1. I substituted RBF with Kriging in HDMR.
   1. Let Kriging segment be a structure instead of a function.
      1. Kriging used to be an individual function, for each call in HDMR member functions’ construction, Kriging would firstly use sample points to train the coefficients theta, then secondly use the trained theta to predict a given point. In this way, for every call in genetic algorithm(GA) of HDMR, the summarization of member functions, the Kriging function as a surrogate for each member function, would be called at every iteration in GA. Thus, if the GA of HDMR iterates for thousands of times and the GA of each Kriging iterates also for thousands of times, there would be thousands of thousands of iterations for each HDMR optimization, which obviously is a huge cost of time.
      2. Actually, once a member function has already been built, training for theta of the member function again and again is a waste of time. This process need to be done only one time. In this way, the member function is designed to be a structure, which contains its theta and the function for prediction. When constructing a member function, the GA optimizing theta is called one time, and theta is then stored in the member function structure.
      3. By using this method, if the GA of HDMR iterates for thousands of times and the GA of each Kriging iterates also for thousands of times, there would be thousands of iterations for each HDMR optimization, which performs better at aspect of time complexity.



1. UML Figure for struct f
   1. Kriging does not accurately go through its sample points but always has a tiny difference, which doesn’t affect its performance when being tested but have a relatively huge difference when called in HDMR.
      1. I still don’t know the answer
   2. Sensitivity analysis…..!!!!!!!!!!