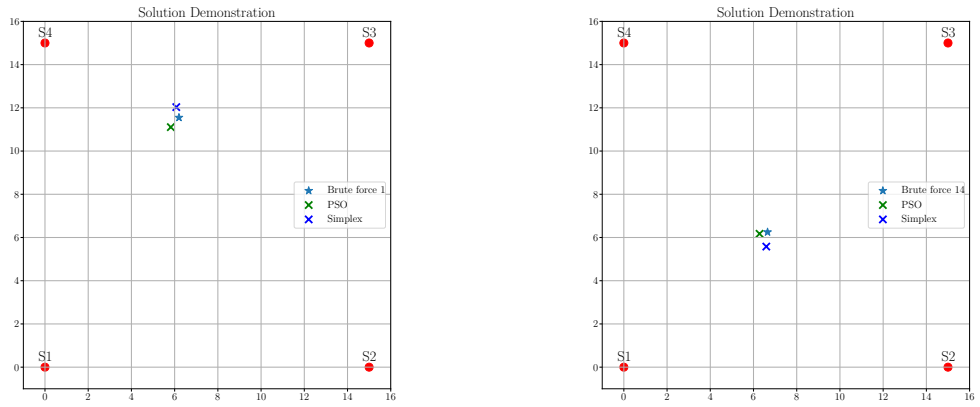
Part I

Results

For the sake of clarity, the two algorithms have been tested for number of iterations and for different hit IDs. To ensure the quality of the optimization method used I have implemented a discrete brute force approach to compute the deviation in both x and y.

0.1 General Testing

The following figure shows the localization done by the particle swarm optimization and the simplex method, besides ,of course, the brute force.



(a) Localization demo

(b) Different Hit

Figure 1: Localization Demo for Two hits

0.2 Deviation Test

Then, we test the deviation of the two algorithms from the brute force to see a visual trend over number of iterations. For the deviation in $y$and x:

$$d_i^x = x_i - x_{bf}$$

$$d_i^y = y_i - y_{bf}$$

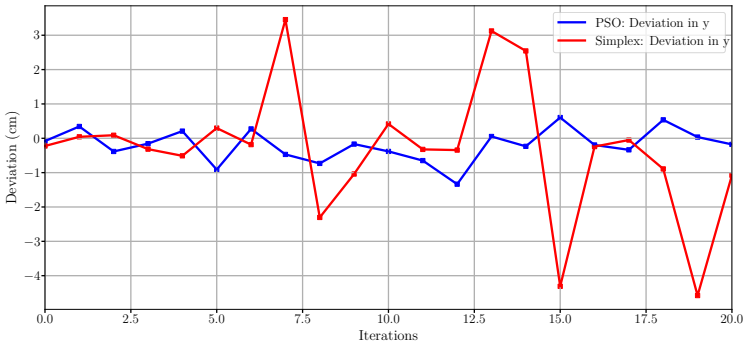


Figure 2: Deviation in y

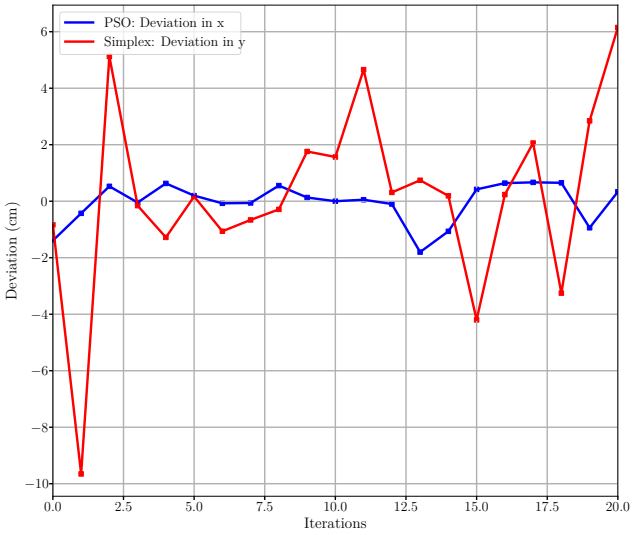


Figure 3: Deviation in x

0.3 Thoughts

Clearly, the PSO has an incredible advantage over simplex optimization in terms of deviation from the true location of the source. But let us take a moment here and analyze the situation more closely.
The algorithmic complexity of PSO is :

$$\theta(n \times m) \quad , \; m = generations \quad , \; n = particles$$

as we have to update every particle within a population over the course of the generation. For instance, I have set the number of generations $(m)$to 100 and number of

particles n to 60 which means around 6000 iterations have been consumed. On the other hand, the simplex method is linear. In fact, its algorithmic complexity can be formulated as :

$$O(n) \quad , \; n = maximim \; iterations$$

Which means that the algorithm can consume them all or any fraction of this depending on the convergence check we discussed last section. However, In average, the simplex optimization took 52 iteration to finish (no longer can better the solution), which considered amazing in terms of computational resources. PSO is superior to simplex optimization due to the swarm intelligence, thanks to the stochastic nature. Nevertheless, we can blend the two algorithms to exploit the fast convergence of the simplex method with the stochastic nature of the particle swarm, which can make (and i conjuncture) a stable algorithm.