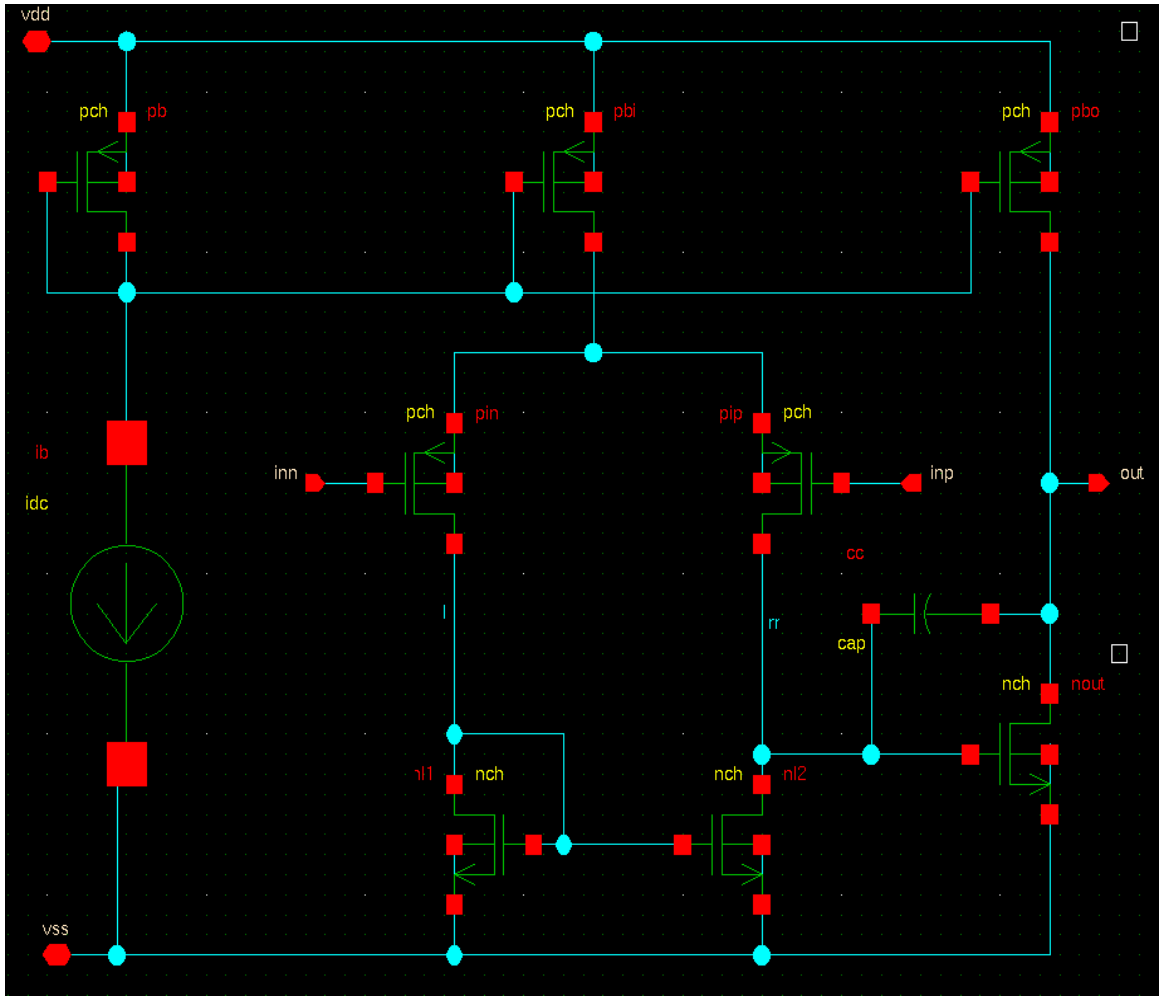
Optimisation of a 2 stage Op-Amp Using Motivated

- Op-amp circuits commonly used across all analogue electronics, in particular for active filters.
- Circuits are re-used and re-targeted for different applications or moved to a new technology node, requiring tweaking of output parameters and performance which is time consuming and repetitive
- How do parametric changes affect the circuit performance? What is the envelope of operation? How are conflicting performance objectives determined?
- Ideal target for MOTIVATED multi-objective automated optimisation process

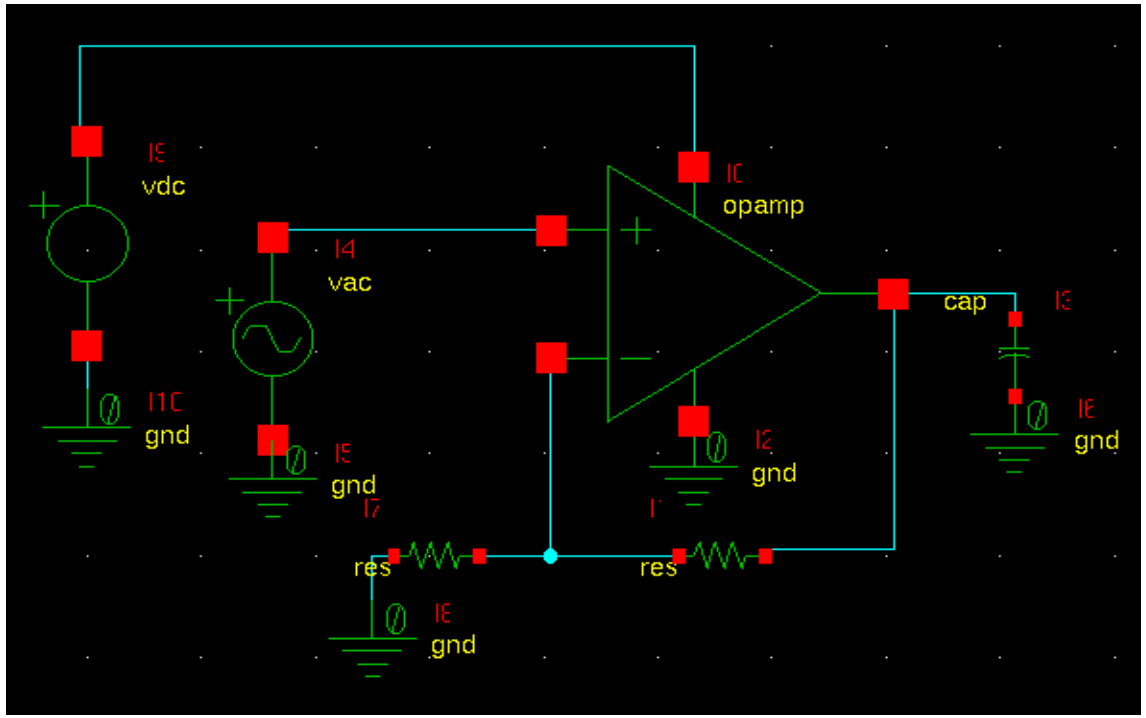
Example Op_amp schematic and netlist




```
*
.subckt amp2 inp inn out vdd vss
+ ib=10u mpb=1 mpbi=4 mpbo=8 mdiff=6 mnload=1 mnout=10 $ "architecture"
+ wpb=10u lpb=1u wpin=20u lpin=1u wnl=10u lnl=4u wout=40u lout=1u ccv=2p

ib ibias vss dc=ib
xpb  ibias ibias vdd vdd pch w=wpb l=lpb m=mpb
xpbi com   ibias vdd vdd pch w=wpb l=lpb m=mpbi
xpbo out   ibias vdd vdd pch w=wpb l=lpb m=mpbo

xpin ll inn com com pch w=wpin l=lpin m=mdiff
xpip rr inp com com pch w=wpin l=lpin m=mdiff

xnl1 ll ll  vss vss nch w=wnl l=lnl m=mnload
xnl2 rr ll  vss vss nch w=wnl l=lnl m=mnload

xnout out rr vss vss nch w=wout l=lout m=mnout
cc out rr ccv
.ends
```

Note: fully parametrised netlist.

Test bench set up




```
.lib './models.sp' ttt
.inc amp2.sp

.inc 'genes_def_set.sp'
*.param wpb=10u lpb=4u wpin=20u lpin=4u wnl=10u lnl=4u wout=40u lout=4u ccv=2p ib=1

xa inp inn out vdd gnd amp2
+ ib=10u mpb=mpb mpbi=mpbi mpbo=mpbo mdiff=mdiff mnload=mnload mnout=mnout
+ wpb=wpb lpb=lpb wpin=wpin lpin=lpin wnl=wnl lnl=lnl wout=wout lout=lout ccv=ccv

.param sup=2.5 rshort=0.1 ropen=1e9 cload=1p
**
rfb inn out r=rshort ac=ropen
rin inn gnd r=ropen ac=rshort

vacin inp gnd dc='sup/2.0' ac=1

cl out gnd c=cload

vsup vdd gnd dc=sup

.control
  ac dec 50 1 1e8
  source satM.sp
  let Gmag=20*log10(mag(out))
  let Gph=phase(out)*180/PI
  plot Gmag Gph xlog xlabel 'Frequency [Hz]' ylabel 'Mag [dB, deg]'

**measurements
  let gdc_obj=Gmag[0]
  let gmag3=gdc_obj-3
  print gdc_obj
  meas ac bw3_obj when Gmag=gmag3
  meas ac bw0_obj when Gmag=0
  meas ac ph0_obj find Gph at=bw0_obj

.endc
```

Includes “objective” measurement/calculations for the optimisation. For this example simply looking at gain, bandwidth & phase

“Genes” file to specify design variables (optimisation parameters)

Define design variables (optimisation parameters), initial values, limits and steps

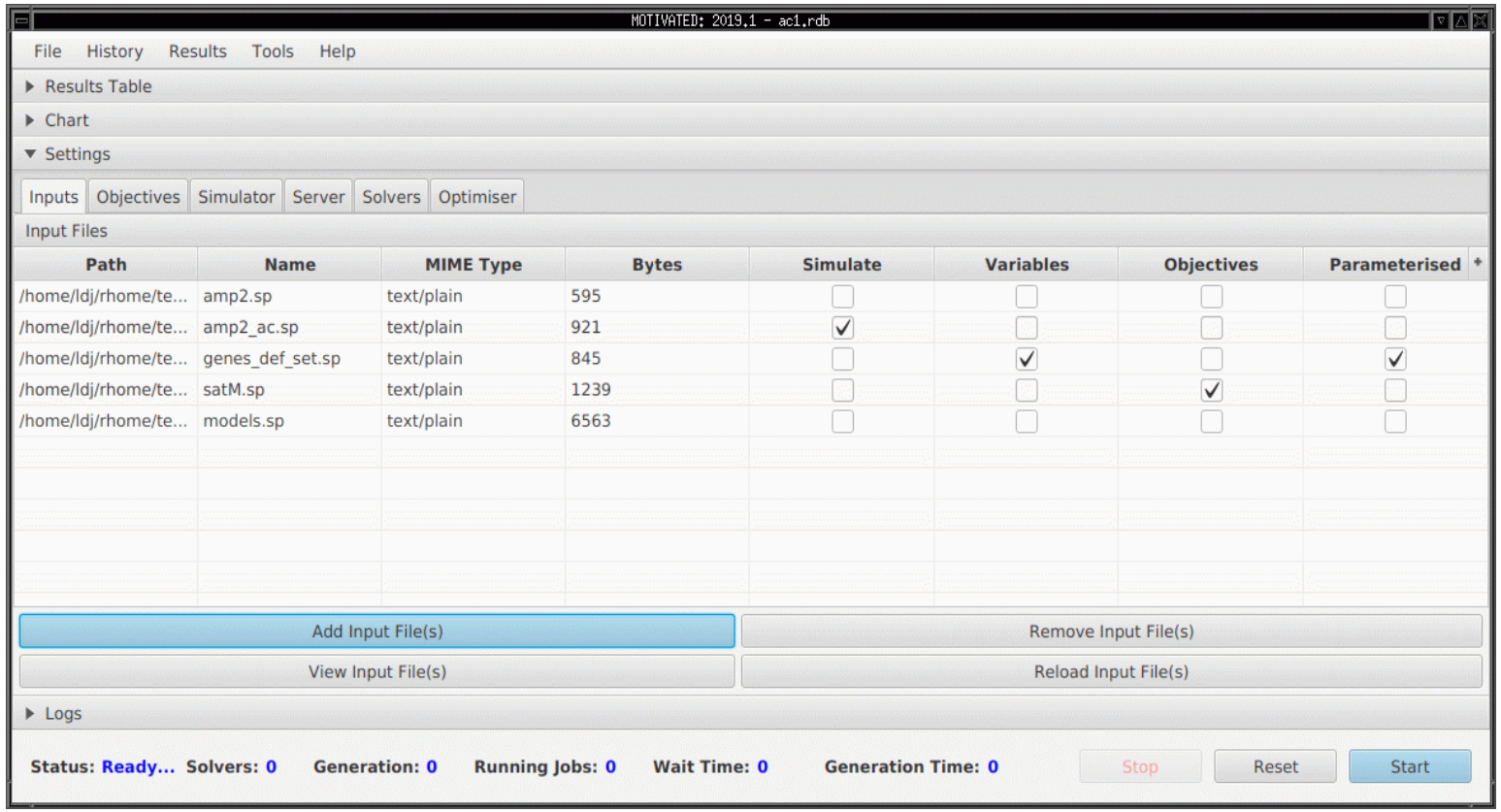
Place-holders for optimisation parameters values

```
* &GENE: 0, 24e-6, 1e-6, 100e-6, 1e-6 $ wpb
* &GENE: 1, 6e-6, 0.4e-6, 20e-6, 0.05e-6 $ lpb
* &GENE: 2, 32e-6, 1e-6, 100e-6, 1e-6 $ wpin
* &GENE: 3, 6e-6, 0.4e-6, 20e-6, 0.05e-6 $ lpin
* &GENE: 4, 24e-6, 1e-6, 100e-6, 1e-6 $ wnl
* &GENE: 5, 6e-6, 0.4e-6, 20e-6, 0.05e-6 $ lnl
* &GENE: 6, 48e-6, 1e-6, 100e-6, 1e-6 $ wout
* &GENE: 7, 6e-6, 0.4e-6, 20e-6, 0.05e-6 $ lout
* &GENE: 8, 2e-12, 0.1e-12, 100e-12, 0.5e-12 $ ccv
* &GENE: 9, 1, 1, 20, 1 $ mpb
* &GENE: 10, 4, 1, 20, 1 $ mpbi
* &GENE: 11, 16, 1, 20, 1 $ mpbo
* &GENE: 12, 6, 1, 20, 1 $ mdiff
* &GENE: 13, 1, 1, 20, 1 $ mnload
* &GENE: 14, 10, 1, 20, 1 $ mnout

.param wpb=&0
.param lpb=&1
.param wpin=&2
.param lpin=&3
.param wnl=&4
.param lnl=&5
.param wout=&6
.param lout=&7
.param ccv=&8
.param mpb=&9
.param mpbi=&10
.param mpbo=&11
.param mdiff=&12
.param mnload=&13
.param mnout=&14
```

MOTIVATED GUI Settings: Inputs

Add in the target circuit, test bench, model files, “genes” file. Can be fully hierarchical.

Select which are simulated, which have the parameters & objectives

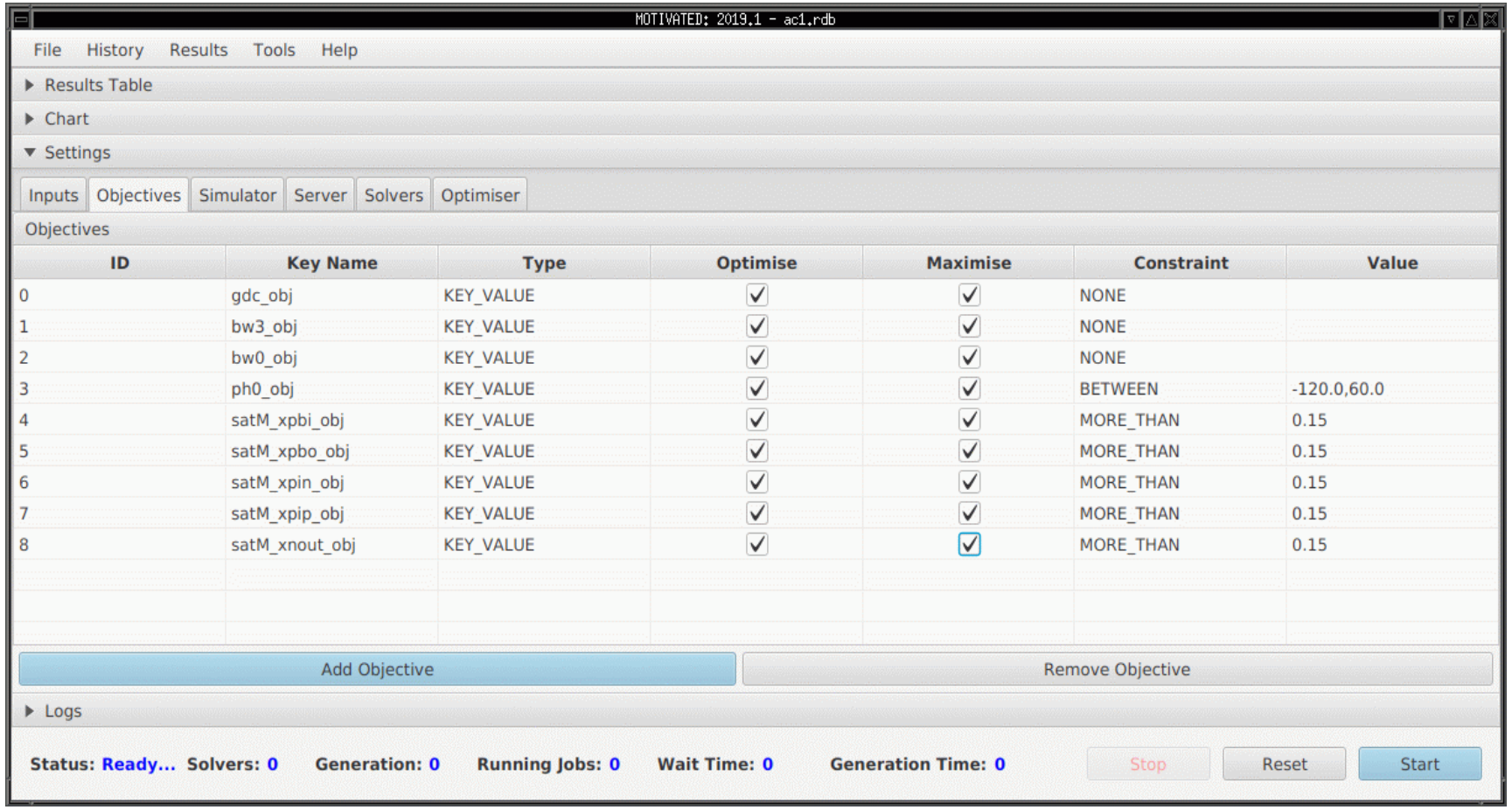


MOTIVATED GUI Settings: Objectives

Objectives are automatically picked up by the Optimiser.

Select which are to be optimised and if they are maximised, minimised or have any constraints

Once set, the Optimiser is started, using a simulation engine of choice

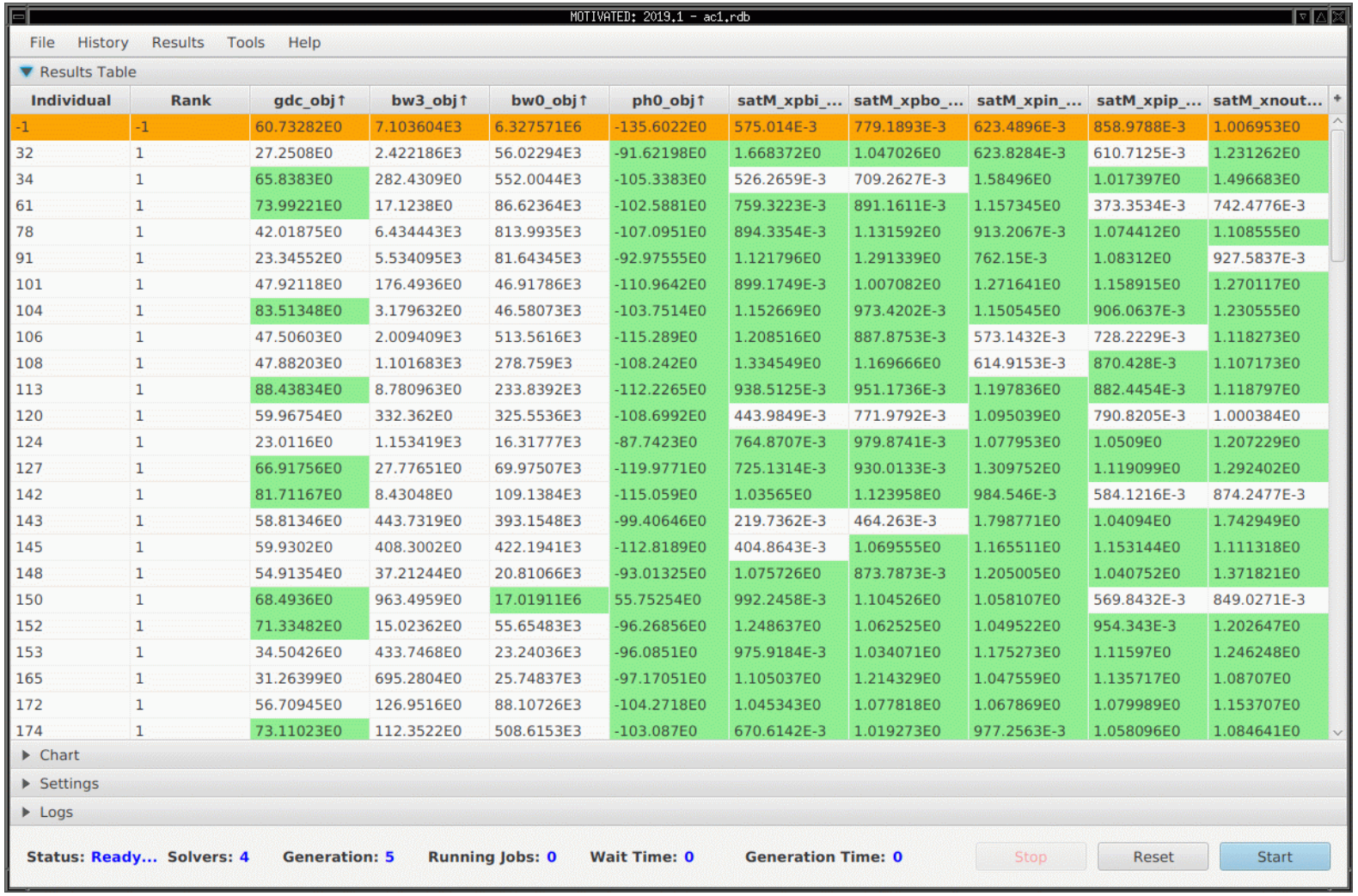


MOTIVATED GUI – Optimisation process results

As the Optimiser progresses, the results of each “objective” are put in a table and marked against a reference (initial / typical performance) – green highlights an improvement

As the number of generations increases, the optimiser selects the “fittest” out of the population to move on to the next generation

Once all generations are complete, the table displays the best objectives of the final generation

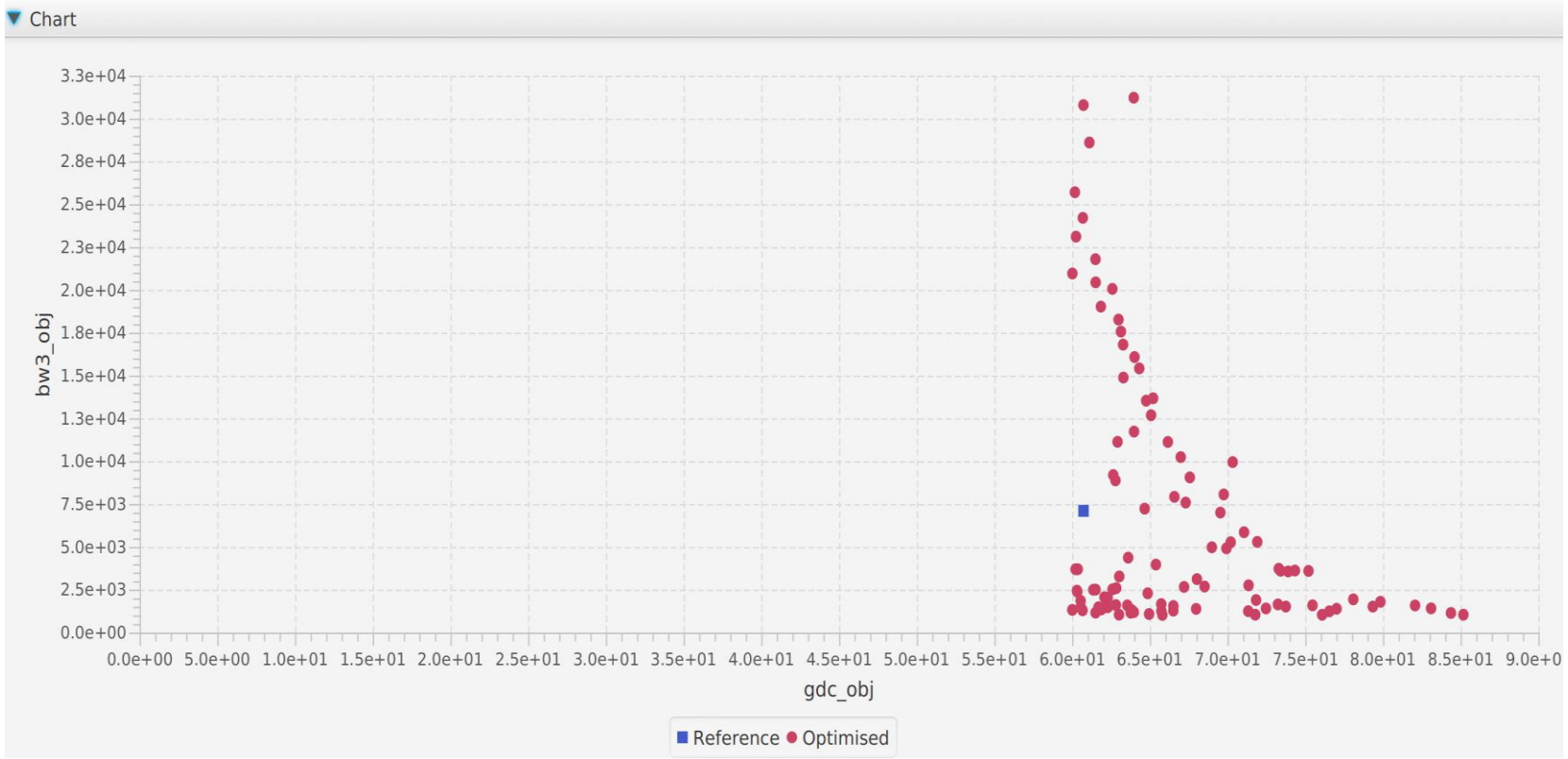


MOTIVATED GUI – Optimisation process results

The results can also be displayed during, and at the end of the optimisation as a plot of one objective against another, effectively a pareto curve.

Each point on the curve represents a fully optimised solution where each objective is at it's maximum and can only be improved at the detriment of a different objective.

This allows the engineer to select the best solution and trade off of all objectives simply



MOTIVATED GUI – Optimisation process results

- Curve “zoomed in” for greater clarity
- Results based on a constraint of Gain >60 and Bw3 >1k