**Input Variables**

*dispatch\_ih*- This contains MWh of heat dispatch for each actor I during each hour h of a given year. It is structured as 2D numpy array with each row representing an actor i and column representing dispatch during each hour h. It is taken from the market module.

*price\_h* - This contains €/MWh price of heat during each hour h of the given year. It is structured as 1D numpy array with each element corresponding to price during each hour h. It is also taken from the market module.

*Opcost\_ih* – This represents operation cost € for each actor i during each hour h of operation for the given year. It is structured as 2D numpy array with each row representing an actor i and column representing the total operation cost incurred during each hour h. It is taken from the market module.

*Capex\_t* – This represents the initial investment into a given capacity of each technology t installed or selected by the techno-economic module. It is structured as 1D numpy array with each element representing capex of each technology.

*opex\_t* - This represents the yearly operation & maintenance cost of each technology t installed or selected by techno-economic module. It is structured as 1D numpy array with each element representing the opex of each technology.

*Projectduration* – This represents the lifetime in years of the whole project. It is taken from the knowledge base as an integer.

*dicountrate\_i* – Represents discount rate for two scenarios: socio economic scenario & private business scenario. It is also taken from the knowledge base as 1D numpy array containing two elements, one discount rate for each scenario.

*rls* – This represents the relationship between actors and technologies that they own. It is structured as 2D array with only two columns with the first representing the actor and the second the corresponding technology owned by that particular actor. Thus, this matrix contains 1-to-1 mapping of actors and their ownership of technologies. This information on ownership is primarily provided by the user however, this matrix should be created either in core functionalities or in the business module – should be sorted out in subsequent integration of different modules.

**Function**

The function is called BM which takes in all the above-mentioned variables as input. The function first does some pre-processing mainly to calculate the capex and opex for each actor as the capex and opex provided by the techno-economic module only corresponds to each technology. This corresponding of actor to installed technology is provided by *rls* matrix and this is used to find the capex and opex for each actor involved.

The net present value (NPV) and internal rate of return (IRR) are calculated for two scenarios: socio-economic and private business. For private business scenario, NPV and IRR are calculated for each actor while for socio-economic scenario only one NPV and IRR is calculated. Further, a sensitivity analysis is performed on the above-mentioned NPVs by increasing and decreasing the discount rate (*discountrate\_i*) by 50%. These values are plotted within the function.

**Function Output**

*IRR\_socio* – Internal rate of return for the socio-economic scenario, a single float value.

*NPV\_socio* – Net present value for the socio-economic scenario, also a single float value.

*NPV\_socio\_sen* – Results of sensitivity analysis conducted on the NPV under socio-economic scenario. It is represented as 1D numpy array with float datatype.

*IRR\_i* – Internal rate of return for each actor i under the private business scenario. It is represented as 1D numpy array of float datatype.

*NPV\_i* - Net Present value for each actor i under the private business scenario. It is represented as 1D numpy array of float datatype.

*NPV\_sen\_i* – It contains the result of sensitivity analysis on the NPVs of each actor i under private business scenario. It is represented as a 2D numpy array of float datatype with rows representing actors and columns containing the NPV values against different discount rates.