Particle Swarm Optimization – Tabu Search Approach to Constrained Engineering Optimization Problems

RITCHIE MAE GAMOT, ARMACHESKA MESA
Department of Mathematics, Physics and Computer Science
University of the Philippines in Mindanao
Mintal, Davao City 8000
PHILIPPINES
ariane43981@gmail.com, arma_cheska@yahoo.com, http://www.upmin.edu.ph

Abstract: - Constraint handling is one of the most difficult parts encountered in practical engineering design optimizations. Different kinds of methods were proposed for handling constraints namely, genetic algorithm, self-adaptive penalty approach and other evolutionary algorithms. Particle Swarm Optimization (PSO) efficiently solved most nonlinear optimization problems with inequity constraints. This study hybridizes PSO with a meta-heuristic algorithm called Tabu Search (TS) to solve the same engineering design problems. The algorithm starts with a population of particles or solution generated randomly and is updated using the update equations of PSO. The updated particles are then subjected to Tabu Search for further refinement. The PSO algorithm handles the global search for the solution while TS facilitates the local search. With embedded hyrbridization, this study which we call PSO-TS, showed better results compared to algorithms reported in Hu et al's study as applied to four benchmark engineering problems. Specifically, this study beat the results of Coello, Deb and Hu.

Key-Words: - constrained engineering optimization problems, particle swarm optimization, tabu search

1 Introduction

Constraint handling is one of the most difficult parts encountered in engineering design optimizations. These constraints often limit the feasible solution to a small subset. Generally, a constrained optimization problem can be described as follows:

Minimize $f(X), X = \{x_1, x_2, ..., x_n\} \in R$ (1)

Subject to $g_i(X) \leq 0, i = 1,2,..., p$
and $h_i(X) \leq 0, i = 1,2,..., m$
where $x_i^{(L)} \leq x \leq x_i^{(U)}, i = 1,2,..., n$

They are composed of three basic components: a set of variables, a fitness function that must be optimized and a set of constraints that specify the feasible spaces of the variables. The objective is to locate the values of the variables that optimize the fitness function while satisfying the constraints.

Different general deterministic solutions were proposed for handling constrains in recent years. However, deterministic methods make strong assumptions on the continuity and differentiability of the objective function. The complexity and unpredictability of constraints also makes a general deterministic solution to this type of problem hard to find. Thus, there is an ongoing popularity for stochastic/heuristic algorithms that can tackle constrained optimization effectively.

Coello [1] used the notion of using co-evolution to adapt the penalty factors of a fitness function incorporated in genetic algorithm for numerical optimization. The proposed approach produced better solutions as compared to studies of Homaifar et al (Genetic Algorithm variant), Himmelblau (Applied Nonlinear Programming) and Gen et al (Genetic Algorithm variant).

Hu et al's, in 2002 [6], showed that a new evolutionary algorithm Particle Swarm Optimization (PSO) is an efficient and general solution to solve most of the twelve nonlinear optimization problems with nonlinear inequality constraints.

Hu et al, in 2003, used Particle Swarm Optimization again in four engineering optimization problems and showed that PSO outperformed the studies of Coello, Gen, Homaifar, Arora and Deb.

The promise of PSO in solving constrained nonlinear optimization problems inspired the creation of this paper. Here, two heuristics namely, Particle Swarm Optimization and Tabu Search is “hybridized” to solve nonlinear engineering optimization problems with constraints.

2 Particle Swarm

Particle Swarm Optimization (PSO) is a relatively new evolutionary computation technique [3]. In fact, the first book dedicated entirely to Swarm

Intelligence was just published in 2001 by Morgan Kaufmann Publisher.

PSO is simple in concept with few parameters to adjust and is easy to implement. The core of PSO is the updating formulae of the particle. The first equation below calculates a new velocity for each particle based on its previous velocity (v_{id}), the particle's location at which the best fitness so far has been achieved (p_{id} or pBest), and the neighbor's best location (p_{gd} or gBest) at which the best fitness in a neighborhood so far has been achieved. The second term in the first equation is called the cognitive part while the third term is the social part. The second equation below updates each particle's position in the solution hyperspace. rand() and Rand() are two random numbers independently generated. c1 and c2 control the influences of pbest and gbest and are called learning factors.

$$v_{id} = w_i * v_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id}) \quad (2)$$

$$x_{id} = x_{id} + vid \quad (3)$$

A traditional, simple PSO algorithm has the following steps:

1. Initialize a population of particles with random positions and velocities on D dimensions in the problem space.
2. For each particle, evaluate the desired optimization fitness function in D variables.
3. Compare the particle's fitness evaluation with its previous best (pBest). If current value is better than pbest, then set pBest equal to the current value, and Pi equals to the current location Xi in D-dimensional space.
4. Identify the particle in the neighborhood with the best success so far, and assign its index to the variable g.
5. Update the velocity and position of the particle
6. Loop to step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

There are two versions of PSO. The local PSO algorithm is preferred for more accurate results while the global version is faster.

The PSO approach has the following advantages [5]:

(1) There are not many parameters to be adjusted and optimized.

(2) Preprocessing and complicated manipulations are not present in the algorithm. Fitness function and constraints are handled separately, thus there are no limitations to the constraints.

(2) The only part of the algorithm that deals with the constraints is to check if a solution satisfies the constraints.

Aside from constrained optimization problems, PSO has proven its effectiveness in solving problems such as integer programming. It was also used as a preprocessor for generating good initial points in a branch and bound technique of an integer programming problem. A study of Zhang et al in 2003 compared PSO with genetic algorithm in solving multiobjective optimization problems. It was mentioned in the study that the proposed algorithm can be understood and performed easily due to the fact that there are no operations such as crossover and mutation used.

3 Tabu Search

Tabu Search, TS, according to the Oxford dictionary, is a social or religious custom prohibiting or restricting a particular practice or forbidding association with a particular person, place, or thing. TS, as described by Glover and Hansen in 1986, "…is a meta-heuristic superimposed on another heuristic. The overall approach is to avoid entrainment in cycles by forbidding or penalizing moves which take the solution, in the next iteration, to points in the solution space previously visited ("hence tabu")". The main principle behind TS is that it has some memory of the states that it has already investigated and it does not re-visit those states again for some time.

This scheme helps in two ways: (1) Avoids the search getting into a loop by continually searching the same area without actually making any progress, and (2) Helps the search explore regions that it might not otherwise explore.

The TS algorithm can be summarized as follows:

Tabu Search (particle X, num of iterations, tabu size)

1. Let *best* be initially equal to particle X.
2. Clear the Tabu List.
3. For 1 to num of iterations
   i. Let bestmove be null.
   ii. Let bestpair be null.
   iii. For each unordered pair (i,j) not in the tabu list and must be within the specified range.
      a. swapped = swap ith and jth element in best.
      b. if fitness (swapped) is better than fitness (bestmove) then

bestmove = swapped and bestpair = (i,j).

c. best = bestmove.

d. if cardinality(tabu list) = tabu size then bestpair replaces oldest pair; else add bestpair to tabu list.

4. If fitness(best) is better than fitness(X), return best; else return X.

Tabu search was used by Ahr and Reinelt in 2006 for the min-max k-Chinese postman problem. Extensive computation results showed that the tabu search algorithm outperforms all known heuristics and improvement procedures mentioned in their study.

4 Hybrid Methods

There are many opportunities to hybridize to develop sophisticated tools for supporting design. Hybridizing algorithms in solving optimization problems have produced better results.

A work of Hwang and He used a hybrid genetic algorithm and simulated annealing. It is a novel adaptive real-parameter simulated annealing genetic algorithm that maintains the merits of genetic algorithm and simulated annealing. Adaptive mechanisms were added to ensure the solution quality and to improve the convergence speed. The performance of the algorithm is tested on a helical spring optimization design case and system identification problem described by the auto regressive and moving average exogenous model. The results was significantly better than the other genetic-algorithm based methods.

Another hybrid presented a solution model for the unit commitment problem using fuzzy logic to address uncertainties in the problem. This study was done by Victoire et al in 2006 were hybrid tabu search, particle swarm optimization and sequential quadratic programming was used to schedule the generating units based on the fuzzy logic decisions. Results showed, based on extensive numerical simulations, when uncertainties are considered, that the presented model improves the secure operation of the system.

5 Constrained Engineering Optimization Problems

Four constrained engineering optimization were used to test the performance of the hybrid algorithm. The same test problems were used in Hu et. al's study. The test problems are the following:

(1) Design of a Pressure Vessel
(2) Welded Beam Design
(3) Minimizatio of the Weight of a Tension/Compression Spring; and
(4) Himmelblau's Nonlinear Optimization Problem

4.1 Design of a pressure vessel (PVD)

The objective of the problem is to minimize the total cost of the material, forming and welding of a cylindrical vessel. There are four design variables: x_1 (T_s, thickness of the shell), x_2 (T_h, thickness of the head), x_3 (R, inner radius), and x_4 (L, length of the cylindrical section of the vessel), T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, R and L are continuous . The problem can be specified as follows:

Minimize

$$f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_2^3 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(X) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \text{ and}$$

$$g_4(X) = x_4 - 240 \le 0$$

The following ranges of the variables were used: $1 \le x_1 \le 99,\ 1 \le x_2 \le 99,\ 10.0 \le x_3 \le 200.0,\ 10.0 \le x_4 \le 200.0$.

4.2 Welded Beam Design (WBD)

The objective is to minimize the cost of the welded beam subject to the constraints on shear stress, bending stress in the beam, bucking load on the bar, end deflection of the beam, and side constraints. The problem can be stated as follows:

Minimize

$$f(X) = 1.10471x_1^2 x_2 + 0.04711x_3x_4(14.0 + x_2)$$

Subject to

$$g_1(X) = \tau(X) - \tau_{max} \le 0$$
$$g_2(X) = \sigma(X) - \sigma_{max} \le 0$$
$$g_3(X) = x_1 - x_4 \le 0$$
$$g_4(X) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \le 0$$
$$g_5(X) = 0.125 - x_1 \le 0$$
$$g_6(X) = \delta(X) - \delta_{max} \le 0 \text{ ; and}$$
$$g_7(X) = P - P_c(X) \le 0$$

where

$$\tau(X) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2}$$
$$\tau' = \frac{P}{\sqrt{2}x_1x_2},\ \tau'' = \frac{MR}{J},\ M = P\left(L + \frac{x_2}{2}\right)$$
$$R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$
$$\sigma(X) = \frac{6PL}{x_4x_3^2},\ \delta(X) = \frac{4PL^3}{Ex_3^2x_4}$$
$$P_c^X = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

P = 6000lb, L = 14in, E = 30 x 10^6 psi, G = 12 x 10^6 psi, τ_{max} = 13600psi, σ_{max} = 30000psi, δ_{max} = 0.25in.

The following ranges of the variables were used: $0.1 \le x_1 \le 2.0,\ 0.1 \le x_2 \le 10.0,\ 0.1 \le x_3 \le 10.0,\ 0.1 \le x_4 \le 2.0$.

4.3 Minimization of the Weight of a Tension/Compression Spring (WTS)

The problem consists of minimizing the weight of a tension/compression spring subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are the mean coil diameter D, the wire diameter d and the number of active coils. The problem can be expressed as follows:

Minimize

$$f(X) = (N + 2)Dd^2$$

Subject to

$$g_1(X) = 1 - \frac{D^3N}{71785d^4} \le 0$$
$$g_2(X) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \le 0$$
$$g_3(X) = 1 - \frac{140.45d}{D^2N} \le 0 \text{ ; and}$$
$$g_4(X) = \frac{D + d}{1.5} - 1 \le 0$$

The following ranges for the variables were used: $0.05 \le x_1 \le 2.0,\ 0.25 \le x_2 \le 1.3,\ 2.0 \le x_3 \le 15.0$.

4.4 Himmelblau's Nonlinear Optimization Problem (HMO)

This problem was proposed by Himmelblau and it has been used before as benchmark for several evolutionary algorithm based techniques. In this problem, there are five design variables, six nonlinear inequality constraints and ten boundary conditions. The problem can be stated as follows:

Minimize

$$f(X) = 5.3578547x_3^2 + 0.8356894x_1x_5 + 37.2932239x_1 - 40.792.141$$

Subject to

$$0 \le 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5 \le 92$$
$$90 \le 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110 \text{ ; and}$$
$$20 \le 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25$$

where

$$78 \le x_1 \le 102,\ 33 \le x_2 \le 45,\ 27 \le x_3 \le 45,\ 27 \le x_4 \le 45,\ 27 \le x_5 \le 45$$

6 PSO-TS Hybrid Algorithm

This sections presents the particle swarm optimization – tabu search hybrid for the constrained engineering optimization design problems. We shall refer to the hybrid as PSO-TS Hybrid Algorithm.

The PSO part of this algorithm handles of the global search of solutions while Tabu Search, which is embedded in the developed algorithm, deals with the local search of solutions.

5.1 PSO – TS

The PSO-TS Hybrid algorithm maintains a single population thru PSO where each member of the population is improved using TS. The PSO Hybrid algorithm can be summarized as follows:

1. For each particle{
   i. REPEAT randomly initialize particle until it satisfies all constraints and variable ranges.
   ii. Compute fitness value (f(X)) of the particle and let the particle be its own initial pBest.
   iii. Set initial velocity of the particle.
   }
2. Let the particle with the optimum fitness value be the gBest.
3. Do{
   For each particle{
   i. Calculate new particle velocity using equation $v_{id.}$
   ii. Calculate new particle position using equation xid.
   iii. Further refine the solution by submitting the particle to Tabu Search.
   }
   While maximum number of iterations or optimal value is attained.

}

5.2 Parameter Values

There is only one set of parameter values settings for PSO. The following are the PSO parameter values:

(1) learning factors = 1.49445, both for c1 and c2, ,
(2) initial weight = [0.5 + (Rand/2.0],
(3) number of iterations = 10,000,
(4) initial population size = 20; and
(5) maximum velocity = set to dynamic range of the variables.

The parameters values used in the study is the same as in Hu et al's study [5] to facilitate fair comparison of results.

The experimental nature of the study resulted into testing the different engineering design problems using several parameter settings for Tabu Search. Thus, the TS parameters are the following:

(1) number of iterations = {10 or 15}; and
(2) tabu size = {10 or 15}.

There is a total of 4 parameter values combinations for TS parameters. Table 1 shows the four parameter settings for the PSO-TS hybrid.

Table 1. Different PSO-TS parameter settings.

Parameter	PSO-TS1	PSO-TS2	PSO-TS3	PSO-TS4
Learning Factors	1.49445	1.49445	1.49445	1.49445
Initial Weight	[0.5+ (Rand/2.0]	[0.5 + (Rand/2.0]	[0.5+ (Rand/2.0]	[0.5 + (Rand/2.0]
Number of iterations	10,000	10,000	10,000	10,000
Population Size	20	20	20	20
Maximum Velocity	dynamic range of the variables	dynamic range of the variables	dynamic range of the variables	dynamic range of the variables
Tabu Search iterations	10	15	15	10
Tabu Search tabu size	10	15	10	15

5 Experimental Results

The following sections show the results of the PSO-TS algorithm compared to its pure counterpart and best results found in literature.

5.1 Performance Criterion

The PSO-TS was initially compared to its pure PSO counterpart using decreased values of parameters to determine if the introduction of Tabu Search improves the solution.

Since the algorithm is probabilistic in nature, several runs must be done. The number of runs done for each parameter setting and for each problem is the same the number of runs done in the study of Hu et al which is eleven.

The different PSO-TS parameter settings were also run and compared to each other in terms of mean solution quality and mean solution time. The best mean solution quality of the four parameter settings was identified. The best solution run found in the PSO-TS results was compared to studies mentioned in Hu et al [5] and He et al's [4] studies.

5.2 PSO-TS versus PSO

Table 2 shows the mean value of the solution for PSO-TS and PSO.

Table 2. Comparison of mean value of the solution for PSO-TS and PSO.

Test Problem	PSO-TS	PSO
PVD	6566.89310349	6613.06155947
WBD	1.75187099	1.75531979
WTS	0.01852839	0.01976658
HMO	-30896.39901488	-30825.23054996

It is evident that PSO-TS beat its pure PSO counterpart in all the test problems in optimizing the four test problems. Although extra loops are needed for the PSO-TS hybrid to finish its iterations, the time complexity is not high as expected.

5.3 Best PSO-TS parameter setting

Tables 3 to 6 show the result of each parameter setting for the four test problems.

All the runs for all parameter settings for both the HMO and WTS obtained the same results. We can say that these two test problems are not sensitive to the different parameter settings we subjected.

The test problems PVD and WBD, on the other hand, obtained different results for each run. The best parameter setting for PVD and WBD are PSO-TS2 and PSO-TS1, respectively (Tables 3 and 4). The PVD test problem thus needs larger number of iterations for tabu search, 15, and larger number of tabu size, 15 to obtain better results. The WBD test problem, on the other hand, needs fewer number of tabu iterations, 10 and larger number of tabu size, 15.

Table 3. Comparison of mean value of the different PSO-TS parameter settings for the Pressure Vessel Design.

Parameter Setting	Mean Value
PSO-TS1	6168.40064528
PSO-TS2	5992.34600472
PSO-TS3	6101.16043578
PSO-TS4	6216.98564958

Table 4. Comparison of mean value of the different PSO-TS parameter settings for the Welded Beam Design.

Parameter Setting	Mean Value
PSO-TS1	1.77073287
PSO-TS2	1.84312439
PSO-TS3	1.82342282
PSO-TS4	1.80538152

Table 5. Comparison of mean value of the different PSO-TS parameter settings for the Weight of Tension/Compression Spring.

Parameter Setting	Mean Value
PSO-TS1	0.00607614
PSO-TS2	0.00607614
PSO-TS3	0.00607614
PSO-TS4	0.00607614

Table 6. Comparison of mean value of the different PSO-TS parameter settings for the Himmelblau Nonlinear Optimization Problem.

Parameter Setting	Mean Value
PSO-TS1	-31025.56149103
PSO-TS2	-31025.56149103
PSO-TS3	-31025.56149103
PSO-TS4	-31025.56149103

5.4 PSO-TS versus previous results

Tables 7, 9, 11, and 13 show detailed results of the best solution run of PSO-TS on each of the test problem with comparison on the values obtained per design variable. The last row on each table is the value of interest for comparison. The minimal the value, the better the approximation is.

Tables 8, 10, 12 and 14 show a line-by-line comparison of the best solution run from the 11 runs for each test problem. The tables show that PSO-TS showed better results for each of the test problem compared to the results of previous methods. The goodness of results shown by PSO-TS in comparison to previous results is very evident in the test problems pressure vessel design and weight of compression/tension spring. Table 9 shows the absolute errors of the PSO-TS with respect to the best solution found so far in literature.

Table 7. Comparison of the results for pressure vessel design problem.

Vars	PSO-TS	Eberhart	Coello	Deb
x1	0.77816864	0.8125	0.8125	0.9875
x2	0.38464916	0.4375	0.4375	0.5000
x3	40.31961872	42.09845	40.3239	48.3290
x4	200.0	176.6366	200.0000	112.6790
g1(X)	0.0	0.0	-0.034324	-0.004750
g2(X)	0.0	-0.03588	-0.052847	-0.038941
g3(X)	0.0	-5.8208E-11	-27.105845	-3652.876838
g4(X)	-40.0	-63.3634	-40.0000	-127.321000
f(X)	**5885.33277362**	**6059.131296**	**6288.7445**	**6410.3811**

Table 8. Comparison of the results of different methods for pressure vessel design.

METHOD	f(X)
This paper (PSO-TS)	5885.33277362
Ebehart (modified PSO)	6059.131296
Coello (self-adaptive penalty approach)	6288.7445
Deb (GeneAs)	6410.3811
He and Wang (CPSO)	6061.0777
Sandgren (1988)	8129.1036
Kannan and Kramer (1994)	7198.0428
Coello and Montes (2002)	2.433116
Coello and Montes (feasibility-based tournament selection)	6059.9463

Table 9. Comparison of the results for welded beam design.

Vars	This paper	Eberhart	Coello	Deb
x1	0.20572964	0.20573	0.2088	0.2489
x2	3.47048867	3.47049	3.4205	6.1730
x3	9.03662391	9.03662	8.9975	8.1739
x4	0.20572964	0.20573	0.2100	0.2533
g1(X)	0.0	-0.023712	0.337812	-5758.603777
g2(X)	0.0	-0.026564	-353.902604	-255.576901
g3(X)	0.0	0.0	-0.00120	-0.004400
g4(X)	-3.43298379	-3.432982	-3.411865	-2.982866
g5(X)	-0.08072964	-0.08073	-0.08380	-0.123900
g6(X)	-0.23554032	-0.235540	-0.235649	-0.234160
g7(X)	0.0	-0.029809	-363.232384	-4465.270928
f(X)	**1.72485231**	**1.72485512**	**1.74830941**	**2.43311600**

Table 10. Comparison of the results of different methods for welded beam design problem.

METHOD	f(X)
This paper (PSO-TS)	1.72485231
Ebehart (modified PSO)	1.72485512
Coello (self-adaptive penalty approach)	1.74830941
Arora (constraint correction at constant cost)	2.43311600
He and Wang (CPSO)	1.728024
Ragsdell and Phillips (Geometric programming)	2.385937
Deb (GA)	2.433116
Coello and Montes (feasibility-based tournament selection)	1.728226

Table 11. Comparison of the results for minimization of the weight of a tension/compression spring.

Vars	This paper	Eberhart	Coello	Arora
x1	0.05	0.05146637	0.051480	0.05396
x2	0.60761419	0.35138395	0.351661	0.399180
x3	2.00000000	11.60865920	11.632201	9.185400
g1(X)	0.0	-0.00333661	-0.002080	0.000019
g2(X)	-0.84560388	-1.0970128E-4	-0.000110	-0.000018
g3(X)	-8.5105566	-4.02631810	-4.026118	-4.123842
g4(X)	-0.56159054	-0.73123933	-0.731239	-0.698283
f(X)	**0.00607614**	**0.01266614**	**0.01270478**	**0.12730274**

Table 12. Comparison of the results of different methods for the minimization of the weight of a tension/compression spring.

METHOD	f(X)
This paper (PSO-TS)	0.00607614
Ebehart (modified PSO)	0.01266614
Coello (self-adaptive penalty approach)	0.01270478
Arora (constraint correction at constant cost)	0.12730274
He and Wang (CPSO)	0.0126747
Belegundu (numerical optimization technique)	0.0128334
Coello and Montes (feasibility-based tournament selection	0.0126810

Table 13. Comparison of the results for himmelblau's nonlinear optimization problem.

Vars	This paper	Eberhart	Coello	Humaifar
x1	78.0	78.0	78.0495	78.0000
x2	33.0	33.0	33.0070	33.0000
x3	27.07099688	27.070997	27.0810	29.9950
x4	45.0	45.0	45.0000	45.0000
x5	44.96924246	44.96924255	44.9400	36.7760
g1(X)	92.00000001	92.0	91.997635	90.714681
g2(X)	100.40478426	100.4047843	100.407857	98.840511
g3(X)	19.9999999	20.0	20.001911	19.999935
f(X)	**-31025.561491**	**-31025.56142**	**-31020.859**	**-30665.609**

Table 14. Comparison of the results of different methods for himmelblau's nonlinear optimization problem.

METHOD	f(X)
This paper (PSO-TS)	-31025.561491
Ebehart (modified PSO)	-31025.56142
Coello (self-adaptive penalty approach)	-31020.859
Humaifar (co-evolutionary PSO)	-30665.609

Table 15. Absolute error of PSO-TS results with respect to the best solution found in literature.

Test Problem	PSO-TS	Best in Literature	Absolute Error
PVD	5885.33277362	6059.131296	173.7985224
WBD	1.72485231	1.72485512	0.00000281
WTS	0.00607614	0.01266614	0.00659
HMO	-31025.561491	-31025.56142	0.000071

6 Conclusion

On all engineering problems the hybrid performed better with no exception. PSO-TS was compared to PSO and results showed that the former outperformed the latter in all test problems.

Different parameter settings were also set and subjected it to all the test problems. It was identified that the problems HMO and WTS were not sensitive to different settings of parameters. PVD and WBD, on the other hand, obtained different results for each run. Of the four parameter settings for PVD, higher value setting for tabu iterations and tabu size were needed to obtain the best average solution. Out of the four parameter settings for WBD, smaller number of tabu iterations and higher tabu size was necessary to obtain the best average solution.

The best solution run for each of the parameter settings were obtained and compared to the results mentioned in previous literatures. All the comparisons showed that PSO-TS outperformed other methods in all test problems.

It is well known that practical engineering optimization involves multiple, nonlinear and non-trivial constraints due to real world limitations. From an engineering standpoint a better, faster, cheaper solution is always desired. With the success of the PSO hybrid in finding the optimal solution, the algorithm offers a better alternative in solving constrained engineering optimization problems.

7 Future Work

The authors will also hybridize PSO with other single solution heuristics for comparison with PSO-TS. Currently, PSO-SA or Particle Swarm Optimization – Simulated Annealing is applied on the same set of test problems. Results of the study will be compared with the results of PSO-TS and the results of previous studies mentioned in this paper.

References:

[1] X1. Coello, C.A.C., Use of a self-adaptive penalty approach for engineering optimization problems, *Computers in Industry*, Vol.41, 2000, pp. 113-127.

[2] X2. Deb, K., GeneAS: A robust optimal design technique for mechanical component design, *In Dasgupta, D., Michalewicz, Z. (Eds.), Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, 1997.

[3] X3. Eberhart, R.C. and J. Kennedy, Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol 4, 1995, pp. 1942-1948.

[4] X4. He, Q. and L. Wang, An effective co-evolutionary PSO for constrained engineering design problems, *Engineering Applications of Artificial Intelligence,* 2006, In press.

[5] X5. Hu, X., R.C. Eberhart and Y. Shi, Engineering optimization with particle swarm, *Proceedings of the IEEE Swarm Intelligence Symposium,* 2003, pp. 53-57.

[6] X6. Hu, X., and R.C. Eberhart, Solving Constrained Nonlinear Optimization Problems with Particle Swarm Optimization, *Proceedings of the Sixth World Multiconference on Systemics, Cybernetics and Informatics 2002 (SCI 2002),* Orlando, USA. 2002