VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

OPTIMIZATION OF PRESSURE VESSEL DESIGN USING PYOPT

Dereje Engida Woldemichael[1] and Abraham Debebe Woldeyohannes[2]
[1]Universiti Teknologi Petronas, Mechanical Engineering Department, Bandar Seri Iskandar, Perak, Malaysia
[2]Caledonian College of Engineering, Department of Mechanical and Industrial Engineering, Muscat, Sultanate of Oman
E-Mail: dereje.woldemichael@utp.edu.my

ABSTRACT

pyOpt is an open source python based object oriented framework for nonlinear constrained optimization problems. In this study, we used pyOpt to solve pressure vessel design problem. Among the available optimizers in pyOpt, SLSQP (Sequential least squares programming), COBYLA (Constrained Optimization by Linear Approximation), ALPSO (Augmented Lagrangian Particle Swarm Optimizer), NSGAII (Non Sorting Genetic Algorithm II), MIDACO (Mixed Integer Distributed Ant Colony Optimization), and ALGENCAN (Augmented Lagrangian with GENCAN) were used. The effect of initial design variables on convergence was investigated for six different regions. The initial design variables for MIDACO and SLSQP should be within the design variable bound while COBYLA and ALPSO provide good result when the initial point is greater than the upper bound. On the other hand, NSGAII and ALGENCAN converge to the optimum value regardless of the initial value. The optimum results from all optimizers were compared with published literatures. Except for ALPSO with mixed discrete variables, the results are in good agreement with maximum percentage error of less than 5%. Therefore, pyOpt can be considered as an alternative option to solve engineering design optimization problems.

Keywords: pyOpt, pressure vessel design, constrained optimization, nonlinear optimization.

INTRODUCTION

Design optimization is the process of finding optimal parameters that leads to obtaining minimum or maximum value of the cost (objective) function subjected to a set of constraints. This type of optimization is known us constrained optimization and can be further classified as linear or nonlinear optimization depending on the type of objective function and constraints. There are a number of algorithms and methodologies to solve such problems with varying ease of use and degree of success. The choice and selection of these algorithms depends on the type and complexity of the design problem (linear/nonlinear, constrained/unconstrained), types of design variables (continuous, discrete, integer), availability of solver (commercial/ open source), and ease of use among others.

In this paper, we investigate the use of pyOpt, an open source optimization framework, for optimal design of pressure vessel design. pyOpt is a python based object oriented framework for nonlinear constrained optimization problems[1]. Python is an open source high-level programming language which can be used to write standalone application models. It can also be interfaced with application models written in low-level programming languages such as C, C++, and Fortran. pyOpt is also an easy-to-use optimization framework where problem formulation and solution by different solvers are defined independently using object oriented constructs. The main capabilities of pyOpt include flexible optimizer integration, operating on multiple platforms, parallelization, and warm-restart for automatic result refinement.

In general, pyOpt can be used to find solution for general constrained nonlinear optimization problems of the form:

$$\min_{x} \; f(x) \qquad (1)$$

Subjected to:

m_e equality constraints

$$h_j(x) = 0 \qquad j = 1,\dots,m_e \qquad (2)$$

m inequality constraints

$$g_j(x) \le 0 \qquad j = m_e + 1,\dots,m \qquad (3)$$

n bounds

$$x_{iL} \le x_0 \le x_{iU} \qquad i = 1,\dots,n \qquad (4)$$

The objective function *f(x)* is assumed to be nonlinear function, and the equality and inequality constraints can be either linear or nonlinear functions of the design variables *x*. Three different variable types namely: continuous, integer, and discrete can be used in pyOpt. However, the use of variable types depends on the optimizers. There are various optimization algorithms integrated with pyOpt for nonlinear constrained optimization. Some of the algorithms require commercial license while most are freely available with Creative Commons (CC) license. We will use the CC licensed algorithms to study their effectiveness in solving pressure vessel design and compare their result with previous studies. Pressure vessel design is one of the most widely used structural design benchmarking problem used by a number of researchers to validate their algorithms [2-8].

PyOpt 1.2 version with Python 2.7.11 which contains a number of constrained optimization solvers designed to solve general nonlinear optimization problems was used. Among the available solvers, we used SLSQP (Sequential least squares programming), COBYLA (Constrained Optimization by Linear Approximation), ALPSO (Augmented Lagrangian Particle Swarm Optimizer), NSGA2 (Non Sorting Genetic Algorithm II), ALHSO (Augmented Lagrangian Harmony Search Optimizer), MIDACO (Mixed Integer Distributed Ant Colony Optimization), and ALGENCAN (Augmented Lagrangian with GENCAN). These solvers are all freely available except MIDACO which requires license when

VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

the variables are more than 4. The derivative estimation for SLSQP and ALGENCAN with finite difference (FD) and complex-step (CS) methods has been investigated. Table 1 lists the optimizers used in this study with supported variable types.

Table-1. Optimizers and variable types used in this study.

Optimizer	Variable type (c - continuous, d – discrete, i – integer)
SLSQP	c
COBYLA	c
ALGENCAN	c
ALPSO	c, d
ALHASO	c, d
MIDACO	c, d, i

PROBLEM FORMULATION

The pressure vessel design problem has been formulated to minimize the total cost which includes the cost of material and cost of fabrication (forming and welding). The pressure vessel is made from a cylindrical vessel caped at both ends with hemispherical heads as shown in Figure-1.There are four design variables namely: thickness of the cylindrical vessel, T_s, thickness of the hemispherical heads, T_h, inner radius of the vessel, R, and the length of the vessel excluding the heads, L. T_s and T_h are considered as discrete variables with values of integer multiples of 0.0625 whereas R and L are considered to be continuous variables.

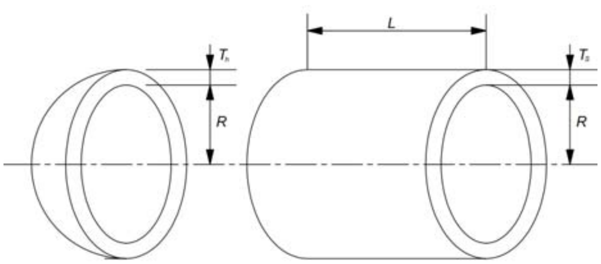

Figure-1. Pressure vessel with design parameters.

According to [8], the optimization problem for pressure vessel design is modeled as

Minimize:

$$f = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_s^2 R \quad (5)$$

Subjected to:

$$g_1 = -T_s + 0.0193R \le 0 \quad (6)$$

$$g_2 = -T_h + 0.00954R \le 0 \quad (7)$$

$$g_3 = -\pi R^2 L - \frac{4\pi R^3}{3} + 1296000 \le 0 \quad (8)$$

$$g_4 = L - 240 \le 0 \quad (9)$$

With bounds:

$$0.0625 \le T_s, T_h \le 99 * 0.0625;\ 10 \le R, L \le 200 \quad (10)$$

There are two upper bounds used in the literature for L; [2, 6, 8, 9] used $L \le 200$ while [5, 6, 10, 11] used $L \le 240$.

SIMULATION MODEL

The numerical model was defined in pyOpt programming environment. The initial values (starting points) for simulation were varied as $x_0 < x_{iL}$; $x_0 = x_{iL}$; $x_{iL} < x_0 < x_{iU}$; $x_0 = x_{iU}$; $x_0 > x_{iU}$ using randomly generated values for each variable. Here, x_{iL} and x_{iU} are the lower and upper bounds of design variable i respectively. The simulation was carried out for both $L \le 200$ and $L \le 240$.

As indicated in the problem description, T_s and T_h are discrete variables. Due to the limitation in SLSQP, COBYLA and ALGENCAN, these variables were considered as continuous for those solvers. Similar assumption were taken by [6] in obtaining the optimal values of the design variables. For ALEPSO, ALHASO and MIDACO we considered both continuous and discrete variable options. An excerpt from the source code for continuous variable definition is shown in Figure-2.

```
# Import Standard Python modules
import numpy as np
from random import randint

# Import Extension module and Optimizers
from pyOpt import Optimization
from pyOpt import SLSQP
from pyOpt import COBYLA
from pyOpt import NSGA2
from pyOpt import ALGENCAN
from pyOpt import ALPSO
from pyOpt import MIDACO
from pyOpt import ALHSO

# Define the Objective function and constraints
def objfunc(x):
    Ts= x[0]
    Th = x[1]
    R = x[2]
    L = x[3]
    f = 0.6224*Ts*R*L + 1.7781*Th*R**2 + 3.1661*(Ts**2)*L + 19.84*(Ts**2)*R
    g = [0.0]*4
    g[0] = -Ts + 0.0193*R
    g[1] = -Th + 0.00954*R
    g[2] = -np.pi*(R**2)*L - (4/3)*np.pi*R**3 + 1296000
    g[3] = L-240
    fail = 0
    return f,g, fail

# Generate random initial design variables
t = randint(0,99) # random number for thickness
Rr= randint(10,200)# random number for radius
Ll= randint(10,200)# random number for length
d=0.0625

# Instantiate Optimization Problem and define the design variables
opt_prob = Optimization("Pressure Vessel Design",objfunc)
opt_prob.addVar('Ts','c',lower=d,upper=99*d,value=d*t)
opt_prob.addVar('Th','c',lower=d,upper=99*d,value=d*t)
opt_prob.addVar('R','c',lower=10.0,upper=200.0,value=Rr)
```

Figure-2. Excerpt from the source code for pressure vessel design.

RESULT AND DISCUSSION

Effect of initial value

We have conducted numerical study with randomly generated initial values for each design variable for three regions ($x_0 < x_{iL}$; $x_{iL} < x_0 < x_{iU}$; $x_0 > x_{iU}$) and at the boundaries to investigate the effect of initial values on the optimal design and computational time. For each

VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

region we repeated the simulation ten times to see the consistency of the result and effect of initial values on the optimal values. Based on the numerical result, NSGAII and ALGENCAN converge to the same numerical result regardless of the initial value. MIDACO requires initial design variables to be set within the bounds ($x_{iL} < x_0 < x_{iU}$). For initial values with values less than the lower bound or greater than the upper bound, MIDACO terminates prematurely. For initial design variables within the bound, MIDACO converges to the same optimal value for all attempts similar to NSGAII and ALGENCAN regardless of the initial value. On the other hand, the results from SLSQP, ALPSO and ALHASO vary with the initial value. Thus, the results from these solvers require statistical analysis such as mean, median and standard deviation. SLSQP gives reasonable values when the initial values are within the bound. COBYLA and ALPSO converge to the optimum value when the initial starting values are in the upper region ($x_0 > x_{iU}$). For all the regions, the results obtained using ALHASO did not converge to optimum value and we could not find any statistical correlation. For instance for twenty randomly generated initial variables within the bound $x_{iL} < x_0 < x_{iU}$ with L< 200 for the fourth design variable, the optimum value for the objective function, f was (best = 6706.89, worst = 9427.99, mean = 855056, median = 9066.30 and standard deviation = 1003.43). Similar trends were observed for the other regions as starting value. Thus, we omit the result from ALHASO from further analysis.

Comparative study with published results

Pressure vessel design optimization problem has been solved by a number of authors in the literature. Kennan and Kramer [8] used augmented lagrange multiplier approach, [9] used self-adaptive penalty approach with genetic algorithm, [5] used Firefly algorithm (FA), [10] used filtered simulated annealing (FSA), and [6] used penalty guided artificial bee colony (ABC) algorithm. A variety of particle swarm optimization such as simple particle swarm optimization (SiC-PSO) algorithm for constrained optimization problem [2], Co-evolutionary particle swarm optimization (CPSO) approach [12] were also used. [13] used hybrid method combining PSO and AC called heuristic particle swarm ant colony optimization (HPSACO). [11] used four algorithms namely Simple Genetic Algorithm(SGA), Struggle Genetic Algorithm (StrGA), Particle Swarm Optimization Algorithm (PSOA), and Particle Swarm Optimization Algorithm with Struggle Selection (PSOStr). The results from these literatures were compared with the results obtained in this paper using PyOpt.

A number of numerical studies were conducted for both conditions ($L \leq 200$ and $L \leq 240$). We classified the study in to two categories. The first category includes NSAGAII, ALGENCAN and MIDACO (both continues [c] and discrete[d] values). These algorithms converge to the same optimal value regardless of the initial set of design variables. The results are shown together with previously published results. Table-2 shows the results of the optimization using pyOpt simulation packages for the design parameters, constraints and the optimal functions in comparison with previously published results.

Table-2. Comparison of results for category I optimizers with other results presented in literature.

	Methods		**Design variables and objective function**				
			T_s	T_h	R	L	$f(x)$
L ≤ 200	SiC-PSO [2]		0.8125	0.4375	42.0984	176.636595	6,059.71
	ABC[6]		0.7781978	0.3846657	40.3211	199.980237	5885.403
	Kannan and Kramer [8]		1.125	0.625	58.291	43.69	7198.2
	Coello [9]		0.8125	0.4375	40.3239	200	6288.7445
	Present study	NSGAII	0.840505	0.415946	43.5387	174.085371	6366.64
		ALGENCAN	0.797934	0.394419	41.3437	200	6230.73
		MIDACO [c]	0.796934	0.393425	41.3437	199.999765	6220.25
		MIDACO[d]	0.0625	0.0625	41.3443	199.99273	6220.27
L ≤ 240	FA [5]		0.75	0.375	38.8601	221.8601	5850.383
	ABC[6]		0.7275958	0.35965529	37.6991	239.999806	5804.449
	Present study	NSGAII	0.817429	0.4043	42.3531	187.641892	6291.25
		ALGENCAN	0.742817	0.367175	38.4879	240	6078.31
		MIDACO [c]	0.741822	0.366183	38.4881	239.996523	6067.69
		MIDACO[d]	0.0625	0.0625	38.4879	239.999993	6067.67

VOL. 11, NO. 24, DECEMBER 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

For $L \leq 200$, [7] used mathematical analysis and Lagrange multiplier method in an attempt to find the true global optimality of the pressure vessel design as 6059.714 which agrees with a number of other publications. For $L \leq 200$, the percentage error in comparison with [7] are 3% for ALGENCAN, MIDACO[c] and MIDACO[d] and 5% for NSGAII. Similar trends were observed for $L \leq 240$. Note that the results from [5] and [6] are better than the true global optimal reported in [7] as the reported values are the best rather than the mean as shown Table-3.

In the second category, the results from SLSQP, COBYLA, ALPSO[c] and ALPSO[d] were considered. These algorithms give slightly different optimal values for each initial set of design variables. Thus, the comparison requires statistical analysis such as mean, median, and standard deviations. Table-3 shows the minimized objective function using pyOpt simulation packages (SLSQP, COBYLA, ALPSO[c] and ALPSO[d]), in comparison with published results.

As shown in Table-3, the mean objective function value obtained from pyOpt packages (SLSQP, COBYLA, ALPSO[c]) for $L \leq 200$ are in good agreement with the true optimal value reported in [7] with maximum of percentage error of 4%. These solvers also give comparable result with those reported in [9, 12, 13] with better standard deviation value. For $L \leq 240$, SLSQP, gives the best result compared to [10, 11]. The mean optimal value obtained from COBYLA and ALPSO[c] is close to the result reported in [10, 11] with better standard deviation. On the other hand, for both $L \leq 200$ and $L \leq 240$, ALPSO[d] gives the worst result among the ones reported in the literature.

Table-3. Comparison of results for category II optimizers with other results presented in literature.

	Method		Worst (Maximum)	Best (Minimum)	Mean	Median	Standard deviation
L<200	Coello [9]		6308.1497	6288.7445	6293.8432	N/A	7.4133
	CPSO [12]		6363.8041	6061.0777	6147.1332	N/A	86.4545
	HPSACO [13]		6135.3336	6059.0925	6075.2567	N/A	41.6825
	Present study	SLSQP	6231.4400	6230.4700	6230.7800	6230.720	0.2500
		COBYLA	6324.1500	6308.3300	6314.4240	6311.485	5.3906
		ALPSO[c]	6276.1500	6224.4500	6251.4970	6246.325	16.3548
		ALPSO[d]	9641.1300	9572.4800	9632.7800	9639.205	20.1209
L<240	FSA [10]		6804.3281	5868.7648	6164.5859	N/A	257.4737
	PSOStr [11]		N/A	N/A	6272.5745	N/A	538.3703
	StrGA [11]		N/A	N/A	6122.0806	N/A	98.3040
	PSOA [11]		N/A	N/A	6292.1732	N/A	528.9008
	Present study	SLSQP	6078.9200	6077.3700	6078.1778	6078.300	0.4006
		COBYLA	6136.5400	6126.5300	6132.2920	6132.585	2.9250
		ALPSO[c]	6604.3800	6075.6600	6180.7220	6122.695	156.1676
		ALPSO[d]	9947.2500	9877.9700	9908.8910	9907.340	15.6692

CONCLUSIONS

There have been many attempts on the methodology of solving the optimization problem in engineering design with various degree of success. In this paper we presented the application of an open source package pyOpt to optimal design of pressure vessel. Among the optimizers in pyOpt, we investigated seven solvers namely: SLSQP, COBYLA, NSGAII, ALGENCAN, MIDACO, ALPSO and ALHASO. Since MIDACO, ALPSO and ALHASO can handle both discrete and continuous design variables; we optimized the pressure design problem using both types of design variables. The effect of initial design variables on convergence of the optimization problem has been identified. The results from these optimizers have been compared with published results. In general, the results were in good agreement with less than 5% error with other optimizers except ALHASO (both continuous and discrete variables) and ALPSO[d] with discrete variable.

Similar to other search algorithms reported in literatures which were tested with different benchmarking problems, it is necessary to investigate the optimizers in pyOPt with various engineering problems. Thus, as continuation to this research, we will conduct detailed investigation on the application of these optimizers on other engineering design problems.

REFERENCES

[1] R. Perez, P. Jansen, and J. R. A. Martins, "pyOpt: a Python-based object-oriented framework for nonlinear constrained optimization," Structural and Multidisciplinary Optimization, vol. 45, pp. 101-118, 2012.

[2] L. C. Cagnina, S. C. Esquivel, and C. A. C. Coello, "Solving engineering optimization problems with the simple constrained particle swarm optimizer," Informatica, vol. 32, 2008.

[3] L. d. S. Coelho, "Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems," Expert Systems with Applications, vol. 37, pp. 1676-1683, 2010.

[4] A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," Engineering with computers, vol. 29, pp. 17-35, 2013.

[5] H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using Firefly Algorithm," Computers & Structures, vol. 89, pp. 2325-2336, 2011.

[6] H. Garg, "Solving structural engineering design optimization problems using an artificial bee colony algorithm," Journal of Industrial and Management Optimization, vol. 10, pp. 777-794, 2014.

[7] X. S. Yang, C. Huyck, M. Karamanoglu, and N. Khan, "True global optimality of the pressure vessel design problem: A benchmark for bio-inspired optimisation algorithms," International Journal of Bio-Inspired Computation, vol. 5, pp. 329-335, 2013.

[8] K. Kannan and S. N. Kramer, "An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design," Journal of Mechanical Design, vol. 116, pp. 405-411, 1994.

[9] A. Coello Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," Computers in Industry, vol. 41, pp. 113-127, 2000.

[10] A.-R. Hedar and M. Fukushima, "Derivative-Free Filter Simulated Annealing Method for Constrained Continuous Global Optimization," Journal of Global Optimization, vol. 35, pp. 521-549, 2006.

[11] G. G. Dimopoulos, "Mixed-variable engineering optimization based on evolutionary and social metaphors," Computer Methods in Applied Mechanics and Engineering, vol. 196, pp. 803-817, 2007.

[12] Q. He and L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," Engineering Applications of Artificial Intelligence, vol. 20, pp. 89-99, 2007.

[13] A. Kaveh and S. Talatahari, "Engineering optimization with hybrid particle swarm and ant colony optimization," Asian journal of civil engineering, vol. 10, pp. 611-628, 2009.