

1 Performance

How to use profile of matlab:

```
profile on;
cmd;
info = profile('info');
save('~\Matlab\Profile\name.mat','info');
load ~\Matlab\Profile\name.mat;
profview(0,info);
```

1.1 Latch

The profile is saved as `~/Matlab/Profile/passgate.mat`. The total time is `38k(38476.615)` seconds.

The running time of some important functions is

Function	Calls	Time	Time per Call
ph_forward	581	36k	62.53
ph_model	1268	33k	26.1
ph_march	1267	2.6k	2.03
model_create	158112	27k	0.1729
linearizeQuadFitLip	1581112	22k	0.0139
linearizeQuadFit	31362	311	0.01
brockettFitComp	316229	1.6k	0.0052
ph_trim	3136	2.65k	0.8457
ph_interval	158112	2.15k	0.0136
lp_project	32011	1.7k	0.0538
lp_project2	98025	0.8k	0.0083
lp_relax	1996712	1.65k	8e-4
lp_norm	1996712	0.9k	4.5e-4
poly_reduce	70560	0.6k	0.0086
poly_union	13587	0.4k	0.0273
poly_intersect	32011	0.4k	0.0124

Figure 1 shows how much time is spent on each function. More than *36k* time is spent on `ph_forward`, with *33k* (91%) on `ph_model`, *2.57k* on `ph_march` and *0.6k* on `ph_trim`. It is called by 581 times, while `ph_model` is called by 1268 times, which means COHO tries a little more than twice to guess an valid value of time and bloat **It seems that bloat is increased to maxBloat first, then is decreased because it does not pass the verify. Use better algorithm to guess it?**

For the `ph_model` function, *27k*(82.6%) is spent on `model_create`, *2.15k* on `ph_interval` and *1k* for `int_maxDot`. The `face_model` is called by 98052 times (one `ph` has 77.3 faces on average) and `model_create` is called by 158112 times, which is 1.6 iterations on average.

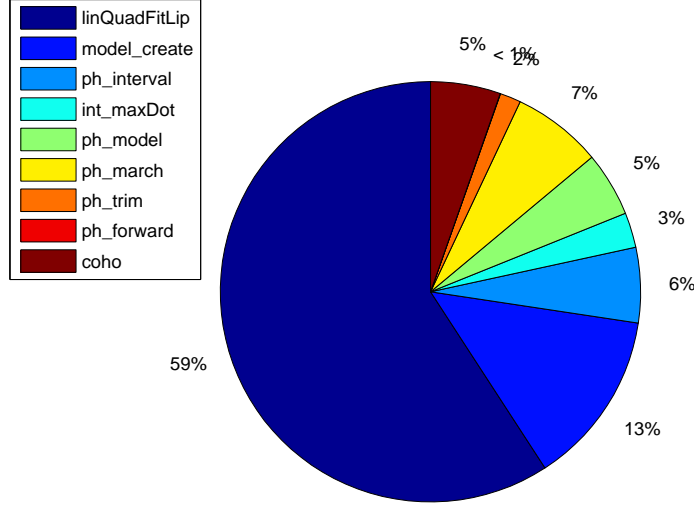


Figure 1: The time spent on coho functions

`face_forward` costs $2k$ seconds, with $0.8k$ (40%) on `lp_project2`, $0.4k$ on `model_forward`.

In the `model_create` function, about $22k$ time is spent on `linearizeQuadFitLip` function. The time spent on computing matrix M is $2.2k$, on computing c, b vector is $2.25k$, on computing the error is $10.4k$, on removing cubes by lp is $3.55k$, and on computing grids related data is $3.5k$, which is also shown in figure 2.

The total time of `quadBound` function is $2.4k(2406.369)$ with $6m(5958801)$ calls, the average time per call is $4e - 4$ seconds. The number of iterations to reduce gap between lower and upper bounds is 5958801 , which means 3.7687 iterations per call. If the number of iteration exceeds 5 or the number of cubes is huge ($> 1e6$), `linearizeQuadFit` is called. It spends 42 seconds on computing M , 45 on b , 127 on error, 40 on lp, and 51 on grids. **The optimized `linearizeQuadFitLip` is even slower than `linearizeQuadFit` function?!**

The lines with most time is

Line	Function	Calls	Time	Percent
L249	quadBounds	5958801	2406	11%
L244	$x \cdot x'$	5958801	1532	7.0%
L97	lp_relax	1581111	1334	6.1%
L266	mat2cell	4377690	1247	5.7%
L77	lower bound of all cubes	4743336	1014	4.6%

The functions called with most time is

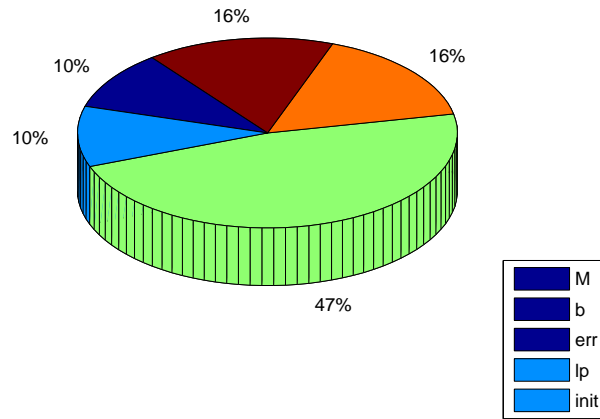


Figure 2: The time spent on `linearizeQuadFitLip` function

Function	Calls	Time	Percent
quadBounds	5958801	2334	10.6%
mat2cell	7539912	2062	9.4%
cell2mat	9121023	1678	7.6%
lp_relax	1581111	1304	5.9%
sub2ind	7905555	1032	4.7%
linearizeQuadFit	31362	312	1.4%

We have another profiling data in `~/Matlab/Profile/passgate2.mat`. The result is similar. The average number of faces per projectagon increases to 100, because COHO runs longer thus a projectagon is more complicated. The number of iterations to reduce error gap in `linearizeQuadFitLip` function is reduced to 3.29.

References

- [NW04] David Harris Neil Weste. *CMOS VLSI Design: A Circuits and System Perspective*. Addison Wesley, 2004.