REPORT
Heuristics and Optimization

CSP and Heuristic Search
Course 2020/2021

DEGREE IN COMPUTER ENGINEERING

GROUP 88

Carlos Gallego Jiménez 100405937
Andrés Langoyo Martín 100405869

TABLE OF CONTENTS

Carlos Gallego Jimenez 100405937
Andrés Langoyo Martín 100405869

1. INTRODUCTION

In this document, we describe how our team has implemented the models for the second lab assignment. Firstly, we describe how the CSP and the Heuristic Search Task have been implemented. Then, we analyze the obtained results and provide an explanation for them. Finally, we make some conclusions about the overall results and the assignment.

2. DESCRIPTION OF THE MODELS

2.1 Part 1: CSP

The first problem presented can be modeled as a constraint satisfaction model. This implies that should have the form:
$R = (X, D, C)$
The variables in this problem can be represented with the combination of satellites and possible time slots that each satellite has.

X = {SAT1 0-12, SAT2 0-12, SAT3 6-12, SAT3 13-16, SAT4 16-0, SAT5 6-13, SAT6 9-13, SAT6 13-19}

The possible values that can be assigned to each value are the antennas:

$D_i = \{ANT\ Y\}\ ,\ Y \in \{1..12\}$

However, the domains can be restricted for each variables:

Variable	**Domain**
SAT1 0-12	ANT1, ANT2, ANT3, ANT4
SAT2 0-12	ANT1, ANT2, ANT3
SAT3 6-12	ANT4, ANT6
SAT3 13-16	ANT7, ANT9, ANT10
SAT4 16-0	ANT8, ANT11, ANT12
SAT5 6-13	ANT1, ANT7, ANT12
SAT6 9-13	ANT7, ANT9
SAT6 13-19	ANT3, ANT4, ANT5

Now, the binary constraints must be specified:

Constraint 1: All satellites must have a transmission antenna assigned to it in all its time slots. Take SAT3 for instance with two visible time slots (6:00–12:00 and 13:00–16:00), so that an antenna has to be assigned to its first time slot and another one for the second, among those with visibility in each time slot.
With the definition of the variables and domains we have made, it is not necessary to post any constraint. This is because the variables are combination of satellites and time slots. As in a CSP problem all the variables must have a value, the constraint is satisfied if the problem is feasible.

Constraint 2: Since SAT1 and SAT2 have similar orbits, it is required to assign them the same antenna.

$R_{SAT1\ 0-12,\ SAT2\ 0-12} = \{(ANT1,\ ANT1), (ANT2,\ ANT2), (ANT3,\ ANT3)\ \}$

Constraint 3: Satellites SAT2, SAT4 and SAT5 should have assigned different antennae.

$R_{SAT2\ 0-12,\ SAT5\ 6-13} =$ *{(ANT1, ANT7),(ANT1, ANT7),*
(ANT2, ANT1), (ANT2, ANT7), (ANT2, ANT12),
(ANT3, ANT1), (ANT3, ANT7), (ANT3, ANT12)}

$R_{SAT4\ 16-0,\ SAT5\ 6-13} =$ *{(ANT12, ANT1),(ANT12, ANT7),*
(ANT8, ANT1), (ANT8, ANT7), (ANT8, ANT12),
(ANT11, ANT1), (ANT11, ANT7), (ANT11, ANT12)}

The constraints between SAT2 0-12 and SAT4 14-0 allow all possible combinations of values as none of them are equal.

Constraint 4: In case SAT5 communicates with ANT12, then SAT4 can not communicate with ANT11.

This constraint removes one combination from the previously specified constraint, leaving:

$R_{SAT4\ 16-0,\ SAT5\ 6-13} =$ *{(ANT12, ANT1),(ANT12, ANT7),*
(ANT8, ANT1), (ANT8, ANT7), (ANT8, ANT12),
(ANT11, ANT1), (ANT11, ANT7)}

Constraint 5: If, in any solution, ANT7 and ANT12 are used, then both must be assigned to time slots beginning before 12:00 or after. It is not allowed, for example, for ANT7 to be assigned the time slot 06:00–12:00 to SAT3, and ANT12 to the time slot 16:00–00:00 for communicating with SAT4.

Before noon	**After noon**
SAT1 00:00 - 12:00	SAT3 13:00 - 16:00 - **ANT7**
SAT2 00:00 - 12:00	SAT4 16:00 - 00:00 - **ANT12**

SAT3 06:00 - 12:00	SAT6 13:00 - 19:00
SAT5 06:00 - 13:00 - **ANT7, ANT12**	
SAT6 09:00 - 13:00 - **ANT7**	

Then, the legal combination between variables from different groups would be:

$R_{SAT3\ 13-16,\ SAT5\ 6-13}$ = *{ (ANT7, ANT1),*
(ANT9, ANT7), (ANT9, ANT7), (ANT9, ANT12),
(ANT10, ANT7), (ANT10, ANT7), (ANT10, ANT12) }

$R_{SAT3\ 13-16,\ SAT6\ 9-13}$ = *{ (ANT7, ANT9),*
(ANT9, ANT7), (ANT9, ANT9),
(ANT10, ANT7), (ANT10, ANT9) }

$R_{SAT4\ 16-0,\ SAT5\ 6-13}$ = *{ (ANT8, ANT1), (ANT8, ANT7), (ANT8, ANT12),*
(ANT11, ANT1), (ANT11, ANT7), (ANT11, ANT12),
(ANT12, ANT1) }

$R_{SAT4\ 16-0,\ SAT6\ 9-13}$ = *{ (ANT8, ANT7), (ANT8, ANT9),*
(ANT11, ANT7), (ANT11, ANT9),
(ANT12, ANT9) }

In any other restriction between two variables not specified above, all the possible combinations would be feasible.

2.2 Part 2: Search Problem

Assumptions

For this problem we made some assumptions that will be described in the following section:

- A first assumption regarding satellite location, we assume that the first satellite location is between the first two bands and the next satellites are one band apart except the last one which is 2 bands apart. With this disposition all visibility bands are covered by at least one satellite.
- A second assumption regarding the satellites was made, we have assumed satellites can turn to any band one movement at a time.
- A third assumption is that, satellite must only be idle if there is no better action to take, meaning that it will always charge before idling.
- Our final assumption regards the state space size, for this task as the hours in which a satellite is not visible would expand nodes with no action but idle, which has no impact on heuristics or battery, we are only counting the first 12 hours.

STATE SPACE

The state spaces are populated by a number of satellites and a number of objects to be measured. So the problem could be represented with two lists of those entities.

Each of the objects can be identified with several attributes:
Object(band, hour, measured)

The band is a number between 0 and the number of satellites plus two. The hour is a number between 0 and 11. Measure and downlinked indicate if a satellite has performed that operation on the object.

Regarding the satellites, we would have:
Satellite(band,hour,battery, cost_downlink, cost_measure, cost_turn, measured_objects)

The band is a number indicating that the satellite can see the objects in that position and the following one (band and band + 1)
The hour is a number between 0 and 11.
It is necessary to keep track of the costs of the different operations although they are not modified in the development of the problem.
Measured_objects is a container that keeps track of the objects that have been measured. This way, the satellite knows which satellite to downlink

The **initial state** of the problem can vary and it is given. In case of the statement, it would be:
Object1(0, 1, False)
Object2(0, 3, False)
Object3(1, 3, False)

Satellite1(0,0,1,1,1,1, measured=empty)
Satellite2(2,0,8,1,1,1, measured=empty)

Also, the number of bands would be 5 there are two more bands than satellites.

In a **goal state**, all the objects must have been measured and downlinked:
Object1(0, 1, True)
Object2(0, 3, True)
Object3(1, 3, True)

Satellite1(Band, Hour, Battery, cost_downlink, cost_measure, cost_turn, measured=empty)
Satellite2(Band, Hour, Battery, cost_downlink, cost_measure, cost_turn, measuerd=empty)

The band, hour and battery of the satellites can be anything in a final state.
The costs are maintained as in the beginning.
The list of measured objects of each satellite must be empty and all the objects marked as empty.

An estimation of the **size of the state space** in a given configuration could be:
(number of satellites + 2 bands for objects) x (11 hours) x (3 combinations of measured and downlinked) x (number of satellites + 1 bands for satellites) x (11 hours) x product(the possible number of energy units for each of the satellites)

Operators

Idle: An operator that does not apply any change, only the hour is updated.

- Precondiciones: battery must not be max_battery.
- Postconditions:
    - The hour is incremented.

For the following operators we will assume as postcondition the increase in the hour.

Measure: A satellite takes a measurement of an object.

- Preconditions:
    - The object must not have been measured already.
    - The satellite and object hours' and bands' must coincide.
    - The battery of the satellite must be enough.
    - Two satellites can not measure the same object at the same time slot
- Postconditions:
    - The object is marked as measured by the satellite.
    - The satellite battery must be decreased by the cost of the measurement operation.
    - The object is stored in the satellite container.
    - The hour is incremented.

Charge: A satellite increases its battery charge counter.

- Preconditions:
    - The energy of the battery must be lower than the maximum battery of the satellite.
- Postconditions:
    - The battery of the satellite increases by the value of the recharge of a satellite. The satellite can never have more than its maximum energy units.
    - The hour is incremented.

Downlink: A satellite downlinks an object.

- Preconditions:
    - The object must have been measured already by the satellite.
    - The battery of the satellite must be enough.
- Postconditions:
    - The object is downlinked by the satellite and, thus, removed by the satellite's container which measured it.
    - The satellite's battery is decreased by the downlink cost of that satellite.

Turn: A satellite changes position to an adjacent band.

- Preconditions:

    - The band must be adjacent to the satellite’s.
    - The battery of the satellite must be enough.
    - If the satellite is in the first band it can only move to the next one
    - If the satellite is in the last band it can only move to the previous one
    - If the satellite is in a non-external band, it can move to the previous one or the last one.
- Postconditions:
    - The satellite must change its band.
    - The satellite’s battery is decreased by the cost of the turning operation.

The minimum branching factor of a node happens when all the satellites are on an edge band and they cannot measure, downlink or recharge. The only possible actions are to remain idle and to turn in only one direction, which leaves us with two nodes. Assuming that satellites move at the same time the minimum branching factor would be 2 x number of satellites.

The maximum branching factor takes place when all the satellites are over an object which hasn’t been measured, they have data to downlink .

Heuristics implemented

For this problem we have implemented two heuristics, for both of them we are using constraint relaxation.
In the first heuristic, **h1**, we are relaxing the battery cost and eliminating the need to recharge.
We are taking as heuristic the average of the sum of the distances to each non-measured object from each one of the satellites and then we are adding the number of non measured objects and subtracting the downlinked objects.
In this way, we can get an approximate value for the steps a satellite must take to get to an object. This heuristic also favours the movement of the satellites towards the bands where there are more objects.

In the second heuristic, **h2**, we are relaxing the hours, as we want to minimize the energy and the measurement and downlink costs’ are unavoidable we are penalising the turns by adding to the heuristic the distance a satellite has from their original position.
Furthermore, we are adding to the heuristic the non measured objects and adding the non-downlinked.
In this way, we get an heuristic that allows the satellite to measure the objects nearest to it.

More heuristics were implemented but these two yielded the best results, these can be found at the end of the code commented.

3. ANALYSIS OF RESULTS

2.1 Part 1: CSP

After implementing our model, we obtained a total of solutions.

One example of those solutions is:

Variables	Value
SAT1 0-12	ANT3
SAT2 0-12	ANT3
SAT3 6-12	ANT6
SAT3 13-16	ANT10
SAT4 16-0	ANT12
SAT5 6-13	ANT1
SAT6 9-13	ANT9
SAT6 13-19	ANT5

Regarding constraints, we can observe that:

1. All satellites have been assigned one antenna for every time slot they have.
2. SAT1 and SAT2 have been assigned the same antenna, ANT3.
3. SAT4, SAT5 and SAT6 have been assigned a different antenna.
4. SAT4 and SAT5 have not been assigned ANT11 and ANT12 respectively.
5. ANT7 does not appear in the solution, so there is no coincidence between ANT7 and ANT12 in forbidden time slots.

As a result, the solution obtained is valid.

In order to generalize this problem we have separated code from data. Our script receives an input with the data in JSON format. The JSON is composed of two main parts: a data section and a constraints section:

The data section contains a dictionary for each Satellite, with a dictionary of time slots each. Every time slot contains a list of possible antennas.

```
"data": {
        "SAT1": {
          "0-12": [
            "ANT1",
            "ANT2",
            "ANT3",
            "ANT4"
          ]
        },
```

The constraint section has one list for every constraint:
The second constraint specifies the satellites that must be equal in all their time slots.

The third constraint specifies the satellites that must never be equal in all their time slots.
The fourth constraint contains a list of satellite-antenna tuples that must never happen simultaneously in a solution.
The fifth constraint specifies the list of antennas that must belong to the same group of time slots.

```
"constraints": {
    "constraint_2": ["SAT1", "SAT2"],
    "constraint_3": ["SAT2","SAT4","SAT5"],
    "constraint_4": [["SAT5","ANT12"],["SAT4","ANT11"]],
    "constraint_5": ["ANT7","ANT12"]
}
```

Test 1: 10 satellites and 20 different time slots with the same constraints. That yielded 6716520 solutions. The goal of the test was to be able to find an assignment for an increased number of satellites and antennas.
Solution:
SAT4 16-0 is assigned to ANT12
SAT10 13-19 is assigned to ANT5
SAT3 13-16 is assigned to ANT10
SAT6 13-19 is assigned to ANT5
SAT8 13-19 is assigned to ANT15
SAT9 16-0 is assigned to ANT17
SAT5 6-13 is assigned to ANT1
SAT2 0-12 is assigned to ANT3
SAT1 0-12 is assigned to ANT3
SAT10 9-13 is assigned to ANT9
SAT6 9-13 is assigned to ANT9
SAT7 0-12 is assigned to ANT20
SAT3 6-12 is assigned to ANT6
SAT8 9-13 is assigned to ANT2
SAT7 6-12 is assigned to ANT13
SAT9 9-13 is assigned to ANT12
SAT10 6-12 is assigned to ANT5

1. All satellites have been assigned one antenna for every time slot they have.
2. SAT1 and SAT2 have been assigned the same antenna, ANT3.
3. SAT4, SAT5 and SAT6 have been assigned a different antenna.
4. SAT4 and SAT5 have not been assigned ANT11 and ANT12 respectively.
5. ANT7 does not appear in the solution, so there is no coincidence between ANT7 and ANT12 in forbidden time slots.

Test 2: Adding SAT3, SAT5, SAT6 to the third constraint leaves 145 solutions.

SAT4 16-0 is assigned to ANT12
SAT5 6-13 is assigned to ANT1
SAT2 0-12 is assigned to ANT3
SAT6 13-19 is assigned to ANT5

SAT3 13-16 is assigned to ANT10
SAT6 9-13 is assigned to ANT9
SAT3 6-12 is assigned to ANT6
SAT1 0-12 is assigned to ANT3

It can be seen that SAT3, SAT5 and SAT6 have been assigned different antennas in their different time slots. Also, the number of solutions have decreased with respect the original problem.

Test 4: Adding SAT2 and SAT1 to the third and second constraints so that the problem results unfeasible.
The result leaves 0 solutions.

Test 5: Placing SAT4 with only ANT11 and SAT5 with ANT12, defying constraint 4.
The result leaves 0 solutions.

Test 6: Placing ANT12 in 16-0 and ANT 7 in 9-13, defying constraint 5.
The result leaves 0 solutions.

Test 7:
In this test, we want to modify constraints one and two to force satellites SAT1 SAT2, SAT4, SAT5 and make SAT3 and SAT6 different. For that, we included in the domain of the first four the ANT1 value.
One of the 40 solutions obtained was:

SAT3 13-16 is assigned to ANT10
SAT6 13-19 is assigned to ANT5
SAT4 16-0 is assigned to ANT1
SAT3 6-12 is assigned to ANT6
SAT6 9-13 is assigned to ANT9
SAT5 6-13 is assigned to ANT1
SAT2 0-12 is assigned to ANT1
SAT1 0-12 is assigned to ANT1

Test 8: In this test we changed the satellites ANT7 and ANT12 by ANT3 and ANT4.
There were obtained 457 solutions and one example is:

SAT4 16-0 is assigned to ANT12
SAT5 6-13 is assigned to ANT7
SAT2 0-12 is assigned to ANT3
SAT6 13-19 is assigned to ANT5
SAT3 13-16 is assigned to ANT10
SAT1 0-12 is assigned to ANT3
SAT3 6-12 is assigned to ANT6
SAT6 9-13 is assigned to ANT9

We can see that ANT3 and ANT4 meet this constraint as ANT4 has not been assigned as this would violate the constraint. Also, it can be seen that ANT7 and ANT12 would violate the original constraint, but they were removed from the constraint.

2.2 Part 2: Search Problem

Regarding the solution to the problem given, we obtain a path with a total cost of 6 energy units with both of our heuristics. This is clearly the minimum cost considering only energy units as there are three objects to be measured and downlinked. The first heuristic takes longer and more time because it gives some priority to turn operations while the second one penalizes them. However, in this situation it just suffices with waiting and measuring when satellites are over the objects.

```
ejemplos/problema1.prob h1
Overall time: 0.41
Overall cost: 6
# Steps: 17
# Expansions: 2365

1. SAT1: IDLE, SAT2: IDLE
2. SAT1: IDLE, SAT2: IDLE
3. SAT1: IDLE, SAT2: IDLE
4. SAT1: Measure O2, SAT2: IDLE
5. SAT1: Charge, SAT2: IDLE
6. SAT1: Downlink O2, SAT2: IDLE
7. SAT1: Charge, SAT2: IDLE
8. SAT1: IDLE, SAT2: IDLE
9. SAT1: IDLE, SAT2: IDLE
10. SAT1: IDLE, SAT2: IDLE
11. SAT1: IDLE, SAT2: IDLE
12. SAT1: IDLE, SAT2: IDLE
13. SAT1: IDLE, SAT2: IDLE
14. SAT1: Measure O1, SAT2: IDLE
15. SAT1: Charge, SAT2: IDLE
16. SAT1: Downlink O1, SAT2: Measure O3
17. SAT1: Charge, SAT2: Downlink O3
```

```
ejemplos/problema1.prob h2
Overall time: 0.03
Overall cost: 6
# Steps: 18
# Expansions: 158

1. SAT1: IDLE, SAT2: IDLE
2. SAT1: Measure O1, SAT2: IDLE
3. SAT1: Charge, SAT2: IDLE
4. SAT1: IDLE, SAT2: Measure O3
5. SAT1: IDLE, SAT2: Downlink O3
6. SAT1: IDLE, SAT2: Charge
7. SAT1: Downlink O1, SAT2: Charge
8. SAT1: Charge, SAT2: IDLE
9. SAT1: IDLE, SAT2: IDLE
10. SAT1: IDLE, SAT2: IDLE
11. SAT1: IDLE, SAT2: IDLE
12. SAT1: IDLE, SAT2: IDLE
13. SAT1: IDLE, SAT2: IDLE
14. SAT1: IDLE, SAT2: IDLE
15. SAT1: IDLE, SAT2: IDLE
16. SAT1: Measure O2, SAT2: IDLE
17. SAT1: Charge, SAT2: IDLE
18. SAT1: Downlink O2, SAT2: IDLE
```

Our implementation has the benefit that it allows us to add as many objects and satellites as required. Therefore, we can test how the program grows with the number of satellites and objects.

Comparison varying the number of objects

FILE	HEURISTIC	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
objects0.prob	H1	0.58	7	17	2365	2	3
objects1.prob	H1	1.14	8	29	5574	2	4
objects2.prob	H1	7.61	11	40	32607	2	5
objects3.prob	H1	17.14	13	52	62738	2	6
objects4.prob	H1	38.20	15	56	114648	2	7

FILE	HEURISTIC	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
objects0.prob	H2	0.03	6	18	158	2	3
objects1.prob	H2	0.07	8	16	277	2	4
objects2.prob	H2	1.54	12	39	5945	2	5
objects3.prob	H2	3	14	52	11574	2	6
objects4.prob	H2	3.26	15	41	11424	2	7

If we compare both heuristics given the satellites like in the example, and increase the number of satellites, we see that when using both heuristics time and the number of expanded nodes grow exponentially. The cost increases proportionally with the number of objects and is different in each heuristics because, although the heuristics are admissible, they might not be consistent. As a consequence, suboptimal solutions can be obtained.

Comparison varying the number of satellites

FILE	HEURISTIC	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
satellites0.prob	H1	1.51	9	32	6210	2	4
satellites1.prob	H1	1.49	9	29	5827	3	4
satellites2.prob	H1	0.03	9	11	93	4	4
satellites3.prob	H1	0.05	8	11	90	5	4
satellites4.prob	H1	0.07	8	12	162	6	4

FILE	HEURISTIC	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
satellites0.prob	H2	0.41	9	24	1483	2	4
satellites1.prob	H2	0.17	8	30	609	3	4
satellites2.prob	H2	0.11	8	13	264	4	4
satellites3.prob	H2	0.29	8	22	529	5	4
satellites4.prob	H2	0.33	8	22	575	6	4

When analysing the previous results we can see that the time decreases when we get the same number of satellites and objects as turn cost is decreased due to complete “parallelization” between each satellite and each object. After getting to the most optimal result in terms of time, it again starts to grow because the satellites that do not measure any object are just expanding nodes that have no purpose and, therefore, wasting time. However, for the second heuristic, they might find a solution in less steps as new satellites could be closer to objects.

Comparison varying the scatter of the objects

FILE	HEURISTIC	SCATTER	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
scatter0.prob	H1	LOW	0.35	20	26	662	6	8
scatter1.prob	H1	MEDIUM	0.38	20	20	562	6	8
scatter2.prob	H1	HIGH	0.25	18	20	398	6	8

FILE	HEURISTIC	SCATTER	TIME	COST	STEPS	EXPANDED	#SAT	#OBJ
scatter0.prob	H2	LOW	2.25	16	43	4087	6	8
scatter1.prob	H2	MEDIUM	2.24	16	43	4087	6	8
scatter2.prob	H2	HIGH	1.23	16	32	2219	6	8

Also the scatter plays an important role in the efficiency of the problem.
Here we placed in “scatter.prob0” all the satellites in the middle bands; in “scatter1.prob”, we placed in all bands but the exterior ones and, in “scatter2.prob”, we placed one object per band.

As the cost is related to battery consumption and the satellites are spread covering all the bands, they benefit greatly when they don't have to make any turning operation. That is why it can be observed that the expanded nodes and the time tend to grow with the dispersion.
With the first heuristic, the turns are penalized in the overall cost. For this reason, the number of steps necessary to reach the solution is reduced.

Regarding the assumptions we have made in the problem with respect to the initial layout of the problem, the second heuristic needs this to be true to perform correctly. If another configuration is given, it runs the risk to never reach the solution. Also, it doesn't take into account costs of different operations.

4. CONCLUSIONS

After working on this lab assignment we have learnt how to implement heuristics in a search task and how to model a CSP constraint satisfaction problem.
However, we found some things that could be improved.
Firstly, we think that the statement could be more specific in all aspects, as we had to make several assumptions at different times when working on this project that were related with the statement.

Secondly, we have thought that establishing the energy as the cost of the operators, the one to be minimized is a way harder task than minimizing the time for searching the solution. As we did not need to minimize the time, we thought that we could have just placed our satellites in a band with an object and make them idle as the measure and downlink cost cannot be avoided but the turn cost can.

Furthermore, we think that the satellites would have been more efficient if we tried to minimize the time as they would have taken as many actions as possible to measure and downlink objects while turning, as they can recharge their battery.

In general terms, while working in the laboratory several problems arose being the most daunting finding an admissible heuristic that completed the problem optimally, even when using constraint relaxation. as there were many parameters to take into account.
Moreover, in the code we have commented at the end the attempts of heuristics with constraint relaxation.

Furthermore, we had to change language as we had first implemented the search task in C++ but the constant difficulties forced us to change the implementation language so we were more familiar with it and we could approach the problem without encountering so many errors.