COGROB HOMEWORK 2: The Dyson Swarm Adventure

Due date: 15.1.2022

Submitters: Nir Manor   305229627

           Liran   308571561

This assignment involves designing and simulating a Dyson Swarm Adventure, which poses an engineering challenge to harness the Sun's energy.
The main objective is to capture **90%** of the sun's Irradiance by the year 2100.
Due to limitations in solar panel efficiency, we aim to maximize the amount of energy that can be obtained.
To accomplish this, we gather and analyze relevant engineering data from the web to formulate the physical properties of this challenge and make informed predictions about its future potential.

preliminary knowledge

- it takes about 2,000kWh to create a 1,000-watt solar panel, which is corresponded to $1[\mathrm{m}^2]$ of solar panel on earth.
- According to NASA's MESSENGER spacecraft, which orbited Mercury from 2011 to 2015, it took approximately 6.5 years to complete its journey from launch to orbit insertion at Mercury
- The maximal theoretical efficacy of solar panels is 68.7%
  Based on the mentioned, the efficacy we will reach in the nearest future will be 30% in Mercury orbit
- The sun Irradiance is a constant $3.95 \cdot 10^{26}[\mathrm{W}]$

General Assumptions:

- The efficiency of the satellite which capture energy, increase linearly as a function of distance from the sun
- We cannot improve the satellites efficiency remotely from earth
- Time for lunching a satellite to the orbit increase over the distance from the lunching point
- The energy that comes from the satellite should be divide between creating more satellite or increase the production ability
- More satellite can produce more energy that enable to increase the production of the satellites
- We assume that mercury has enough materials to produce the number of satellites we need for the Dyson swarm and we focused on this mission on the question of in which distance we will set the orbit.

Numerical Assumptions:

- According to the solar panel's maximal theoretical efficacy, we estimate that the efficacy in the nearest future will reach 30% in Mercury orbit
- Based on the amount of energy we know that is required to create solar panels, we estimate that it will be 20 times 'chipper' in energy terms, to produce our satellite on Mercury.
- We assuming the travel time in space will be shorter in the future and velocities of rockets will be 10 times faster in 10 years.

Implementation

To achieve our goal, we assumed that once an autonomous crew is deployed on Mercury and an initial number of energy-harvesting satellites are in orbit, the actions would be stable and the growth rate would be predictable and exponential. Using this assumption, we used PDDL to plan the initial stage of the mission and, once stability is reached, we plan to use a Python script to predict growth and energy absorption.
For that we divided our plan into 2 parts:

PDDL:
Using the UPF we created a PDDL for construction the initial satellite and crew, which will reach Mercury and will start to harvest the sun's energy for creating more satellite and harvest more energy. All the numbers that used for calculations were changed to month.

The predicates on the PDDL file is separated into 2 groups: Boolean Fluents and numerical Fluents. The purpose of the Boolean Fluents is to check that actions that dependent on previous ones can act. For example: we can't build satellites on mercury if an autonomous crew hasn't sent there.

The numerical Fluents purpose is to keep track on the important variables to this mission like power absorbed, number of satellites on mercury and on orbit and crew capabilities.

The actions the PDDL can choose:

1. Build satellites on Earth – building Initial number of satellites on earth (100k satellites, fix duration).
2. Build Deployment crew – building autonomous crew that will sent and deploy on mercury and will be able to construct more satellites (fix duration).
3. Launching satellites from Earth to Orbit - duration is dependent on orbit distance.
4. Lunching the autonomous crew from Earth to Mercury. (Fix duration)
5. Building satellites on Mercury – building satellites according to the amount of power given by the sun and the construction ability of the autonomous construction team.
   The duration of this action is fix to 6 months and the number of satellites is growing with the team capabilities.
   Parameters dependencies:
   - Power
   - Autonomous construction team capabilities.
6. Launching satellites from Mercury to orbit – the duration of this actions is dependent on the distance to orbit. The cost of this action is dependent on the distance of the orbit and on the number of satellites that sent.
   Parameters dependencies:

- Power
- Number of satellites on mercury
- Orbit distance
7. Expand the autonomous construction team capabilities – the cost and duration are fixed.

Python script:
In this part we continued what we accomplish in the first part.

We can assume that in certain point the process of creating new satellite and harvest more energy, is a repetitive process.
We created a function in python, which take in account the amount of satellite we already have and all our assumptions and requirements.
The function input variables:

- **n**: The number of satellites at the start of the iteration.
- **s**: The surface area of each satellite (in m^2)
- **e**: The base capture efficiency of each satellite (a value between 0 and 1)
- **es**: The energy required to create a single satellite (in J)
- **c**: The energy conversion efficiency (a value between 0 and 1)
- **target surface**: The target surface area that you want to cover with satellites (in [m^2])
- **distance**: The orbit distance from the sun at which the satellites are placed (in [m]

In the function, we evaluated the orbital characteristics of a satellite with respect to its distance from the sun. Specifically, we analyzed the time required for the satellite to travel from its initial position at Mercury to a desired orbit and the energy efficiency of the satellite on that orbit. Our analysis assumed a linear relationship between the deployment distance and these orbital parameters.
In terms of time and power that required for producing satellites on Mercury, we assumed that every week, we produce new stock of satellites.
We evaluate the amount of power for that by dividing the required energy over a week time.

The Equations we used in our function:

- The amount of power to produce 1 $[m^2]$ of Solar cell:

$$2{,}000{,}000[watt - hour] \cdot \frac{3600 \left[\frac{sec}{hour}\right]}{3600 \cdot 24 \cdot 7 \left[\frac{sec}{week}\right]} = 11904.76 \approx 11904[watt - week]$$

- The amount of power to produce 1 satellite with surface of 1 $[km^2]$:

$$E = 11904[watt - week] \cdot 10^6 = 1.1904 \cdot 10^{10}[watt - week]$$
$$= 0.2976 \cdot 10^{10}[watt - month]$$

- Travel time from mercury to the orbit:
$$T_{travel}\left(d_{dep}\right) = \frac{t_{mer2sun}}{d_{mer2sun}} \cdot \left(d_{dep}\right) + t_{min}[Sec]$$

- Satellite energy efficiency with respect to distance:

$$e_{eff}(d_{orbit2sun}) = e_s \frac{d_{orbit2sun} - d_{mer2sun}}{d_{mer2sun}} + e_s$$

- The surface area that required to capture the sun's power output:
$$S_{orbit} = 0.9 \cdot 4\pi(d_{orbit2sun})^2 [m^2]$$

- The energy that falls on each satellite with respect to the distance from the sun:
$$I_{solar-panel}(d_{orbit2sun}) = \frac{I_{solar}}{4\pi(d_{orbit2sun})^2} \left[\frac{W}{m^2}\right]$$

- The total energy we receive from all the satellites:
$$E_{total}(d_{orbit2sun}) = \frac{e_{eff}(r) \cdot N_{sat} \cdot S_s \cdot I_{solar}}{4\pi(d_{orbit2sun})^2} [W]$$

❖ $t_{mer2sun}$ – time travel from Mercury to the sun [Sec]
❖ t_{min} – it takes at least 1 day to travel to any orbit [Sec]
❖ $d_{mer2sun}$ – distance from mercury to the sun [m]
❖ $d_{orbit2sun}$ – distance from the orbit to the sun [m]
❖ d_{dep} – the absolute deployment distance from mercury to the orbit [m]
❖ e_s – efficiency on mercury
❖ I_{solar} – the sun total irradiance $3.95 \cdot 10^{26}$ [W]

Results:

We couldn't solve the PDDL part of the problem with the existing UPF solvers that we know.
We did manage to get a plan when we used relaxed assumptions that cancel the action dependencies, but the plans were logically inconsistent. We believe that we succeed in defining the problem and the domain in the correct way and with a better solver we would manage to program a plan.
The python script produced an approximation of the intake of the sun energy with the growth of the Dyson swarm in years. According to our assumptions and simplifications we assume that the initial

part of the mission is less relevant to starting point on Mercury and more corelated with the orbit distance from the Dyson swarm construction base.
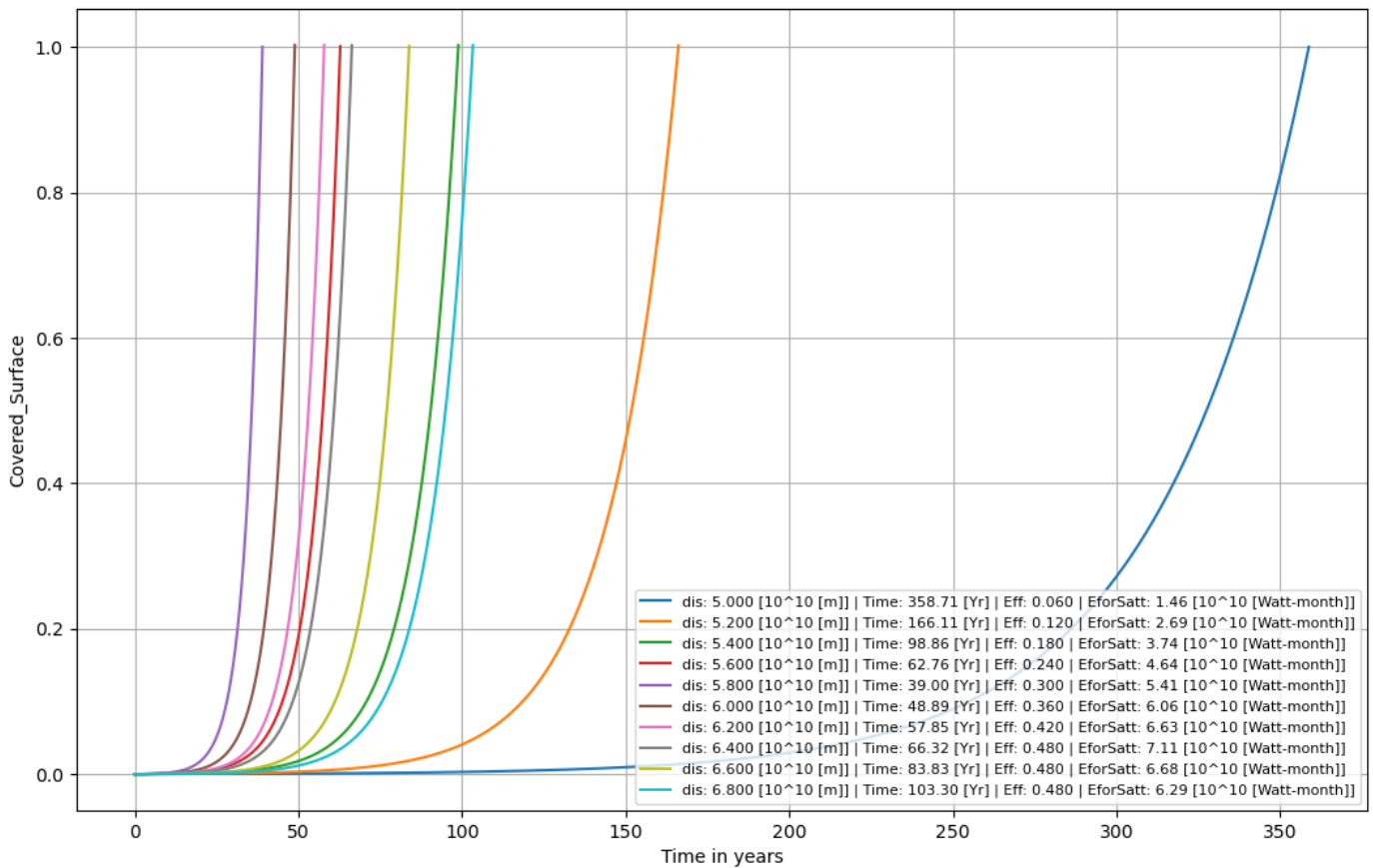

Figure 1: The covered surface of the sun in percentage through time in years with different distance of orbits from the sun. the distance is represented in $10^{-10}[m]$, the Time in years, Eff it the efficiency of each satellite in specific orbit and EforSatt is the energy absorbed in $[Watt - month]$.

Conclusions: In this assignment we asked to plan the construction of the complex structure that will facilitate an energy source in the future. The mission contains many factors which complicates the problem in many levels. We decided to isolate factors and check how to distance of the orbit of the Dyson swarm will affect the problem. We used many simplifications and relaxed assumptions in order to construct a simple model. The chosen method to solve this problem was with PDDL with another python script to approximate growth rate of the swarm in the stable part of the problem. The idea behind the combination was that in the second part of the mission the growth rate of the swarm is predictable and it can simplify the calculations.

In contrary to our efforts, we saw that the existed solvers that we know in UPF library, can't find a solution to the first part of the problem. We hope that a more complex solver will be able to accomplish it.

Assuming our approximations we can see from the python script that proximity to that construction base of the satellites is very important factor in the building process of the Dyson swarm.