Hyper-parameter tuning to improve existing software

Alexander Brownlee, University of Stirling

Collaborators

Queen Mary
University of London

Edmund K. Burke

Lancaster University

Michael Epitropakis

AIRFRANCE KLM

Jeroen Mulder

Marc Paelinck

Outline

- The software
- What to improve?
- A systematic approach:
  - Statistical analysis
  - Single-objective tuning
  - Multi-objective tuning
- What about GI?

AIRFRANCE KLM

KLM Royal Dutch Airlines

AIRFRANCE

2004

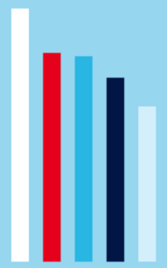
87.4
million passengers carried in 2014

316
destinations in 115 countries

546
aircraft in operation at December 31, 2014

1st
Air France-KLM is the number one in terms of intercontinental traffic on departure from Europe

Number of passengers carried by the European airlines in 2014



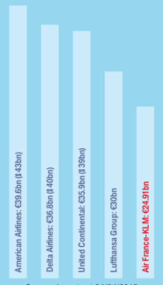
- Lufthansa Group 106 million
- Air France-KLM 87.4 million
- Ryanair 86.4 million
- IAG 77.3 million
- Easyjet 65 million

(source : company communication)

3.3
billion passengers carried in 2014
↓
7.3
billion in 2034
(source : IATA.)

4%
Airline traffic is expected to see 4% annual growth over the next 15 years
(source: Boeing)

Global airline ranking



Conversion rate at 24/04/2015

5th
The Air France-KLM Group ranks number five in terms of revenues
(source: company communication)

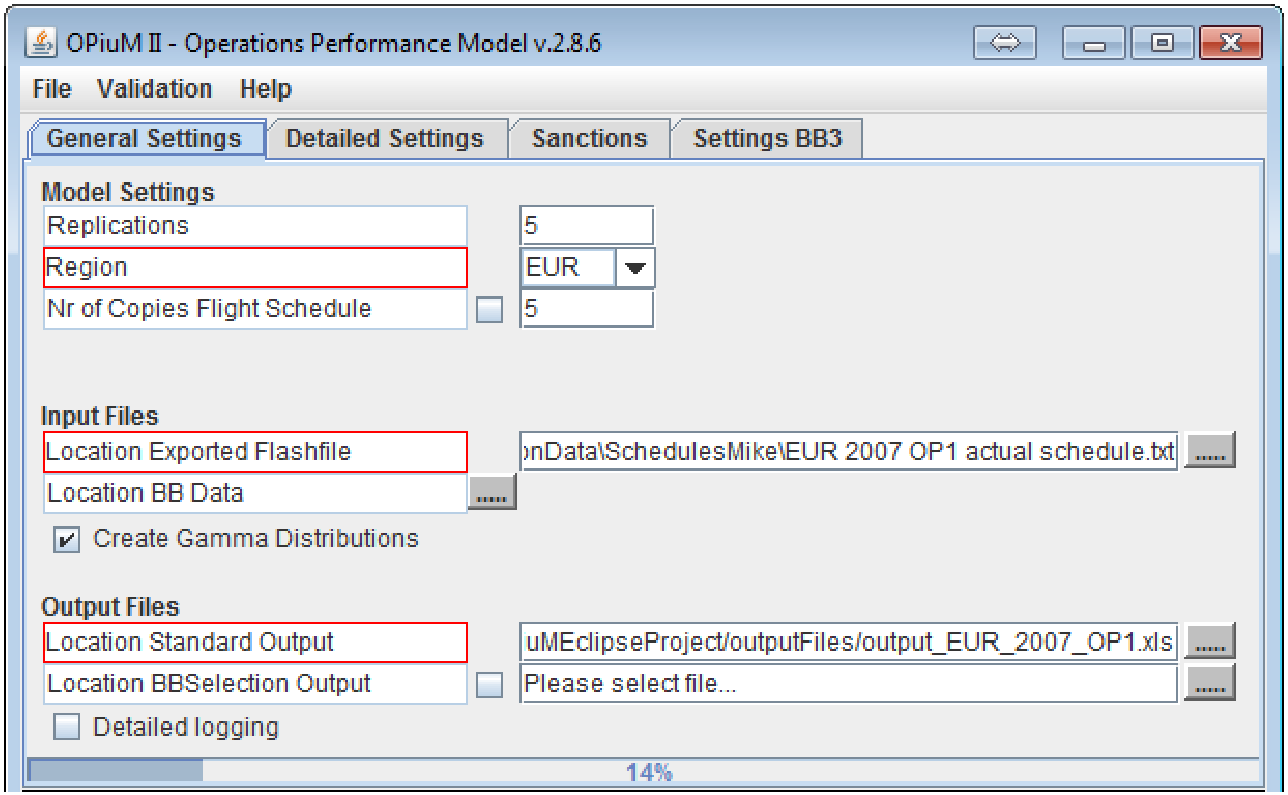
Air France — KLM Annual Report 2014: http://www.airfranceklm.com/sites/default/files/publications/annual_report_2014.pdf

Discover the Air France-KLM world

Air France — KLM Annual Report 2014: http://www.airfranceklm.com/sites/default/files/publications/annual_report_2014.pdf

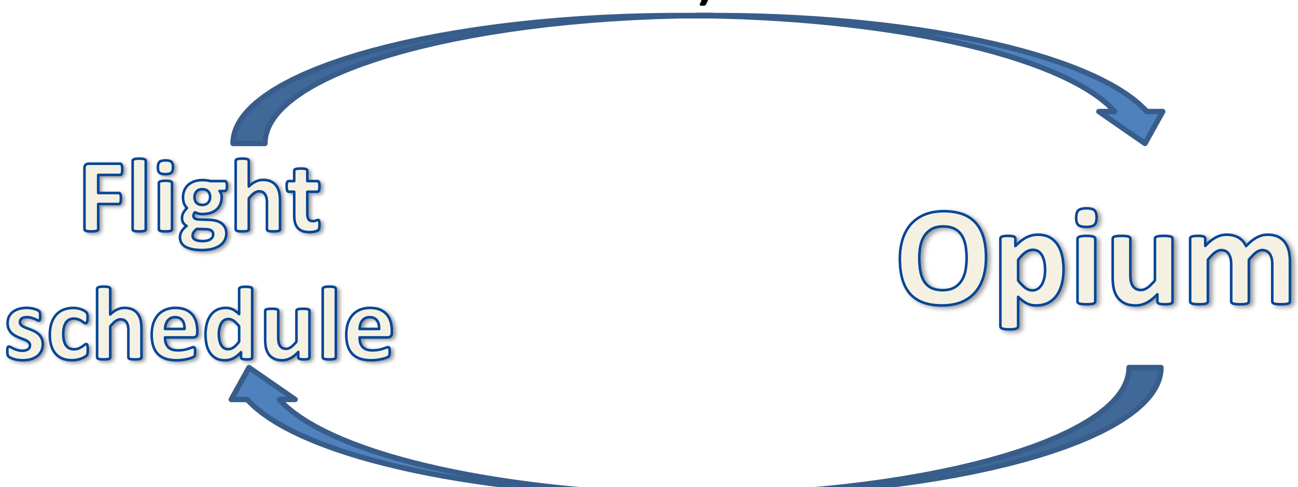
Software

- OPiuM – Java based simulator, developed in-house at KLM
- Built on DSOL library, developed at TU Delft

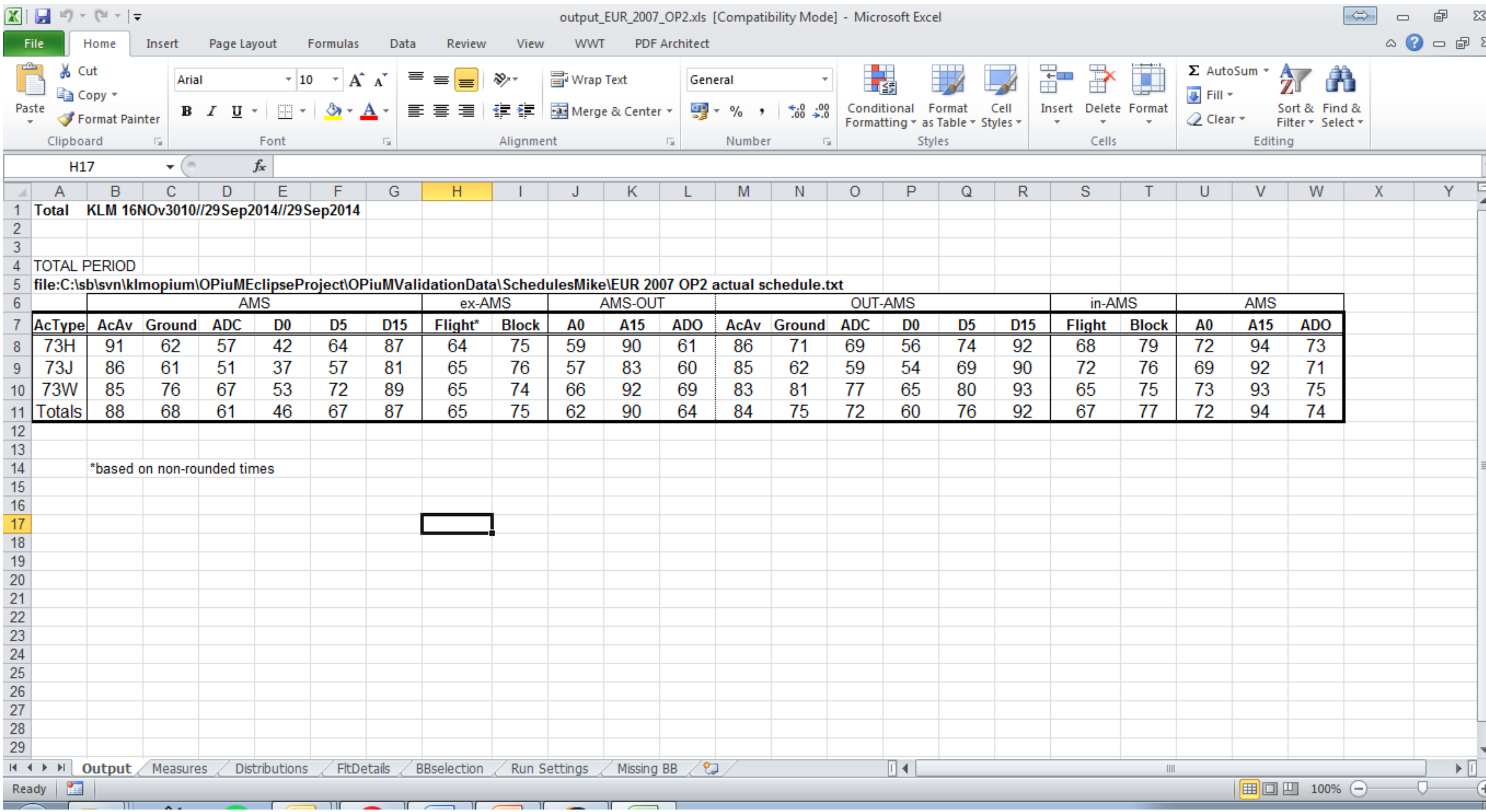


Software

- Simulates aircraft movements given a schedule, estimates possible delays
- One flight schedule:
  - E.g. Europe, 3 months, ~17k flights
- All KLM flight schedules pass through Opium (soon to include Air France too)



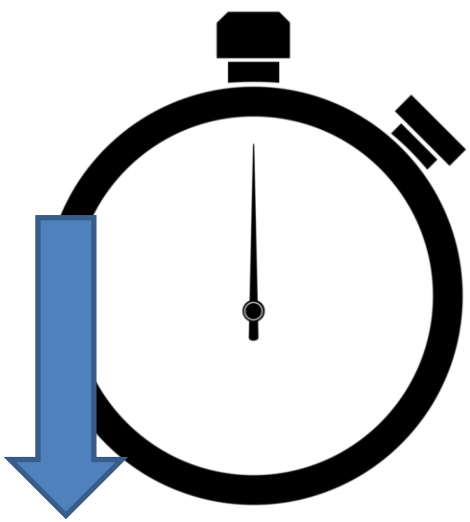
Software



	AMS						ex-AMS		AMS-OUT			OUT-AMS						in-AMS		AMS		
AcType	AcAv	Ground	ADC	D0	D5	D15	Flight*	Block	A0	A15	ADO	AcAv	Ground	ADC	D0	D5	D15	Flight	Block	A0	A15	ADO
73H	91	62	57	42	64	87	64	75	59	90	61	86	71	69	56	74	92	68	79	72	94	73
73J	86	61	51	37	57	81	65	76	57	83	60	85	62	59	54	69	90	72	76	69	92	71
73W	85	76	67	53	72	89	65	74	66	92	69	83	81	77	65	80	93	65	75	73	93	75
Totals	88	68	61	46	67	87	65	75	62	90	64	84	75	72	60	76	92	67	77	72	94	74

What to improve?

- Opium software is part of a loop of improving and testing schedules
- so, **faster**, and **at least the same accuracy**

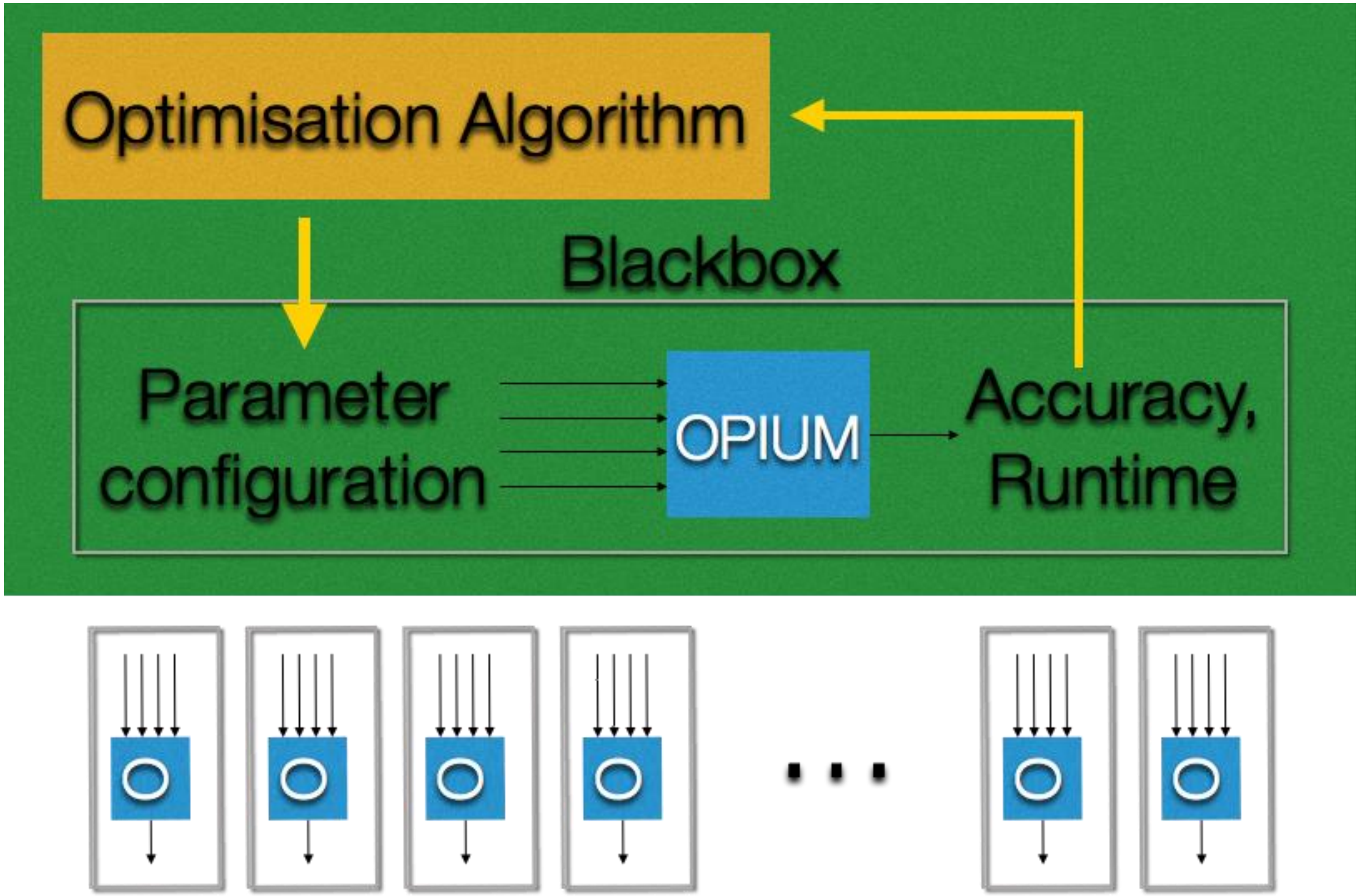


Parameter tuning

- We were provided with real-world schedules and results covering 2007-2010
- Starting point: Opium has 14 external parameters
  - These have been manually tuned over about 10 years, and are now mostly "don't touch"
  - Tune these to improve simulation accuracy (fit to historical data) and simulation run time

Wrapper

- Needed for any kind of automated improvement



A systematic approach

1. Statistical analysis of the parameters
2. Single objective tuning & model based analysis
3. Seeded multi-objective optimisation

Results:

high-performing configurations, with explanation

Stage 1: statistical analysis

1. Statistical Screening
   – Design of experiments / fractional factorial
   – Uses lower and upper bounds for each parameter
   – Screens out insensitive parameters
2. Exploring the sensitive parameters
   – Fine-grained exploration of each parameter
   – Exhaustive: accuracy
   – Response surface: time

Statistical Screening (Accuracy)

Parameter	LB	UB	P-value
Max Maintenance Reduction	0	0.2	0.177
Ground Factor Out	1	1.3	0.311
Slack Selection BB3	0	50	0.505
Max Legs Swap	2	6	0.404
HSF threshold Out	0	5	0.794
HSF threshold In	0	15	0.789
Max Legs Cancel	**1**	**7**	**0.018**
HSF threshold	0	15	0.625
Cancel Measure On	**0**	**1**	**0.006**
Break Maintenance Measure On	0	1	0.980
Create Gamma	**0**	**1**	**0**
Rounding off method	Regular	None	0.514
Swap Measure On	**0**	**1**	**0**
HSF Measure On	0	1	0.714

Optimal values: Accuracy

- Exhaustive search
  - Search space of 112

Parameter	LB	UB
Max Legs Cancel	1	14
HSF threshold	False	True
Create Gamma	False	True
Swap Measure On	False	True

- Matches default params acc=271.628)
- Importance, high to low:
  - Swap Measure On
  - Create Gamma
  - Cancel Measure On (negligible?)
  - Max Legs Cancel (negligible?)

MLC	CMO	CG	SMO	MSE
1	1	1	1	271.6
2	1	1	1	271.6
3	1	1	1	271.6
4	1	1	1	271.6
5	1	1	1	271.6
6	1	1	1	271.6
7	1	1	1	271.6
8	1	1	1	271.6
9	1	1	1	271.6
10	1	1	1	271.6
11	1	1	1	271.6
12	1	1	1	271.6
13	1	1	1	271.6
14	1	1	1	271.6
1…14	0	1	1	271.6
2…14	1	0	1	292.7
1	1	0	1	306.9
1…14	0	0	1	306.9
2…14	1	1	0	366.2
2…14	1	0	0	453.3
1	1	1	0	564.0
1…14	0	1	0	564.0
1	1	0	0	646.9
1…14	0	0	0	646.9

Time

- Same process for time, but second stage was a response surface experiment (6 params, 520 solutions)
- Optimal config:
  - Run time 476.5s (default was 1406.7)
  - Accuracy (MSE) 426.988 (default was 271.628)
- So some potential for improvement

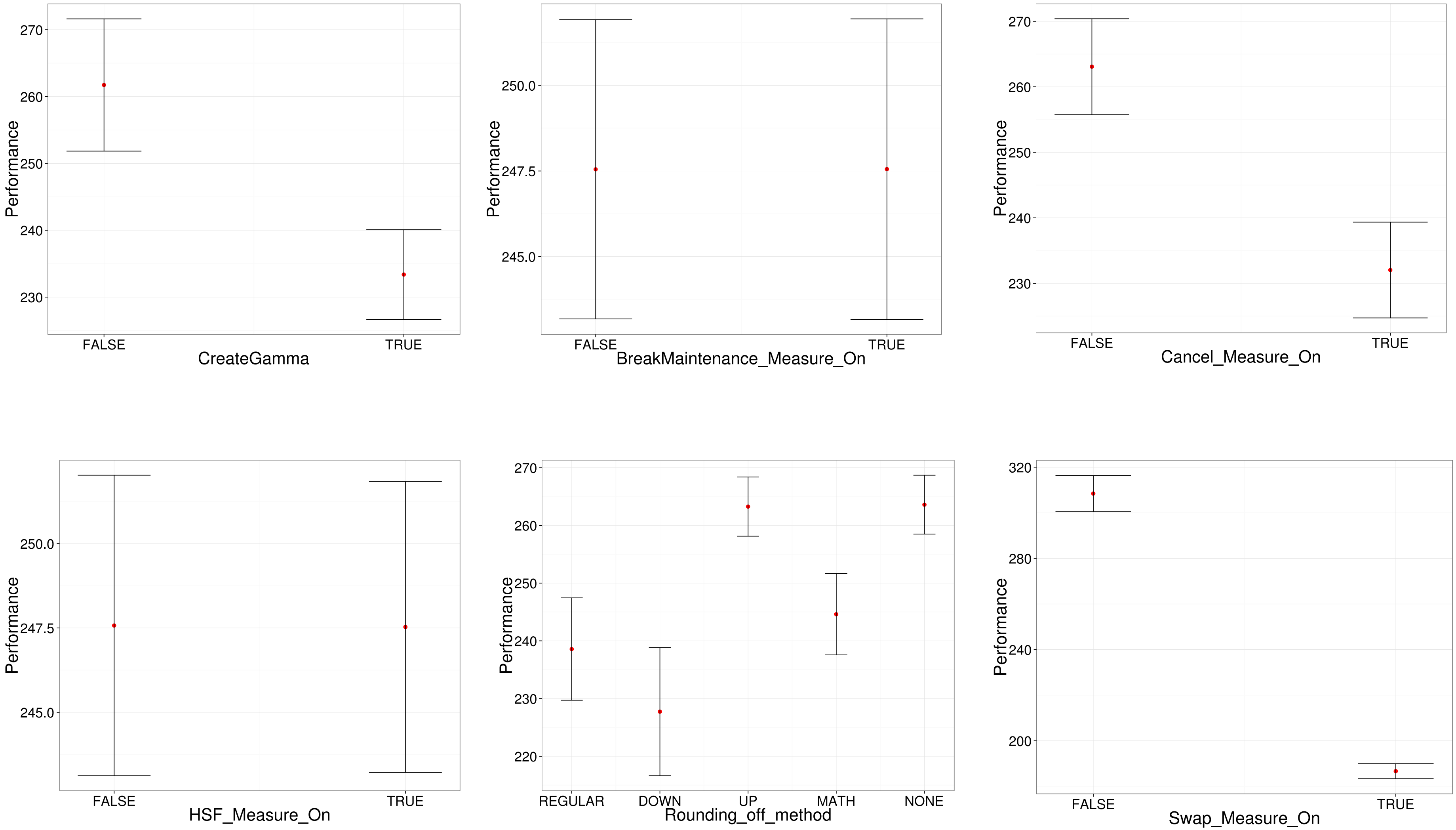
Stage 2: single-objective tuning

- Automatic Hyper-parameter Optimization
  - Optimization with **irace**
  - Optimization with **SMAC**
  - "Optimal" configurations found
    - Best was acc 241.268 vs 271.628
    - Probably because of interactions
  - Functional ANOVA (fANOVA) main/pairwise interactions

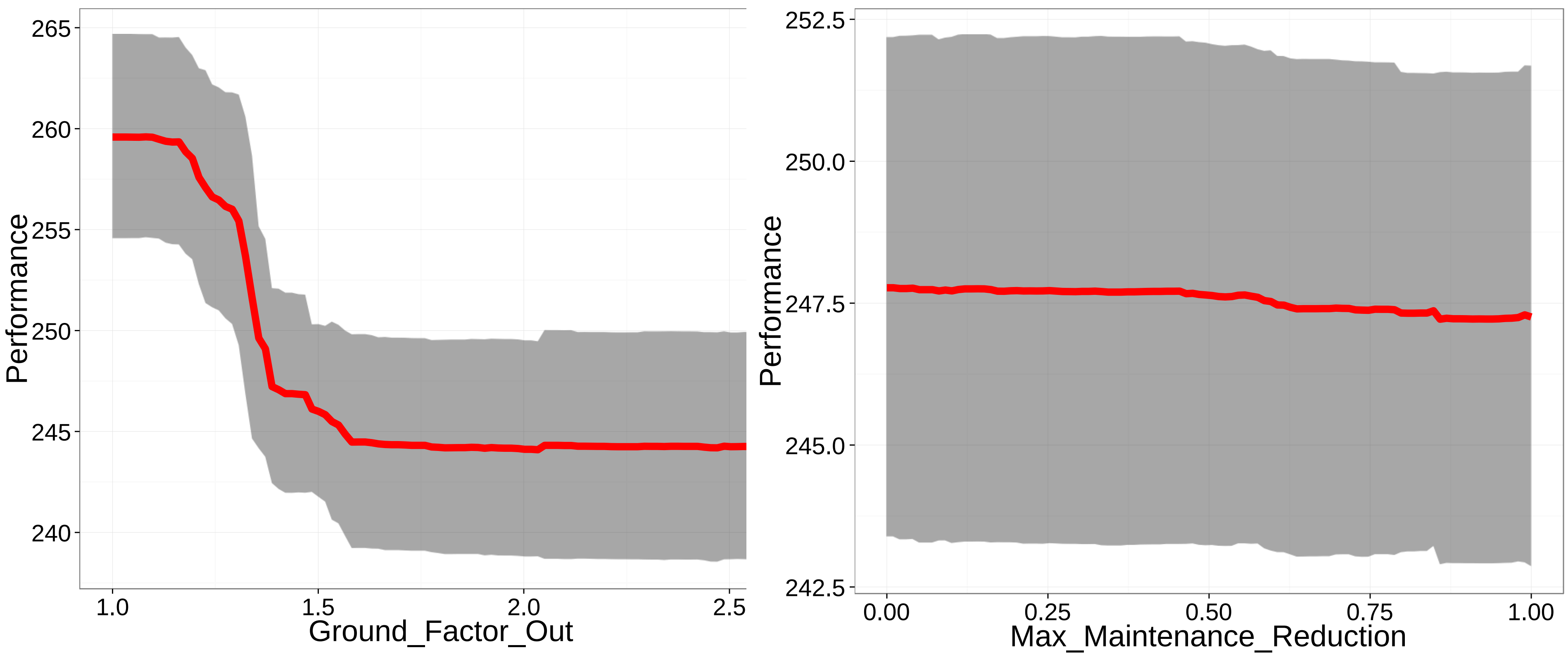
fANOVA main/pairwise effects

Sum of fractions for main effects 68.91%	
Sum of fractions for pairwise interaction effects 16.30%	
54.25% due to main effect	Swap_Measure_On
4.05% due to interaction	Swap_Measure_On x Cancel_Measure_On
4.02% due to main effect	Cancel_Measure_On
3.57% due to main effect	CreateGamma
3.55% due to main effect	Rounding_off_method
2.16% due to interaction	Swap_Measure_On x Slack_Selection_BB3
2.13% due to main effect	Slack_Selection_BB3
1.35% due to interaction	Slack_Selection_BB3 x Cancel_Measure_On
1.28% due to interaction	Swap_Measure_On x Rounding_off_method
0.84% due to interaction	Swap_Measure_On x CreateGamma
0.82% due to interaction	Slack_Selection_BB3 x CreateGamma
0.75% due to interaction	CreateGamma x Cancel_Measure_On
0.63% due to main effect	Ground_Factor_Out
0.55% due to interaction	Slack_Selection_BB3 x Rounding_off_method
0.48% due to interaction	Slack_Selection_BB3 x HSF_threshold
0.44% due to interaction	Slack_Selection_BB3 x HSF_threshold_In
0.36% due to interaction	Rounding_off_method x CreateGamma
0.33% due to main effect	HSF_threshold
0.33% due to main effect	HSF_threshold_In
0.33% due to interaction	Swap_Measure_On x HSF_threshold_In
0.31% due to interaction	Swap_Measure_On x Ground_Factor_Out
0.31% due to interaction	Swap_Measure_On x HSF_threshold
0.25% due to interaction	Rounding_off_method x Cancel_Measure_On
0.24% due to interaction	HSF_threshold_In x Cancel_Measure_On
0.21% due to interaction	HSF_threshold x Cancel_Measure_On
0.15% due to interaction	Rounding_off_method x HSF_threshold_In
0.15% due to interaction	HSF_threshold_In x CreateGamma
0.13% due to interaction	Rounding_off_method x Ground_Factor_Out
0.12% due to interaction	HSF_threshold x CreateGamma
0.10% due to interaction	Slack_Selection_BB3 x Ground_Factor_Out

Integer marginal distributions

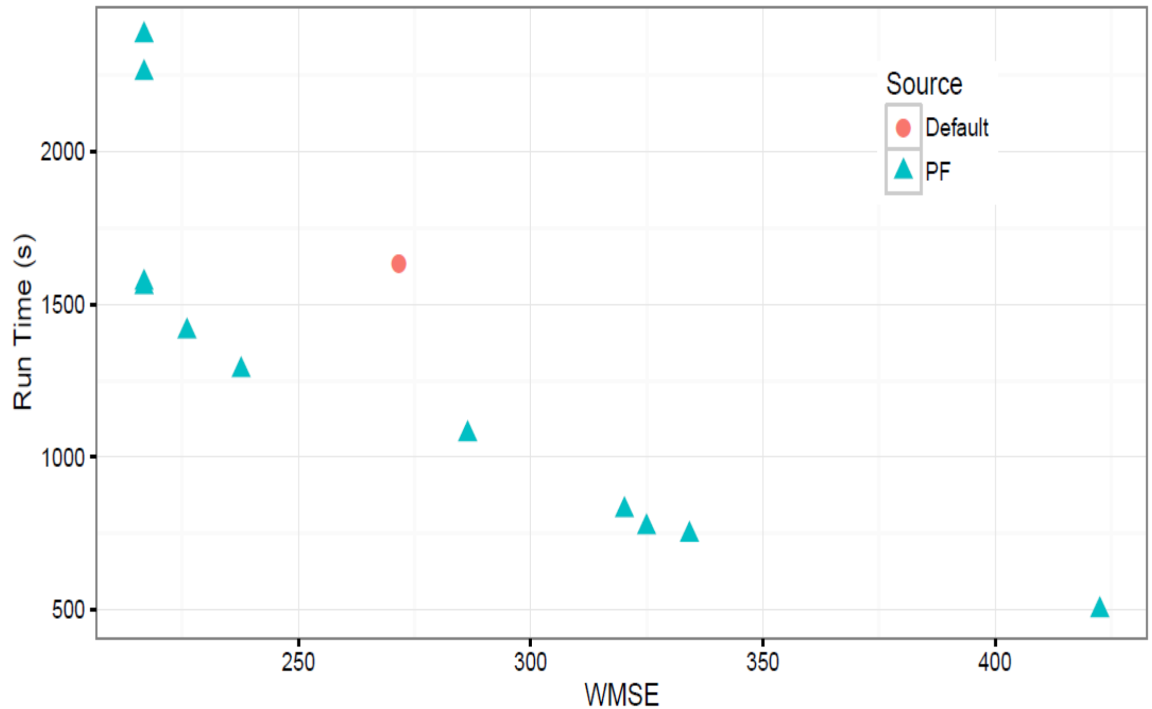


Continuous marginal distributions



Stage 3: Multi-objective Optimisation

- Improvement in both objectives!
- Highlighted params correspond with statistical analysis



MMR	GFO	SSBB3	MLS	HSFTO	HSFTI	MLC	HSFT	CMO	BMMO	CG	ROM	SMO	HSFMO	MSE	RunTime
0.25	1.8	90	9	3	49	9	13	1	0	1	1	1	0	216.748	2382.6
0.25	1.8	90	8	3	48	5	0	1	0	1	1	1	0	216.748	2258.4
0.2	1.8	90	8	12	51	10	2	1	0	1	1	1	0	216.748	1570.9
0.2	1.8	90	8	3	49	5	2	1	0	1	1	1	0	216.748	1557.2
0.25	1.8	50	9	28	51	2	0	1	0	1	1	1	0	225.988	1411.4
0.35	1.8	40	3	9	50	5	1	1	0	1	0	1	0	237.648	1284.5
0.25	1.55	60	12	25	47	9	8	1	1	0	1	1	0	286.428	1075.0
0.25	1.6	100	7	2	48	10	2	1	0	1	1	0	0	320.188	825.8
0.2	1.6	100	4	5	12	10	15	1	0	1	0	0	0	324.948	769.4
0.5	1.3	100	12	6	40	10	14	1	1	1	1	0	1	334.188	745.0
0.25	1.7	10	12	24	46	10	7	1	0	0	1	0	0	422.548	498.0

Where next?

- The results are good, but can we do better?
- Possible deep parameter tuning
  - Hundreds of parameters internally
  - Relatively simple to identify and apply further search
- Genetic improvement
  - DSOL library is open source, currently developing a project to explore GI on this
  - Prime candidates are searching the space of Java API classes such as containers, and lower-level improvements to source code

Conclusions

- Start simple! Having written the wrapper, parameter tuning is fairly easy to try
- The results were better than expected: improving both speed and accuracy
- *Value-added* optimisation – we added deeper analysis of the parameters that has been fed back to developers
- Ready for deeper GI improvement at code level

Thanks for listening

sbr@cs.stir.ac.uk

Questions?