Block Coordinate Descent

Max Iterations = Roof of (attributes / n); where n is the number of attributes in 1 block

X0 = Starting Pt

Var\_order = Order for Iterations

While Epochs < Max Epochs:

Do {While Iterations < Max Iterations:

Do {x = Sampling(n, method); Sampling: Creates ‘n’ shaped data based on method

X\_data = Sampling\_fn(X0, x, Iterations);

[Sampling\_fn: Based on x and Iterations creates X\_data which has constant values for attributes not in that Var\_order]

Y\_data = execute\_C\_file(X\_data)

model = Alamo\_train(X\_data, Y\_data)

X\_min, Y\_min = scipy.optimize(model)

Store X\_min, Y\_min in excel file

X0 = X\_min}

Issues:

1. Using the thumb rule of scaling down the sampling size: As we get close to the solution with increasing success, we narrow down the sampling area. In later stages, the sampling region becomes too small for improvements. Should there be a cut-off for the minimum sampling area.

Also, is the scaling factor of 2 a good one?

1. If we randomize the attribute blocks in each epoch then we cannot use the previously sampled data. So, should we keep the pairings same?
2. What should be used to create a global model? ALAMO itself