

BEFORE DOING EXPERIMENTS

Presented by

Bing Xue

Bing.Xue@ecs.vuw.ac.nz

&

Harith Al-Sahaf

Harith.Al-Sahaf@ecs.vuw.ac.nz

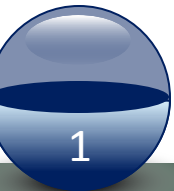
Evolutionary Computation Research Group

Additional presentations – 2016



Outline

- Parameter Settings
- Results
- Codes and Programs
- Grid Computing
- Others
- 10-fold Cross Validation



Parameter Settings

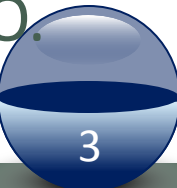
- Please **start** with **commonly used** parameter settings from the literature, or settings recommended by good papers.
 - Do not randomly pick up some parameters values **unless** you have good reasons to use them.
- Usually, your EC algorithms need to run **at least 30** runs (using 30 different seeds)
 - I always suggest at least 50 runs unless your experiment is too time-consuming.



Parameter Settings (continue...)

- DO record your random seeds to make sure you can **re-produce** the same results later if needed
 - Do NOT use **clock time** as the random seed

- Use the **same** random seeds to **compare** two different versions of the same approach:
 - E.g. two different GP algorithms: GP1 and GP2, run both of them for 50 times. Please make sure you use the same 50 random seeds for GP1 and GP2 to let them have the same starting points for fair comparisons.
 - It is not necessary to use the same random seeds if you compare GP with PSO.



Results

- Please **record all** useful results
 - Eg, the *gbest* in PSO, the best program from GP, the training, testing performances in each run (you may further check with your supervisors)
 - **Computational (training) Time of each run**: first generation to the last generation — not include test process
- Keep the results of the standard algorithm
- Do **keep all** the original results from **each run** of your GP/PSO/LCSs
- **Perform statistical significance tests**: Students T-test; Wilcoxon test
- **Do NOT delete** results unless they use too much memory, or they are wrong



Codes and Programs

- **Backup** different versions of your codes
- Make clear **documentation** of your codes
- Make clear **Reademe** documentation
- Do NOT perform operations **manually**
- Organise your files
- Best programming practices



Grid Computing

- Try to **learn** and **use** grid computing **unless** you experiments are super fast:
 - START: <https://ecs.victoria.ac.nz/Support/TechNoteEcsGrid>
 - HELP:
 - ...
 - Then Kevin Buckley
- Submit 35 jobs for 30 runs, or 55 for 50 runs **BUT**
- Make your executable files dynamic, i.e., avoid hardcoding, by using:
 - Command line arguments
 - Configuration files
- Results in different paths to avoid overwrite
 - clearly distinguish results from different codes
- Optimise the use of 120/150 **limited jobs** on Grid
 - Use array of jobs, e.g., 1 job comprises 30 runs



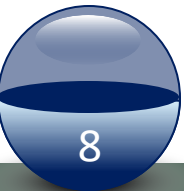
Others

- Andrew Lensen
 - **Multithreading** features in ECJ (or in PSO). I've always used it in my experiments and found it makes my experiments **3x quicker** with not much work, especially in ECJ (since you can evaluate multiple individuals in parallel).
 - The only **potential issue** is using too much CPU when other people are using their own machines, but I've **never** had any complaints using 3/4 of the CPU or so.



10-fold Cross Validation

- Deterministic
- Stochastic





THANK

You for your attention

Questions?