Reference Report

Oscar Wu

1 Introduction

This report is for Prof Gleb referencing codes in repository:

GitHub

Most of codes based on package *pymoo*. Reason of using this package to working this project show the table below:

Table 1: Multi-objective Optimization Frameworks in Python

Name	Focus on multi-objective	Pure python	Visualization	Decision Making
jMetalPy	$\surd$	$\surd$	$\surd$	$\times$
PyGMO	$\surd$	$\times$	$\times$	$\times$
Platypus	$\surd$	$\surd$	$\times$	$\times$
DEAP	$\times$	$\surd$	$\times$	$\times$
Inspyred	$\times$	$\surd$	$\times$	$\times$
pymoo	$\surd$	$\surd$	$\surd$	$\surd$

2 Architecture

Probelms: Optimization problems in framework are categorized into single, multi, and many-objective test problems. Gradients are available through automatic differentiation and parallelization can be implemented by using a variety of techniques [1].

Optimization: Most of the algorithms are based on evolutionary computations [1], operators such as sampling, mating selection, crossover and mutation have to be chosen or implemented. Moreover, when the algorithm is used to solve the problem, a termination criterion must be defined either explicitly or implicitly by the implementation of the algorithm.

Analytics: During and after an optimization run analytics support the understanding of data[1]. First, intuitively the design space, objective space, or other metrics can be explored through visualization. Moreover, to measure the convergence and/or diversity of a Pareto optimal set performance indicators can be used.

3 Problems

3.1 Problem Table

Table 2 shows the problems list. Clicking the problem name to access formula webpage. The first four problems, from *"Binh and Korn function"* to *"Constr-Ex problem"* define by users.

Table 2: Problem table

Problems List			
Problem Name	# Variables	# Objectives	# Constraints
Binh and Korn function	2	2	2
BNH	2	2	2
Chankong and Hamies function	2	2	2
Constr-Ex problem	2	2	2
OSY	6	2	6
Truss2D	3	2	1
Welded beam	4	2	4
zdt1	30	30	None
zdt2	30	2	None
zdt3	30	2	None
zdt4	10	2	None
zdt5	80	2	None
DASCMOP1	30	2	11

The rest of problems can be called from *pymoo* package. There are other available problems in package (Other availabel problems).

3.2 Problems Codes

The location of problems code is here Repository.

1. **Only need users to give the input range (Random search).**
2. **Then run the python problem script**
3. **The script will generate four files: a)Feasible X, b)Feasible objective value, c)Infeasible X, d)Infeasible objective value.**

3.3 Outcome of problem files

Purpose of problems files is given random input, splitting input into feasible and infeasible, and visualizing the output (Visualization deleted in loop files). Because in pymoo package, algorithms files only outputs best objective function value. In other words, algorithm only return the non-dominated set of solution.

4 Algorithm

4.1 Algorithm Table

Table 3: Algorithm table

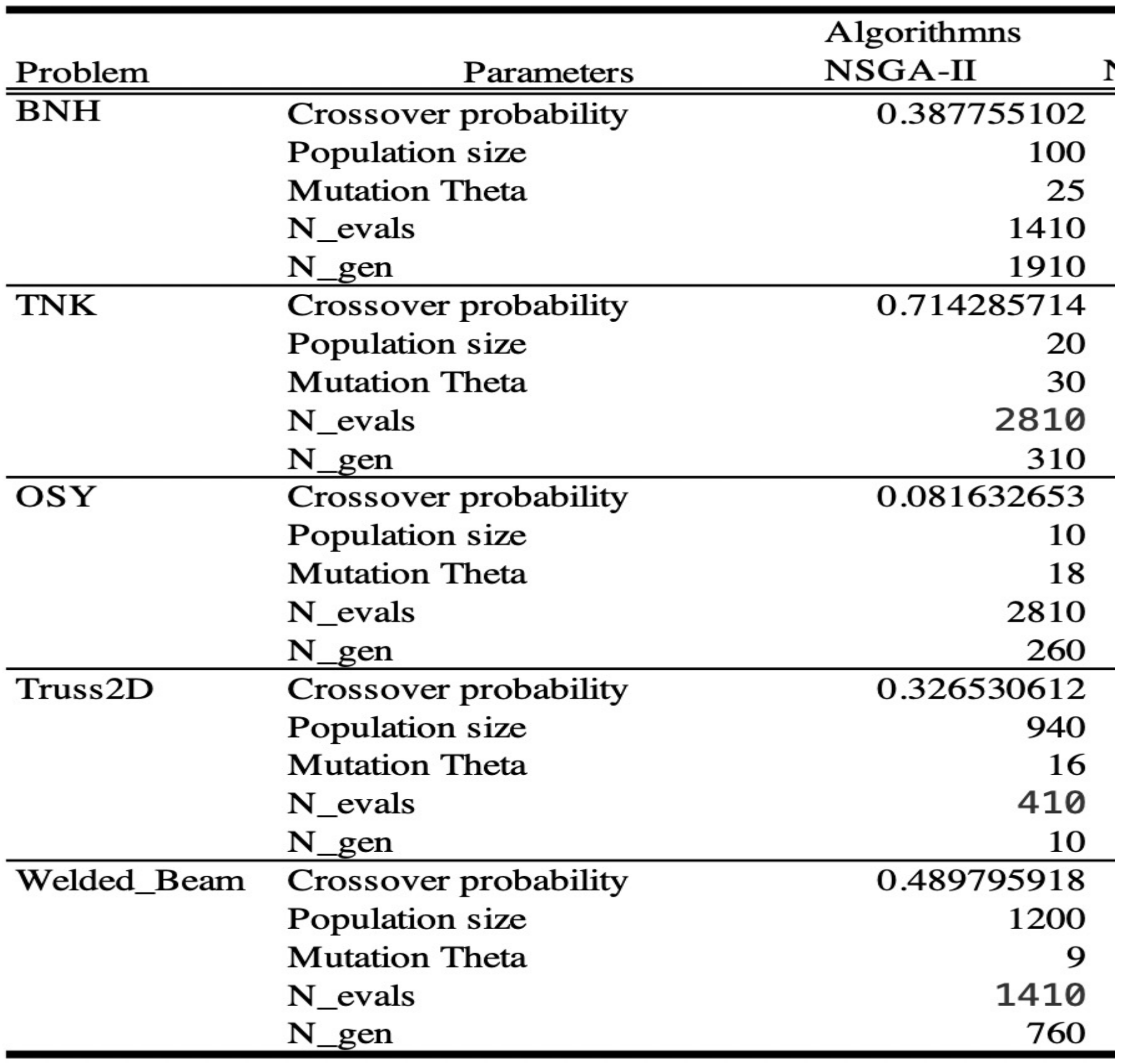
Algorithm List	
Algorithm Name	# Parameter
NSGAII	crossover, #evaluzations, #generate, Mutation, Population
RNSGA-II	Episilon, Refer_point,Weight
NSGAIII	#N_partions

4.2 Algorithm Codes

1. **Search the optimal Parameter of specific algorithm**
2. **For example, to search optimal crossover of NSGA-II by clicking crossover file in Table 3.**
3. **Users need to define Parameter-search domain and import problem inside files**
4. **The algorithm files will find the best parameter corresponding to that problem.**
5. **The same with problem files, the algorithm files will output Objectives X into files.**

4.3 Parameter result

Table 4: Parameter result table



Problem	Parameters	Algorithmns NSGA-II
BNH	Crossover probability	0.387755102
	Population size	100
	Mutation Theta	25
	N_evals	1410
	N_gen	1910
TNK	Crossover probability	0.714285714
	Population size	20
	Mutation Theta	30
	N_evals	2810
	N_gen	310
OSY	Crossover probability	0.081632653
	Population size	10
	Mutation Theta	18
	N_evals	2810
	N_gen	260
Truss2D	Crossover probability	0.326530612
	Population size	940
	Mutation Theta	16
	N_evals	410
	N_gen	10
Welded_Beam	Crossover probability	0.489795918
	Population size	1200
	Mutation Theta	9
	N_evals	1410
	N_gen	760

4.4 Outcome of algorithm files

For different problems, algorithms files try to find various parameters. And then using parameters founded to process MOOP.

5 Further Work

- For hypervolumn, calculating the Pareto front give data set
- From given Pareto front, calculating hypervolumn relative to some fixed points
- Studying two extra programs.
- To figure out NCI compile problem.

6 Improvement

The main problem of experiment is that I did not understand some algorithms very well. There may some potential errors in problem and algorithm files. Furthermore, not sure if I need to add visualization to each loop files.

References

[1] Julian Blank and Kalyanmoy Deb. Pymoo: Multi-objective optimization in python. *IEEE Access*, 8, 2020.