

```

# Created by Darren Holland
# Modified by Darren Holland 2020-11-02
#*****
# File for Dakota optimization.  Runs a multiple objective genetic
# algorithm (MOGA) optimization.  The design with the lowest required
# counts is passed into the mesh adaptive search to find a local
# optimal design.  Settings are modified by StartDakota.sh.
#*****

environment
  output_precision = 15
  # Create summary of parameters and objective functions
  tabular_data
    tabular_data_file = 'Summary.dat'
  # Use method 'HS' (hybrid sequential)
  top_method_pointer = 'HS'

method
  #Define method 'HS'
  id_method = 'HS'
  hybrid sequential
    method_pointer_list = 'MOGA' 'MAS'
    # Perform MOGA first, then MAS
    # (in practice MAS was not used since MC approach never evaluated
    enough designs)

method
  id_method = 'MOGA'
  # Define MOGA sub-method
  moga
    # NOTE: population_size must be >= replacement_type below_limit value
    # Settings
    population_size = 175
    max_function_evaluations = 10000
    initialization_type unique_random
    crossover_type shuffle_random
    num_offspring = 2 num_parents = 2
    crossover_rate = 0.8
    mutation_type replace_uniform
    mutation_rate = 0.1
    fitness_type domination_count
    replacement_type below_limit = 5
    shrinkage_percentage = 0.9
    convergence_type metric_tracker
    percent_change = 0.01 num_generations = 100
  output silent
  model_pointer = 'MOGA_M'

method
  id_method = 'MAS'
  # Define MAS sub-method
  mesh_adaptive_search
  # Settings
  initial_delta = 10.0
  variable_tolerance = 1.0e-6
  function_precision = 1.0e-7
  model_pointer = 'MAS_M'
  output debug

model
  id_model = 'MOGA_M'
  # Set evaluation model
  single
    interface_pointer = 'MOGA_E'

```

```

# Set response model
responses_pointer = 'MOGA_R'
# Set variables
variables_pointer = 'V1'

model
  id_model = 'MAS_M'
  # Set evaluation model
  single
    interface_pointer = 'MAS_E'
  # Set response model
  responses_pointer = 'MAS_R'
  # Set variables
  variables_pointer = 'V2'

interface
  # MOGA evaluation settings
  fork
    asynchronous
    evaluation_concurrency = 1
  # Script to call to start design analysis and evaluation
  id_interface = 'MOGA_E'
  analysis_drivers = 'MOGA_simulator_script'
  # File to save the design parameters
  parameters_file = 'MOGAparameters.in'
  # File to load evaluation results (generated by simulation script)
  results_file = 'MOGAresults.out'
  # Location to save files
  work_directory directory_tag
  named 'MOGAworkdir' file_save directory_save
  dprepro

interface
  # MAS evaluation settings
  fork
    asynchronous
    evaluation_concurrency = 1
  # Script to call to start design analysis and evaluation
  id_interface = 'MAS_E'
  analysis_drivers = 'MAS_simulator_script'
  # File to save the design parameters
  parameters_file = 'MASparameters.in'
  # File to load evaluation results (generated by simulation script)
  results_file = 'MASresults.out'
  # Location to save files
  work_directory directory_tag
  named 'MASworkdir' file_save directory_save
  dprepro

responses
  id_responses = 'MOGA_R'
  # Set MOGA objective functions
  objective_functions = 2
  no_gradients
  no_hessians

responses
  id_responses = 'MAS_R'
  # Set MAS objective function (only 1 allowed, single objective method)
  objective_functions = 1
  no_gradients
  no_hessians

```