***ENAE625 2nd Half Project Read Me***

***Group 1***

Joe Southgate, Matthew Weiner, Jon Gabriel

***Objectives:***

The objective of this project has been to develop a custom-built MOGA algorithm based on principles taught in ENAE625. A code base has been developed in MATLAB to test the functionality of this custom-built MOGA against several test problems. The code base also allows for the comparison of the custom-built MOGA vs MATLAB’s implemented MOGA. This Read-Me file describes how to utilize the code base

**File Descriptions:**

* Final Code Base: This folder contains all the code that will be required to run the test problems
  + MATLAB MOGA:
    - This folder contains all the run files for each test problem solved using the built-in MATLAB MOGA
  + Developed MOGA:
    - This folder contains all the fitness calculation and run files for each test problem solved using the MOGA algorithm built for this project
  + Supporting Functions:
    - This folder contains all the supporting functions that enable the use of the developed MOGA algorithm, such as shared fitness calculation and pareto front quality metric calculation
* GA\_Test\_Problem\_Caller
  + This file is the primary user interface. The different test function run functions are called from this file – see “How-To Use” for more information on use specifics

**How-To Use:**

Upon opening the GA\_Test\_Problem\_Caller.m, the user will be presented with instructions on how to call different MOGA test problems. 2 sets of function libraries are presented, one of which names the functions to run the test problems using our developed MOGA, and the second set details the test functions for running the test problems with MATLAB’s MOGA. An example is included below.

A picture containing text, screenshot, font

Description automatically generated

The left side details the test problem associated with the right side, which is the call function that a user should copy depending on which test problem is desired to be called. The arguments for each test function are the number of GA runs to conduct, the initial population size, and the number of generations respectively. The numbers found in the functions are the recommended values for these parameters, but can be changed depending on the user’s needs. The right hand function call should be copied and pasted in the area shown below.

A picture containing text, font, screenshot, line

Description automatically generated

The user can then simple click run and the GA should begin calculating the pareto front.