International Journal of Energy Applications and Technologies
Available online at http://dergipark.gov.tr/ijeat/issue/30402
Research Article
Vol. 4 Issue 2 pp. 34-52 Year 2017

A COMPARISON OF SEQUENTIAL QUADRATIC PROGRAMMING, GENETIC ALGORITHM, SIMULATED ANNEALING, PARTICLE SWARM OPTIMIZATION AND HYBRID ALGORITHM FOR THE DESIGN AND OPTIMIZATION OF GOLINSKI'S SPEED REDUCER

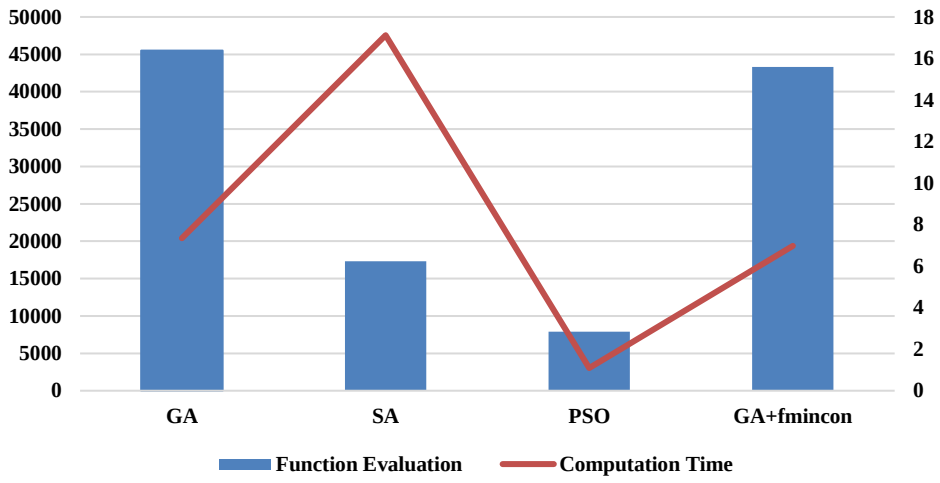
Cenker Aktemur[1]*, Islam Gusseinov[1]

[1] Department of Mechanical Engineering, Eastern Mediterranean University, via Mersin 10, Turkey

Abstract

This article provides information on different optimization methods such as Sequential Quadratic Programming (SQP), Genetic Algorithm (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO) and Hybrid Algorithm (HA). Optimization is a method of designing a system in such a manner that it falls into all limitations on design and satisfies all the design parameters provided. In this particular study, Matlab software is used to perform these optimization methods. It is very helpful software with wide range of applications. One of the such applications is optimization toolbox which is called optimtool. It contains readily written codes for different optimization tools. After conducting optimization on Golinski's speed reducer with five various optimization method, the results are in kilograms for the weight optimization which are SQP = 2994.355 kg, GA = 2994.914 kg, SA = 2730.74 kg, HA = 2994.355 kg, and PSO = 2905.677. The figure below, which is thought graphically abstract, represents a result of all optimizations.

Key words: *Optimization, Golinski's speed reducer, Design parameter, Limitation*



Paper submitted: March 05, 2017
Paper accepted: July 13, 2017

Corresponding author, cenkeraktemur_41@hotmail.com

International Journal of Energy Applications and Technologies, Year 2017, Vol. 4, No. 2, pp. 34-52

1. Introduction

The Golinski's speed reducer is one of the well-studied optimization problems. It was first studied by Golinski in 1970-73, as a simple problem of minimization of speed reducer gear box's weight. This problem appears to be the benchmark for application of new methods of optimization. The transmission can be used between the two rotating parts to allow them to rotate with torque and speed separately for maximum efficiency [1]. Generally, there are seven design variables related to dimensions and material properties of the shafts and gears. These variables can vary between lower and upper bounds which appear to be the limitations on design variables. Moreover, there are constraints which are made up of different operations on design variables. That is to say, the constraints on design variables are bounds [2]. The whole idea of applying different optimization methods on speed reducer problem is to obtain the best results with regard to minimum weight of a gearbox and the minimum time spent for optimization [3]. The detailed information on each method of optimization and the results will be provided below.

2. System Representation

Drawing of the speed reducer is indicated in Figure 1 below. Design variables are denoted by X's, they are related to the dimensions of the parts as well as to the number of teeth on the gear.

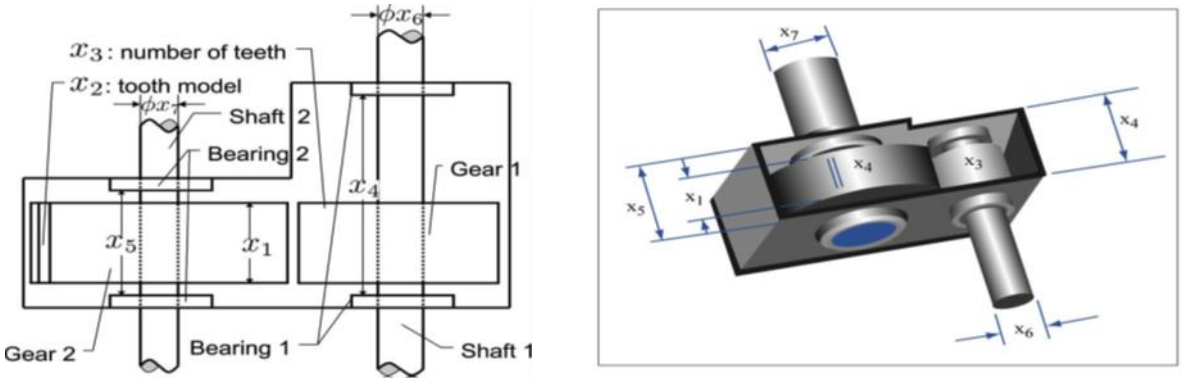


Fig. 1. The Speed Reducer labelled with the Design variables. [4]

3. Design Variables

Design parameters are the parameters which can be changed by designer in order to achieve optimum solution and at the same time to satisfy all the constraints. The design of the Golinski's speed reducer is a more challenging benchmark, since it involves seven various design variables.

b – Face width of gear	(x1)
m – Module	(x2)
z – Number of pinion teeth	(x3)
l1 – Length of shaft 1	(x4)
l2 - Length of shaft 2	(x5)
d1 – Diameter of shaft 1	(x6)
d2 - Diameter of shaft 2	(x7)

Table 1. Design Variables.

Variable	Symbol	Units	Lower Bound (LB)	Upper Bound (UB)
Face width of gear face	x_1	cm	2.6	3.6
Teeth module	x_2	cm	0.7	0.8
Number of teeth of pinion	x_3	-	17	28
Length of shaft 1 between bearings	x_4	cm	7.3	8.3
Length of shaft 2 between bearings	x_5	cm	7.3	8.3
Diameter of shaft 1	x_6	cm	2.9	3.9
Diameter of shaft 2	x_7	cm	5.0	5.5

The variable bounds for the problem are as follows:

$$2.6 \leq x_1 \leq 3.6$$
$$0.7 \leq x_2 \leq 0.8$$
$$17 \leq x_3 \leq 28$$
$$7.3 \leq x_4 \leq 8.3$$
$$7.3 \leq x_5 \leq 8.3$$
$$2.9 \leq x_6 \leq 3.9$$
$$5.0 \leq x_7 \leq 5.5$$

4. Design Constraints

Design constraints are the limitations on design variables, also called bounds. Constraints can be of two types, equality constraints and inequality constraints. Optimization problems with equality constraints are difficult to solve since there are no many options for design variables limitations [5]. The following Table 2 indicates that the inequality constraints are given as some point solutions between lower and upper bounds.

Table 2. Design Constraints.

Constraint	Symbol	Units	Limit
$27x_1^{-1}x_2^{-2}x_3^{-1}$	*G1*	cm^{-3}	≤1
$397.5x_1^{-1}x_2^{-2}x_3^{-2}$	*G2*	cm^{-3}	≤1
$1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4}$	*G3*	cm^{-2}	≤1
$1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4}$	*G4*	cm^{-2}	≤1
$[(745x_4x_2^{-1}x_3^{-1})^2 + 16.9x10^6]^{\frac{1}{2}}/[110.0x_6^3]$	*G5*	cm^{-3}	≤1
$[(745x_5x_2^{-1}x_3^{-1})^2 + 157.5x10^6]^{\frac{1}{2}}/[85.0x_7^3]$	*G6*	cm^{-3}	≤1
$x_2x_3/40$	*G7*	cm	≤1

5. Problem Formulation

As it was mentioned previously, the main idea of optimization of any system and particularly the speed reducer is to obtain optimum results by considering all the constraints while changing the design variables. In order to do so, there is need for the formulation of objective function. This function will variate and control design variables in such a way that they will remain within given upper and lower bounds. Objective function and constraints are provided below.

Minimize [6]:

$$f(x) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.5079x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$$

Subject to (constraints in right-hand column) [7]

$27x_1^{-1}x_2^{-2}x_3^{-1} \leq 1$	G_1
$397.5x_1^{-1}x_2^{-2}x_3^{-2} \leq 1$	G_2
$1.93x_2^{-1}x_3^{-1}x_4^3x_6^{-4} \leq 1$	G_3
$1.93x_2^{-1}x_3^{-1}x_5^3x_7^{-4} \leq 1$	G_4
$[(745x_4x_2^{-1}x_3^{-1})^2 + 16.9x10^6]^{\frac{1}{2}}/[110.0x_6^3] \leq 1$	G_5
$[(745x_5x_2^{-1}x_3^{-1})^2 + 157.5x10^6]^{\frac{1}{2}}/[85.0x_7^3] \leq 1$	G_6
$x_2x_3/40 \leq 1.0$	G_7
$5x_2/_{x_1} \leq 1.0$	G_8
$x_1/12x_2 \leq 1.0$	G_9
$(1.5x_6 + 1.9)x_4^{-1} \leq 1.0$	G_{25}
$(1.1x_7 + 1.9)x_5^{-1} \leq 1.0$	G_{25}

6. Results and Discussion

6.1. Sequential Quadratic Programming

SQP is an acronym for sequential quadratic programming which is an iterative method for nonlinear optimization. SQP methods are used on mathematical problems which allows to take the quadratic objective function values by showing the linear equality or inequality [8]. The following Table 3 shows Run 1, 2 and 3 used optimizer parameters including initial variables. Figure 2 shows a diagram of the sequence of movements.

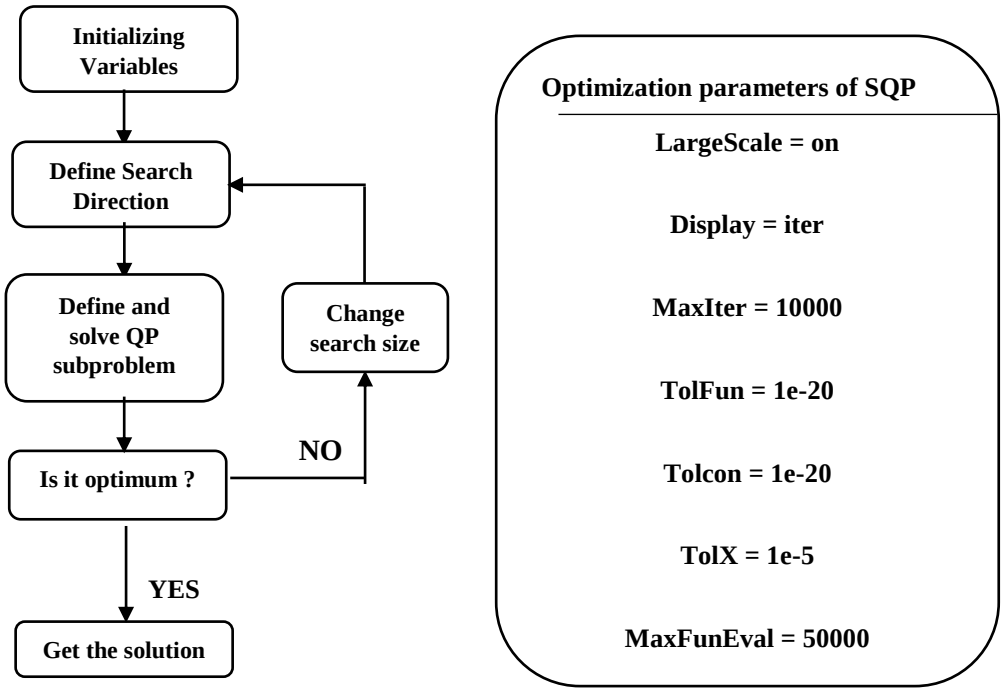


Fig. 2. Flow Chart and Optimization Parameters of SQP.

Table 3. Optimization Results using SQP.

Solution	Units	Run 1	Run 2	Run 3
x_1	cm	3.49999998375537	3.49999999499952	3.49999999999265
x_2	cm	0.700000000462526	0.700000000104668	0.700000000000056
x_3	-	17.0000000173360	17.0000000038690	17.0000000000021
x_4	cm	7.30000032190827	7.30000007099798	7.30000000003957
x_5	cm	7.71532003798962	7.71531994355954	7.71531991149378
x_6	cm	3.35021466782917	3.35021466838174	3.35021466609829
x_7	cm	5.28665446549233	5.28665446596900	5.28665446498077
G1	cm^{-3}	-0.0739152782678250	-0.0739152795624820	-0.0739152803981837
G2	cm^{-3}	-0.197998526115150	-0.197998526601016	-0.197998527142273
G3	cm^{-3}	-0.499172183725132	-0.499172235045034	-0.499172248098522
G4	cm^{-3}	-0.904643905746147	-0.904643909157525	-0.904643904555754
G5	cm^{-3}	-1.03377173488184e-09	-1.93226890043974e-09	-7.07101044383762e-13
G6	cm^{-3}	-4.49723206452646e-08	-4.52589901112077e-08	-1.72639680329212e-13
G7	cm	-0.702499999500047	-0.702499999887809	-0.702499999999967
G8	-	5.30207366900015e-09	1.57823398794221e-09	-1.34003919072256e-13
G9	-	-0583333335542531	-0.583333333990931	-0.583333333333278
G10	cm^{-1}	-0.0513257950194953	-00513257622988151	-0.0513257535444649
G11	cm^{-1}	-5.09285891236999e-09	7.21439730178020e-09	-1.17295062551648e-12
Objective	kg	2994.35503119890	2994.35502783063	2994.35502611209
Optimizer parameters		*x0 = [2.7 0.71 17.5 7.4 7.4 2.95 5.1]* *Default options*	*x0 = [2.8 0.76 17.9 7.7 7.6 3.15 5.4]* *TolFun = 1e-30,Tolcon=1e-25*	*x0 = [2.9 0.79 22.9 7.9 7.8 3.45 5.45]* *Tolcon=1e-10, TolX=1e-10*

* The violated Constraints are highlted in Bold.

6.2. Genetic Algorithm

Genetic algorithm is a search and optimization method working in a manner similar to the evolutionary process observed in nature in order that solving both constrained and unconstrained optimization problems [9]. It changes the population of solutions in a sequence. At each step, it selects individuals as parents arbitrarily from the current population. Then, next generation is produced by parents. Over successive generations, the population evolves toward an optimal solution.

6.3. Simulated Annealing

Simulated annealing algorithm is a global optimization technique which is designed to find the largest or the smallest of the function having many variables, and especially non-linear function having many local minimum value [9].

6.4. Particle Swarm Optimization

Simulated annealing algorithm is a global optimization technique which is designed to find the largest or the smallest of the function having many variables, and especially non-linear function having many local minimum value [9].

6.5. Particle Swarm Optimization

Hybrid optimization is an option given in GA method. All these tools are provided within matlab library and there is no need in combining any method together or writhe the codes for optimization, seperately [10]. Hybrid optimization allows us to exploide more, when the optimum point is found. It takes longer

to perform the optimization by hybrid algorithm, for example, with SQP method. However, it will give more accurate and optimal solution on global scale.

Table 4. Optimization Results using GA

Solution	Units	Run 1	Run 2	Run 3
x_1	cm	3.49999999917056	3.49999999558368	3.49999999635221
x_2	cm	0.700000000000000	0.700000000000000	0.700000000000000
x_3	-	17.0000000046170	17.0043476128025	17.0000000000018
x_4	cm	7.30003882860316	7.37761322612151	7.30000000000000
x_5	cm	7.71727481160603	7.71531917795673	7.71531991147802
x_6	cm	3.35021473858727	3.35035680792981	3.35021894810512
x_7	cm	5.28723002555150	5.28665379812875	5.28665446498004
G1	cm^{-3}	-0.0739152804299184	-0.0741520573686602	-0.0739152794327809
G2	cm^{-3}	-0.197998527387514	-0.198408579630337	-0.197998526306249
G3	cm^{-3}	-0.499164299855089	-0.483247364757583	-0.499174808585474
G4	cm^{-3}	-0.904612948632791	-0.904668263963307	-0.904643904556074
G5	cm^{-3}	1.78459469424297e-11	2.62884158885868e-10	-3.83437912998552e-06
G6	cm^{-3}	-0.000326165612403506	1.47618806067840e-10	1.05693231944315e-13
G7	cm	-0.702499999919203	-0.702423916775957	-0.702499999999969
G8	-	2.36984210033597e-10	1.26180554893551e-09	1.04222497299133e-09
G9	-	-0.583333333432077	-0.583333333859086	-0.583333333767594
G10	cm^{-1}	-0.0513307846054230	-0.0612770011615877	-00513248736770292
G11	cm^{-1}	-0.000171275940231030	-1.95732319241415e-12	1.99840144432528e-15
Objective	kg	2994.9142752518287	2995.7747732760727	2994.355026185286
Optimiter parameters		Default options	Population size=60 Crossover function=intermediate	Migration direction = both Population size=80

* The violated Constraints are highlted in Bold.

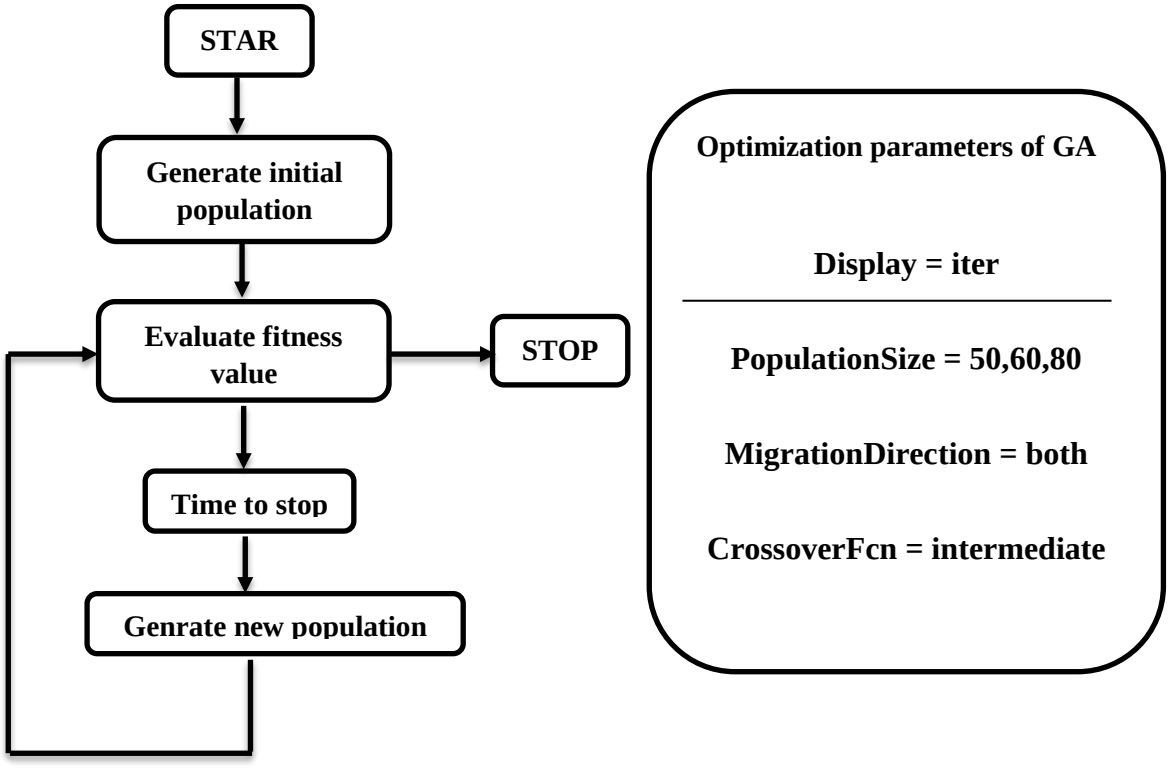


Fig. 3. Flow Chart and Optimization Parameters of GA.

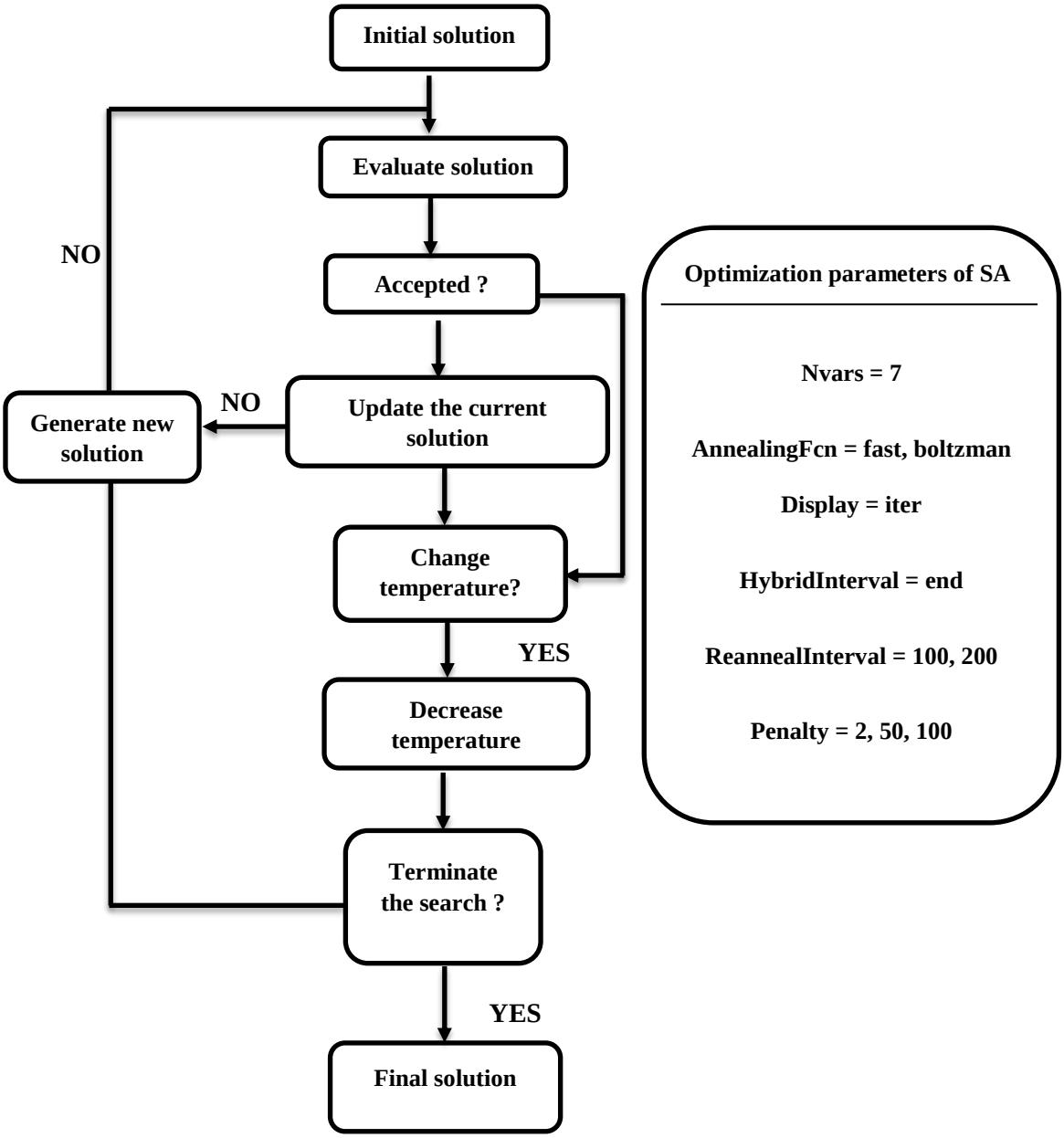


Fig. 4. Flow Chart and Optimization Parameters of SA.

Table 5. Optimization Results using SA.

Solution	Units	Run 1	Run 2	Run 3
x_1	cm	3.18049276183517	3.09588220243017	2.60272767086836
x_2	cm	0.700000000000000	0.700000000000000	0700000000000205
x_3	cm	17.2897637943420	17.5625865367410	21.1708821579682
x_4	cm	7.86445885380494	7.81112597235225	7.47959779350768
x_5	cm	7.66011833837421	8.23866977987855	7.98097724225297
x_6	cm	3.35104610078262	3.35071201930736	3.34800904706652
x_7	cm	5.28659318090479	5.28674366737047	5.28625032478952

Solution	Units	Run 1	Run 2	Run 3
G1	cm^{-3}	**4.91620648852892**	0.357860619241329	-0.000745812323867057
G2	cm^{-3}	**17.3756108394523**	-70.4048829446124	-174.129552759518
G3	cm^{-3}	-736.087161914335	-629.845564675242	-1054.45511900595
G4	cm^{-3}	-8530.83906678581	-8524.45979848900	-10591.5487046114
G5	cm^{-3}	-2.11059514185763e-05	-0.00762946796567121	-0.0159799897628545
G6	cm^{-3}	-9.57518568611704e-06	-0.00388310844391526	-0.00904190375877079
G7	cm	-27.9510250410049	-27.7061874710848	-25.1801957336391
G8	-	0.881663101627737	0.404118479873925	0.897310727895238
G9	-	-5.78166314963480	-5.30411921076036	-5.79737294890711
G10	cm^{-1}	-0.473861067845282	-0.885057313390134	-0.557582020357852
G11	cm^{-1}	0.0164156308606751	-0.523250631461757	-0.266098171530905
Objective	kg	2730.74	3070.68	3332.17
Optimizer parameters		*x0 = [2.7 0.71 17.5 7.4 7.4 2.95 5.1]* *Bolztman annealing* *Penalty=2*	*x0 = [2.8 0.76 17.9 7.7 7.6 3.15 5.4]* *Fast annealing* *Penalty = 50*	*x0=[2.9 0.79 22.9 7.9 7.8 3.45 5.45]* *Bolztman annealing* *Reannealing interval=200* *Penalty = 100*

* The violated Constraints are highlted in Bold

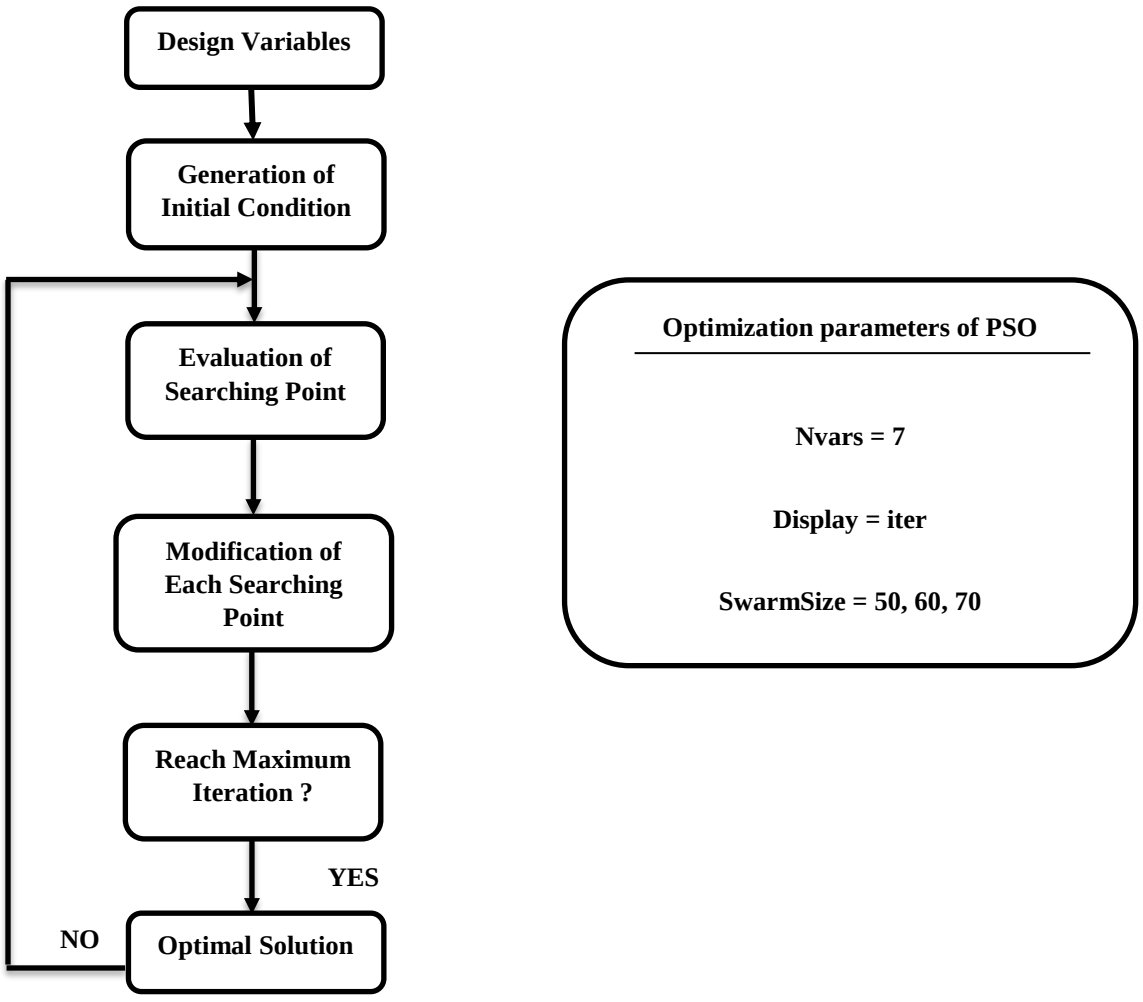


Fig. 5. Flow Chart and Optimization Parameters of PSO.

Table 6. Optimization Results using PSO.

Solution	Units	Run 1	Run 2	Run 3
x_1	cm	2.600000293615250	2.806982922281113	3.241295065039231
x_2	cm	0.700000000000000	0.700002776012913	0.700000010114834
x_3	-	17.000000039716184	17.000000000000000	17.000000000000000
x_4	cm	7.300000296645011	7.300000000000000	7.300009349959121
x_5	cm	7.300001816109539	7.300001147904924	7.715320149635029
x_6	cm	3.350214745752692	3.350214988606270	3.350214747225644
x_7	cm	5.286518028298517	5.286518100940317	5.286654521528211
G1	cm^{-3}	**534200000000000**	**3.61766392752051**	2.84215601489279e-05
G2	cm^{-3}	**29.3140000000001**	0.258723164020111	-61.4995168334775
G3	cm^{-3}	-748.320237162150	-748.320635444769	-748.319892126360
G4	cm^{-3}	-8543.69357638908	-8543.71302730395	-8409.08292162352
G5	cm^{-3}	-0.000829275864816736	0.000312305503030075	0.00110988645064936
G6	cm^{-3}	-0.00143479676989955	-0.00167048769253597	0.000506414220581064
G7	cm	-28.1000000000000	-28.0999753507152	-28.0999989833588
G8	-	0.900000000000000	0.693005022350219	0.258707746177129
G9	-	-5.80000000000000	-5.59300683875728	-5.15870816479410
G10	cm^{-1}	-0.374681482567630	-0.374688864058732	-0.374680746872511
G11	cm^{-1}	0.415169935353485	0.415169966415443	1.65268929839257e-05
Objective	kg	2703.591990187170	2775.691594721565	2905.677289186828
Optimizer parameters		*SwarmSize=50* *Penalty=2*	*SwarmSize=100* *Penalty=10*	*SwarmSize=200* *Penalty=50*

* The violated Constraints are highlted in Bold

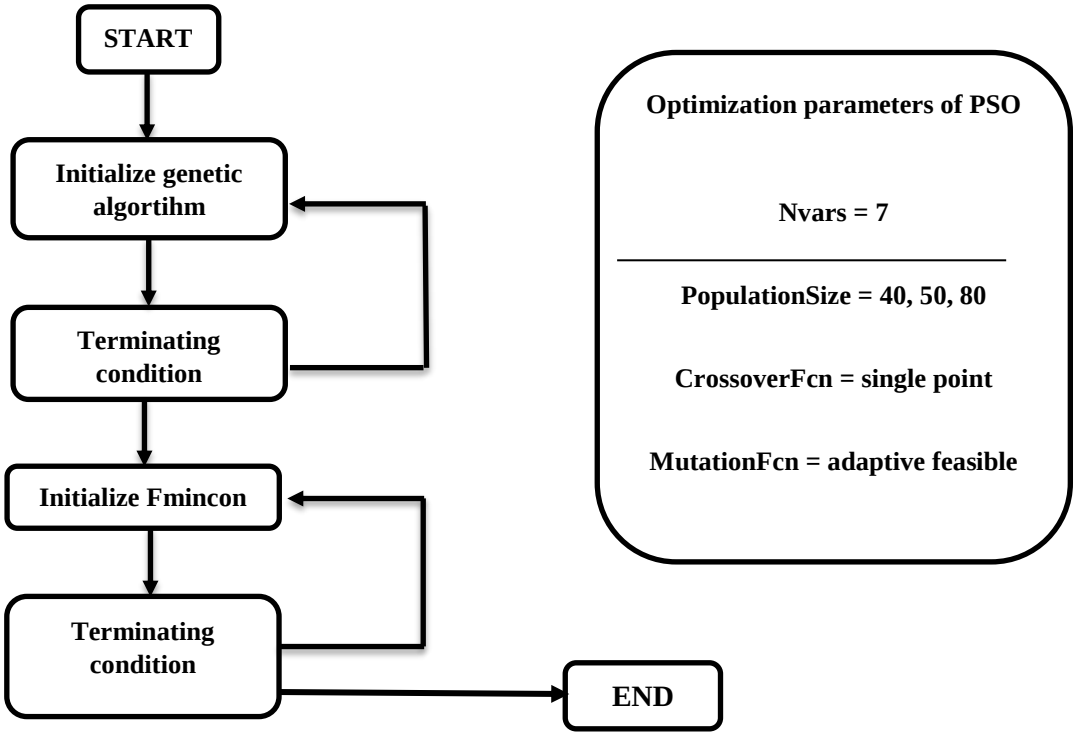


Fig. 6. Flow Chart and Optimization Parameters of GA+Fmincon.

Table 7. Optimization Results using GA.

Solution	Units	Run 1	Run 2	Run 3
x_1	cm	3.49999999991348	3.49999999999646	3.49999999999291
x_2	cm	0.700000000000000	0.700000000000000	0.700000000000000
x_3	-	17	17.0000000000027	17
x_4	cm	7.30000000000000	7.30000000000000	7.30000000000000
x_5	cm	7.71531990724617	7.71531991150423	7.71531991146436
x_6	cm	3.35021466139672	3.35021466622022	3.35021466609643
x_7	cm	5.28665446492610	5.28665446500385	5.28665446496775
G1	cm^{-3}	-0.0739152803749812	-0.0739152803970836	-0.0739152803959985
G2	cm^{-3}	-0.197998527122124	-0.197998527141393	-0.197998527140325
G3	cm^{-3}	-0.499172245292149	-0.499172248176510	-0.499172248102407
G4	cm^{-3}	-0.904643910392746	-0.904643910240491	-0.904643910239350
G5	cm^{-3}	4.20843870863052e-09	-1.10834563749052e-10	1.82076576038526e-14
G6	cm^{-3}	-4.46735785031294e-08	-4.47168828632272e-08	-4.46964094624747e-08
G7	cm	-0.702500000000000	-0.702499999999953	-0.702500000000000
G8	-	2.47193376878840e-11	1.01230135385322e-12	2.02460270770644e-12
G9	-	-0.583333333343633	-0.583333333333755	-0.583333333334177
G10	cm^{-1}	-0.0513257545075229	-0.0513257535163936	-0.0513257535418300
G11	cm^{-1}	1.17723639725398e-08	1.12315523548290e-08	1.12315743372449e-08
Objective	kg	2994.3550260801335	2994.3549922651923	2994.355026076736
Optimiter parameters		*Default options*	*Population size=40* *Mutation function=adaptive feasible*	*Population size=80* *Crossover function=single point*

* The violated Constraints are highlted in Bold

The following Table 8 compares all optimization methods.

Table 8. Comparison of Optimizers.

Solution	SQP	GA	SA	PSO	GA+fmincon
Objective Value	2994.355	2703.592	2730.74	2905.677	2994.355
Function Evaluation	80	45499	17339	7910	43299
Computation Time	0.642023	7.347204	17.119729	1.093691	6.970902
Ease of Implementation	Difficult	Medium	Medium	Medium	Easy

7. Problem Formulation

Optimization is a strong tool in finding the best solution for particular problem at given constraints and with the help of design variables, which can be altered. In this research, comparison between 5 different types of optimization methods is conducted on a benchmark problem, namely, the Golinski's Speed Reducer based on the results, one can notice that the fastest method of optimization is SQP method, in spite of the fact that it will not give the most accurate result, it is still applicable in cases quick result is required. SQP is gradient based method of optimization and that is why even after few runs, results will

be the same, in case of same initial values of design variables. So, the most difficult part in SQP method is to guess proper initial values for design variables in order to not violate the constraints. In this term, other optimization methods like GA, SA etc., are better since they do not require initial guessing. Also, SQP appears to be local optimizer whereas GA, SA, PSO are global ones. That is to say, that the result from SQP method will be for local minima or maxima, and it cannot be the best solution for the problem.

References

[1] Ray, T., 2003, " Golinski's speed reducer problem revisited ", *AIAA journal*, 41(3), 556-558.

[2] Datseris, P., 1982, "Weight minimization of a speed reducer by heuristic and decomposition techniques", *Mechanism and Machine Theory*, 17(4), 255-262.

[3] Deb, K., 2002, "A fast and elitist multiobjective genetic algorithm: NSGA-II", *IEEE transactions on evolutionary computation*, 6(2), 182-197.

[4] Kennedy, J., 2011, "Particle swarm optimization." Encyclopedia of machine learning, Springer US, 760-766.

[5] Dorigo, M. and Gambardella, L., 1997, "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE Transactions on evolutionary computation*, 1(1), 53-66.

[6] Hassan, R., 2005, "A comparison of particle swarm optimization and the genetic algorithm", 46th AIAA/ASME/ASCE/AHS/ASC Structures, *Structural Dynamics and Materials Conference*.

[7] Yang, C. and Simon, J., 2005, "A new particle swarm optimization technique", Systems Engineering, *18th International Conference on*, IEEE.

[8] Eberhart, C and Kennedy, J., 1995, "A new optimizer using particle swarm theory", Micro Machine and Human Science, *Proceedings of the Sixth International Symposium on*, IEEE.

[9] Eberhart, C. and Yuhui Shi., 1998, "Evolving artificial neural networks", *Proceedings of the International Conference on*, Neural Networks and Brain, 1(998).

[10] Eberhart, C., and Yuhui S., 1998, "Comparison between genetic algorithms and particle swarm optimization", *International Conference on Evolutionary Programming*, Springer Berlin Heidelberg.

International Journal of Energy Applications and Technologies, Year 2017, Vol. 4, No. 2, pp. 34-52

APPENDIX

SA

```
function Fout = Speed_Reducer_obj_sqp(x)


Fout = ((0.7854*x(1)*x(2)^2)*(3.3333*x(3)^2+14.9334*x(3)-43.0934)) -
((1.5079*x(1))*(x(6)^2+x(7)^2)) + (7.477*(x(6)^3+x(7)^3)) +
(0.7854*((x(4)*x(6)^2)+(x(5)*x(7)^2)));

    ceq = zeros(50,1);          % Equality constraints

    C = zeros(50,1);          % Non-equality constraints

%%25 Inequality constraints and 0 Equality constraints
C(1) = 27-x(1)*x(2)^2*x(3);                                  %Upper bound on the bending
stress of the gear tooth
C(2) = 397.5-x(1)*x(2)^2*x(3)^2;                             %Upper bound on the contact
stress of the gear tooth
C(3) = 1.93*x(4)^3-x(2)*x(3)*x(6)^4;                         %Upper bound on the
transverse deflection of the shaft 1
C(4) = 1.93*x(5)^3-x(2)*x(3)*x(7)^4;                         %Upper bound on the
transverse deflection of the shaft 2
C(5) = (((745*x(4)*(x(2)^-1)*(x(3)^-1))^2+16.9e6)^0.5)-110*x(6)^3;          %Upper bound
on the stresses in shaft 1
C(6) = (((745*x(5)*(x(2)^-1)*(x(3)^-1))^2+157.5e6)^0.5)-85*x(7)^3;          %Upper bound
on the stresses in shaft 2
C(7) = x(2)*x(3)-40;                                         %Dimensional restrictions based on
space and experience
C(8) = 5*x(2)-x(1);                                          %Dimensional restrictions based on
space and experience
C(9) = x(1)-12*x(2);                                         %Dimensional restrictions based on
space and experience
%{
C(10) = 2.6-x(1);                                            %Side constraints on x1
C(11) = x(1)-3.6;                                            %Side constraints on x1
C(12) = 0.7-x(2);                                            %Side constraints on x2
C(13) = x(2)-0.8;                                            %Side constraints on x2
C(14) = 17-x(3);                                             %Side constraints on x3
C(15) = x(3)-28;                                             %Side constraints on x3
C(16) = 7.3-x(4);                                            %Side constraints on x4
C(17) = x(4)-8.3;                                            %Side constraints on x4
C(18) = 7.3-x(5);                                            %Side constraints on x5
C(19) = x(5)-8.3;                                            %Side constraints on x5
C(20) = 2.9-x(6);                                            %Side constraints on x6
C(21) = x(6)-3.9;                                            %Side constraints on x6
C(22) = 5-x(7);                                              %Side constraints on x7
C(23) = x(7)-5.5;                                            %Side constraints on x7
```

```
%}
C(10) = 1.5*x(6)+1.9-x(4);                    %Dimensional requirements for
shafts based on experience
C(11) = 1.1*x(7)+1.9-x(5);                    %Dimensional requirements for
shafts based on experience
ceq = [];


SUM_OF_ALL_CONSTRAINT_VIOLATIONS = sum(C(find(C>0)));

assignin('base','cons',C);

Penalty = 2;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      end
```

RUN 1

```
function [x,fval,exitflag,output] = sa4(x0,lb,ub,ReannealInterval_Data)
tic
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.9 0.79 22.9 7.9 7.8 3.45 5.45]

%% Start with the default options
options = optimoptions('simulannealbnd');
%% Modify options setting
options = optimoptions(options,'AnnealingFcn', @annealingboltz);
options = optimoptions(options,'Display', 'iter');
options = optimoptions(options,'HybridInterval', 'end');
[x,fval,exitflag,output]=simulannealbnd(@Speed_Reducer_obj_sqp,x0,lb,ub,options);

Penalty = 2;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      end
toc
```

RUN 2

```
function [x,fval,exitflag,output] = sa1(x0,lb,ub)
tic
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
```

```
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.8 0.76 17.9 7.7 7.6 3.15 5.4];
%% Start with the default options
options = optimoptions('simulannealbnd');
%% Modify options setting
options = optimoptions(options,'Display', 'iter');
options = optimoptions(options,'HybridInterval', 'end');
[x,fval,exitflag,output] = simulannealbnd(@Speed_Reducer_obj_sqp,x0,lb,ub,options);

Penalty = 50;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      End
toc
```

RUN 3

```
function [x,fval,exitflag,output] = sa4(x0,lb,ub,ReannealInterval_Data)
tic
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.9 0.79 22.9 7.9 7.8 3.45 5.45]

%% Start with the default options
options = optimoptions('simulannealbnd');
%% Modify options setting
options = optimoptions(options,'AnnealingFcn', @annealingboltz);
options = optimoptions(options,'Display', 'iter');
options = optimoptions(options,'HybridInterval', 'end');
options = optimoptions(options,'ReannealInterval', 200);
[x,fval,exitflag,output]=simulannealbnd(@Speed_Reducer_obj_sqp,x0,lb,ub,options);

Penalty = 100;             %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      End
toc
```

PSO- RUN 1

```
function [x,fval,exitflag,output] = pso(nvars,lb,ub)
tic
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7;
```

```
% options = optimoptions ('Display', 'iter');
options = optimoptions ('particleswarm','SwarmSize',50);
[x,fval,exitflag,output]= particleswarm(@Speed_Reducer_obj_sqp,nvars,lb,ub)

Penalty = 2;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      End
toc
```

RUN 2

```
function [x,fval,exitflag,output] = pso(nvars,lb,ub)
tic
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7;
% options = optimoptions ('Display', 'iter');
options = optimoptions ('particleswarm','SwarmSize',60);
[x,fval,exitflag,output]= particleswarm(@Speed_Reducer_obj_sqp,nvars,lb,ub)

Penalty = 10;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      End
toc
```

RUN 3

```
function [x,fval,exitflag,output] = pso(nvars,lb,ub)
tic
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7;
% options = optimoptions ('Display', 'iter');
options = optimoptions ('particleswarm','SwarmSize',70);
[x,fval,exitflag,output]= particleswarm(@Speed_Reducer_obj_sqp,nvars,lb,ub)

Penalty = 50;              %   try: 2, 10, 50, 100
      if isempty(SUM_OF_ALL_CONSTRAINT_VIOLATIONS)
         Fout = Fout;
      else
         Fout = (Fout) +  Penalty*SUM_OF_ALL_CONSTRAINT_VIOLATIONS;
      End
toc
```

GA+Fmincon- RUN 1

```
function [x,fval,exitflag,output,population,score] = hybrid_main2(nvars,lb,ub)
% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7
% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'HybridFcn', {  @fmincon [] });
options = gaoptimset(options,'Display', 'iter');
[x,fval,exitflag,output,population,score]=ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,
@Speed_Reducer_con_sqp,[],options);
```

RUN 2

```
function [x,fval,exitflag,output,population,score] =
hybrid_main1(nvars,lb,ub,PopulationSize_Data)
tic

% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7

% Start with the default options
options = gaoptimset;
% Modify options setting
options = gaoptimset(options,'PopulationSize', 40);
options = gaoptimset(options,'MutationFcn', @mutationadaptfeasible);
options = gaoptimset(options,'HybridFcn', {  @fmincon [] });
options = gaoptimset(options,'Display', 'iter');
[x,fval,exitflag,output,population,score] =
ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,[], options);
toc
```

RUN 3

```
function [x,fval,exitflag,output,population,score] =
adadsasdasd(nvars,lb,ub,PopulationSize_Data)
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7
%% Start with the default options
options = optimoptions('ga');
%% Modify options setting
```

```
options = optimoptions(options,'PopulationSize', 80);
options = optimoptions(options,'CrossoverFcn', @crossoversinglepoint);
options = optimoptions(options,'HybridFcn', {  @fmincon [] });
options = optimoptions(options,'Display', 'iter');
[x,fval,exitflag,output,population,score] =
ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,[],options);
```

SQP

```
function Fout = Speed_Reducer_obj_sqp(x)

Fout = ((0.7854*x(1)*x(2)^2)*(3.3333*x(3)^2+14.9334*x(3)-43.0934)) -
((1.5079*x(1))*(x(6)^2+x(7)^2)) + (7.477*(x(6)^3+x(7)^3)) +
(0.7854*((x(4)*x(6)^2)+(x(5)*x(7)^2)));

function  [c,ceq] =Speed_Reducer_con_sqp(x)

ceq=0;
c = zeros(50,1);        %   Non-equality constraints
      c(1) = (27*x(1)^-1*x(2)^-2*x(3)^-1) - 1;
      c(2) = (397.5*x(1)^-1*x(2)^-2*x(3)^-1) - 1;
      c(3) = (1.93*x(2)^-1*x(3)^-1*x(4)^3*x(6)^-4) - 1;
      c(4) = (1.93*x(2)^-1*x(3)^-1*x(5)^3*x(7)^-4) - 1;
      c(5) = ((((745*x(4)*x(2)^-1*x(3)^-1)^2) + (16.9*10^6))^0.5) / (110.0*x(6)^3) - 1;
      c(6) = ((((745*x(5)*x(2)^-1*x(3)^-1)^2) + (157.5*10^6))^0.5) / (85.0*x(7)^3) - 1;
      c(7) = ((x(2)*x(3)) / 40) - 1;
      c(8) = ((5*x(2)) / x(1)) - 1;
      c(9) = (x(1) / (12*x(2))) - 1;
      c(10) = (((1.5*x(6)) + 1.9)*x(4)^-1) - 1;
      c(11) = (((1.1*x(7)) + 1.9)*x(5)^-1) - 1;
 assignin('base','Constraint',c);
```

RUN 1

```
clear all; close all; clc;
format short
tic

%% SQP
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.7 0.71 17.5 7.4 7.4 2.95 5.1]

Options = optimset('LargeScale','on','Display','iter','MaxIter',10000,'TolFun',1e-20,'Tolcon',1e-
20,'TolX',1e-5,'MaxFunEval',50000);
```

```
[X1,fval,exitflag,output]=fmincon(@Speed_Reducer_obj_sqp,x0,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,Options);
 toc
```

RUN 2

```
clear all; close all; clc;
format short
tic

%% SQP
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6     0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.8 0.76 17.9 7.7 7.6 3.15 5.4]

Options = optimset('LargeScale','on','Display','iter','MaxIter',10000,'TolFun',1e-30,'Tolcon',1e-25,'TolX',1e-5,'MaxFunEval',50000);
[X1,fval,exitflag,output]=fmincon(@Speed_Reducer_obj_sqp,x0,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,Options);
 toc
```

RUN 3

```
clear all; close all; clc;
format short
tic

%% SQP
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6     0.8   28    8.3   8.3   3.9   5.5];
x0 = [2.9 0.79 22.9 7.9 7.8 3.45 5.45]

Options = optimset('LargeScale','on','Display','iter','MaxIter',10000,'TolFun',1e-20,'Tolcon',1e-10,'TolX',1e-10,'MaxFunEval',50000);
[X1,fval,exitflag,output]=fmincon(@Speed_Reducer_obj_sqp,x0,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,Options);
 toc
```

GA - RUN 1

```
function [x,fval,exitflag,output,population,score] = ga_main1(nvars,lb,ub)
tic
% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6     0.8   28    8.3   8.3   3.9   5.5];
nvars=7
% Start with the default options
options = gaoptimset;
```

```
% Modify options setting
options = gaoptimset(options,'Display', 'iter');
[x,fval,exitflag,output,population,score] =
ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,[],options);
toc
```

RUN 2

```
function [x,fval,exitflag,output,population,score] = sa(nvars,lb,ub,PopulationSize_Data)
tic
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7
%% Start with the default options
options = optimoptions('ga');
%% Modify options setting
options = optimoptions(options,'PopulationSize', 60);
options = optimoptions(options,'CrossoverFcn', {  @crossoverintermediate [] });
options = optimoptions(options,'Display', 'iter');
[x,fval,exitflag,output,population,score] =
ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,[],options);
toc
```

RUN 3

```
function [x,fval,exitflag,output,population,score] = sa(nvars,lb,ub,PopulationSize_Data)
tic
%% This is an auto generated MATLAB file from Optimization Tool.
lb = [2.6    0.7   17    7.3   7.3   2.9   5.0];
ub = [3.6    0.8   28    8.3   8.3   3.9   5.5];
nvars=7

%% Start with the default options
options = optimoptions('ga');
%% Modify options setting
options = optimoptions(options,'PopulationSize', 80);
options = optimoptions(options,'MigrationDirection', 'both');
options = optimoptions(options,'CrossoverFcn', {  @crossoverintermediate [] });
options = optimoptions(options,'Display', 'iter');
[x,fval,exitflag,output,population,score] =
ga(@Speed_Reducer_obj_sqp,nvars,[],[],[],[],lb,ub,@Speed_Reducer_con_sqp,[],options);
toc
```