Baselines_notebook

June 21, 2016

0.0.1 INITIALIZATION

```
In [13]: from ipywidgets import interact

         from baselines_plot import frequency_plot, schedule, scatter_plot, time_table
         from util import *
         import time

         %matplotlib inline

         bc = KTIBruggCables()
```

0.0.2 Get Opportunities and Batches

```
In [14]: opportunities = bc.get_opportunities()
         batches = bc.get_batches()
```

0.0.3 BASELINE GENERATION

```
In [15]: start = time.clock()
         baselines = bc.get_baselines()
         print(len(baselines), 'Baselines Generated in {0:.3f} sec.\n'
                                                              .format(time.clock()-start))
```

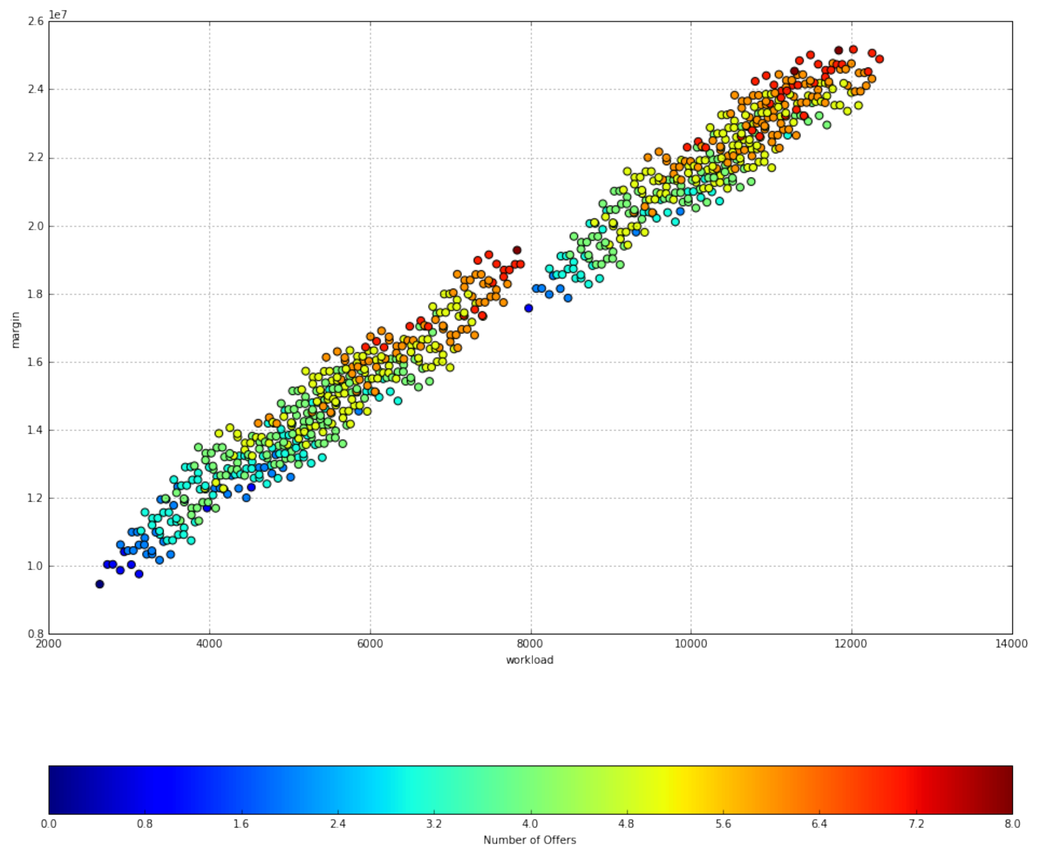
```
896 Baselines Generated in 1.790 sec.
```

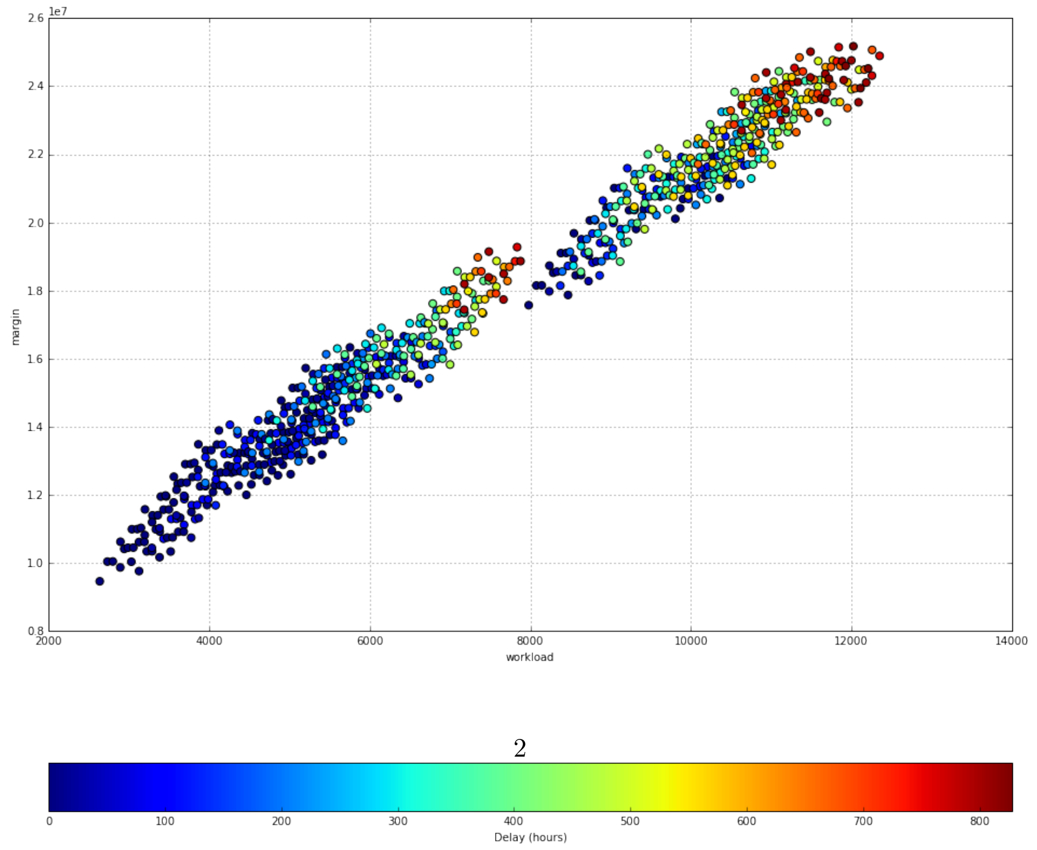
0.0.4 FITNESS: compute the fitness for the baselines

```
In [16]: start = time.clock()
         fitness_b = bc.compute_fitness(baselines)
         print('Fitness Computed in {0:.3f} sec.\n'.format(time.clock()-start))
```

```
Fitness Computed in 38.315 sec.
```

```
In [17]: def plot_scatter_dynam(delay):
             fitness_ = fitness_b[fitness_b.delay <= delay]
             scatter_plot(fitness_.baselines.tolist(), 'workload', 'margin', df = fitness_)

         interact(plot_scatter_dynam, delay=(0, fitness_b.delay.max() , 10))
```

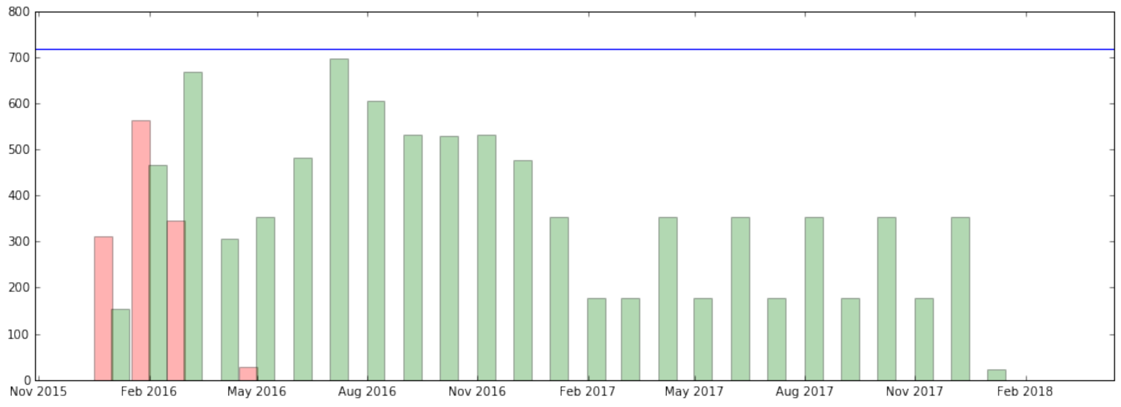

1e7
margin
workload
Number of Offers
0.0
0.8
1.6
2.4
3.2
4.0
4.8
5.6
6.4
7.2
8.0

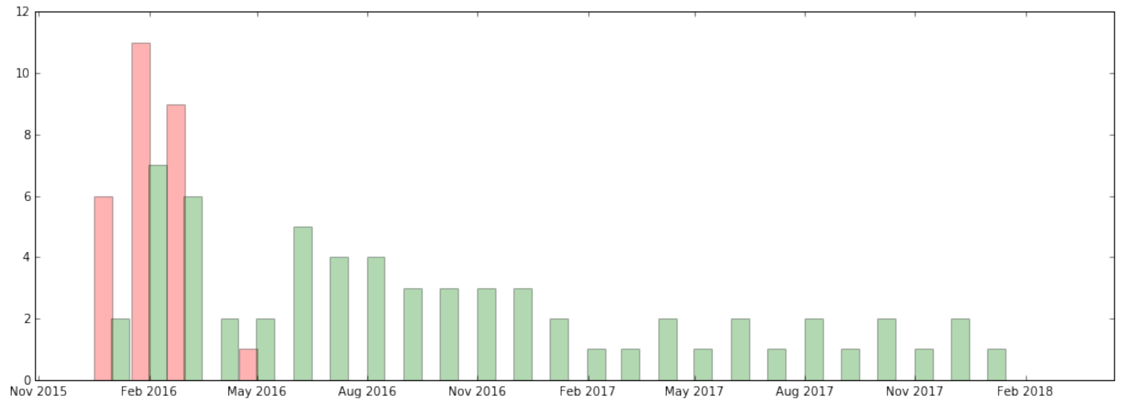

1e7
margin
workload
Delay (hours)
0
100
200
300
400
500
600
700
800

```
Out[17]: <function __main__.plot_scatter_dynam>
```

0.0.5 SELECTION: preliminary selection rule for the best baseline

```
In [6]: fitness_ = fitness_b[fitness_b.delay <= 0]
        fitness_ = fitness_.sort_values(['margin','workload','number'],ascending=[False,False,True])
        baseline = fitness_.iloc[0].baselines

        schedule(baseline)
```





0.0.6 FILLING: combine the fillers to get the schedules

```
In [7]: start = time.clock()
        schedules = bc.filling(baseline)
        print(len(schedules), 'Schedules Filled in {0:.3f} sec.\n'
                                                    .format(time.clock()-start))

Number of Fillers: 1
Number of Fillers: 0
Number of Fillers: 2
Number of Fillers: 6
Number of Fillers: 7
Number of Fillers: 7
Number of Fillers: 7
100 Schedules Filled in 23.793 sec.
```

0.0.7 FITNESS: compute the fitness for the FILLED SCHEDULES

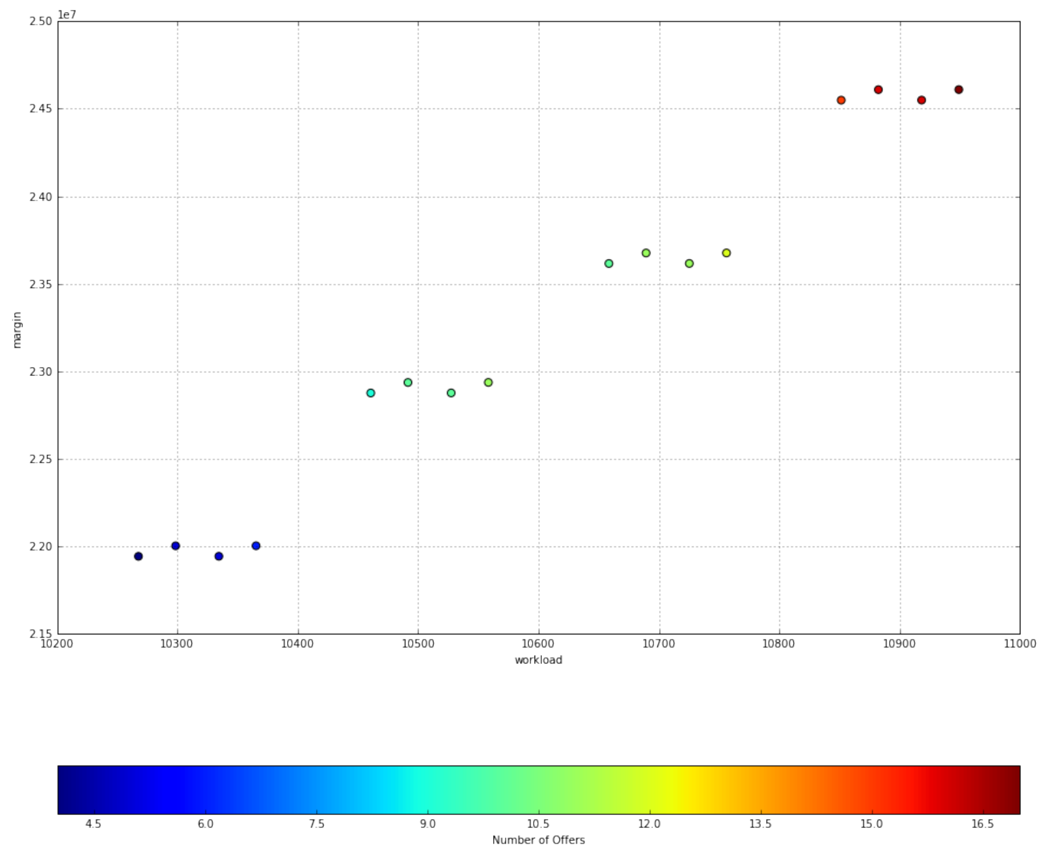
```
In [8]: start = time.clock()
        fitness = bc.compute_fitness(schedules, opportunities=opportunities,\
                                                                  batches = batches)

        print('Fitness Computed in {0:.3f} sec.\n'.format(time.clock()-start))
```

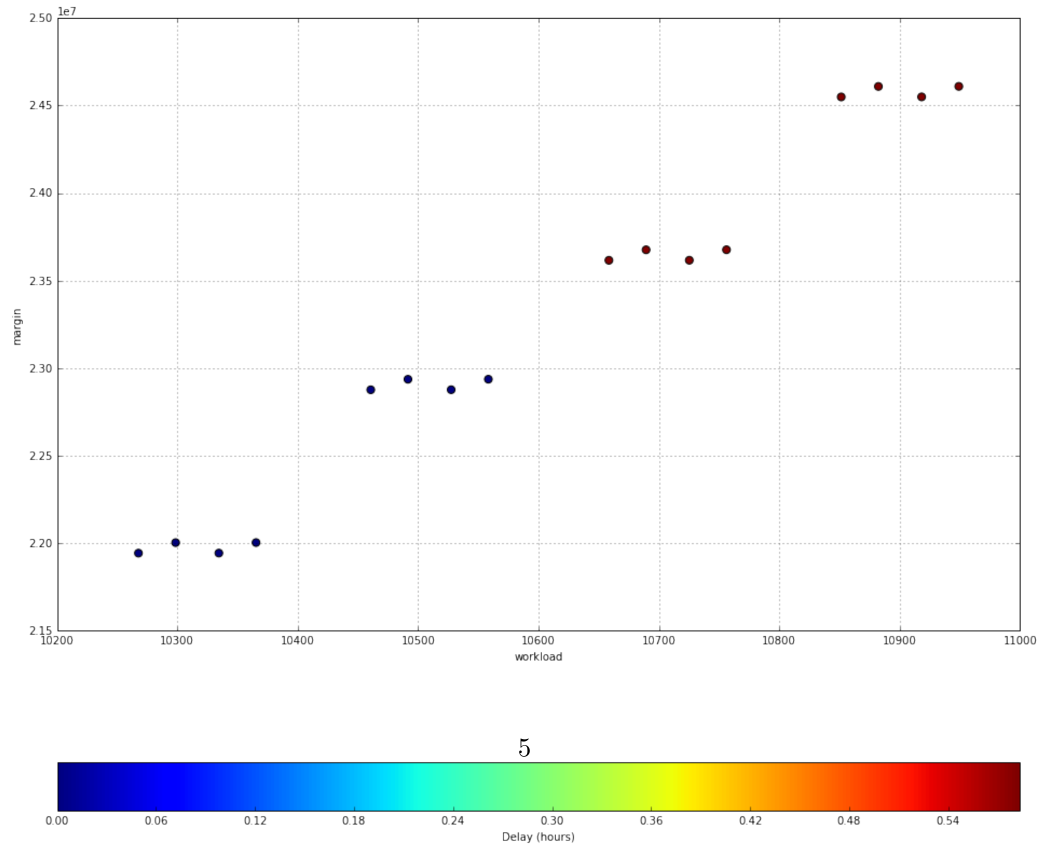
```
Fitness Computed in 5.169 sec.
```

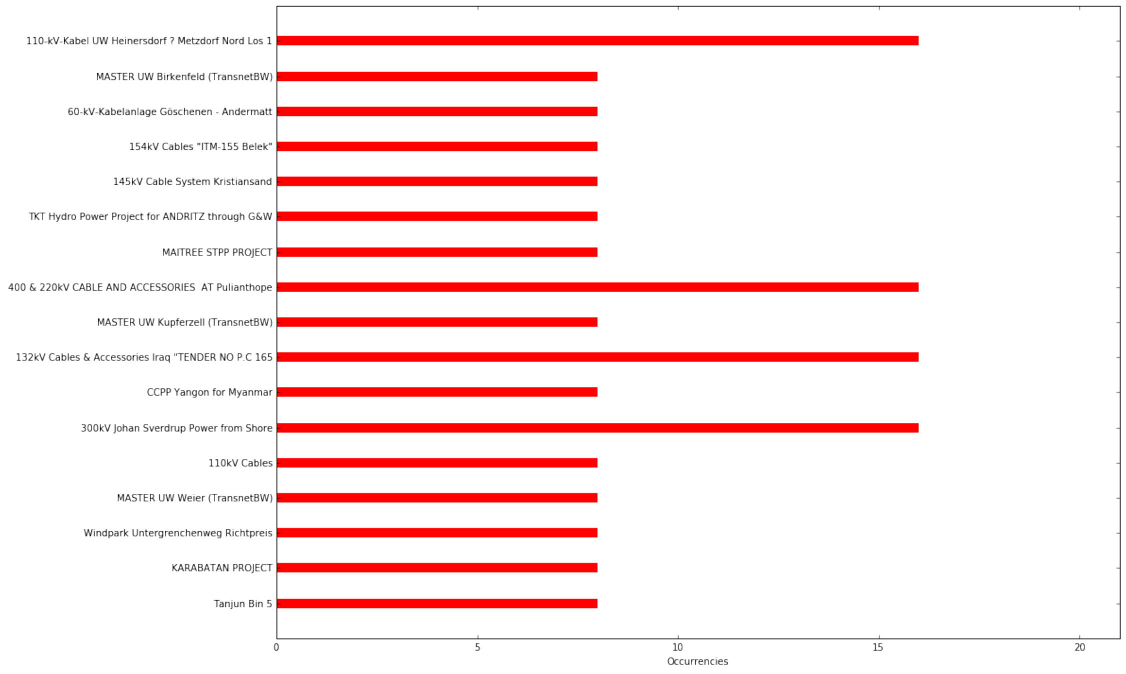
1 Diagnostic Plot

1.0.1 Distribution of the Combinations

```
In [9]: def plot_scatter_dynam(delay):
            fitness_p = fitness[fitness.delay <= delay]
            scatter_plot(fitness_p.baselines.tolist(), 'workload', 'margin', df = fitness_p)

            frequency_plot(fitness_p.baselines.tolist(),opportunities = opportunities)

        interact(plot_scatter_dynam, delay=(0, fitness.delay.max() , 10))
```


1e7
margin
workload
Number of Offers


1e7
margin
workload
Delay (hours)

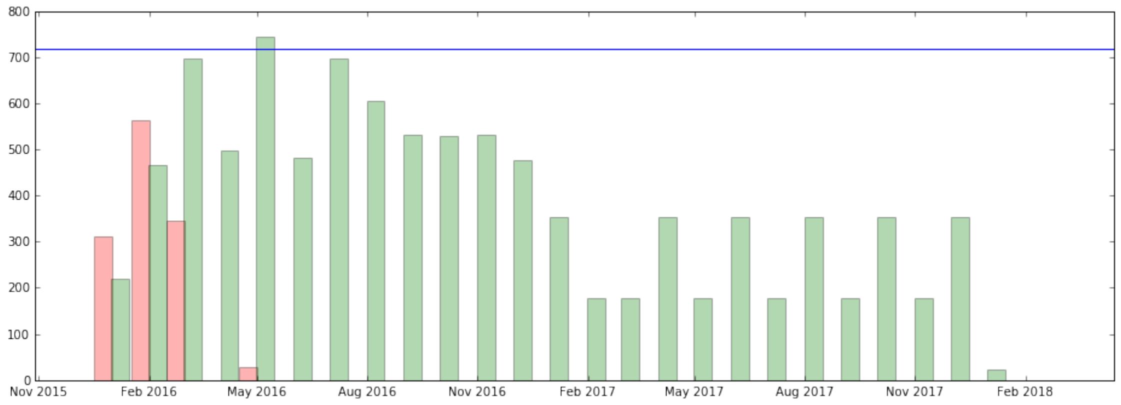


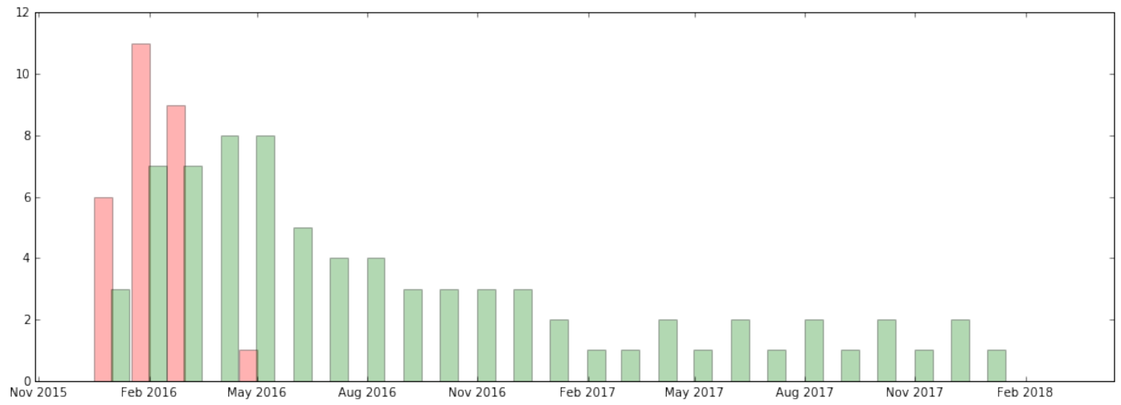
1.0.2 Selection of the "Best" Combination of Offers

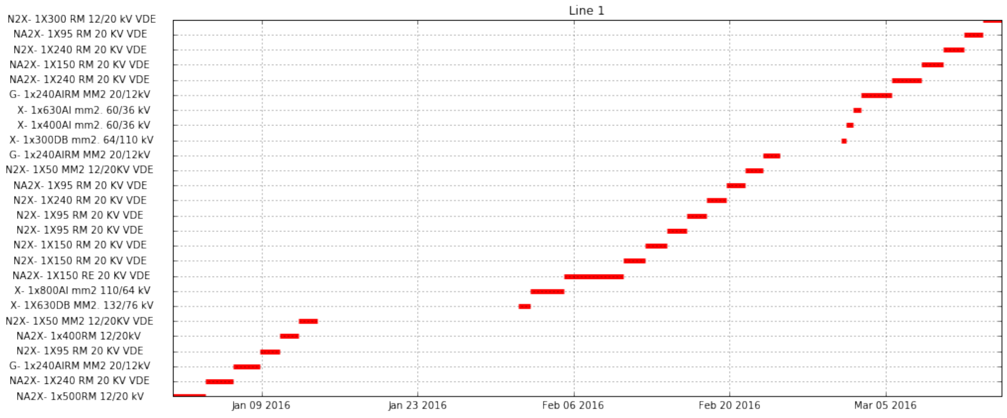
```
In [11]: sel_fit = copy(fitness)
         sel_fit = sel_fit[sel_fit.delay < 20]

         sel_fit = sel_fit.sort_values(['workload', 'margin'],ascending=[False,False])
```

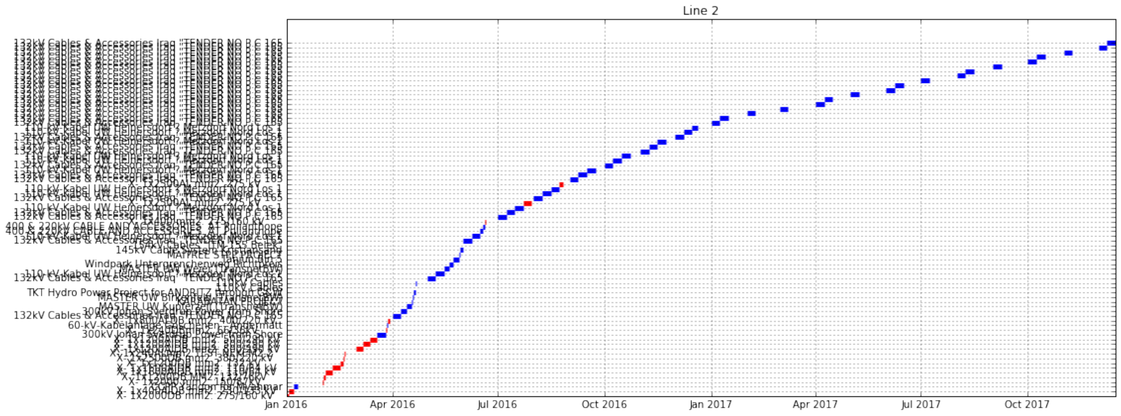
1.0.3 Workload Distribution

```
In [12]: def distr_overwiew(num):
             tmp_sel_fit = sel_fit.iloc[num]
             schedule(tmp_sel_fit.baselines)
             time_table(tmp_sel_fit.baselines, batches = batches)
             tmp_sel_fit

         interact(distr_overwiew, num=(0,len(sel_fit)-1 , 1))
```


800
700
600
500
400
300
200
100
0
Nov 2015
Feb 2016
May 2016
Aug 2016
Nov 2016
Feb 2017
May 2017
Aug 2017
Nov 2017
Feb 2018


12
10
8
6
4
2
0
Nov 2015
Feb 2016
May 2016
Aug 2016
Nov 2016
Feb 2017
May 2017
Aug 2017
Nov 2017
Feb 2018





1.0.4 Questions:

- Strategy for the Baseline selection
- Strategy for the Schedule selection
- More diagnostic plots

```
In [ ]:
```