicate is defining a function with parameters (F1, F2 and K12, all of type Integer). We only need 2 predicates because we only have 2 constraints type (= & >). The first one called “EQ” is computing |F1-F2|=K12 and the second named “SUP” is computing |F1-F2|>K12. We will see why we needs those predicates in the next and last part.



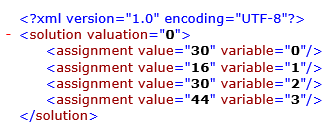
Here we are defining the constraints, each one has a name (simply counted 1,2,3…, n), the type of the the constraints referring to the associated predicate defined above (EQ or SUP). The scope is telling which parameters are variable, the others are considerate constants. The arity is the number of variables in the constraint. Finally, we give as parameters the parameters expected by the predicate (2 frequencies/variables and the K12).

For example, the first constraint is telling: |F0-F1|>10 and F0 and F1 are variables, 10 is a constant.

### Firsts results

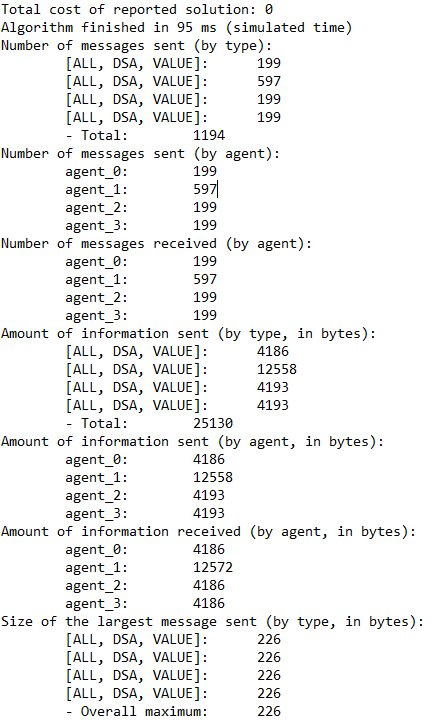
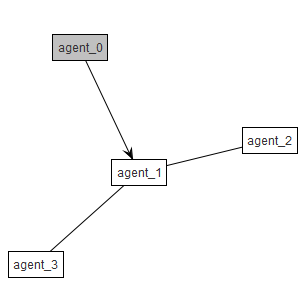
Once FRODO has solved the XML problem we gave him and the algorithm, we can analyse the results and the messages exchanges between agents during the resolution.

Here are the results for scen00 solved by the DSA algorithm



A quick look at the constraint allow us to show that the scenario is solved correctly.

Also, we have information about the number of send messages, their size, what is the cost of the solution and the time solving.



# 3. Analyse des résultats

We have compared some algorithms on some of the scenarios.

# Conclusion