

# **Instructions of Implementing Constraint-Handling Techniques (CHTs) for P3GA**

Author: Ying-Kuan (Rick) Tsai  
Principal Investigator: Dr. Richard Malak

December 11, 2023

## Table of Contents

Main.m.....	3
Function Syntax: .....	3
Description from Matlab file: .....	3
Function 1: run_p3ga.m.....	3
Function Syntax: .....	3
Description from Matlab file: .....	3
Function 2: Initialize_LHS_GP.m.....	4
Function Syntax: .....	4
Description from Matlab file: .....	4
Function 3: p3ga_Bayesian.m .....	5
Function Syntax: .....	5
Description from Matlab file: .....	5
Function 4: train_GPC.m.....	5
Function Syntax: .....	5
Description from Matlab file: .....	6
Function 5: RepairInfeasible.m .....	6
Function Syntax: .....	6
Description from Matlab file: .....	6

## Main.m

### Function Syntax:

None

### Description from Matlab file:

This script is to define the problem, CHT, and any parameters for running the algorithm and saving the results and call the functions to run P3GA. Before optimization, it is highly recommended that the user does some sampling or other design of experiment (DOE) techniques to **make sure feasible points exist**. Otherwise, the Bayesian technique would fail since we cannot train a gaussian process classification (GPC) model with only one-class data.

- Section 1: Setting
  - `classif_err_allowance`: allowance of classification error,  $\epsilon$ , used for Bayesian CHT. The repair mechanism selects “moving infeasible points” or “querying new potential points” based on this.
  - `m`: number of objectives
  - `p`: number of parameters
  - `ProbType`: Problem Type (Types I, II, III, or IV), category of constraints defined by MW<sup>1</sup>
  - `W_constr_handling`: Selection of CHTs (1: CDP, 2: Bayesian, 3: C-TAEA, 4: c-DPEA)
  - `run_trial`: trial number if the user wants multiple runs (the value can be 1, 2, ...)
- Section 2: Define Objective, Parameter, and Constraint Functions
  - `obj_fun`: objective function
  - `para_fun`: parameter function
  - `Prob`: `funland2` (attribute function), `par` (index of parameters), `dom` (index of dominance objectives), `lb` (lower bound), `ub` (upper bound), `nvars` (number of variables), `Generations`, `PopulationSize`
- Section 3: Add paths
- Section 4: Run

## Function 1: run\_p3ga.m

### Function Syntax:

```
[pHVI_value, x_opt, fval, M, exp_classif_err] =  
run_p3ga(Prob, W_constr_handling, classif_err_allowance, ProbType, run_trial)
```

### Description from Matlab file:

This function is to run P3GA given the defined input arguments  
(`Prob`, `W_constr_handling`, `classif_err_allowance`, `ProbType`, `run_trial`)

- Section 1: Define ‘options’

---

<sup>1</sup> Ma, Z., & Wang, Y. (2019). Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Transactions on Evolutionary Computation*, 23(6), 972-986.

## Instructions of implementing constraint-handling techniques (CHTs) for P3GA

- If Bayesian CHT is used, initializing population is needed by calling the function `'Initialize_LHS_GP'`.
- `'options.Log'` and `'options.GenerationData'` are Boolean to determine if the user want to save the data for each generation; if so, the user needs to specify the saving dictionary by defining `'options.SaveLoc'` and the increment of saving data by generation (`options.GenerationDataIncrement`).
- Section 2: Run P3GA  
One of the p3ga functions is selected based on which CHT the user specified in the `'main.m'` script. The resulting figures will be saved right after the computation.
- Section 3: Save data

## Function 2: Initialize\_LHS\_GP.m

### Function Syntax:

```
[X_new, feasible_shifted, X, feasible, cEval] =  
Initialize_LHS_GP(lb,ub,A,b,Aeq,beq,nonlcon,Npop,nvars,PlotOrNot,classif_err_  
allowance,TolCon,X_initial)
```

### Description from Matlab file:

This function is to generate the initial populations using LHS and GP techniques. The goal is to generate the initial population for P3GA through identifying the nonlinear constraint boundary and repairing those infeasible points to the predicted boundary. The user can also provide the initial population by `X_initial`

### Procedure:

1. Define `X` (if the user does not provide the initial population, then generate it via LHS).
2. Identify feasibility.
3. Classify based on the feasibility (two pools: `X_feasible`, `X_infeasible`)
4. Two conditions:
  - a. If no infeasible solutions or low feasibility occurs, skip training GPC model and output the `X_new` and `feasible_repaired`
  - b. Otherwise, train the GPC model using the function `'train_GPC'` and calculate the expected classification error, defined by `exp_classif_err`. The GPC model at each individual is evaluated by the function `'gp_Eval'`. Next, repair the infeasible solutions.
5. `counter` is to count how many initialization we tried to avoid endless initialization.

### Input Arguments:

- `A,b,Aeq,beq`: linear constraints for both inequality and equality functions
- `nonlcon`: nonlinear constraint functions
- `Npop`: number of population members (Population size)

Author: Ying-Kuan Tsai ([yktsai0121@tamu.edu](mailto:yktsai0121@tamu.edu))

PI: Dr. Richard Malak ([rmalak@tamu.edu](mailto:rmalak@tamu.edu))

## Instructions of implementing constraint-handling techniques (CHTs) for P3GA

- PlotOrNot: index for showing the plot in the design space ('Plot' or 'DontPlot')
- TolCon: hyperparameter for P3GA (tolerance of constraint violation; 1e-6)
- X\_initial: initial population given by the user (if applicable)

### **Outputs:**

- X\_new: repaired (new) population (set as the initial population to P3GA for the next step)
- feasible\_repaired: feasibility for the repaired population {0,1} or {false, true}
- X: original initial population before the repairment
- feasible: feasibility for the original population {0,1} or {false, true}
- cEval: number of constraint evaluations

## Function 3: p3ga\_Bayesian.m

### Function Syntax:

```
[x, fval, M] =  
p3ga_Bayesian(fun, dom, par, nvar, A, b, Aeq, beq, lb, ub, nonlcon, options, textarea, Fro  
ntier)
```

### Description from Matlab file:

This function is doing the same thing as the original p3ga function. However, Bayesian CHT is included in Line 253 – 255 and Line 274 – 395. Also, the crossover and mutation operations use 'crossoverfcn2' and 'mutationfcn2'. Other than those, there is no difference to p3ga.

Procedure starting at Line 274:

1. Collect the infeasible solutions after the selection, crossover, and mutation operators:  
infeasibleKids
2. A threshold is set up to skip the Bayesian CHT process if no infeasible solutions or the infeasibility rate is very low (<0.1). The original algorithm can perform well without the CHT under those situations. If not, go Step 3.
3. Define new feasibility index,  $y$ , because the GPC model only accepts 1 (feasible) or -1 (infeasible).
4. Train the GPC model using the function 'train\_GPC' and calculate the expected classification error, defined by exp\_classif\_err. The GPC model at each individual is evaluated by the function 'gp\_Eval'. Next, repair the infeasible solutions.

## Function 4: train\_GPC.m

### Function Syntax:

```
[fun1, fun2, hyp, inffunc, meanfunc, covfunc, likfunc, post] = train_GPC(X,y)
```

Author: Ying-Kuan Tsai ([yktsai0121@tamu.edu](mailto:yktsai0121@tamu.edu))

PI: Dr. Richard Malak ([rmalak@tamu.edu](mailto:rmalak@tamu.edu))

### Description from Matlab file:

This function is to train the gaussian process classification (GPC) model using the toolkit developed by Rasmussen and Williams<sup>2</sup>. In the function, the inference function, mean function, etc. are defined and use the dataset (X,y) to produce a GP classifier. Then, the loss functions for repair mechanism are defined (maximize entropy function and minimize the distance to the approximate boundary).

#### **Input:**

- X: (N-by-n) array, where N is the number of samples and n is the dimension of the design variables
- y: (N-by-1) array with +1 or -1

#### **Output:**

- fun1
- fun2
- GP model outputs (hyp, inffunc, meanfunc, covfunc, likfunc, post)

## Function 5: RepairInfeasible.m

### Function Syntax:

```
[xQ,xQ_infea,shift_index] = RepairInfeasible(funQ,funP,x2,X,y,lb,ub,...  
    hyp, inffunc, meanfunc, covfunc, likfunc, post, exp_classif_err,  
    classif_err_allowance)
```

### Description from Matlab file:

The function is to repair the infeasible points using Bayesian methods (Bayesian active learning and Gaussian process classifier). The repair method depends on if  $\text{exp\_classif\_err} < \text{classif\_err\_allowance}$ . If so, use 'opt\_fun\_project' (move the infeasible points to the nearest constraint boundaries). If not, use 'opt\_fun\_query' to query new points that are the most informative and optimally improve the model accuracy.

#### **Input:**

- funQ: the entropy function `entropy_fun`, defined by BALD<sup>3</sup>. This function uses the trained GPC model.
- funP: the lost function, `GPC_boundary_fun`, to be minimized. This function is to find the closest point on the constraint boundary.
- x2: infeasible points to be repaired

#### **Output:**

- xQ: repaired points

---

<sup>2</sup> <https://gaussianprocess.org/gpml/code/matlab/doc/>

<sup>3</sup> Houlisby, N., Huszár, F., Ghahramani, Z., & Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

## Instructions of implementing constraint-handling techniques (CHTs) for P3GA

- xQ\_infea: unrepaired points (for some reasons, the points fail to be repaired)

### Function 6: opt\_fun\_query.m

#### Function Syntax:

```
[xQ,fval,f0,eval_func] = opt_fun_query(fun,x0,lb,ub)
```

#### Description from Matlab file:

This function is to find the query for classification by GA optimizer. The maximum of generations and population size are  $\min(\max(10 * (\text{length}(x0) - 1), 10), 50)$ .

#### Input:

- fun: objective function for the optimization = - (predictive output variance for classification) because the optimal query would be the point that have the highest uncertainty for the classification
- x0: initial for optimization

#### Output:

- xQ: queried points
- fval: objective value
- eval\_func: number of function evaluations (surrogate GPC model, not the original one)

### Function 7: opt\_fun\_project.m

#### Function Syntax:

```
[xQ,fval,eval_func] = opt_fun_project(x0,lb,ub,funP)
```

#### Description from Matlab file:

This function is to find the nearest point on the approximated boundary given the infeasible point. The maximum number of optimization evaluations is  $100 * (\text{length}(x0) - 1)$ . The constraint tolerance is set  $1e-3$ .

#### Input:

- funP: objective function for minimizing the distance between the infeasible point and the predicted boundary
- x0: initial for optimization

#### Output:

- xP: projected points
- fval: objective value

Author: Ying-Kuan Tsai ([yktsai0121@tamu.edu](mailto:yktsai0121@tamu.edu))

PI: Dr. Richard Malak ([rmalak@tamu.edu](mailto:rmalak@tamu.edu))

## Instructions of implementing constraint-handling techniques (CHTs) for P3GA

- eval\_func: number of function evaluations (surrogate GPC model, not the original one)