In [1]:

```
import copy
import random
import re

import pandas
import numpy

from curve_fits import Call

from curve_fits import fits
from curve_fits import frames
```

In [2]:

```
def random_ints_sample(size, max_abs=10000):
    return random.sample(range(-max_abs, max_abs), k=size)
```

In [3]:

```
size_range = range(1050, 1750)
overfit = 1  # HIGH to differentiate from single-line shape
init_calls = [Call('set', random_ints_sample(size)) for size in size_range]
profile = frames.TimeComplexityProfile(init_calls, Call('copy'), Call('pop'),
    index=pandas.Index(size_range, name='List length'), overfit=overfit, fraction=0.9, loops=100)
```

In [4]:

```
profile.fit_all_with(Call('polynomial', 1), Call('polynomial_stair', [1], [[x] for x in range(1300
, 1420)]))
```

In [5]:

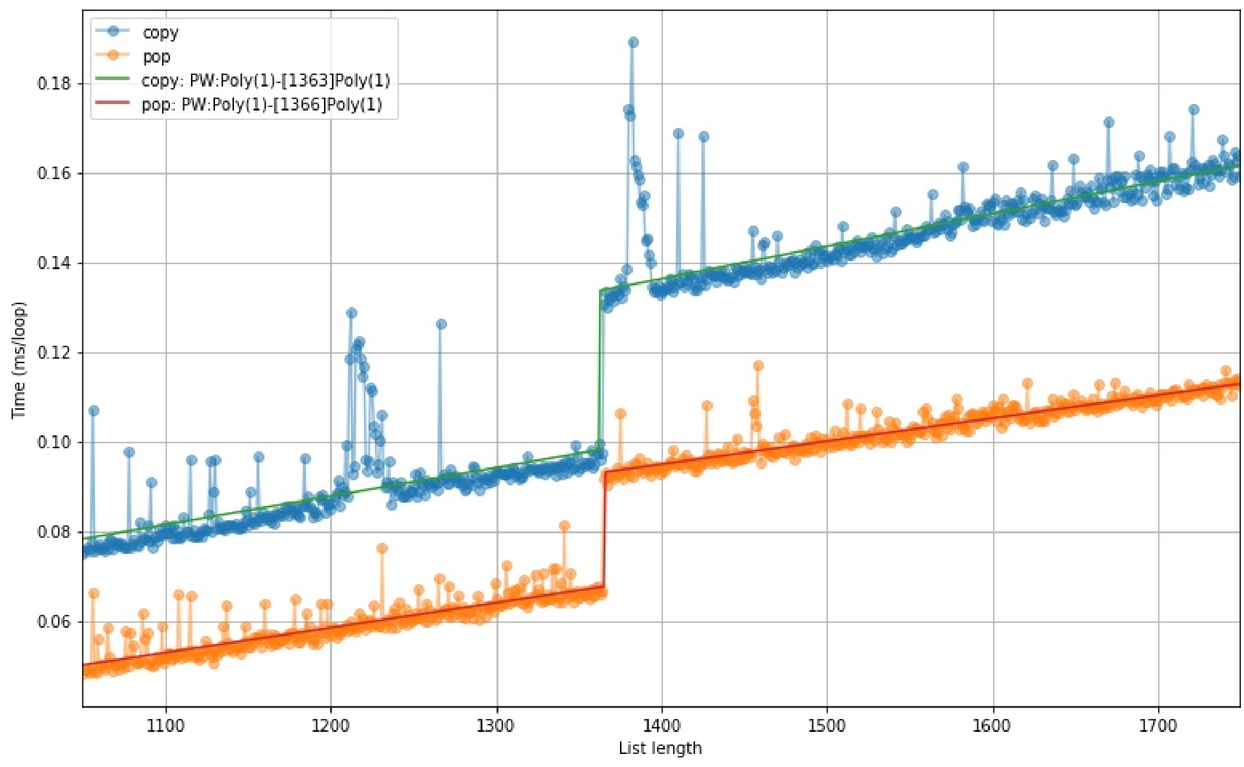
```
profile.best_fits(limit=3)
```

Out[5]:

		kind		fit	DOF
key	cost				
copy	0.017196	PW:Poly(1)-[1363]Poly(1)	(0.0114 ± 0.0056) + (0.0000637 ± 0.0000046)x |...		4
	0.017221	PW:Poly(1)-[1365]Poly(1)	(0.0085 ± 0.0054) + (0.0000661 ± 0.0000044)x |...		4
	0.017359	PW:Poly(1)-[1364]Poly(1)	(0.0105 ± 0.0057) + (0.0000645 ± 0.0000047)x |...		4
pop	0.006520	PW:Poly(1)-[1366]Poly(1)	(-0.0083 ± 0.0021) + (0.0000556 ± 0.0000018)x ...		4
	0.006703	PW:Poly(1)-[1365]Poly(1)	(-0.0084 ± 0.0023) + (0.0000558 ± 0.0000019)x ...		4
	0.007240	PW:Poly(1)-[1364]Poly(1)	(-0.0100 ± 0.0022) + (0.0000570 ± 0.0000018)x ...		4

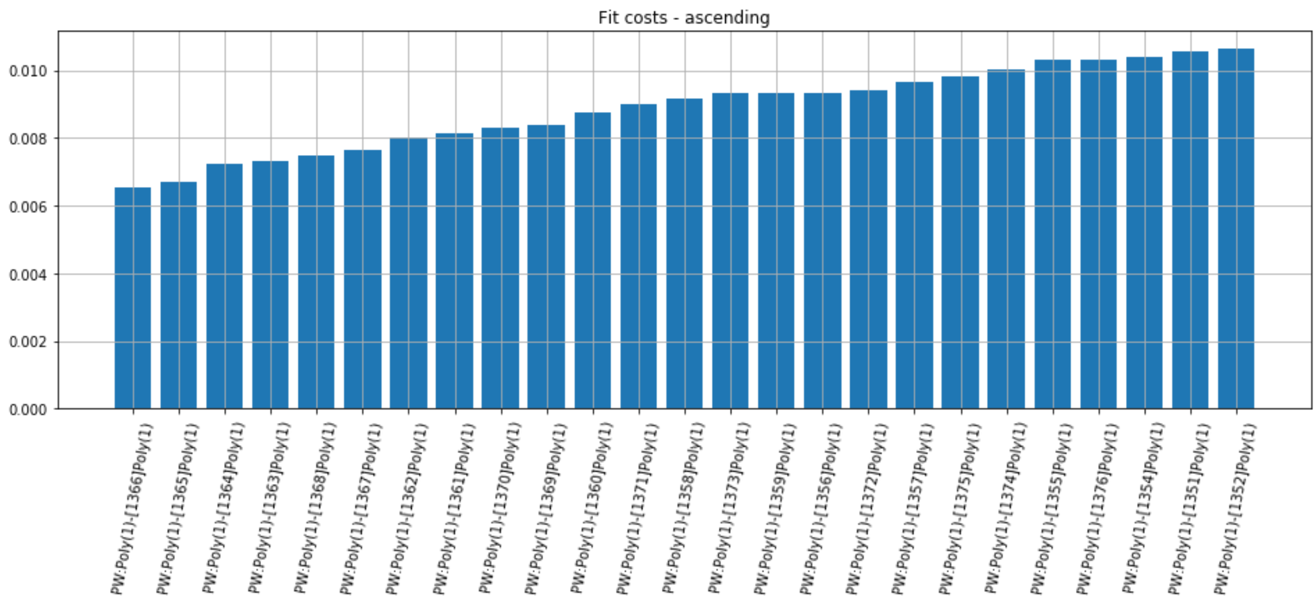
In [6]:

```
profile.plot(limit=1)
```



In [7]:

```
profile.plot_costs('pop', limit=25, rotation=80)
```



In [8]:

```
pop_fit = profile.best_fit('pop')
pop_fit
```

Out[8]:

```
<PiecewiseFit: (-0.0083 ± 0.0021) + (0.0000556 ± 0.0000018)x | (0.0231 ± 0.0015) + (0.00005132 ± 9.
9E-7)x>
```

In [9]:

```
pop_fit.fits, pop_fit.jumps_at
```

Out[9]:

```
((<PolynomialFit: (-0.0083 ± 0.0021) + (0.0000556 ± 0.0000018)x>,
  <PolynomialFit: (0.0231 ± 0.0015) + (0.00005132 ± 9.9E-7)x>),
 (1366,))
```

In [10]:

```
pop_fit.fits[1].measures
```

Out[10]:

```
(<Measure: 0.0231 ± 0.0015>, <Measure: 0.00005132 ± 9.9E-7>)
```

In [11]:

```
pop_fit_copy = copy.deepcopy(pop_fit)
pop_fit_copy.series = None   # Mocked to show equality is not affected
assert pop_fit == copy.deepcopy(pop_fit)
```

In [12]:

```
assert pop_fit != fits.PiecewiseFit(profile.data['pop'], 1, pop_fit.jumps_at, fits.PolynomialFit)
```

In [13]:

```
assert [re.sub(r'\[[0-9]{3,}\]', '', kind) for kind in profile.best_fits(limit=1)['kind']] == [
    'PW:Poly(1)-Poly(1)', 'PW:Poly(1)-Poly(1)']
```

In [ ]: