Analysis

In [1]:

```
import pandas as pd,\
  seaborn as sns,\
  matplotlib.pyplot as pypl,\
  numpy as np

#  plotly.express as px,\
```

In [2]:

```
pypl.style.use('bmh')
```

In [3]:

```
df_historical_pulls = pd.read_csv(
  '../dba5101_gp3/data/pulls_output.csv',
  header='infer',
  index_col=False
)
```

In [4]:

```
## add_uniform_algo

df_uniform_pulls_input = pd.read_json(
  '../dba5101_gp3/data/uniform_pulls_1.json',
  orient='records',
  lines=True
)
```

In [5]:

```
## add_greed_algo

df_greedy_pulls_input = pd.read_json(
  '../dba5101_gp3/data/greedy_pull_3.json',
  orient='records',
  lines=True
)
```

In [6]:

```
df_greedy_pulls_input['index_pull'] = df_greedy_pulls_input['Pull'].rank() - 1
df_uniform_pulls_input['index_pull'] = df_uniform_pulls_input['Pull'].rank() - 1
```

In [7]:

```
df_greedy_pulls = df_greedy_pulls_input[df_greedy_pulls_input['index_pull'] >= 1]
```

In [8]:

```
df_uniform_pulls =  df_uniform_pulls_input[df_uniform_pulls_input['index_pull'] >= (24*3 )]
```

In [9]:

```
df_ucb1_exploit = df_historical_pulls[df_historical_pulls['global_pull'] >= 24 ]
```

In [ ]:

```

```

In [10]:

```
df_uniform_pulls['local_pull'] = df_uniform_pulls['Pull'].rank() - 1
```

```
/var/folders/88/st0km2xx06b1nm8trg753nqm0000gn/T/ipykernel_19265/1785843218.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
ing-a-view-versus-a-copy
  df_uniform_pulls['local_pull'] = df_uniform_pulls['Pull'].rank() - 1
```

In [ ]:

```

```

In [11]:

```
df_greedy_pulls['local_pull'] = df_greedy_pulls['Pull'].rank() - 1
```

```
/var/folders/88/st0km2xx06b1nm8trg753nqm0000gn/T/ipykernel_19265/3050684879.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
ing-a-view-versus-a-copy
  df_greedy_pulls['local_pull'] = df_greedy_pulls['Pull'].rank() - 1
```

In [12]:

```
df_greedy_pulls['rolling_sum'] = df_greedy_pulls['Reward'].cumsum()

df_uniform_pulls['rolling_sum'] = df_uniform_pulls['Reward'].cumsum()

df_historical_pulls['rolling_sum'] = df_historical_pulls['arm_reward'].cumsum()
```

```
/var/folders/88/st0km2xx06b1nm8trg753nqm0000gn/T/ipykernel_19265/4265629572.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
ing-a-view-versus-a-copy
  df_greedy_pulls['rolling_sum'] = df_greedy_pulls['Reward'].cumsum()
/var/folders/88/st0km2xx06b1nm8trg753nqm0000gn/T/ipykernel_19265/4265629572.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
ing-a-view-versus-a-copy
  df_uniform_pulls['rolling_sum'] = df_uniform_pulls['Reward'].cumsum()
```

In [13]:

```
df_ucb1_exploit['rolling_sum'] = df_ucb1_exploit['arm_reward'].cumsum()
```

```
/var/folders/88/st0km2xx06b1nm8trg753nqm0000gn/T/ipykernel_19265/3231549739.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#return
ing-a-view-versus-a-copy
  df_ucb1_exploit['rolling_sum'] = df_ucb1_exploit['arm_reward'].cumsum()
```

In [14]:

```
df_comparative = df_ucb1_exploit.merge(
  right=df_uniform_pulls,
  left_on ='global_pull',
  right_on = 'local_pull',
  how='left' ,
  suffixes=('_ucb1', '_uniform')
)

df_comparative = df_comparative.merge(
  right=df_greedy_pulls,
  left_on ='global_pull',
  right_on = 'local_pull',
  how='left' ,
  suffixes=('', '_greedy')
)
```

In [15]:

```
df_comparative
```

Out[15]:

	global_pull	arm	arm_pull	arm_reward	next_chosen_arm	rolling_sum_ucb1	Arm	NetReward	Pull	Reward	index_pull	local_pu
0	24.0	3.0	1.0	17.0	NaN	17.0	0	194485	8959	27	96.0	24.
1	25.0	8.0	1.0	24.0	NaN	41.0	1	194501	8960	16	97.0	25.
2	26.0	8.0	2.0	19.0	NaN	60.0	2	194509	8961	8	98.0	26.
3	27.0	0.0	1.0	18.0	NaN	78.0	3	194533	8962	24	99.0	27.
4	28.0	8.0	3.0	20.0	NaN	98.0	4	194541	8963	8	100.0	28.
...	...	...	...	...	...	...	...	...	...	...	...	.
1995	2019.0	0.0	1821.0	22.0	NaN	48414.0	0	242273	10954	22	2091.0	2019.
1996	2020.0	0.0	1822.0	27.0	NaN	48441.0	0	242295	10955	22	2092.0	2020.
1997	2021.0	0.0	1823.0	31.0	NaN	48472.0	0	242315	10956	20	2093.0	2021.
1998	2022.0	0.0	1824.0	20.0	NaN	48492.0	0	242337	10957	22	2094.0	2022.
1999	2023.0	0.0	1825.0	29.0	NaN	48521.0	0	242366	10958	29	2095.0	2023.

2000 rows × 20 columns

In [19]:

```
df_comparative [df_comparative['global_pull'] <= 2000 ] [
  ['global_pull','rolling_sum_ucb1','rolling_sum_uniform','rolling_sum']
].rename(columns={'rolling_sum':'greedy_epsilon','rolling_sum_uniform':'uniform','rolling_sum_ucb1':'ucb1'}) .plot
  kind ='line',
  x = 'global_pull'
)



pypl.title("Comparision of UCB1 +  Uniform MAB + Greedy Episilon - Exploitation Phase Cumulative Reward")

pypl.gcf().set_size_inches(17, 10)

pypl.ylabel('Cumulative Reward')
pypl.xlabel('Pull')

#pypl.show()

pypl.savefig(fname='./model_cumulative_sum.png')

pypl.close('all')
```

In [ ]:

```

```

In [ ]:

```

```

In [4]:

```
# Add in historical pulls over all time

df_total_pulls = df_historical_pulls.groupby(['arm'])['global_pull'].count().reset_index()

df_total_pulls.rename(columns={"global_pull": "total_pulls"},inplace=True)

df_historical_pulls =  df_historical_pulls.merge(right =df_total_pulls , how ='left', on ='arm' )
```

In [5]:

```
df_arms = pd.DataFrame( {'arm': df_historical_pulls['arm'].unique().tolist()   } )

df_pulls = pd.DataFrame( {'global_pull': df_historical_pulls['global_pull'].unique().tolist()   } )

df_cartesian = df_arms.merge(right=df_pulls, how='cross')

df_cartesian= df_cartesian.merge( right=df_historical_pulls, on=['global_pull','arm'] ,how='left'  )

df_cartesian['arm_reward'].fillna(value=0,inplace=True)

df_cartesian['cumulative_reward'] = df_cartesian.groupby(['arm'])['arm_reward'].cumsum(skipna=True)

df_cartesian['arm_pull'].ffill(inplace=True)
```

In [6]:

```
df_cartesian = df_cartesian[df_cartesian['global_pull'] >= 24]
```

In [7]:

```
#df_cartesian['ucb_index'] =
df_cartesian['contemporary_linucb'] = (df_cartesian['cumulative_reward'] / (df_cartesian['arm_pull'] + 1) )  + np.
  (2 * np.log( df_cartesian['global_pull']  ) ) / (df_cartesian['arm_pull'] + 1)
)

df_cartesian['ucb_mean'] = (df_cartesian['cumulative_reward'] / (df_cartesian['arm_pull'] + 1) )

df_cartesian['penalty'] = np.sqrt(
  (2 * np.log( df_cartesian['global_pull']  ) ) / (df_cartesian['arm_pull'] + 1)
)
```

In [8]:

```
sns.lineplot(
  x='global_pull',
  y='contemporary_linucb',
  hue='arm',
  data=df_cartesian[
    (df_cartesian['global_pull'] <= 220)
    & (df_cartesian['global_pull'] >= 24)
    & (df_cartesian['arm'].isin([0,3,8,23]) )
  ]
)

pypl.title("UCB Over Time")

pypl.gcf().set_size_inches(17, 10)

pypl.ylabel('UCB Index')
pypl.xlabel('Pull')

#pypl.show()

pypl.savefig(fname='./ucb_exploitation_line.png')

pypl.close('all')
```

In [9]:

```
df_cartesian[
  df_cartesian['global_pull'] == 2020
].sort_values(
  by=['contemporary_linucb'],
  ascending=False
)[['arm','arm_pull','cumulative_reward','contemporary_linucb','ucb_mean','penalty']].head(n=5)
```

Out[9]:

	arm	arm_pull	cumulative_reward	contemporary_linucb	ucb_mean	penalty
2020	0.0	1822.0	44651.0	24.584520	24.493143	0.091377
16188	7.0	0.0	19.0	22.901500	19.000000	3.901500
42500	20.0	0.0	19.0	22.901500	19.000000	3.901500
20236	9.0	0.0	19.0	22.901500	19.000000	3.901500
36428	17.0	1.0	40.0	22.758777	20.000000	2.758777

In [10]:

```
sns.lineplot(
  x='global_pull',
  y='contemporary_linucb',
  hue='arm',
#  color='b',
  data=df_cartesian[
    (df_cartesian['global_pull'] >= 24)
    & (df_cartesian['arm'].isin([0,7,17]) )
  ]
)

pypl.title("UCB Over Time - Candidates For Next Arm, Due To Uncertainity")

pypl.gcf().set_size_inches(17, 10)

pypl.ylabel('UCB Index')
pypl.xlabel('Pull')

#pypl.show()
pypl.savefig(fname='./ucb_future_line.png')

pypl.close('all')
```

In [11]:

```
f, ax = pypl.subplots(figsize=(17, 10))

sns.lineplot(
  x='global_pull',
  y='ucb_mean',
  hue='arm',
#  color='b',
  data=df_cartesian[
    (df_cartesian['global_pull'] >= 24)
    & (df_cartesian['arm'].isin([0,7,17]) )
  ]
)

sns.lineplot(
  x='global_pull',
  y='penalty',
  hue='arm',
  data=df_cartesian[
    (df_cartesian['global_pull'] >= 24)
    & (df_cartesian['arm'].isin([0,7,17]) )
  ]
)


pypl.title("Arm Mean & Penalty Term Over Time")

pypl.ylabel('UCB Index Component Quantity')
pypl.xlabel('Pull')


#pypl.show()
pypl.savefig(fname='./ucb_penalty_line.png')

pypl.close('all')
```

In [12]:

```
#Create combo chart
fig, ax1 = pypl.subplots(figsize=(17,10))


#bar plot creation
ax1.set_title('UCB Index & Reward - Exploitation Phase', fontsize=16)
ax1.set_xlabel('Pull')
ax1.set_ylabel('Reward')
ax1 = sns.scatterplot(x='global_pull', y='arm_reward', hue='arm', data = df_historical_pulls[
(df_historical_pulls['global_pull'] <= 250)
& (df_historical_pulls['global_pull'] >= 24)
#& (df_historical_pulls['arm'].isin([0,3,8,23]) )
])
ax1.tick_params(axis='y')
#specify we want to share the same x-axis
ax2 = ax1.twinx()

#line plot creation
ax2.set_ylabel('UCB Index')
ax2 = sns.lineplot(
  x='global_pull',
  y='contemporary_linucb',
  hue='arm',
  data=df_cartesian[
    (df_cartesian['global_pull'] <= 250)
    & (df_cartesian['global_pull'] >= 24)
#    & (df_cartesian['arm'].isin([0,3,8,23]) )
  ]
)


#show plot
#pypl.show()
pypl.savefig(fname='./ucb_index_reward_line.png')

pypl.close('all')
```

In [48]:

```
#sns.scatterplot( x='global_pull', y='arm_reward',   hue='arm',
#  data = df_historical_pulls[   (df_historical_pulls['global_pull'] <= 250)
#    & (df_historical_pulls['global_pull'] >= 24) ])

#pypl.gcf().set_size_inches(17, 10)

#pypl.show()

#pypl.close('all')
```

In [47]:

```
#sns.displot(  data=df_historical_pulls[ df_historical_pulls['global_pull'] >= 24] ,
#  x="arm_reward", hue="arm", kind="kde")

#pypl.gcf().set_size_inches(17, 10)

#pypl.show()

#pypl.close('all')
```

In [13]:

```
sns.violinplot(
  cut=0,
  inner='quartiles',
  data=df_historical_pulls[ (df_historical_pulls['global_pull'] >= 24) & (df_historical_pulls[ 'total_pulls'] > 2
  x='arm',
  y='arm_reward',
)

pypl.gcf().set_size_inches(17, 10)
pypl.ylim([0,45])

pypl.title("Arm Reward Violinplot, Exploitation Phase - With Quartiles")
pypl.ylabel('Reward')
pypl.xlabel('Arm')

#pypl.show()
pypl.savefig(fname='./reward_violin.png')

pypl.close('all')
```