Solution for OPA is based on optimization problem for phase shifters.

For optimization, Grouping based method, GA based method and OPA based method are developed.

Grouping based method is a common widely used method for OPA.

For each grouping stage, we should solve the optimization problems for multi variables.

When the number of antennas is 64, we should run the 6 stages for full phase compensation.

We can reduce the stages by grouping antennas.

For example, we can set number of stages as 5.

For this case, we should solve 5 optimization problems.

At first stage, we solve the optimization problems for two variables that represent the phase compensation for subgroups which contained 32 antennas respectively.

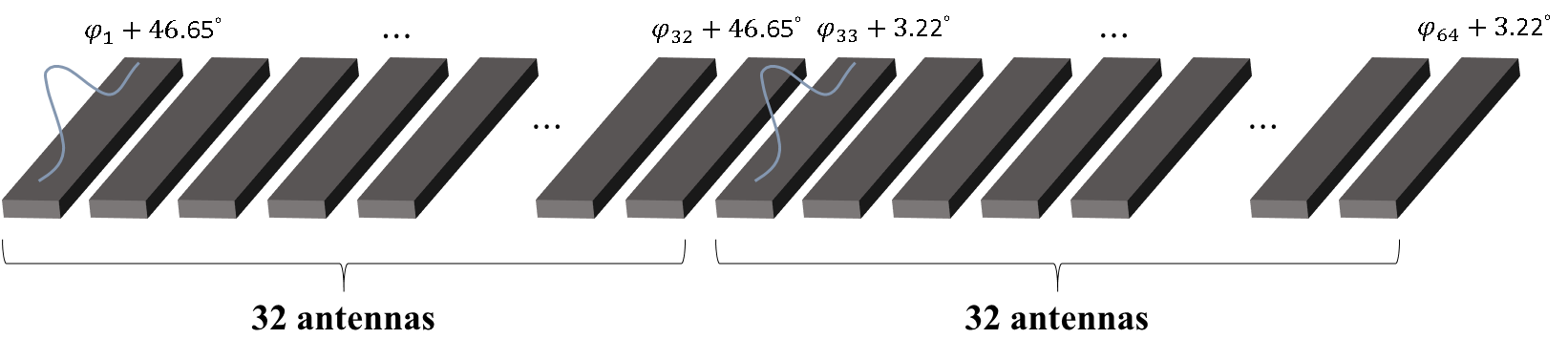


Fig 1. Optimization for first stage of grouping

At second stage, we solve the optimization problems for four variables that represent the phase compensation for subgroups which contained 16 antennas respectively.

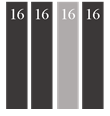


Fig 2.

In the same way, we should solve optimization problems for 32 variables.

For old grouping algorithm, they find the optimization points using iterating each variable.

For example, let’s see the case when we solve the optimization problem at second stage.

At this stage, there are 4 subgroups of 16 antennas and we should optimize 4 compensation angles.

Firstly, we iterate the first variable which represents the compensation angle for the first sub group by step of small angles and fixed other variables.

According to the reference paper, they select the step as 0.0001.

We find the optimization value for selected variable to optimize the fitness function.

Once iteration finished, we save the optimization value for the selected variable and select the second variable and do the same process for the second variable.

We repeated this process for all variables.

Fig 3 shows the flowchart of this process.

We repeated this for all stages.

To compare the performance for algorithms, we use the same phase errors and also same computer since we also compare the time.

We use core i 5 4590 to simulate algorithms.

At firstly, we use the PSLL as the fitness function and show the result in fig 10.

As you can see the performance of grouping algorithm is very low and takes 358.77s.

To upgrade the performance of grouping algorithm, we change the fitness function.

Since the heights of peaks are much different, we make the fitness function considering the height of peak.

The fitness function we proposed is as following

, where w is the weight factor.

We tested this fitness function for original grouping algorithm and get good result for w=300.

Fig 11 shows the result of this case.

From the result we can see that the performance is updated largely.

The psll is about -12db.

To upgrade it, we tested iterating.

We iterated this algorithm several times and check if the psll is upgraded.

However the psll is not upgraded.

To upgrade psll, we developed new iterating algorithm.

We set the fitness function as different by changing the weight factor in the fitness function for all iterations.

Fig 4 shows the flowchart and fig 12 shows the result of this algorithm when we set the iternum as 12.

As you can see, the psll is upgraded to -14.69db however the time is increased as 1628s.

We get the result for GA and PSO to compare the performance with grouping algorithm.

We use the PSLL as fitness function and show the results in fig 13 and 14.

They take time 248s and 328s.

From the testing of old grouping algorithm, we can see that it has bad performance when it use the same fitness function with PSO and GA since PSO and GA has the ability to avoid local optimization points.

To upgrade performance, we propose the iteration of fitness function and in this case we get similar result with PSO.

However the calculation time takes 5 times more than PSO.

From this result, we are going to propose the new optimization method based on grouping algorithm which can upgrade the performance without increasing the time.

In old grouping algorithm, they iterate all variables by step angle for each stage.

For each iteration value, we should calculate the psll value.

Since step angle is too small it takes much time to iterate all.

For instance at the fifth stage, we should iterate 32 variables since there are 32 subgroups at this stage.

For each variable, if we set the step as 0.0006 the iteration number is calculated as

So we should calculate the psll for

times.

To reduce time, we can use muti-variable optimization methods such as GA and PSO.

However GA and PSO also take much time as the above result shows.

So i select the simple optimization method and combine it with the iteration of fitness function.

As we can see in the above tests, iteration of fitness function makes to avoid the disadvantages of local search algorithm.

I select the 1-Dimension search algorithm to optimize the fitness function at each stage.

Fig 5 shows the flow chart.

Firstly we test the grouping algorithm combined 1-dimenstion search algorithm without iteration of fitness function.

The result of this is shown in fig 15.

From the result we can see that it reduces the time much.

At last, we combined fitness function iteration algorithm with it and show the result in figure 16.

We get psll of -15.3db and also take time only 154s.

Table 1 shows the total results.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | old grouping  (using psll as fitness value) | old grouping  (updated fitness function) | Old grouping  (combined iteration of fitness function) | GA  (psll as fitness value) | PSO  (psll as fitness value) | grouping algorithm+1-d search algorithm  (updated fitness function) | grouping algorithm+1-d search algorithm  +iteration of fitness function |
| Psll(db) | -0.128 | -12.08 | -14.69 | -13.8 | -14.96 | -12.57 | 15.25 |
| Time(s) | 358.77 | 359.02 | 1628 | 248 | 328 | 64 | 154 |

From the table, we can see that the new algorithm has best performance of psll and time.

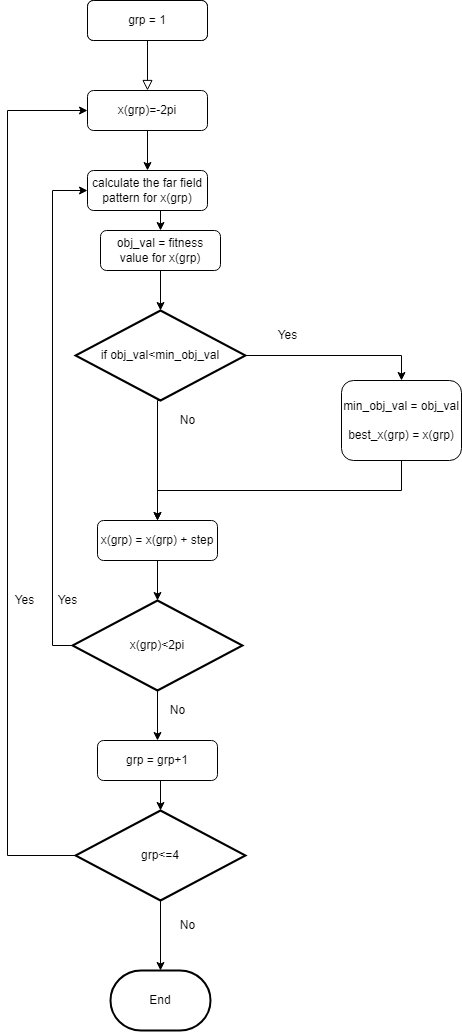


Fig 3. Flowchart for old grouping optimization

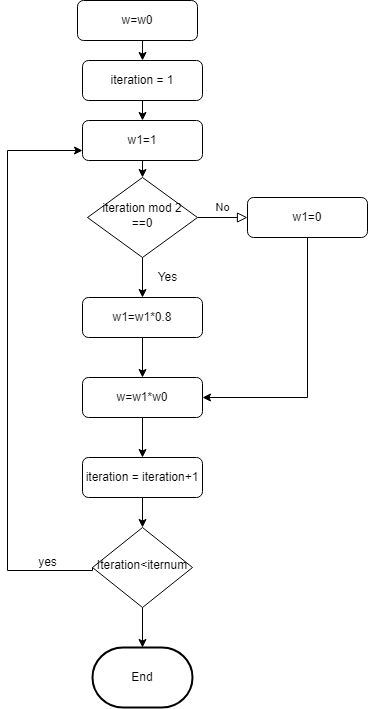


Fig 4. Iteration algorithm of weight factor in fitness function

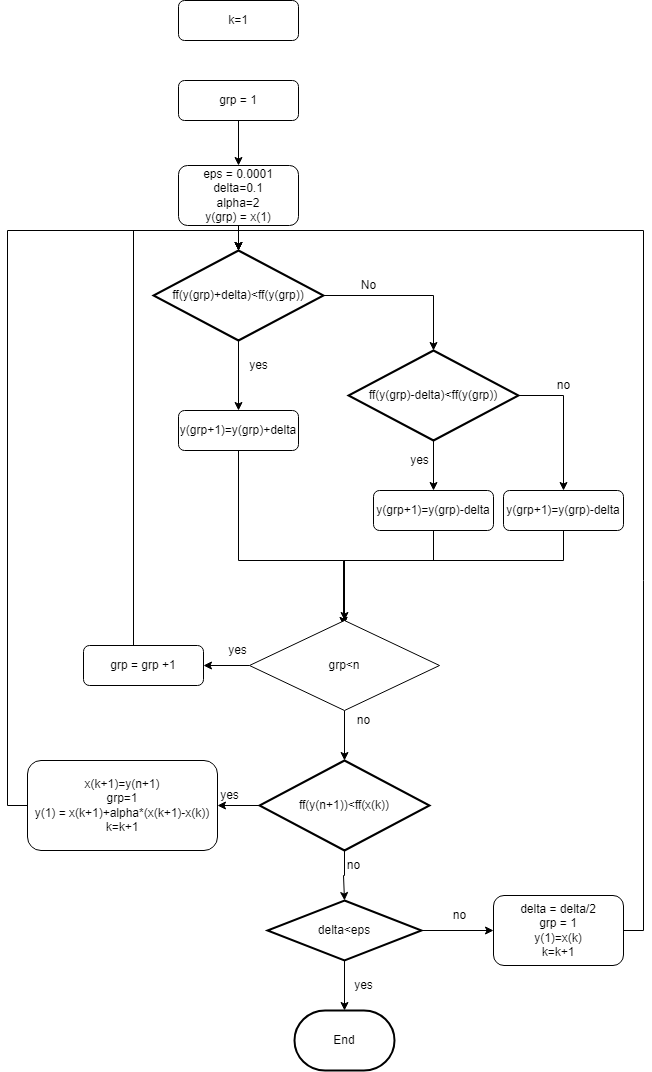


Fig 5. new search algorithm

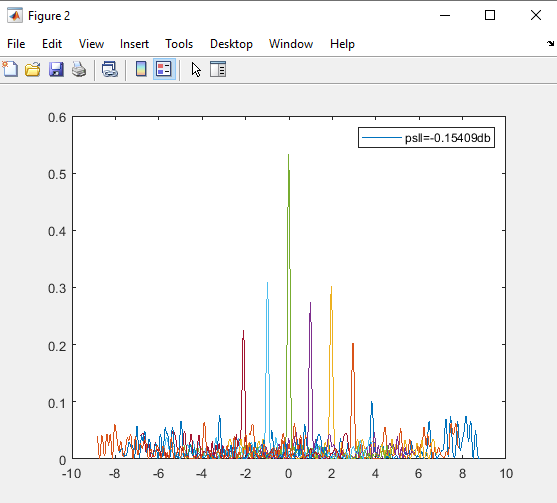


Fig 10. Result for original old grouping (358.77s)

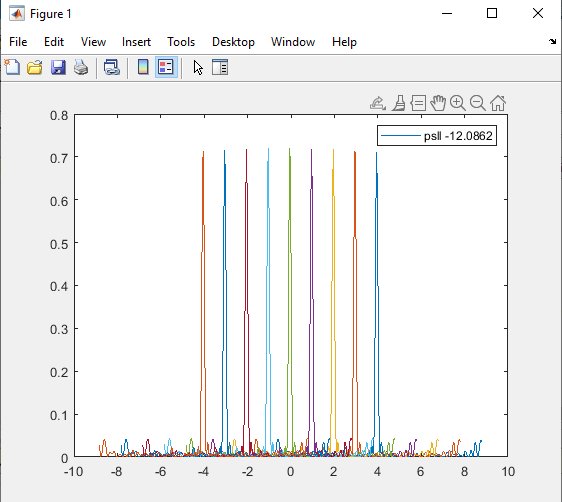


Fig 11. Result of old grouping and fitness function(359.01s)

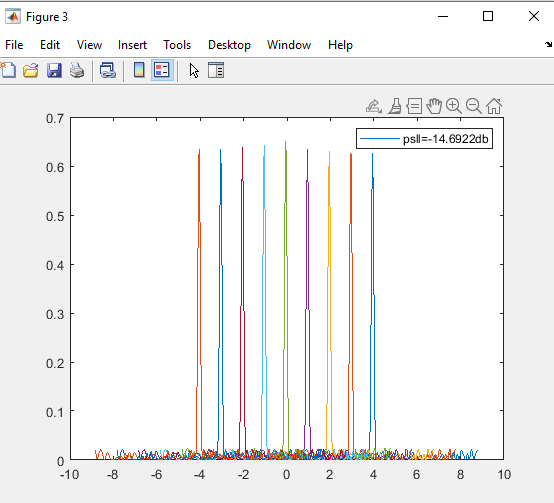


Fig 12. Result for old grouping and iteration(1628s)

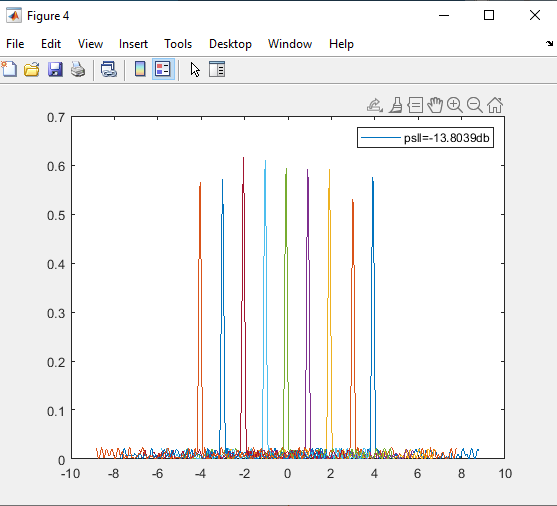


Fig 13. Result of GA(248s)

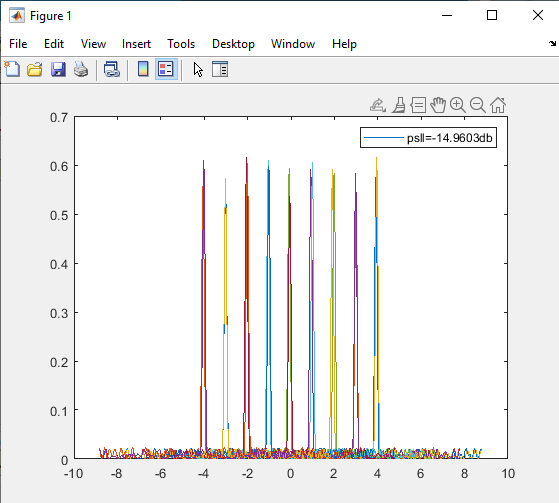


Fig 14. Result for PSO(328s)

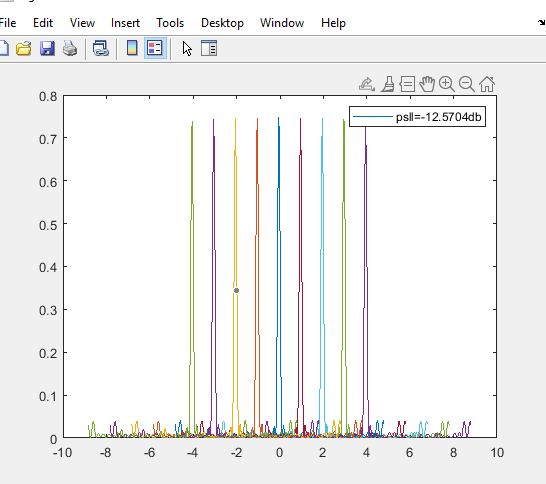


Fig 15. New grouping algorithm with 1 iteration(64s)

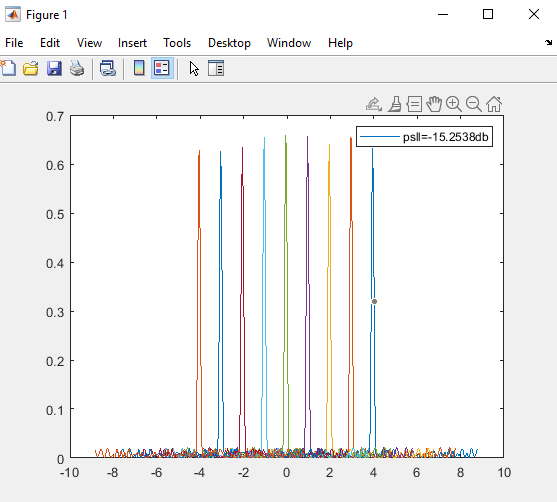


Fig 16. Result for new grouping with iteration of fitness function(154s)