Shapley Coalitions for Prosumers in NRG-X-Change Network

Paying Prosumers Using NRG-X-Change

What is NRG-X-Change? "In this paper we propose NRG-X-Change — anovel mechanism for trading of locally produced re-newable energy that does not rely on an energy mar-ket or matching of orders ahead of time. In our modellocally produced energy is continuously fed into thegrid and payment is received based on actual usage,rather than predicted, as consumption is measured bythe DSO (Distribution Service Operator) and billed in near real-time." (Mihail Mihaylov)

Modeling a Prosumer Market

We will leverage an NRGX-Change based market of prosumers on an micro-grid. Any excess energy that is not consumed by the micro-grid is not considered at this time. The goal of each prosumer would be to offset the demand of the micro-grid. In most cases the total demand would need to be supplemented by the larger Grid at some retail electricity price. The network will use the payout and consumption functions for NRGX-Change as each prosumer generates every month.

The prosumers (P) are indicated on the diagram as generators of electricity. In this scenario solar PV is shown as a generation source. The Distribution Service Operator (DSO) is in charge of consuming excess Net Energy that is generated by the prosumer and it is not consumed by the prosumer. A consumer (C) is the term for a residence that has to consumer more energy that it can produce. It will need to pay for electricity and the DSO would charge it at some price. The NRG-X-Change function will charge the consumer and pay the prosumer for electricity consumed or generated.

Prosumer Payment Function

The NRG-X-Change as described by the authors performs a dynamic payment to prosumers that are capable of meeting the demand of the micro-grid. The micro-grid is made of prosumers and consumers. As the load demand spikes the pricing for net generation also spikes to meet the demand. When there is too much generation on the grid the pricing drops encouraging prosumers to generate less and consumer to consume more. The payout function g(.) , utilizes a normalization component in the denominator to account for over or under generation distributing the payout along the curve. The payment is at its highest when generation meets the total demand and at its lowest as generation starts to saturate the market because of low demand.

$$g(x, t_p, t_c) = \frac{x^n * q_{t_p=t_c}}{e^{\frac{(t_p-t_c)^2}{a}}}$$

In [105…

```
import math
# NRGXChange Payment g(.) Function
def g(price,X,tp,tc,a,n):
    q = price
    try:
        #print(f"n{n},tp{tp},tc{tc},a{a},q{q}")
        pay = (pow(X,n)*q)/math.exp(pow((tp-tc),2)/a)
    except OverflowError:
        pay = float('inf')
    return pay
```

Consumer Charge/Cost Function

Where x, is the net energy of the prosumer. q, is the maximum price allowed. t_p ,is the total produced energy of all prosumers. t_c is the total consumption of all the prosumers. a , is a scaling constant to adjust the pay out. Similarly lets consider the cost of energy for consumers to purchase based on pricing set by the h(.) function. In tandem these incentives are non-linear because of the distribution curve. The shape of that curve can be adjusted to the size of the network and the volatility of the network.

$$h(y, t_p, t_c) = \frac{y * r_{t_c >> t_p} * t_c}{t_c + t_p}$$

Where y is the withdrawn energy, and $r_{t_c >> t_p}$ is the maximum cost of energy delivered by the utility when the energy supply by prosumers is low. Again, t_p is the total production and t_c is the total consumption of the prosumers in the network. The minimum payment by the utility in the historical payment prices would indicate the minimum amount willing to charge customers for energy in order to cover the cost of delivering the energy. We will use the minimum price in our list for r.

In [106…

```
# NRGXChange Charge h(.) Function
def h(price,c,tp,tc):
    y = c
    r = (0.01*price)
    try:
        cost = (y*r*tc)/(tc+tp)
    except OverflowError:
        cost = float('inf')
    return cost
```

Creating Coalitions Using Shapley Value

Review of Game Theory and Shapley Value

The game is in terms of a **characteristic function**, which specfies for every group of players the total payoff that the members of S can by signing an greement among themselves; this payoff is available for distribution among the members of the group. A coalitional game with transferable payoff is a pair $< N, v >$ where $N = \{1, \ldots, n\}$ is the set of players and for every subset S of I (called a coalition) $v(S) \in \mathbb{R}$ is the total payoff that is available for division among members of S (called the worth of S). We assume that the larger the coalition the larger the payoff (this property is called superadditivity).

An agreement amongst players is a list $(x_1, x_1, \ldots, x_n)$ where x_1, is the proposed payoff to individual i. Shapley value is interpreted in terms of **expected marginal contribution**. It is calculated by considering all the possible orders of arrival of the players into a room and giving each player his marginal contribution.

In [117…

```
# Shapley Value Python Logic
# Authored by Susobhan Ghosh
# https://github.com/susobhang70
# Committed on 02/01/2020
from itertools import combinations
import bisect
#Create Combinatorial from List
def power_set(List):
    PS = [list(j) for i in range(len(List)) for j in combinations(List, i+1)]
    return PS
#Calculate Shapley from Characteristic Value list
def get_shapley(n,v):
    tempList = list([i for i in range(n)])
    N = power_set(tempList)
    shapley_values = []
    for i in range(n):
        shapley = 0
        for j in N:
            if i not in j:
                cmod = len(j)
                Cui = j[:]
                bisect.insort_left(Cui,i)
                l = N.index(j)
                k = N.index(Cui)
                temp = float(float(v[k]) - float(v[l])) *\
                           float(math.factorial(cmod) * math.factorial(n - cmod - 1)) / float(math.factorial(n))
                shapley += temp
        cmod = 0
        Cui = [i]
        k = N.index(Cui)
        temp = float(v[k]) * float(math.factorial(cmod) * math.factorial(n - cmod - 1)) / float(math.factorial(n))
        shapley += temp
        shapley_values.append(shapley)
    return shapley_values
```

Pecan Street Data

Pecan Street is a research and development orgnaization that gathers data from active homes, solar homes and electric vehicle owners. According to Spandana Vadam , the pecan street data can be used to build out prosumers and calculating shapley value for coalitional contributions.

The data ranges between 2015-09-23 and 2015-12-22 and is segmented by hour. The data is of 6 single family homes located in Austin, texas. Each home was installed with a PV system for the year of 2015. Each home has a 'Gen',power generated from PV systems, and a 'Use',whole-home electrical usage, value for every hour of the day during Fall, Spring and Winter.

Define Mathematical Model

The monthly average energy production is the individual energy production summed up across the entire Fall season and then devided by 3 for each month. The Gen_{ai} in kWh is the average generation of the i^th prosumer.

The generalized characteristic function, $v(i) = \frac{X*q}{e^{\frac{[Gen_a - Use_a]^2}{a}}}$. When we apply it specifically to a single prosumer we must use the single prosumers net generation for X and then consider all prosumer generation and all prosumer usage as a sum of the averages. Note, we will also choose the $q = \$10/kWh$ for the pricing and the scale factor , $a = 10^6$.

$$v(i) = \frac{(Gen_{ai} - Use_{ai}) * 10}{e^{\frac{[\sum_{i=1}^{6}(Gen_{ai}) - \sum_{i=1}^{6}(Use_{ai})]^2}{10^6}}}$$

Computing Shapley value with data collected from Pecan Street

i(House)	**FALL**			**Spring**			**Winter**		
Prosumer	**Monthly average Energy Production for Fall (kWh) *Gen_ai***	**Monthly average Energy Consumption for Fall(kWh) *Use_ai***	**Energy offered by each prosumer (kWh) (X)**	**Monthly average Energy Production for Spring (kWh) *Gen_ai***	**Monthly average Energy Consumption for Spring (kWh) *Use_ai***	**Energy offered by each prosumer (kWh) (X)**	**Monthly average Energy Production for Winter (kWh) *Gen_ai***	**Monthly average Energy Consumption for Winter (kWh) *Use_ai***	**Energy offered by each prosumer (kWh) (X)**

i(House)	FALL			Spring			Winter		
1	1473	1523	-50.00	1514	1431	83.00	1239	1138	101.00
2	1215	1056	159.00	1937	1279	658.00	1045	903	142.00
3	1006	367	639.00	1337	378	959.00	871	288	583.00
4	643	970	-327.00	903	1087	-184.00	575	810	-235.00
5	1737	676	1061.00	1996	749	1247.00	1447	406	1041.00
6	1518	1098	420.00	1560	1346	214.00	1273	677	596.00
Total	7592.00	5690.00	1902.00	9247.00	6270.00	2977.00	6450.00	4222.00	2228.00

In [229…

```
# Convert Pecan Data into dataset by trial to analyze
pecan_data = [
    {'id':1,'time':'fall','generation':1473,'consumption':1523},
    {'id':2,'time':'fall','generation':1215,'consumption':1056},
    {'id':3,'time':'fall','generation':1005,'consumption':367},
    {'id':4,'time':'fall','generation':643,'consumption':970},
    {'id':5,'time':'fall','generation':1737,'consumption':676},
    {'id':6,'time':'fall','generation':1518,'consumption':1098},
    {'id':1,'time':'spring','generation':1514,'consumption':1431},
    {'id':2,'time':'spring','generation':1937,'consumption':1279},
    {'id':3,'time':'spring','generation':1337,'consumption':1279},
    {'id':4,'time':'spring','generation':903,'consumption':378},
    {'id':5,'time':'spring','generation':1996,'consumption':1087},
    {'id':6,'time':'spring','generation':1560,'consumption':1346},
    {'id':1,'time':'winter','generation':1239,'consumption':1138},
    {'id':2,'time':'winter','generation':1045,'consumption':903},
    {'id':3,'time':'winter','generation':871,'consumption':288},
    {'id':4,'time':'winter','generation':575,'consumption':810},
    {'id':5,'time':'winter','generation':1447,'consumption':406},
    {'id':6,'time':'winter','generation':1273,'consumption':677}
]
```

```
pecan_df = DataFrame(pecan_data)
pecan_df['net_energy'] = pecan_df['generation'] - pecan_df['consumption']
```

In [230…

```
from pandas import DataFrame
#####################################################################
# Calculate the shapley value given the net energy and the average
# generation and consumption values
#####################################################################
def get_nrg_payments( df,price,a=0,n=1):
    for t in df:
        tp = t['generation'].sum()
        tc = t['consumption'].sum()
        t['shapley_w_coalition'] = t['net_energy'].apply(lambda x: g(price=price,X=x,tc=tc,tp=tp,n=n,a=a))
        t['shapley_wo_coalition'] = t['net_energy'].apply(lambda x: g(price=price,X=x,tc=tc,tp=tp,n=1,a=a))
    return df
```

The characteristic function results in a shapley value for each of the prosumers for each of the time periods.

In [231…

```
from IPython.display import display, HTML
pecan_df_by_t = [DataFrame(y) for x, y in pecan_df.groupby('time', as_index=False)]
x1_pecan_df_by_t = get_nrg_payments(pecan_df_by_t,price=10,a=1000000)
tb = 1
for t in x1_pecan_df_by_t:
    t = t.reindex(columns=['id','time','generation','consumption','net_energy','shapley_w_coalition','shapley_wo_coaliti
    display(HTML(f"</br> Table {tb}: X^1 </br>{t.to_html(index=False)}"))
    tb=tb+1
```

Table 1: X^1

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
1	fall	1473	1523	-50	-13.474609	-13.474609
2	fall	1215	1056	159	42.849257	42.849257
3	fall	1005	367	638	171.936011	171.936011
4	fall	643	970	-327	-88.123943	-88.123943
5	fall	1737	676	1061	285.931203	285.931203
6	fall	1518	1098	420	113.186715	113.186715

Table 2: X^1

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
1	spring	1514	1431	83	2.082599	2.082599
2	spring	1937	1279	658	16.510244	16.510244
3	spring	1337	1279	58	1.455310	1.455310
4	spring	903	378	525	13.173067	13.173067
5	spring	1996	1087	909	22.808224	22.808224
6	spring	1560	1346	214	5.369593	5.369593

Table 3: X^1

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
1	winter	1239	1138	101	7.054894	7.054894
2	winter	1045	903	142	9.918762	9.918762
3	winter	871	288	583	40.722806	40.722806
4	winter	575	810	-235	-16.414853	-16.414853
5	winter	1447	406	1041	72.714307	72.714307
6	winter	1273	677	596	41.630862	41.630862

Modify the Linearity of the Characteristic Function (X^n)

The negactive shapley valyue cannot be used to make a fair distribution of gains in a coalition.

In [246…

```
from pandas import DataFrame
#####################################################################
# Calculate the shapley value given the net energy and the average
# generation and consumption values
#####################################################################
def get_nrg_payments( df,price,n,a):
    for t in df:
        t.drop(t[t['net_energy'] <= 0].index, inplace = True)
        tp = t['generation'].sum()
        tc = t['consumption'].sum()
        print(f"generation {tp}, consumption {tc}")
        t['shapley_wo_coalition'] = t['net_energy'].apply(lambda x: g(price=price,X=x,tc=tc,tp=tp,n=1,a=a) )
```

```
        #t['shapley_wo_coalition'] = t['net_energy'].apply(lambda x: g(price=price,X=x,tc=tc,tp=tp,n=1,a=a) )
    return df
```

In [247…

```
from IPython.display import display, HTML
pecan_df_by_t = [DataFrame(y) for x, y in pecan_df.groupby('time', as_index=False)]
x1p5_pecan_df_by_t = get_nrg_payments(pecan_df_by_t,price=10,a=1000000,n=1.5)

tb = 1
for t in x1p5_pecan_df_by_t:
    t = t.reindex(columns=['id','time','generation','consumption','net_energy','shapley_w_coalition','shapley_wo_coaliti
    display(HTML(f"</br> Table {tb}: X^1.5 </br>{t.to_html(index=False)}"))
    tb=tb+1
```

```
generation 5475, consumption 3197
generation 9247, consumption 6800
generation 5875, consumption 3412
```

Table 1: X^1.5

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
2	fall	1215	1056	159	NaN	8.865837
3	fall	1005	367	638	NaN	35.574866
5	fall	1737	676	1061	NaN	59.161337
6	fall	1518	1098	420	NaN	23.419191

Table 2: X^1.5

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
1	spring	1514	1431	83	NaN	2.082599
2	spring	1937	1279	658	NaN	16.510244
3	spring	1337	1279	58	NaN	1.455310
4	spring	903	378	525	NaN	13.173067
5	spring	1996	1087	909	NaN	22.808224
6	spring	1560	1346	214	NaN	5.369593

Table 3: X^1.5

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition

id	time	generation	consumption	net_energy	shapley_w_coalition	shapley_wo_coalition
1	winter	1239	1138	101	NaN	2.342776
2	winter	1045	903	142	NaN	3.293804
3	winter	871	288	583	NaN	13.523153
5	winter	1447	406	1041	NaN	24.146831
6	winter	1273	677	596	NaN	13.824699

In [254…

```
pay = (pow(83,1.5)*10)/math.exp(pow((9247-6800),2)/1000000)
pay
```

Out[254… 18.973381148762577

Prosumer Synthesized Data from EIA.gov

The data gathered is from EIA.gov. The data was then used to synthesize typical prosumer consumption and generation over the course of 12 months. Volatitlity in the usage and generation was added as a normal distribution with a given variance to simulate real world conditions. The data is pulled from a local file and then sorted by timestamp. The fields are grouped by id and by time. Grouping by time allows for settlement calculations to occur at each time interval.

In [121…

```
import pandas as pd
import numpy as np
#####################################################################
# Data Gathering
#####################################################################
# Data gathering and synthesization has been done in a seperate module.
# import the external data gathering set. Pull data from a dataset
# of randomly insantiated prosumer, synthesized from EIA.gov data trends
#
import p0_data_gather as p0
# Set path of dataset file containing all parameterized values
data_set_path = 'data/prosumer_N10_all_20210305_1129.csv'
# Set initial conditions for prosumers dataset
dem_mean = 1100
gen_mean = 1300
# Set number of prosumers, an array of N[] values for multiple experiments
number_of_prosumers = [6]
# Trials [] holds session data for each N itteration of prosumers
trials = []
for N in number_of_prosumers:
```

```
    # Call the data as a query with the instantiated values
    eia_prosumer_data = p0.get_data(path=data_set_path,query=f'id > 0 & id<={N} & demand_std == {dem_mean*0.20} & genera
    ######################################################################
    # Data Wrangling : Split data by time 't'
    ######################################################################
    # Wrangle data into monthly timesteps so that each time step
    # could be processed individually as a market payment for each prosumer
    eia_prosumer_data_by_t = [pd.DataFrame(y) for x, y in eia_prosumer_data.groupby('time', as_index=False)]
    trials.append({"N":N, "prosumers_n":eia_prosumer_data, "prosumers_n_t":eia_prosumer_data_by_t})
print(trials)
```

```
20.0           260.0              1300         1100
25913    11.71       220.0            260.0              1300          1100
25914    11.63       220.0            260.0              1300          1100
25915    11.76       220.0            260.0              1300          1100
25916    11.72       220.0            260.0              1300          1100
25917    11.62       220.0            260.0              1300          1100
25918    12.09       220.0            260.0              1300          1100
25919    11.66       220.0            260.0              1300          1100
1235745  11.86       220.0            260.0              1300          1100
1235746  12.00       220.0            260.0              1300          1100
1235747  11.71       220.0            260.0              1300          1100
1235748  11.97       220.0            260.0              1300          1100
1235749  11.61       220.0            260.0              1300          1100
1235750  11.71       220.0            260.0              1300          1100
1235751  11.53       220.0            260.0              1300          1100
1235752   9.84       220.0            260.0              1300          1100
1235753  11.71       220.0            260.0              1300          1100
1235754  11.63       220.0            260.0              1300          1100
1235755  11.76       220.0            260.0              1300          1100
1235756  11.72       220.0            260.0              1300          1100
1235757  11.62       220.0            260.0              1300          1100
1235758  12.09       220.0            260.0              1300          1100
1235759  11.66       220.0            260.0              1300          1100
...        ...         ...              ...               ...           ...
4865265  11.86       220.0            260.0              1300          1100
4865266  12.00       220.0            260.0              1300          1100
4865267  11.71       220.0            260.0              1300          1100
4865268  11.97       220.0            260.0              1300          1100
4865269  11.61       220.0            260.0              1300          1100
4865270  11.71       220.0            260.0              1300          1100
4865271  11.53       220.0            260.0              1300          1100
4865272   9.84       220.0            260.0              1300          1100
4865273  11.71       220.0            260.0              1300          1100
4865274  11.63       220.0            260.0              1300          1100
4865275  11.76       220.0            260.0              1300          1100
4865276  11.72       220.0            260.0              1300          1100
4865277  11.62       220.0            260.0              1300          1100
4865278  12.09       220.0            260.0              1300          1100
4865279  11.66       220.0            260.0              1300          1100
```

```
6075105  11.86       220.0            260.0              1300         1100
6075106  12.00       220.0            260.0              1300         1100
6075107  11.71       220.0            260.0              1300         1100
6075108  11.97       220.0            260.0              1300         1100
6075109  11.61       220.0            260.0              1300         1100
6075110  11.71       220.0            260.0              1300         1100
6075111  11.53       220.0            260.0              1300         1100
6075112   9.84       220.0            260.0              1300         1100
6075113  11.71       220.0            260.0              1300         1100
6075114  11.63       220.0            260.0              1300         1100
6075115  11.76       220.0            260.0              1300         1100
6075116  11.72       220.0            260.0              1300         1100
6075117  11.62       220.0            260.0              1300         1100
6075118  12.09       220.0            260.0              1300         1100
6075119  11.66       220.0            260.0              1300         1100

[90 rows x 11 columns], 'prosumers_n_t': [         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119   6 2019-10-01   693.494570  588.374290   105.120280         0.0

         price  demand_std  generation_std  generation_mean  demand_mean
25919    11.66       220.0            260.0              1300         1100
1235759  11.66       220.0            260.0              1300         1100
2445599  11.66       220.0            260.0              1300         1100
3655439  11.66       220.0            260.0              1300         1100
4865279  11.66       220.0            260.0              1300         1100
6075119  11.66       220.0            260.0              1300         1100  ,          id       time      demand  generatio
n  consumption  net_energy  \
25918     1 2019-11-01  856.581711  427.340324   429.241387     0.00000
1235758   2 2019-11-01  722.211216  868.127766     0.000000  -145.91655
2445598   3 2019-11-01  817.316715  707.889217   109.427498     0.00000
3655438   4 2019-11-01  827.408281  605.949567   221.458714     0.00000
4865278   5 2019-11-01  986.823771  692.295160   294.528610     0.00000
6075118   6 2019-11-01  782.905659  772.206093    10.699567     0.00000

         price  demand_std  generation_std  generation_mean  demand_mean
25918    12.09       220.0            260.0              1300         1100
1235758  12.09       220.0            260.0              1300         1100
2445598  12.09       220.0            260.0              1300         1100
3655438  12.09       220.0            260.0              1300         1100
4865278  12.09       220.0            260.0              1300         1100
6075118  12.09       220.0            260.0              1300         1100  ,          id       time       demand  generati
on  consumption  net_energy  \
25917     1 2019-12-01   956.424678  562.262330   394.162348    0.000000
1235757   2 2019-12-01   436.158803  797.827130     0.000000 -361.668327
2445597   3 2019-12-01   512.060530  663.954644     0.000000 -151.894114
3655437   4 2019-12-01  1022.271106  644.253416   378.017690    0.000000
```

```
4865277   5 2019-12-01   625.994411  481.237716   144.756695    0.000000
6075117   6 2019-12-01   611.713281  867.306418     0.000000 -255.593137

         price  demand_std  generation_std  generation_mean  demand_mean
25917    11.62       220.0           260.0             1300         1100
1235757  11.62       220.0           260.0             1300         1100
2445597  11.62       220.0           260.0             1300         1100
3655437  11.62       220.0           260.0             1300         1100
4865277  11.62       220.0           260.0             1300         1100
6075117  11.62       220.0           260.0             1300         1100  ,           id        time        demand   generat
ion  consumption  net_energy  \
25916     1 2020-01-01   712.969492   712.567943     0.401549    0.000000
1235756   2 2020-01-01   518.965013  1084.608756     0.000000 -565.643743
2445596   3 2020-01-01   640.004559   831.239223     0.000000 -191.234664
3655436   4 2020-01-01  1032.366027   339.594765   692.771262    0.000000
4865276   5 2020-01-01   577.141083   785.472474     0.000000 -208.331391
6075116   6 2020-01-01   621.511333   758.046211     0.000000 -136.534878

         price  demand_std  generation_std  generation_mean  demand_mean
25916    11.72       220.0           260.0             1300         1100
1235756  11.72       220.0           260.0             1300         1100
2445596  11.72       220.0           260.0             1300         1100
3655436  11.72       220.0           260.0             1300         1100
4865276  11.72       220.0           260.0             1300         1100
6075116  11.72       220.0           260.0             1300         1100  ,           id        time       demand   generati
on  consumption  net_energy  \
25915     1 2020-02-01  741.830771   700.851761     40.97901    0.000000
1235755   2 2020-02-01  723.828069   738.891573      0.00000  -15.063505
2445595   3 2020-02-01  812.585379   917.494903      0.00000 -104.909524
3655435   4 2020-02-01  529.188718  1100.123378      0.00000 -570.934660
4865275   5 2020-02-01  584.356003   834.353270      0.00000 -249.997266
6075115   6 2020-02-01  869.572795   909.884944      0.00000  -40.312149

         price  demand_std  generation_std  generation_mean  demand_mean
25915    11.76       220.0           260.0             1300         1100
1235755  11.76       220.0           260.0             1300         1100
2445595  11.76       220.0           260.0             1300         1100
3655435  11.76       220.0           260.0             1300         1100
4865275  11.76       220.0           260.0             1300         1100
6075115  11.76       220.0           260.0             1300         1100  ,           id        time       demand   generati
on  consumption  net_energy  \
25914     1 2020-03-01  600.908845  1232.947114          0.0 -632.038269
1235754   2 2020-03-01  660.007831   805.269837          0.0 -145.262005
2445594   3 2020-03-01  680.838258  1095.724158          0.0 -414.885900
3655434   4 2020-03-01  667.738498   744.210478          0.0  -76.471981
4865274   5 2020-03-01  624.732258   985.682326          0.0 -360.950068
6075114   6 2020-03-01  672.766573  1300.000000          0.0 -627.233427

         price  demand_std  generation_std  generation_mean  demand_mean
25914    11.63       220.0           260.0             1300         1100
1235754  11.63       220.0           260.0             1300         1100
```

```
2445594  11.63       220.0            260.0             1300         1100
3655434  11.63       220.0            260.0             1300         1100
4865274  11.63       220.0            260.0             1300         1100
6075114  11.63       220.0            260.0             1300         1100  ,          id       time       demand   generat
ion  consumption  net_energy  \
25913     1 2020-04-01   840.259808  1100.604230     0.000000 -260.344422
1235753   2 2020-04-01   631.583253  1254.538710     0.000000 -622.955457
2445593   3 2020-04-01   642.718380  1127.648368     0.000000 -484.929988
3655433   4 2020-04-01  1207.234188  1142.545229    64.688959    0.000000
4865273   5 2020-04-01   849.707278  1300.000000     0.000000 -450.292722
6075113   6 2020-04-01   550.462581   799.488549     0.000000 -249.025969

         price  demand_std  generation_std  generation_mean  demand_mean
25913    11.71       220.0            260.0             1300         1100
1235753  11.71       220.0            260.0             1300         1100
2445593  11.71       220.0            260.0             1300         1100
3655433  11.71       220.0            260.0             1300         1100
4865273  11.71       220.0            260.0             1300         1100
6075113  11.71       220.0            260.0             1300         1100  ,          id       time      demand   generati
on  consumption  net_energy  \
25912     1 2020-05-01  912.760967  1300.000000          0.0 -387.239033
1235752   2 2020-05-01  739.712334  1300.000000          0.0 -560.287666
2445592   3 2020-05-01  775.732607  1077.631521          0.0 -301.898914
3655432   4 2020-05-01  715.108925  1300.000000          0.0 -584.891075
4865272   5 2020-05-01  876.251881  1159.808066          0.0 -283.556185
6075112   6 2020-05-01  816.994069  1227.743378          0.0 -410.749309

         price  demand_std  generation_std  generation_mean  demand_mean
25912     9.84       220.0            260.0             1300         1100
1235752   9.84       220.0            260.0             1300         1100
2445592   9.84       220.0            260.0             1300         1100
3655432   9.84       220.0            260.0             1300         1100
4865272   9.84       220.0            260.0             1300         1100
6075112   9.84       220.0            260.0             1300         1100  ,          id       time       demand   generat
ion  consumption  net_energy  \
25911     1 2020-06-01  1038.681500  1020.879060    17.802440    0.000000
1235751   2 2020-06-01   842.573368  1300.000000     0.000000 -457.426632
2445591   3 2020-06-01   962.018220   910.318008    51.700212    0.000000
3655431   4 2020-06-01  1033.504748  1300.000000     0.000000 -266.495252
4865271   5 2020-06-01  1004.271183  1300.000000     0.000000 -295.728817
6075111   6 2020-06-01   910.162557   948.475309     0.000000  -38.312753

         price  demand_std  generation_std  generation_mean  demand_mean
25911    11.53       220.0            260.0             1300         1100
1235751  11.53       220.0            260.0             1300         1100
2445591  11.53       220.0            260.0             1300         1100
3655431  11.53       220.0            260.0             1300         1100
4865271  11.53       220.0            260.0             1300         1100
6075111  11.53       220.0            260.0             1300         1100  ,          id       time       demand   generat
ion  consumption  net_energy  \
25910     1 2020-07-01  1165.090456  1300.000000          0.0 -134.909544
```

```
1235750   2 2020-07-01   772.752521  1300.000000          0.0 -527.247479
2445590   3 2020-07-01   735.696415  1300.000000          0.0 -564.303585
3655430   4 2020-07-01  1017.990935  1025.455744          0.0   -7.464809
4865270   5 2020-07-01  1117.175491  1300.000000          0.0 -182.824509
6075110   6 2020-07-01  1185.948752  1300.000000          0.0 -114.051248

         price  demand_std  generation_std  generation_mean  demand_mean
25910    11.71       220.0           260.0             1300         1100
1235750  11.71       220.0           260.0             1300         1100
2445590  11.71       220.0           260.0             1300         1100
3655430  11.71       220.0           260.0             1300         1100
4865270  11.71       220.0           260.0             1300         1100
6075110  11.71       220.0           260.0             1300         1100 ,           id       time       demand   generat
ion  consumption  net_energy  \
25909     1 2020-08-01  1236.711727  1300.000000     0.000000  -63.288273
1235749   2 2020-08-01  1320.799390   975.817933   344.981457    0.000000
2445589   3 2020-08-01  1162.721526  1013.844404   148.877122    0.000000
3655429   4 2020-08-01   738.076595  1300.000000     0.000000 -561.923405
4865269   5 2020-08-01  1261.774214  1107.427719   154.346494    0.000000
6075109   6 2020-08-01   990.265411  1300.000000     0.000000 -309.734589

         price  demand_std  generation_std  generation_mean  demand_mean
25909    11.61       220.0           260.0             1300         1100
1235749  11.61       220.0           260.0             1300         1100
2445589  11.61       220.0           260.0             1300         1100
3655429  11.61       220.0           260.0             1300         1100
4865269  11.61       220.0           260.0             1300         1100
6075109  11.61       220.0           260.0             1300         1100 ,           id       time       demand   generat
ion  consumption  net_energy  \
25908     1 2020-09-01   891.556610  1298.843324     0.000000 -407.286714
1235748   2 2020-09-01   455.297268  1300.000000     0.000000 -844.702732
2445588   3 2020-09-01   796.070701  1098.566634     0.000000 -302.495932
3655428   4 2020-09-01   860.703025  1297.552831     0.000000 -436.849806
4865268   5 2020-09-01  1112.831083  1109.574168     3.256915    0.000000
6075108   6 2020-09-01   878.165805   979.311590     0.000000 -101.145784

         price  demand_std  generation_std  generation_mean  demand_mean
25908    11.97       220.0           260.0             1300         1100
1235748  11.97       220.0           260.0             1300         1100
2445588  11.97       220.0           260.0             1300         1100
3655428  11.97       220.0           260.0             1300         1100
4865268  11.97       220.0           260.0             1300         1100
6075108  11.97       220.0           260.0             1300         1100 ,           id       time       demand   generat
ion  consumption  net_energy  \
25907     1 2020-10-01  1130.832481   783.993174   346.839307    0.000000
1235747   2 2020-10-01  1235.486678  1300.000000     0.000000  -64.513322
2445587   3 2020-10-01   859.968470  1300.000000     0.000000 -440.031530
3655427   4 2020-10-01   805.362242  1252.774736     0.000000 -447.412494
4865267   5 2020-10-01   313.363713   744.759198     0.000000 -431.395485
6075107   6 2020-10-01   604.781916   978.225231     0.000000 -373.443315
```

```
         price  demand_std  generation_std  generation_mean  demand_mean
25907    11.71       220.0           260.0             1300         1100
1235747  11.71       220.0           260.0             1300         1100
2445587  11.71       220.0           260.0             1300         1100
3655427  11.71       220.0           260.0             1300         1100
4865267  11.71       220.0           260.0             1300         1100
6075107  11.71       220.0           260.0             1300         1100  ,           id       time      demand   generati
on  consumption  net_energy  \
25906      1 2020-11-01  806.965579  1103.241179          0.0 -296.275600
1235746    2 2020-11-01  790.826705   979.045000          0.0 -188.218295
2445586    3 2020-11-01  722.210544  1152.441576          0.0 -430.231032
3655426    4 2020-11-01  779.390564  1012.187101          0.0 -232.796536
4865266    5 2020-11-01  883.079338  1033.644700          0.0 -150.565363
6075106    6 2020-11-01  748.715813   815.914776          0.0  -67.198963

         price  demand_std  generation_std  generation_mean  demand_mean
25906     12.0       220.0           260.0             1300         1100
1235746   12.0       220.0           260.0             1300         1100
2445586   12.0       220.0           260.0             1300         1100
3655426   12.0       220.0           260.0             1300         1100
4865266   12.0       220.0           260.0             1300         1100
6075106   12.0       220.0           260.0             1300         1100  ,           id       time      demand   generati
on  consumption  net_energy  \
25905      1 2020-12-01  788.236454  1300.000000     0.000000 -511.763546
1235745    2 2020-12-01  750.823892   463.303099   287.520793    0.000000
2445585    3 2020-12-01  293.502601  1013.254735     0.000000 -719.752134
3655425    4 2020-12-01  484.845007  1300.000000     0.000000 -815.154993
4865265    5 2020-12-01  601.377546  1133.709775     0.000000 -532.332230
6075105    6 2020-12-01  524.675582   947.526710     0.000000 -422.851128

         price  demand_std  generation_std  generation_mean  demand_mean
25905    11.86       220.0           260.0             1300         1100
1235745  11.86       220.0           260.0             1300         1100
2445585  11.86       220.0           260.0             1300         1100
3655425  11.86       220.0           260.0             1300         1100
4865265  11.86       220.0           260.0             1300         1100
6075105  11.86       220.0           260.0             1300         1100  ]}]
```

Apply Payments and Charges to Prosumers Using NRG-X-Change

The scalar for the nrg payment would need to be self-tracking. A sweep of possible a values was done to track when any of the value goes past the "inf" value.

In [119…

```
####################################################################
# Calculate and update the dataframe with nrg payments at time t
####################################################################
def get_nrg_payments(df_by_t,max_price,min_price,a=0,n=1):
    for t in df_by_t:
        tc = t['consumption'].sum()
```

```python
        tp = abs(t['net_energy']).sum()
        t['nrg_v'] = t['net_energy'].apply(lambda x: g(price=max_price,p=abs(x),tc=tc,tp=tp,n=n,a=a) if abs(x) > 0 else (
        t['prosumer_debit'] = t['consumption'].apply(lambda x: -(h(price=min_price,c=x,tc=tc,tp=tp)) if abs(x) > 0 else (
        t['prosumer_revenue'] = t['nrg_v'] + t['prosumer_debit']
    return df_by_t
```

In [120…

```python
def apply_nrg_payments(trials,n=1):
    for trial in trials:
        ####################################################################
        # Apply NRG-X change payments for each time 't'
        ####################################################################
        prosumers_n = trial['prosumers_n']
        prosumers_n_t =trial['prosumers_n_t']
        # Set historical pricing limits for NRG
        max_price = prosumers_n['price'].max()
        min_price = prosumers_n['price'].min()
        # Applying payments to each record
        prev_std = 0
        a_scaled = 0
        # Scaling the 'a' until payments are no longer sensitive
        for a in np.arange(start=10000,stop=(10000*1000),step=10000):
            prosumers_n_t = get_nrg_payments(prosumers_n_t,max_price,min_price,a=a,n=n)
            df = pd.concat(prosumers_n_t)
            std = df['nrg_v'].std()
            pct_c = ((std - prev_std)/std)
            prev_std = std
            if  pct_c < 0.001:
                a_scaled = a
                break
        # store the scaled a value for this trail given N proumers
        trial['a_scaled'] = a_scaled
    return trials
```

In [122…

```python
####################################################################
# Calcualate the coalitional Pay out at each time step of t
####################################################################
def get_coalitional_payments(df_by_t,max_price,min_price,a=0,n=1):
    for t in df_by_t:
        tc = t['consumption'].sum()
        tp = abs(t['net_energy']).sum()
        # identify number of prosumers at every time step of t,
        # that have provided net_energy > 0
        ids_net_energy_given = t[abs(t['net_energy']) > 0]['id']
        N_c=len(ids_net_energy_given)
```

```
        # sum the absolute value of all the net energy given
        # by the indentified IDs
        # abs(t.loc[t['id'].isin(ids_net_energy_given)]['net_energy']).sum()
        t['coalition_v'] = 0
        # if number of members is 1 set it to the characteristic value
        if N_c == 1:
            t['coalition_v'] = t['nrg_v']
        # if number of members is greather than 1 calc shapley
        if N_c > 1:
            List = ids_net_energy_given
            # get a power set with combinatorial elements as a list of lists
            PS = [list(j) for i in range(len(List)) for j in combinations(List, i+1)]
            char_vals = []
            # locate all ids in time step within the powerset, (factorial), and sum up
            for nn in PS:
                contribution = abs(t.loc[t['id'].isin(nn)]['net_energy']).sum()
                char_func_val = g(price=max_price,p=contribution,tc=tc,tp=tp,n=n,a=a)
                char_vals.append(char_func_val)
            # use the number of members in the coalition and the
            # characteristic values to calc shapley
            shapleys = get_shapley(N_c,char_vals)
            # add the individual shappley value to each of the id's that generated energy
            for i in range(N_c):
                t.loc[t.index[ids_net_energy_given.values[i]-1], 'coalition_v'] = shapleys[i]
    return df_by_t
```

In [123…

```
def apply_coalitional_payments(trials,n=1):
    for trial in trials:
        ####################################################################
        # Apply Coalitional payments for each time 't'
        ####################################################################
        prosumers_n = trial['prosumers_n']
        prosumers_n_t =trial['prosumers_n_t']
        a =trial['a_scaled']
        # Set historical pricing limits for NRG
        max_price = prosumers_n['price'].max()
        min_price = prosumers_n['price'].min()
        prosumers_n_t = get_coalitional_payments(prosumers_n_t,max_price,min_price,a=a,n=n)
    return trials
```

In [124…

```
####################################################################
# NRG & Coalitional payment processing for each trial(N) prosumers
####################################################################
# Y=X^(1) , linear
```

```python
trials = apply_nrg_payments(trials=trials)
trials = apply_coalitional_payments(trials=trials)
# Visualize/Sample of Data
for trial in trials:
    print(f"\nN={trial['N']}")
    print(trial['prosumers_n_t'][0])
```

```
N=2
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0

N=3
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0

N=4
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
```

```
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0

N=5
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100
4865279  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0
4865279      0       -2.428493         -2.428493            0

N=6
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119   6 2019-10-01   693.494570  588.374290   105.120280         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100
4865279  11.66       220.0           260.0             1300         1100
6075119  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0
4865279      0       -2.428493         -2.428493            0
6075119      0      -10.343836        -10.343836            0
```

```
N=7
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119   6 2019-10-01   693.494570  588.374290   105.120280         0.0
7284959   7 2019-10-01   974.670409  725.090088   249.580321         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100
4865279  11.66       220.0           260.0             1300         1100
6075119  11.66       220.0           260.0             1300         1100
7284959  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0
4865279      0       -2.428493         -2.428493            0
6075119      0      -10.343836        -10.343836            0
7284959      0      -24.558704        -24.558704            0

N=8
         id       time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119   6 2019-10-01   693.494570  588.374290   105.120280         0.0
7284959   7 2019-10-01   974.670409  725.090088   249.580321         0.0
8494799   8 2019-10-01  1040.271045  894.004707   146.266338         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100
4865279  11.66       220.0           260.0             1300         1100
6075119  11.66       220.0           260.0             1300         1100
7284959  11.66       220.0           260.0             1300         1100
8494799  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
```

```
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0
4865279      0       -2.428493         -2.428493            0
6075119      0      -10.343836        -10.343836            0
7284959      0      -24.558704        -24.558704            0
8494799      0      -14.392608        -14.392608            0

N=9
         id        time       demand  generation  consumption  net_energy  \
25919     1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759   2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599   3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439   4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279   5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119   6 2019-10-01   693.494570  588.374290   105.120280         0.0
7284959   7 2019-10-01   974.670409  725.090088   249.580321         0.0
8494799   8 2019-10-01  1040.271045  894.004707   146.266338         0.0
9704639   9 2019-10-01   948.540002  695.130495   253.409507         0.0

         price  demand_std  generation_std  generation_mean  demand_mean  \
25919    11.66       220.0           260.0             1300         1100
1235759  11.66       220.0           260.0             1300         1100
2445599  11.66       220.0           260.0             1300         1100
3655439  11.66       220.0           260.0             1300         1100
4865279  11.66       220.0           260.0             1300         1100
6075119  11.66       220.0           260.0             1300         1100
7284959  11.66       220.0           260.0             1300         1100
8494799  11.66       220.0           260.0             1300         1100
9704639  11.66       220.0           260.0             1300         1100

         nrg_v  prosumer_debit  prosumer_revenue  coalition_v
25919        0       -5.726267         -5.726267            0
1235759      0      -71.538637        -71.538637            0
2445599      0      -55.132233        -55.132233            0
3655439      0      -12.445495        -12.445495            0
4865279      0       -2.428493         -2.428493            0
6075119      0      -10.343836        -10.343836            0
7284959      0      -24.558704        -24.558704            0
8494799      0      -14.392608        -14.392608            0
9704639      0      -24.935496        -24.935496            0

N=10
          id        time       demand  generation  consumption  net_energy  \
25919      1 2019-10-01   757.174284  698.980517    58.193767         0.0
1235759    2 2019-10-01  1230.621807  503.603134   727.018673         0.0
2445599    3 2019-10-01  1241.295996  681.009070   560.286926         0.0
3655439    4 2019-10-01   843.911097  717.432484   126.478613         0.0
4865279    5 2019-10-01   672.649709  647.969906    24.679803         0.0
6075119    6 2019-10-01   693.494570  588.374290   105.120280         0.0
7284959    7 2019-10-01   974.670409  725.090088   249.580321         0.0
```

8494799	8	2019-10-01	1040.271045	894.004707	146.266338	0.0
9704639	9	2019-10-01	948.540002	695.130495	253.409507	0.0
10914479	10	2019-10-01	766.927736	385.539889	381.387847	0.0

	price	demand_std	generation_std	generation_mean	demand_mean \
25919	11.66	220.0	260.0	1300	1100
1235759	11.66	220.0	260.0	1300	1100
2445599	11.66	220.0	260.0	1300	1100
3655439	11.66	220.0	260.0	1300	1100
4865279	11.66	220.0	260.0	1300	1100
6075119	11.66	220.0	260.0	1300	1100
7284959	11.66	220.0	260.0	1300	1100
8494799	11.66	220.0	260.0	1300	1100
9704639	11.66	220.0	260.0	1300	1100
10914479	11.66	220.0	260.0	1300	1100

	nrg_v	prosumer_debit	prosumer_revenue	coalition_v
25919	0	-5.726267	-5.726267	0
1235759	0	-71.538637	-71.538637	0
2445599	0	-55.132233	-55.132233	0
3655439	0	-12.445495	-12.445495	0
4865279	0	-2.428493	-2.428493	0
6075119	0	-10.343836	-10.343836	0
7284959	0	-24.558704	-24.558704	0
8494799	0	-14.392608	-14.392608	0
9704639	0	-24.935496	-24.935496	0
10914479	0	-37.528564	-37.528564	0

Example of Shapley value Calculation for N Prosumers

We consider a comunity of solar prosumers $P = 1, 2, 3..N$, who agree to form a coalition and produce energy. The number of possible coalitions are 2^n and the number of ways to build the grand coalition is $N!$.

In [125…

```
####################################################################
# Consider Convex , Linear & Concave characteristic functions
####################################################################
# Adjust the value of 'X^n' used in NRG payment function
X_n= [-2,-0.5,1,0.5,2]
x_n_trials = []
for n in X_n :
    trials = apply_nrg_payments(trials=trials,n=n)
    trials = apply_coalitional_payments(trials=trials,n=n)
    x_n_trials.append({'X_n':n,'trials':trials})
```

Comparison of Shapley value for Convex, Linear and Concave characteristic functions

Energy produced by individual Prosumers

Shapley value calculation with and without coalition for different characteristic functions

In [126…

```
table = {}
for x_n_trial in x_n_trials :
    x_n = x_n_trial['X_n']
    col_title = f'Y=X^({x_n})'
    trials = x_n_trial['trials']
    rows=[]
    for trial in trials:
        df = pd.concat(trial['prosumers_n_t'])
        avg_wo_co = df['nrg_v'].mean()
        avg_w_co = df['coalition_v'].mean()
        rows.append({"X_n":x_n, "Number of Prosumers":trial['N'],"With Coalition":avg_w_co,"Without Coalition":avg_w_co}
    table[col_title] = rows

lines = []
for header in table.keys():
    line=""
    for i in range(len(table[header])):
        for row in table[header]:
            print(row)


print(lines)
```

```
.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 1, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 1, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 1, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 1, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 1, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 1, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 1, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 1, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 1, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 1, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
```

{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}

{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 0.5, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 0.5, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 0.5, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 0.5, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 0.5, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 0.5, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 0.5, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 0.5, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 0.5, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}

```
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
{'X_n': 2, 'Number of Prosumers': 2, 'With Coalition': 16668.612473211433, 'Without Coalition': 16668.612473211433}
{'X_n': 2, 'Number of Prosumers': 3, 'With Coalition': 22587.925924124862, 'Without Coalition': 22587.925924124862}
{'X_n': 2, 'Number of Prosumers': 4, 'With Coalition': 23514.918469542215, 'Without Coalition': 23514.918469542215}
{'X_n': 2, 'Number of Prosumers': 5, 'With Coalition': 24173.35492380917, 'Without Coalition': 24173.35492380917}
{'X_n': 2, 'Number of Prosumers': 6, 'With Coalition': 21073.363980216473, 'Without Coalition': 21073.363980216473}
{'X_n': 2, 'Number of Prosumers': 7, 'With Coalition': 15214.08840843334, 'Without Coalition': 15214.08840843334}
{'X_n': 2, 'Number of Prosumers': 8, 'With Coalition': 13931.69348528232, 'Without Coalition': 13931.69348528232}
{'X_n': 2, 'Number of Prosumers': 9, 'With Coalition': 9774.919591173862, 'Without Coalition': 9774.919591173862}
{'X_n': 2, 'Number of Prosumers': 10, 'With Coalition': 5716.662792516543, 'Without Coalition': 5716.662792516543}
[]
```

In [ ]:

In [ ]: