

Sample Experimental Results for A#1

We recommend two basic experiments, which will be enough for a good report. In both cases, we empirically investigate the scalability of ACT (Mailbox Actors) vs CSP (Hopac) models. Please feel free to investigate and discuss additional interesting scenarios, but this won't be necessary.

Test #1. Inputs are large and fixed; number of bands is variable.

All experiments use the same two strings, given by input1.txt and input2.txt. We only vary the number of bands: **b1,b2** = 50,50; 100,100; 150,150; ... 2000,2000 (keeping the same numbers of bands for each string).

Fig.1 shows possible results, if we follow the 1D model of Agents1Mailbox, which uses just one task for each horizontal band, i.e. **b1** tasks.

Fig.3 shows alternate results, if we follow the 2D model of Agents2Mailbox, which uses one task for each grid cell, i.e. **b1** × **b2** tasks - a dramatically larger number of tasks. Explanations? Conclusions?

Note that you were of course free to use any of these two models.

Test #2. Number of bands is fixed; size of strings is variable.

We use a fixed low number of bands - 100,100 – for which both Actors and CSP showed similar performance. We dynamically generate strings of varying size: 5000; 10000; 15000; ... 100000, and we use the same string for both sides (sidebar: in this case, the LCS is the string length :).

Fig.2 shows possible results, if we follow the 1D model, i.e. we use 100 tasks. The results based on the 2D model do not essentially differ, just show a wider gap between ACT and CSP. Explanations? Conclusions?

Experimental environment. These experiments have been run on a supercomputer. Your own conclusions need not be the same, e.g. your results will depend on the model you used (1D vs 2D), the size of your inputs, the numbers of bands, the capabilities of your home computer or lab machine.

Our test machine: 48 logical cores

24 physical cores: Intel Xeon Silver 4116 @ 2.10 Gz

F# Compiler version 10.2.3 for F# 4.5

Disclaimer. A professional testing would also require: (i) running a warm-up before doing the measurements (cf. JIT compiling, processor caches); and (ii) averaging test results for several runs. You are welcome to do this, however you are not required to do this for this report.

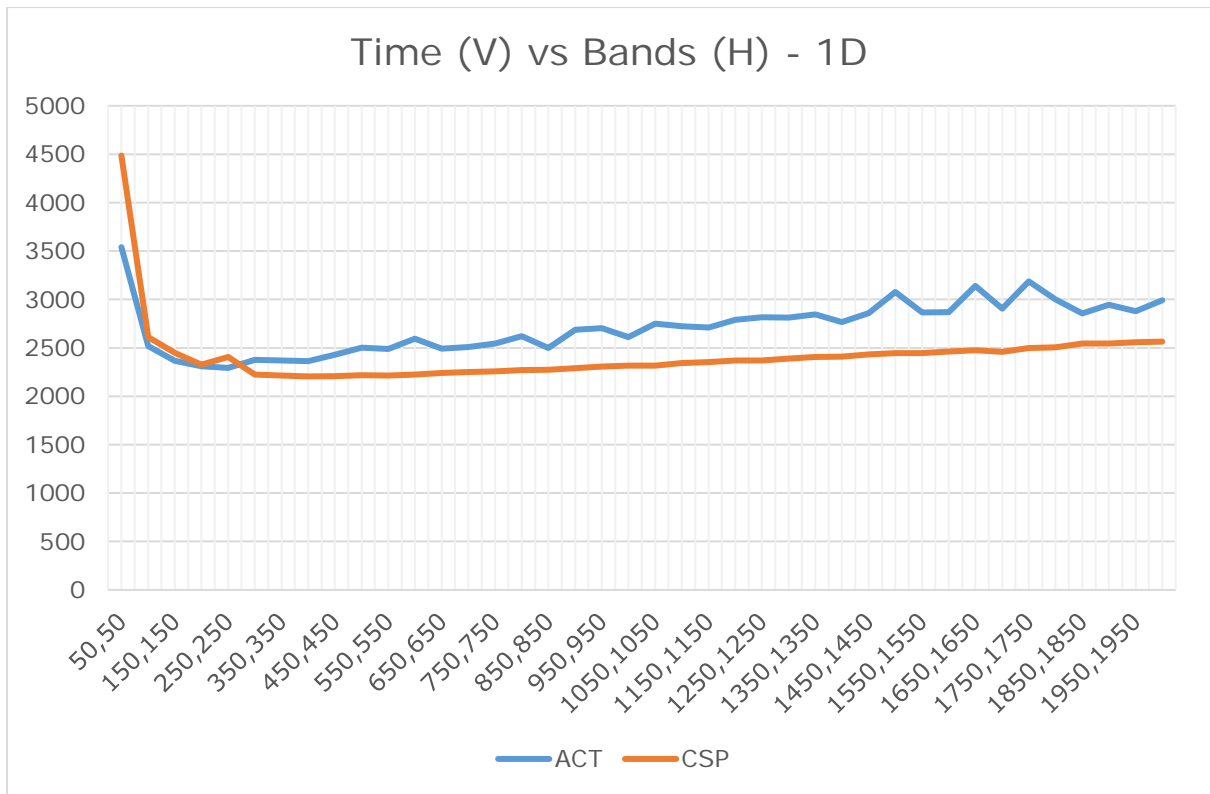


Figure 1 – 1D array of tasks

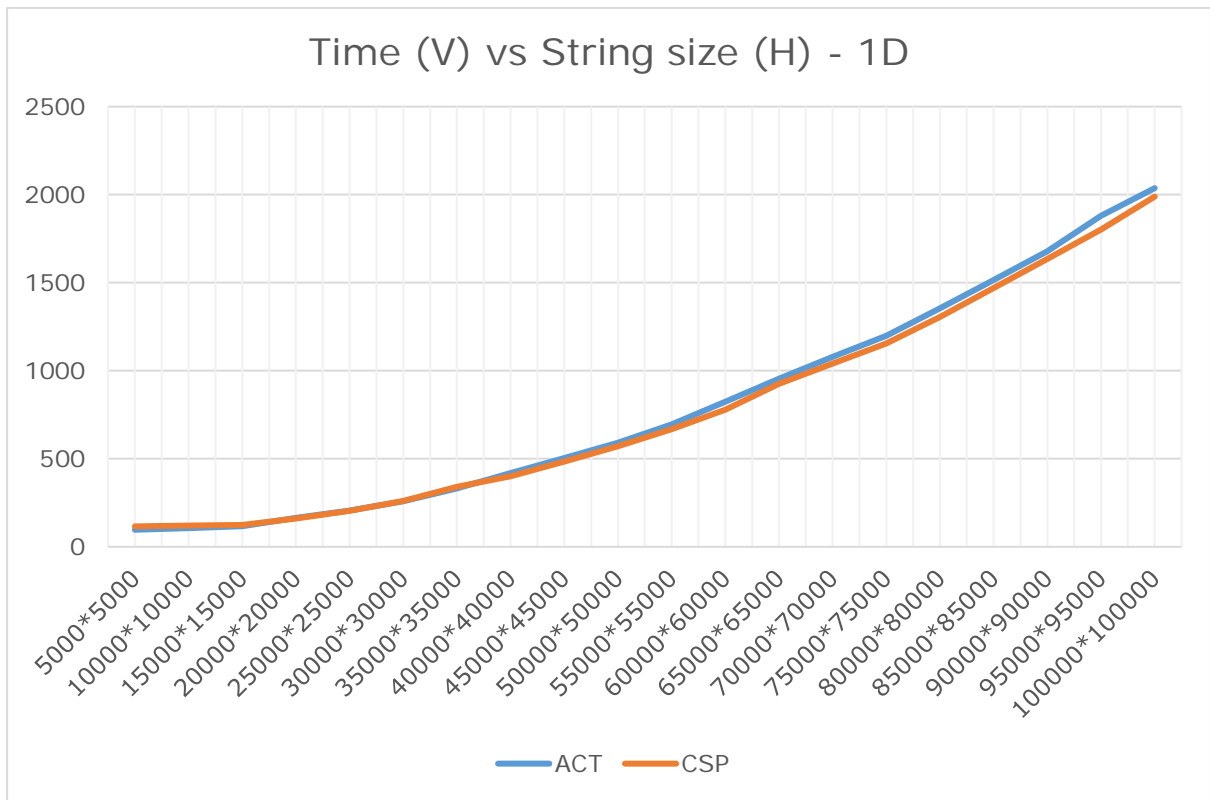


Figure 2 – 1D array of tasks

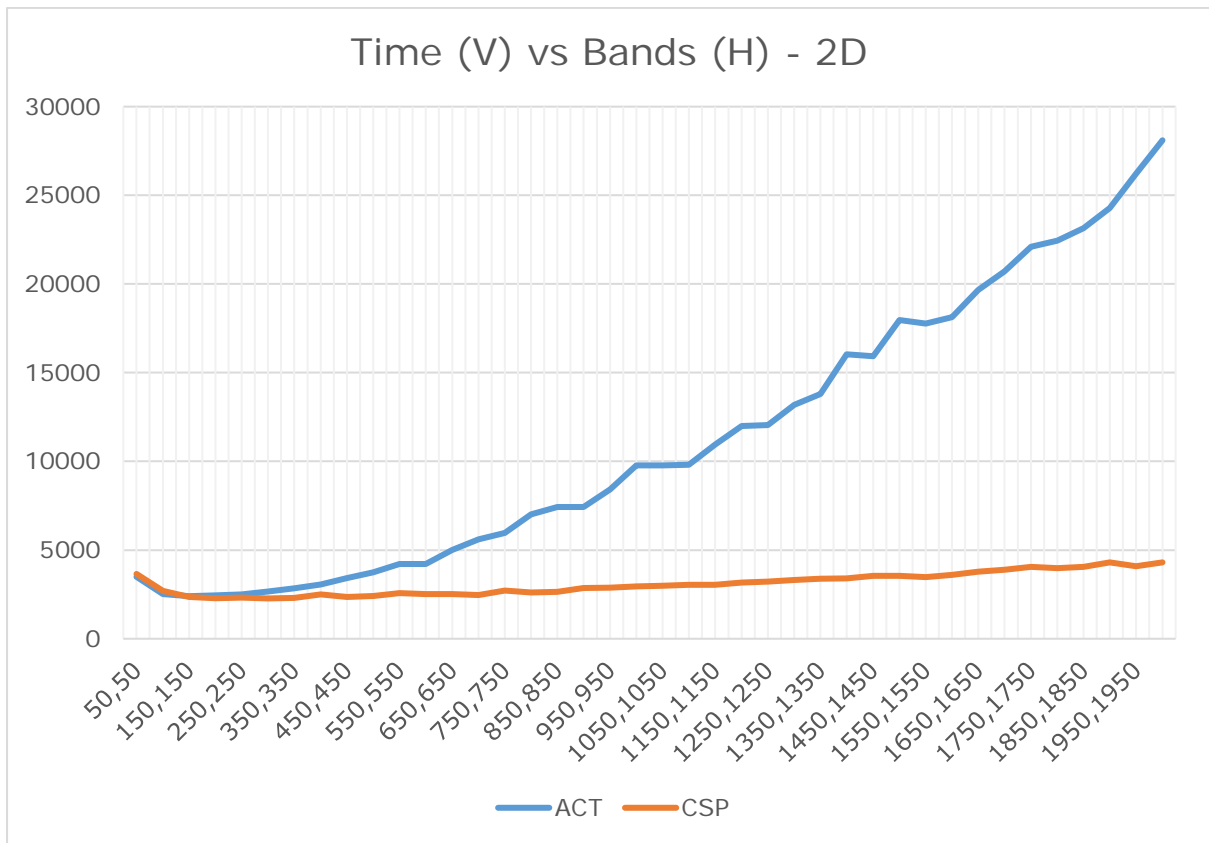


Figure 3 - 2D array of tasks