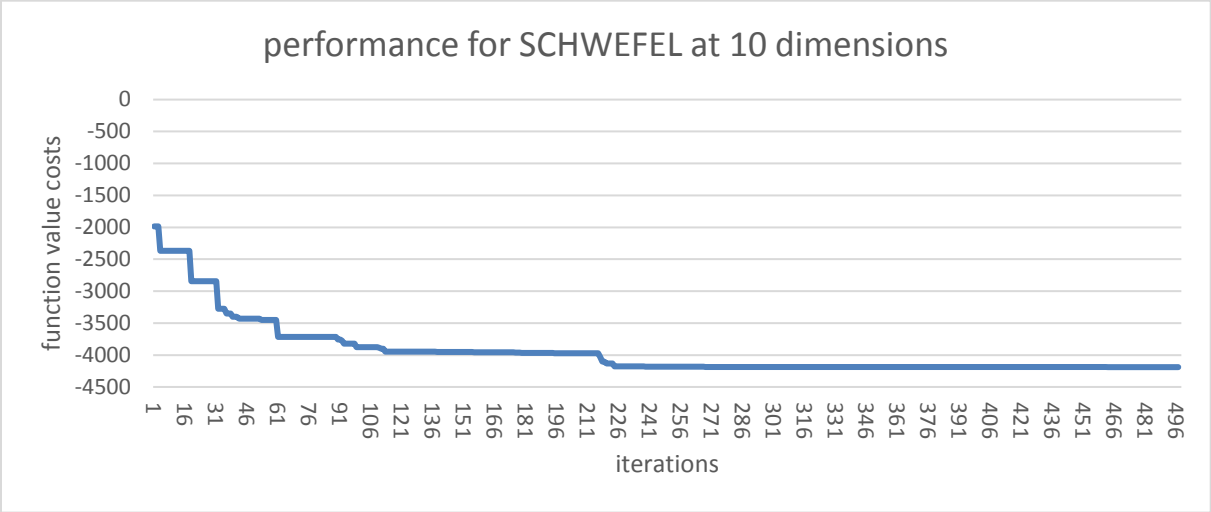
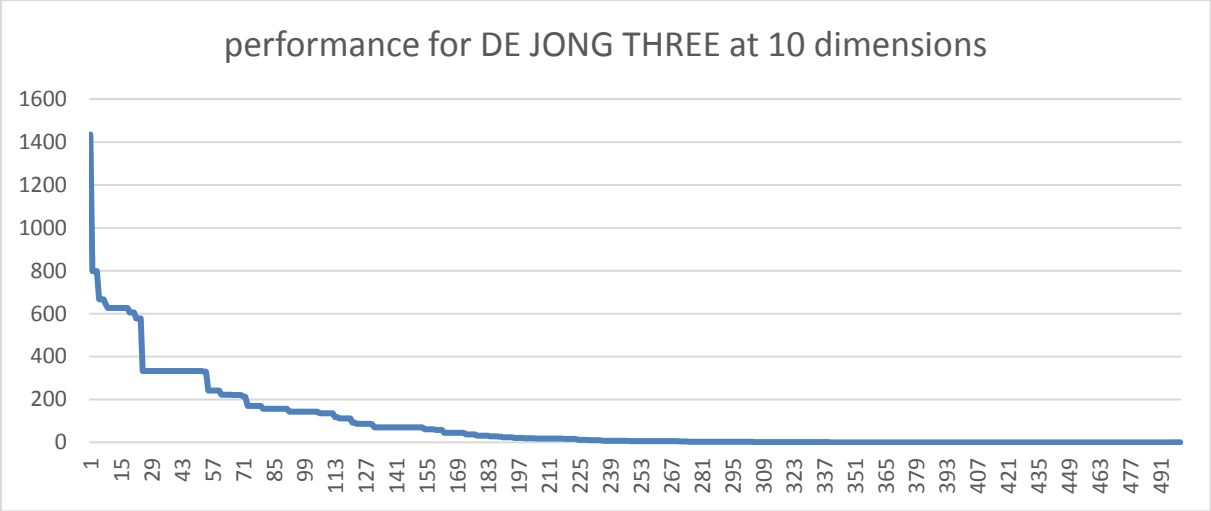
Evolutionary Algorithms: Particle Swarm Optimization in CUDA

Floyd Kretschmar

Performance:





Improvements:

velocityC:

I introduced a new parameter called velocityC (values between 0 and 2). It is multiplied with the entire new velocity when calculating the new velocity. Also exponentiateVelocityC is added as a parameter (Values 0 or 1) which indicates whether velocityC is exponentiated with (1 – currentIteration/numberofIterations). This exponentiation means, that the velocity is smaller in the beginning, and will only reach its full speed (or faster than full speed) in the last iterations.

Random values r1 and r2:

In the original implementation of the PSO the random values are independent from each other which can lead to situations where neither the global best or the personal best are

pursued or both are pursued at the same time. I improved this by making them dependend using r1 and (1 – r1)

Making pursuit of personal and global best dependent on the iteration:
In the first iterations of the algorithm there is usually very much improvement when pursuing the global best. But as soon as the population has locked in on a global best, better results are found when changing towards more of a local search around the personal bests of each population.
Therefore I made following changes:

double quotient = (double)currentIteration / (double)iterations;
double globalBestInfluence = 1 - pow(random, 1 - quotient);
double personalBestInfluence = pow(random, 1 - quotient);

These changes make sure that a larger emphasis is put on the global best in the beginning, but later on a bigger focus is put on the personal best, always using the random value as a basis.

Instead of letting solutions overreach bounds, change velocity using the bound range:
When a solution would exceed its bounds the following formula is used:

$$\vec{v}_i(\tau)_{new} = -\frac{d}{D} \cdot \vec{v}_i(\tau)$$

With d being the distance between the bounds and the point outside of the bounds and D being the entire range of the bounds.

<u>Parameters:</u>

Schwefel:		*DE JONG THREE:*
10	=> Dimensions	10
65	=> Number of Solutions/Particles	65
500	=> Iterations	500
0.9	=> The initial velocity coefficient	0.9
1.5	=> the personal best coefficient	1.5
0.9	=> the global best coefficient	0.9
1.1	=> velocityC	1
1	=> exponentiateVelocityC	0
1	=> ProblemId	1
-500 500	=> Bounds	-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500
-500 500		-500 500