

	x	y
min	69	6
max	128	122
mean	126.557268722	19.3248898678
var	62.330420734	94.5651514196
stddev	7.894961731	9.72446149767

Table 1: Results for tile sizes, iterations=1000, population=50, matrix size=128x128

	x	y
min	128	5
max	128	128
mean	128.0	19.4601769912
var	0.0	121.854608818
stddev	0.0	11.0387775056

Table 2: Results for tile sizes, iterations=1000, population=100, matrix size=128x128

The objective was to find optimal tile sizes for the matrix multiplication, the tile sizes for dimension x and y were allowed to be different.

The matrix size is 128x128 and the number of workers range from 1 to 18. This time the multiplication  $C = A * B$  was performed with matrix A in row format and matrix B in column format (better cache utilization) - in contrast to the tests of the last week. SBX was used for the crossover, “parameter based mutation” was used as mutation operator.

128x128: For each parameter setting, the test was repeated 50 times to get more confidence in the results, 900 tests were performed (50 \* 18):

- 50 tests per setting
- 18 different worker sizes (from 1 to 18)

1000\_50\_256x256: For each parameter setting, the test was repeated 50 times to get more confidence in the results, 900 tests were performed (50 \* 18):

- 50 tests per setting
- 18 different worker sizes (from 1 to 18)

1000\_100\_256x256 and more: For each parameter setting, the test was repeated 5 times to get more confidence in the results, 90 tests were performed (5 \* 18):

- 50 tests per setting
- 18 different worker sizes (from 1 to 18)

	x	y
min	128	7
max	128	126
mean	128.0	19.0143646409
var	0.0	113.070511889
stddev	0.0	10.6334618958

Table 3: Results for tile sizes, iterations=1000, population=200, matrix size=128x128

	x	y
min	142	4
max	256	256
mean	254.686600221	25.1716500554
var	116.328137168	2519.74573006
stddev	10.7855522421	50.1970689389

Table 4: Results for tile sizes, iterations=1000, population=50, matrix size=256x256

	x	y
min	256	6
max	256	245
mean	256.0	20.3956043956
var	0.0	2163.71162903
stddev	0.0	46.5157137861

Table 5: Results for tile sizes, iterations=1000, population=100, matrix size=256x256

	x	y
min	256	6
max	256	251
mean	256.0	16.8823529412
var	0.0	1495.68027682
stddev	0.0	38.6740258677

Table 6: Results for tile sizes, iterations=1000, population=200, matrix size=256x256

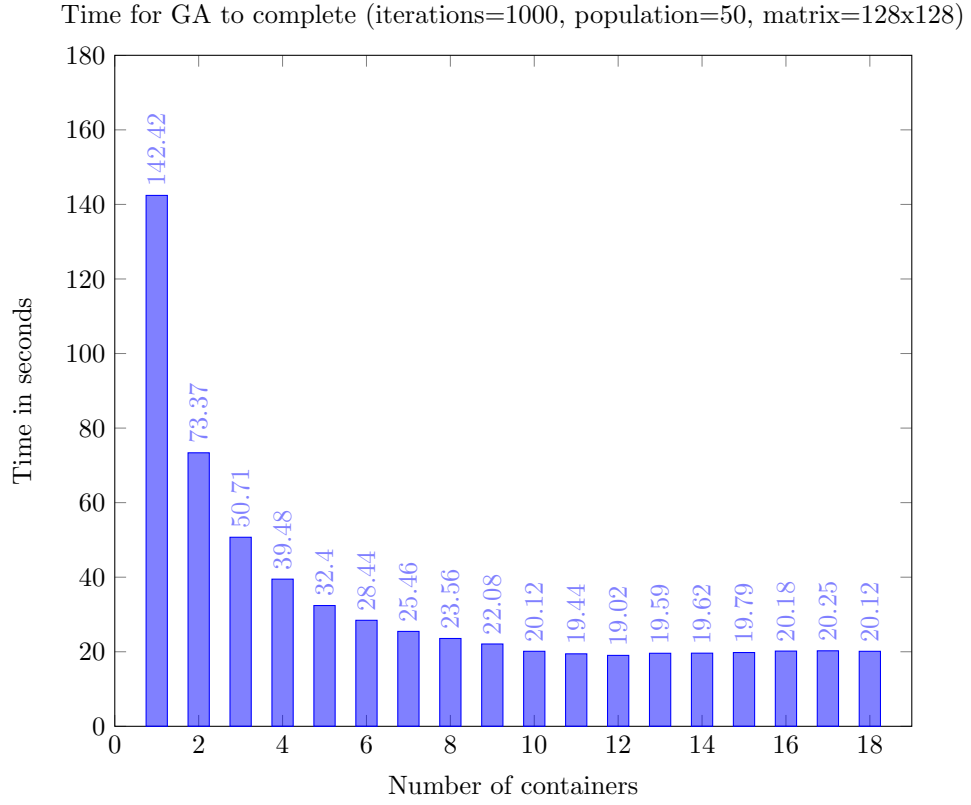


Figure 1: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=50, matrix size=128x128

Time for GA to complete (iterations=1000, population=50, matrix=128x128)

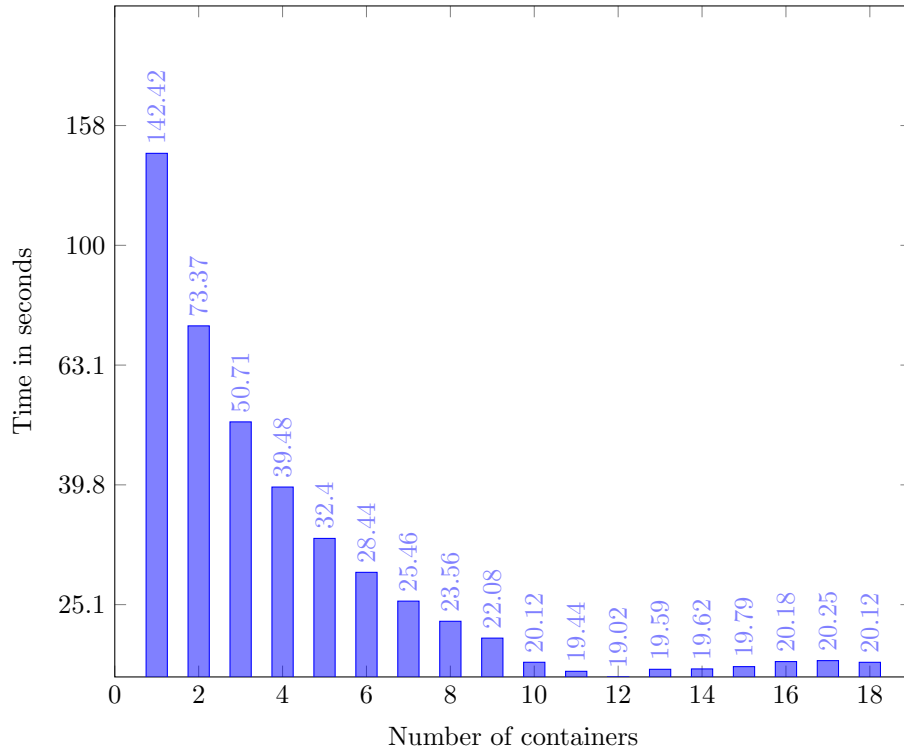


Figure 2: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=50, matrix size=128x128

Time for GA to complete (iterations=1000, population=100, matrix=128x128)

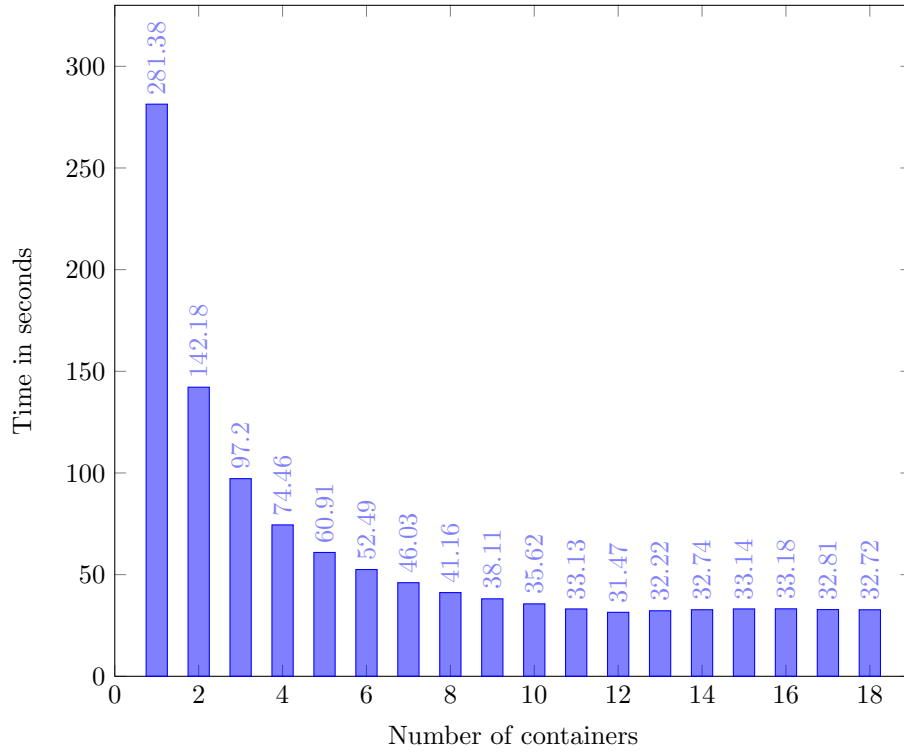


Figure 3: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=100, matrix size=128x128

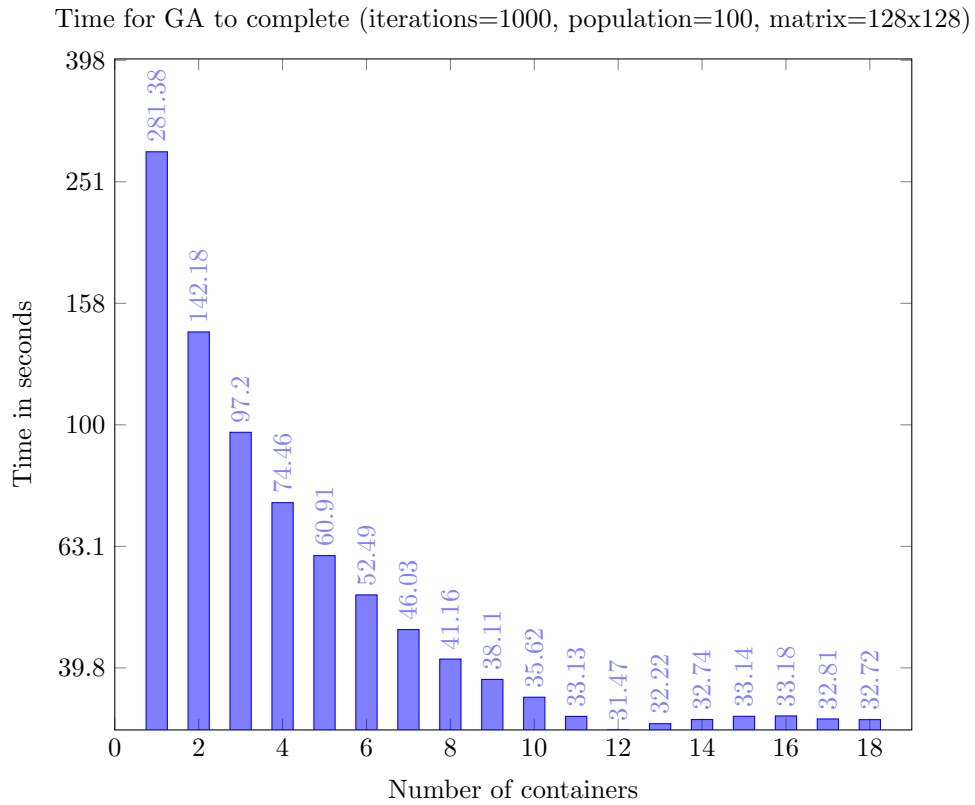


Figure 4: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=100, matrix size=128x128

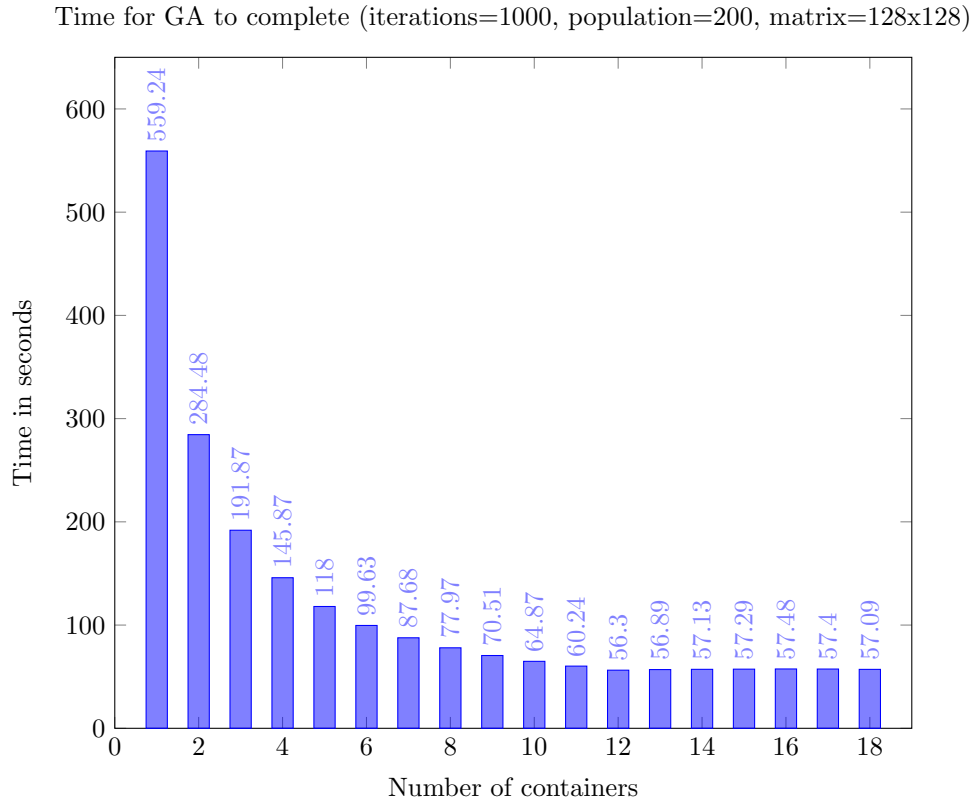


Figure 5: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=200, matrix size=128x128

Time for GA to complete (iterations=1000, population=200, matrix=128x128)

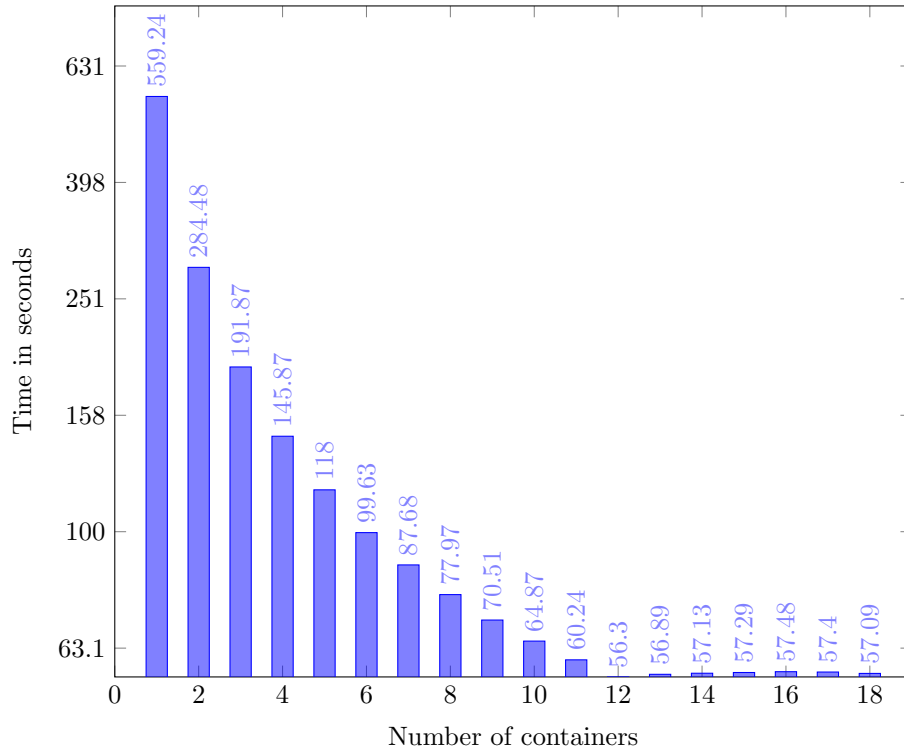


Figure 6: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=200, matrix size=128x128



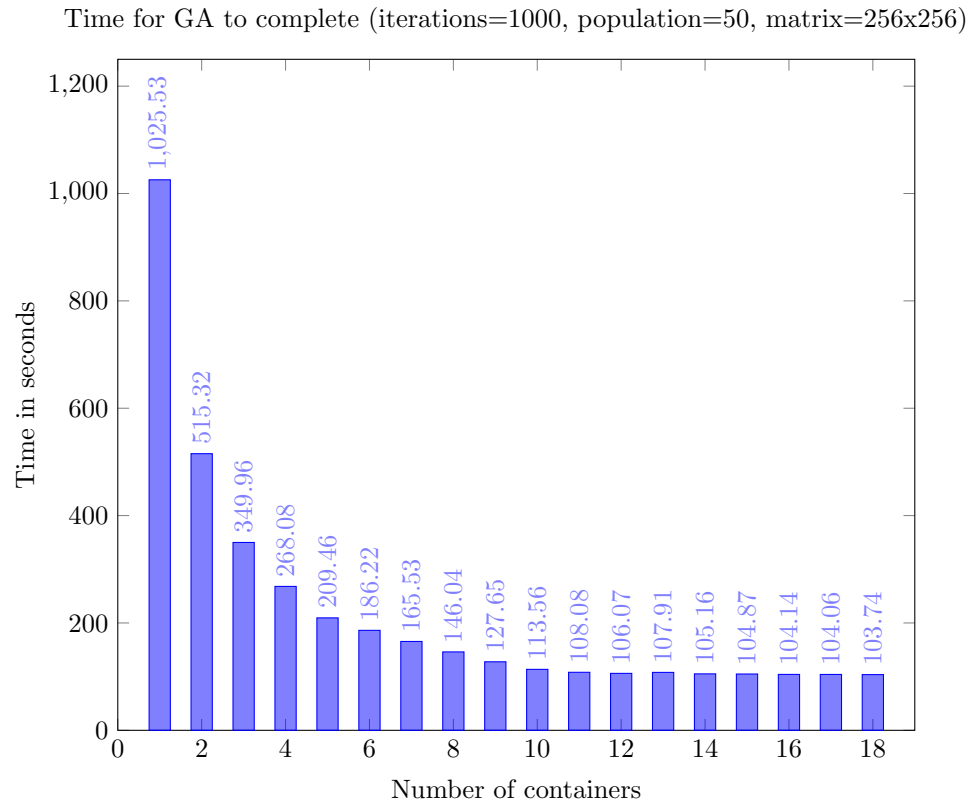


Figure 7: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=50, matrix size=256x256

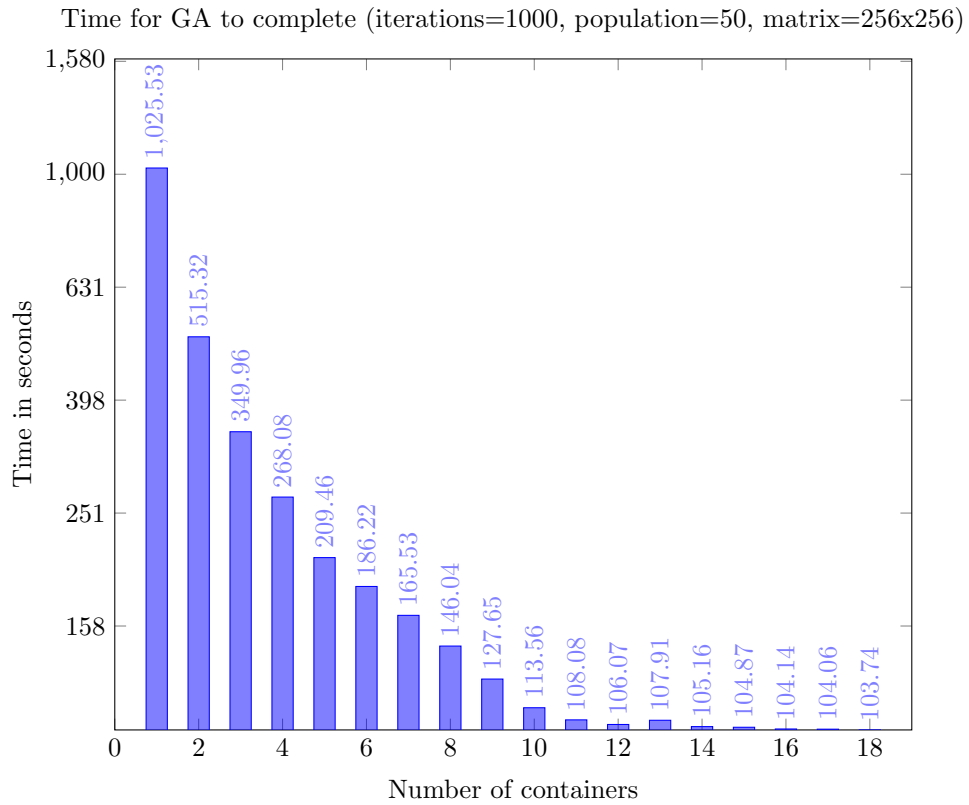


Figure 8: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=50, matrix size=256x256

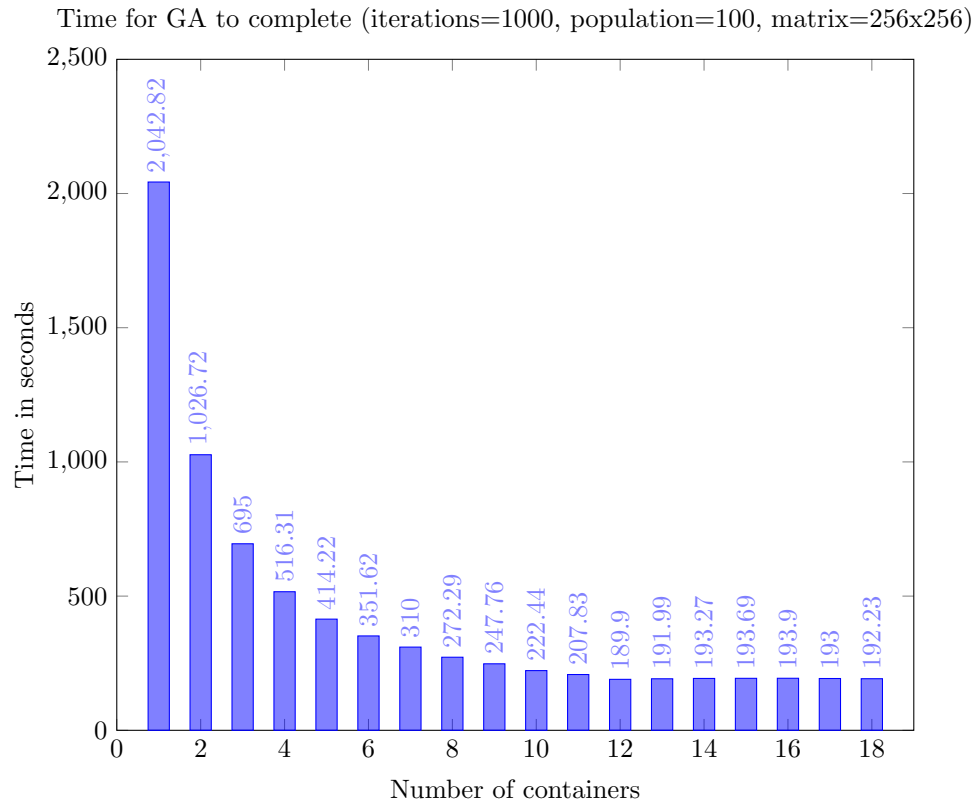


Figure 9: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=100, matrix size=256x256

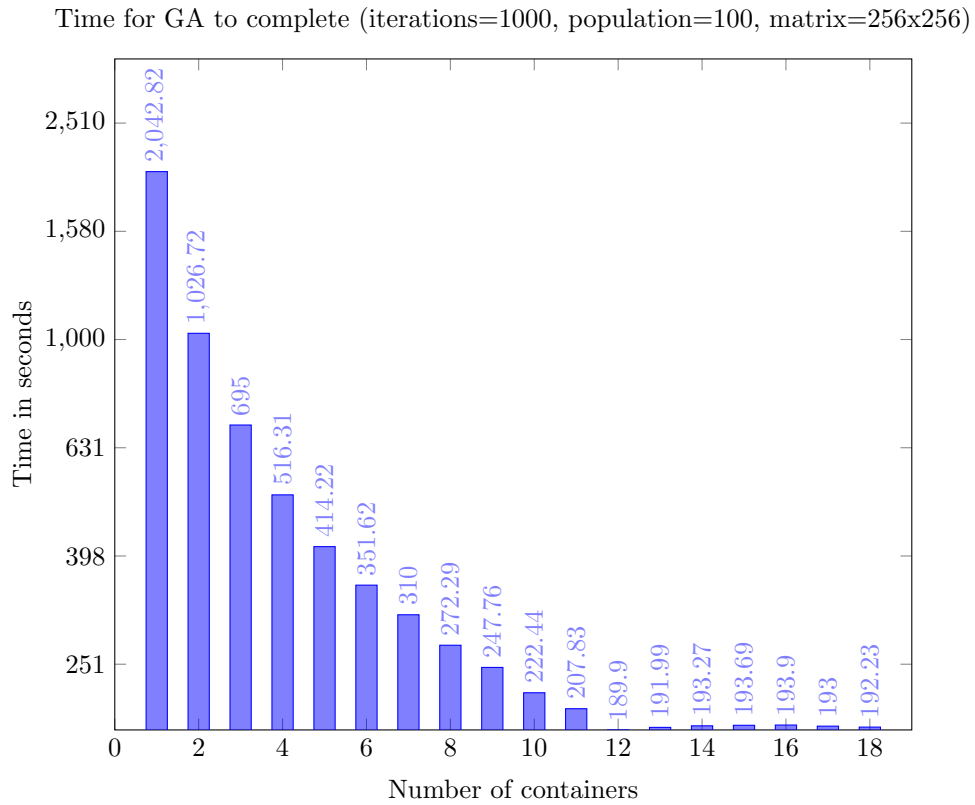


Figure 10: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=100, matrix size=256x256

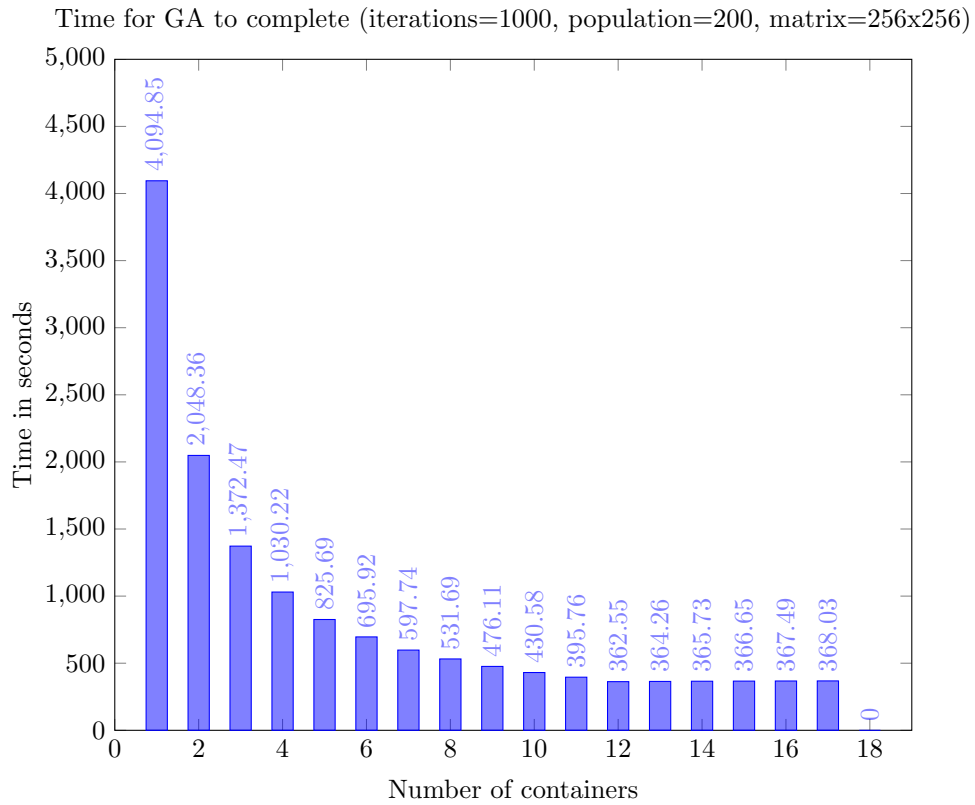


Figure 11: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=200, matrix size=256x256

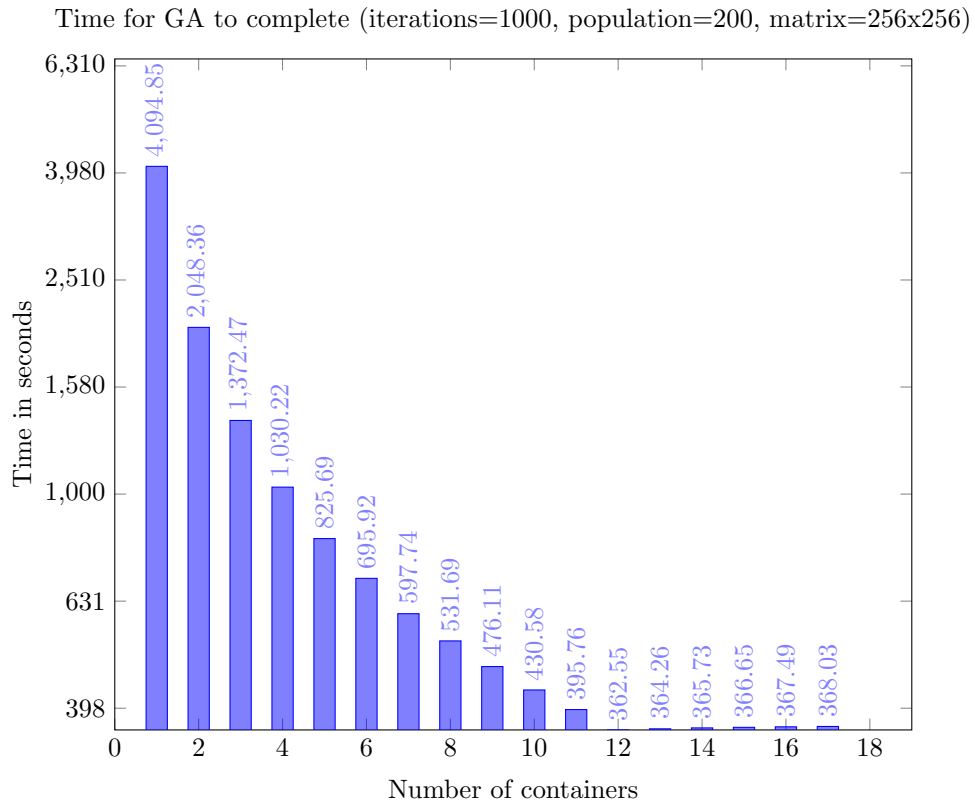


Figure 12: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=200, matrix size=256x256

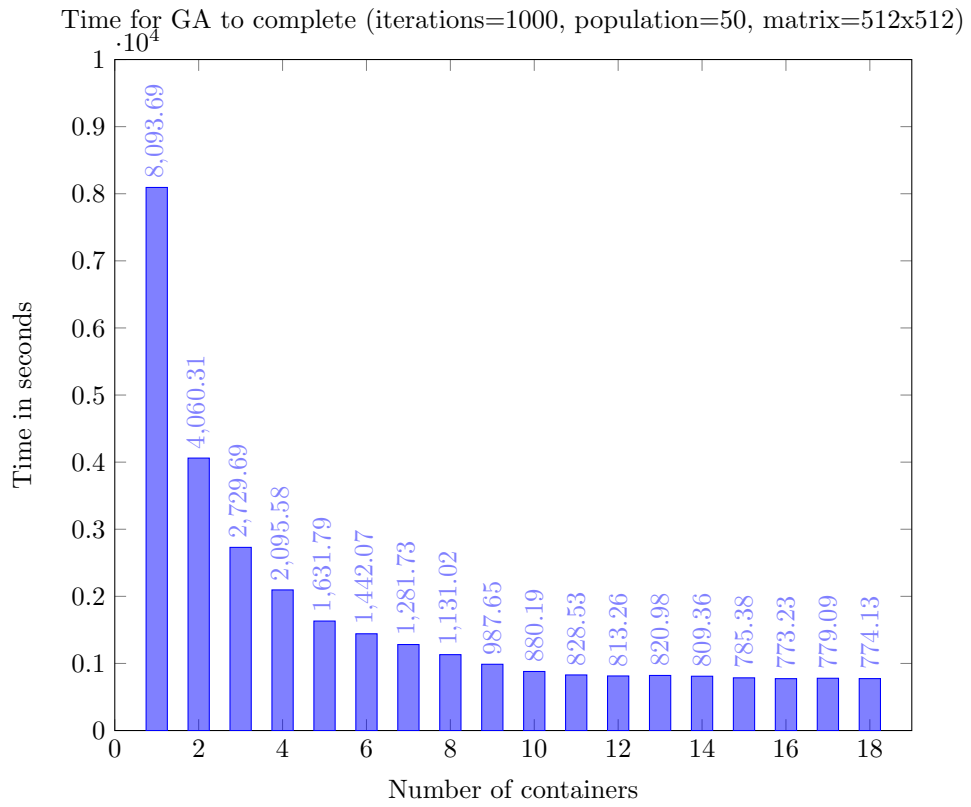


Figure 13: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish. iterations=1000, population=50, matrix size=512x512

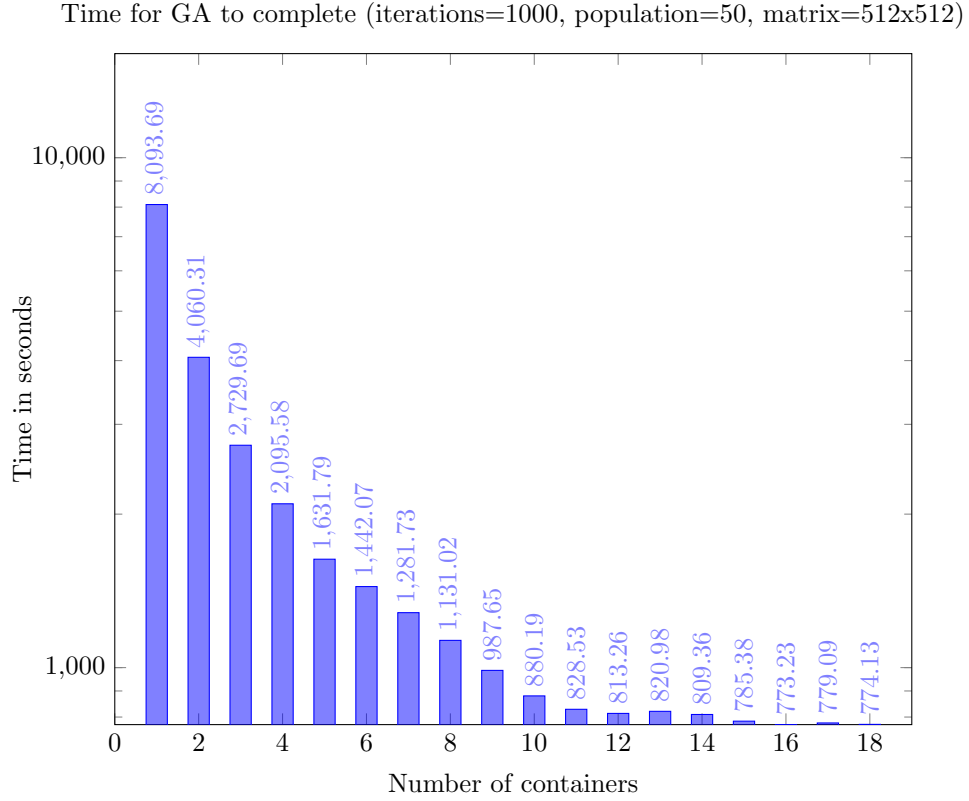


Figure 14: Find an optimal tile size for the tiled matrix multiplication. The x-axis shows the number of containers (workers), the y-axis shows the time it took to finish on a logarithmic scale. iterations=1000, population=50, matrix size=512x512

	x	y
min	258	3
max	512	512
mean	479.522222222	115.355555556
var	6292.11617284	22195.3846914
stddev	79.322860342	148.981155491

Table 7: Results for tile sizes, iterations=1000, population=50, matrix=512x512