OLLSCOIL TEICNEOLAÍOCHTA BHAILE ÁTHA CLIATH
TU DUBLIN
TECHNOLOGICAL UNIVERSITY DUBLIN

Performance comparison between a distributed particle swarm algorithm and a centralised algorithm

Ciarán O'Loughlin - C13358011 -September 2021

Technical University Dublin
MSc in Advanced Software Development – TU060

Motivation & Research Question

Research Question:

- *“At what point are performance gains in running a particle swarm optimisation algorithm in a distributed environment outweighed by the time lost in network communications between multiple swarms?”*

Motivation for this Dissertation:

Discover the potential upper limit for PSO algorithms running on a singular machine.

Aim of the dissertation:

Generate results set of distributed and centralised PSO with increasing numbers of particles/Swarms

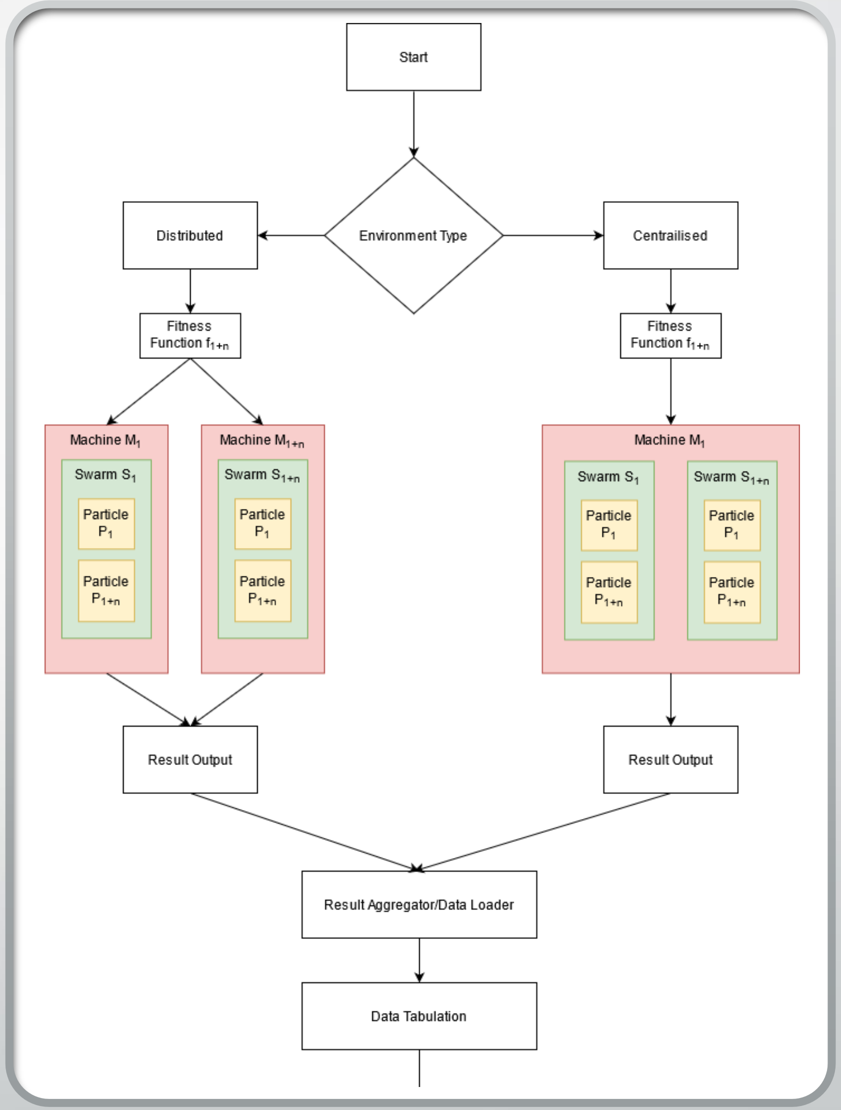
Cross compare the two, determine if there is an OTP(optimisation tipping point) where distributed is more time efficient than centralised.

Literature Review & Background

- PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles.
- Particles are then arranged into a ”swarm”.
- PSO is commonly used in search and optimisation problems. Plenty of research utilizing it in robotics/ real world applications, not just simulations.
- PSO update formula has 4 elements, social, cognitive, inertia and a random element
- Literature has a focus on various off shoots of PSO, dPSO, AGLDPSO etc. to solve specific problems
- Gap in Literature around distributed vs centralised.

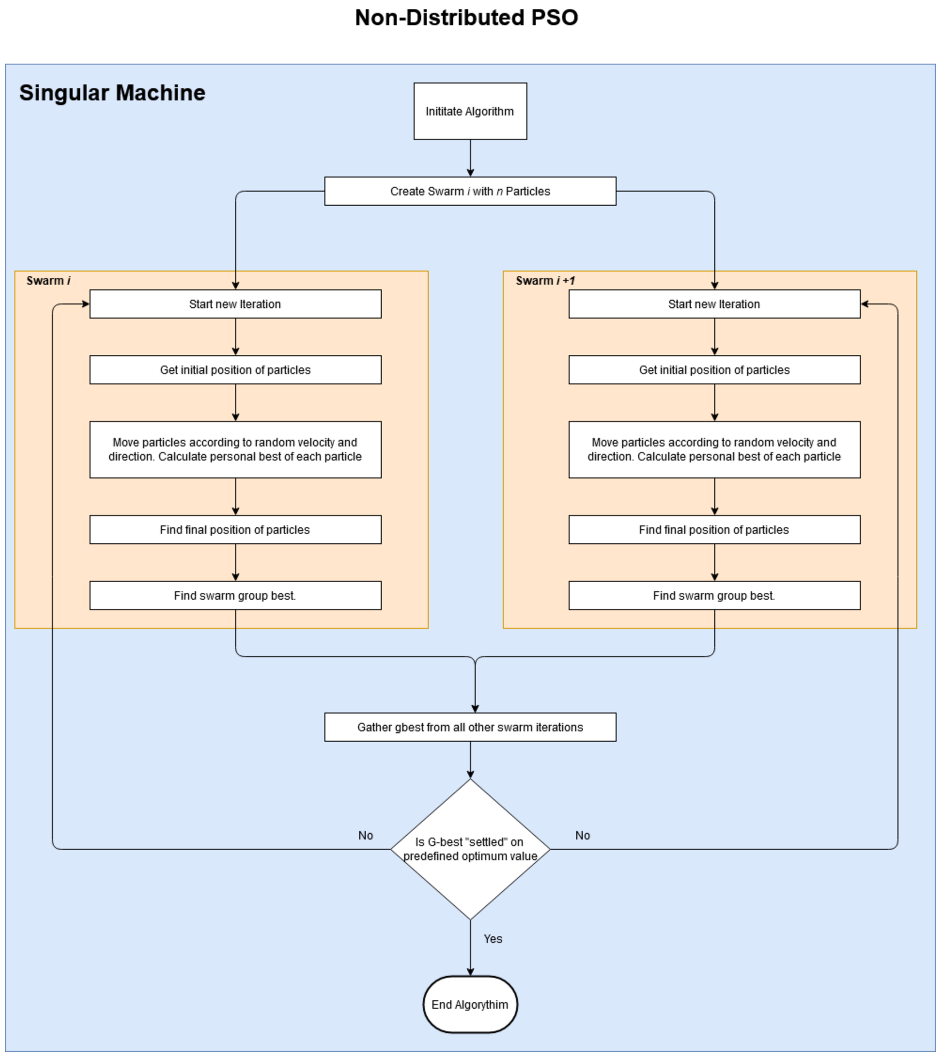
Design & Methodology

- Build two algorithm implementations
- Generate both results set
- Benchmark results between two results sets.
- Self build data aggregation/loader. Excel used to examine results.



Implementation

- Algorithm initializes x swarms with y particles.
- Particles move according to PSO formula.
- Current position is tested against specific fitness function.
- If current position is better than last position, global best for that swarm is updated.
- Iterations continue until Gbest value is “settled” across all swarms



Non-Distributed PSO

Singular Machine

Inititate Algorithm

Create Swarm *i* with *n* Particles

Swarm *i*

Start new Iteration

Get initial position of particles

Move particles according to random velocity and direction. Calculate personal best of each particle

Find final position of particles

Find swarm group best.

Swarm *i +1*

Start new Iteration

Get initial position of particles

Move particles according to random velocity and direction. Calculate personal best of each particle

Find final position of particles

Find swarm group best.

Gather gbest from all other swarm iterations

Is G-best "settled" on predefined optimum value

No

No

Yes

End Algorythim

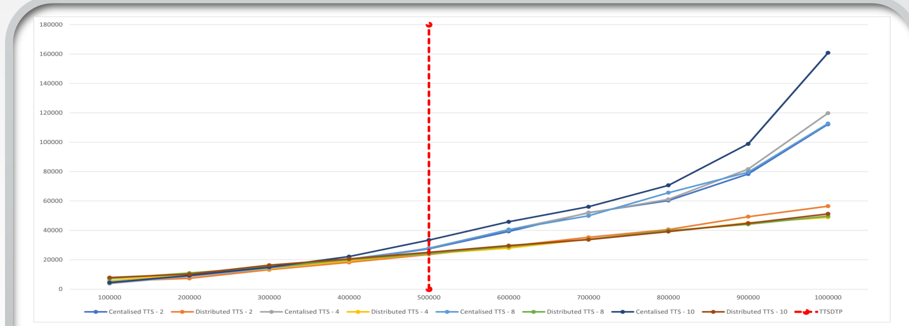
Implementation

- Java used for algorithm implementations.
- Deployed to a container in the cloud. In centralised implementation all swarms run on a single container. In distributed each swarm runs on a separate container. Digital ocean used as the cloud provider.
- Testing performed using postman/newman, with NodeJS used to aggregate results.
- Fitness functions tested:
  - Beales Function
  - Booths Function
  - Easoms Function

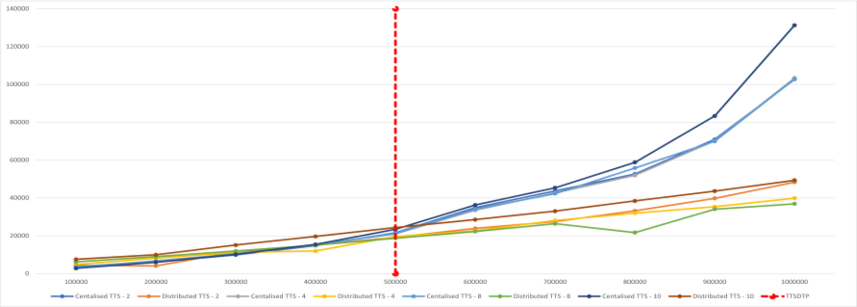
DigitalOcean

Results & analysis

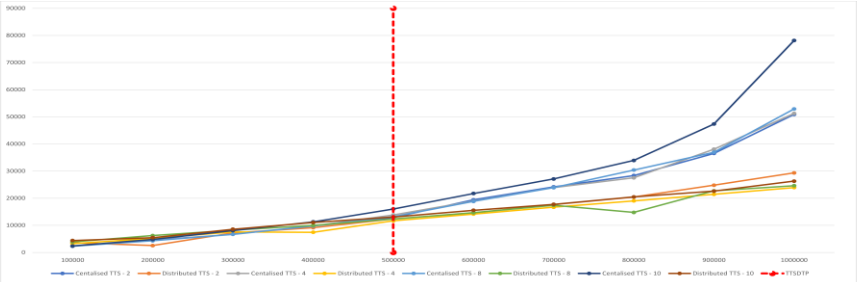
- 3 Fitness functions tested, with swarms ranging from 1-10, and particles ranging from 100,000-1,000,000
- Distributed implementation has a better TTS across all FF's at the 500,000 particle mark.
- At 1,000,000 particles
  - Beale's function -145%.
  - Booth's function - 152%
  - Easom's function - 123%.



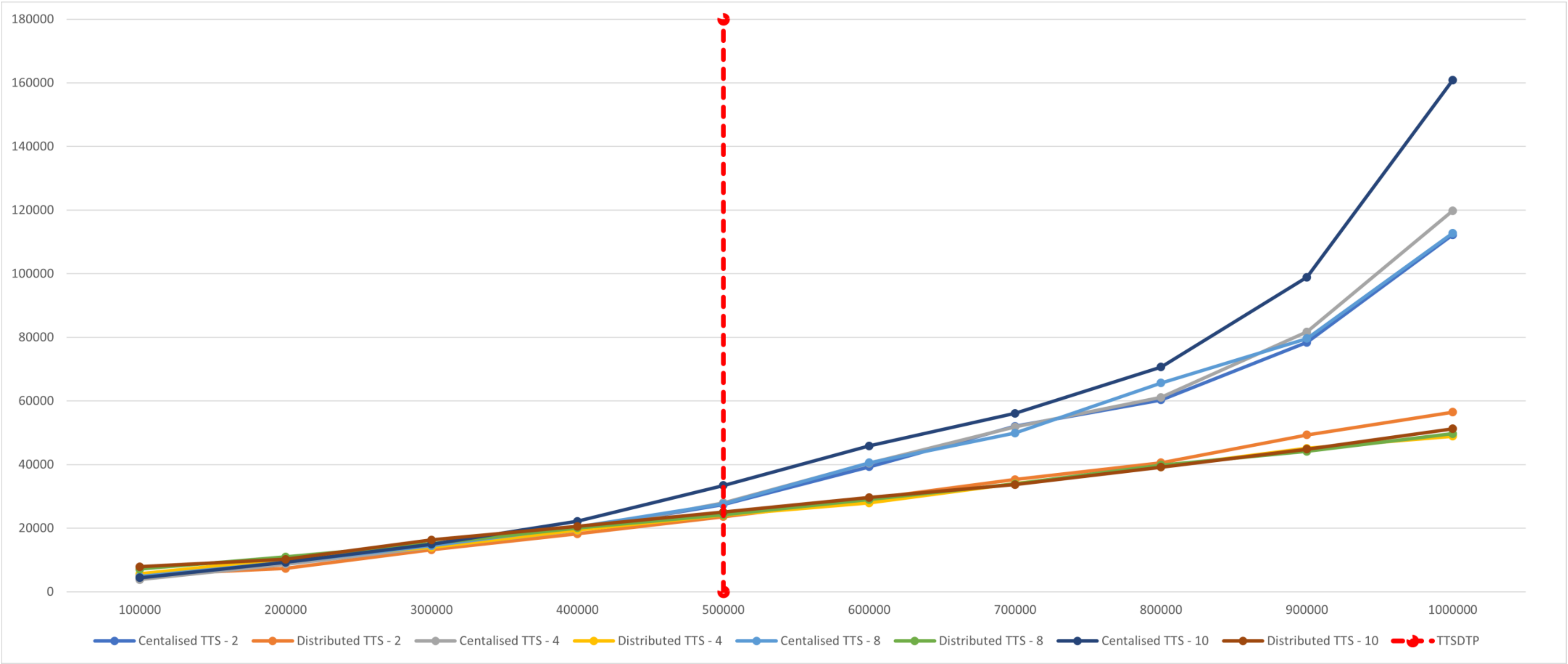
Beale Results Chart



Booths Results Chart



Easom Results Chart



Beale TTS Comparison Chart

Results & analysis

100,000 Particles

	TTS - 2 Swarms	TTS - 4 Swarms	TTS - 8 Swarms	TTS - 10 Swarms
Beale -Centralised	4205	3842	4775	4417
Beale -Distribributed	5291	5630	7153	7860
Booths -Centralised	2910	3243	3634	2805
Booths -Distribributed	4331	4846	6197	7541
Easom -Centralised	2364	2297	2313	2388
Easom -Distribributed	3551	3287	3925	4385

1,000,000 Particles

	TTS - 2 Swarms	TTS - 4 Swarms	TTS - 8 Swarms	TTS - 10 Swarms
Beale -Centralised	112258	119800	112751	160899
Beale -Distribributed	56502	48906	49696	51248
Booths -Centralised	102709	103364	102856	131270
Booths -Distribributed	48322	39878	36861	49330
Easom -Centralised	50887	51263	52918	78160
Easom -Distribributed	29327	23923	24623	26300

Contributions and Impact

- Validated Hypothesis and proved that there is a point where a distributed implementation is more time efficient than a centralised implementation.
- OTP found to be around 500,000 particles for multiple optimisation functions.
- Where this could have the most impact is for within a more practical experiment of a PSO, where very high levels of particles are required, possibly leaning towards more of a simulation problem than an optimisation problem.

Future work

Adapting

Adapting the implementation types to different languages/frameworks

Adding

Adding additional optimisation/search problems to test with.

Upgrading

Upgrading the container to have additional RAM/CPU power and retest to see changes in the OTP.

Utilising

Utilising different stopping criteria to test solution diversity between Centralised/Distributed.

Questions?