Increasing Successful Offspring and Diversity in Differential Evolution for Engineering Design

Efrén Mezura-Montes†, Carlos A. Coello Coello‡ and Jesús Velázquez-Reyes‡
†Apizaco Institute of Technology, Computer Science and Systems Department, Apizaco, Tlaxcala, MEXICO
emezura@itapizaco.edu.mx
‡CINVESTAV-IPN, Computer Science Section, Electrical Eng. Department, México D.F. MEXICO
ccoello@cs.cinvestav.mx, jesus.velazquez@gmail.com

ABSTRACT

We propose a modified version of the differential evolution approach to solve engineering design problems. The aim is to allow each parent in the population to generate more than one offspring at each generation and therefore, to increase its probability of generating a better offspring. To deal with constraints, we use some criteria based on feasibility and a diversity mechanism to maintain infeasible solutions in the population. The approach is tested against penalty function approaches and its performance is also compared against state-of-the-art approaches.

1. INTRODUCTION

Many engineering design problems can be stated like the general nonlinear programming problem in which we want to:

Find $\vec{x}$ which optimizes $f(\vec{x})$

subject to:

$$g_i(\vec{x}) \leq 0,\ i = 1, \ldots, m$$
$$h_j(\vec{x}) = 0,\ j = 1, \ldots, p$$

where $\vec{x} \in \Re^n$ is a vector of solutions $\vec{x} = [x_1, x_2, \ldots, x_n]^T$, where each x_k, $k = 1, \ldots, n$ is bounded by lower and upper limits $L_i \leq x_k \leq U_k$, "m" is the number of inequality constraints and "p" is the number of equality constraints (in both cases, constraints could be linear or nonlinear).

Differential Evolution (DE) is a novel evolutionary algorithm proposed by Storn and Price [8]. The approach works with a mutation operator which is based on the current distribution of solutions in the population, instead of being based on a fixed (usually Gaussian) distribution such as other Evolutionary Algorithms (EAs) like Evolution Strategies [1]. At each generation in DE, each parent will generate one offspring. If this child is better than its parent, it will replace him in the population; if not, the parent will remain and the child is eliminated. Three random individuals are selected to calculate the mutation values by computing a linear combination which involves the scaled difference of two of them added to the values of the third one (lines 9 and 13 in Figure 1).

Four parameters are used by DE: "NP" (population size), number of generations , "F" and "CR". The "F" parameter scales the influence of the set of pairs of solutions (one pair in our case) selected to calculate the mutation values (line 13 in Figure 1). The "CR" parameter controls the influence either of the original parent or the mutation values in the generation of the offspring by a binomial recombination (lines 12 to 15 in Figure 1). Several DE models have been proposed [8]. In our study we use the most used model called "DE/rand/1/bin" (the individuals for mutation are selected at random, just one pair of solutions is used to calculate the scaled difference and a binomial (discrete) recombination is chosen).

The motivation of this paper is two-fold: (1) to increase the probability of a parent to generate a fitter offspring and (2) to avoid the use of a penalty function to deal with the constraints of the problem. The first objective is reached by allowing each parent to generate more than one offspring at each generation. In this way, a pre-selection mechanism is incorporated to select, by using a deterministic process, the best solution among the offspring of one parent, and only this best solution will compete against its parent in order to remain in the population. The second objective refers to handling the objective function value and the constraints of the problem separately and to use a mechanism to keep solutions with a good value of the objective function, regardless of their feasibility, in the population. This issue is important because maintaining promising infeasible solutions will increase the exploration of new regions of the search space and the feasible region and it will decrease the chances of getting trapped in local optima.

The paper is organized as follows: In Section 2, we present the previous related work. Section 3 provides the description of our approach. Afterwards, in Section 4, we detail the experimental design and we present and discuss the results obtained. Finally, in Section 5 some conclusions are drawn and the future work is established.

2. RELATED WORK

EAs have been widely used to solve engineering design problems. More precisely, the approaches discussed in this section do not use a penalty function with penalty factors to be calibrated as a constraint-handling mechanism. Storn [11] proposed a constraint-relaxation mechanism coupled with the aging concept to solve optimization problems using DE. He explored the idea of allowing a solution to generate more than one offspring, but in his approach, once a child is better than its parent, the multiple offspring generation ends. Furthermore, the comparison between solutions is always deterministic and the constraint-handling mechanism is based on relaxing the constraint at the beginning in order to consider all the solutions as feasible.

Mezura-Montes & Coello Coello [7] have explored the idea of avoiding the use of penalty functions to solve engineering design problems. Instead, they proposed to use multiobjective optimization concepts [7]. The approach provided good results, however , it required to perform a considerable high number of evaluations of the objective function of the problem.

Among the state-of-the-art approaches available to solve engineering design problems, we present the following: He et al. [4] proposed a PSO-based approach to solve engineering design problems. The main advantage of the approach is its low computational cost measured in terms of the number of objective function evaluations. However, He's approach only works with feasible points; therefore, there is no diversity mechanism at all and an initial population of feasible solutions is always required. This is obviously the main disadvantage of the technique.

Ray & Liew [9] used a social model to solve engineering optimization problems. In their model, the population of solutions is seen as a civilization and it is divided into sub-populations known as societies. There are leaders in each society and the leaders are also grouped in a leaders' society. Then, solutions can follow its corresponding leader, a leader of another society or the leader of leaders. In this way, the approach aims to maintain diversity. Constraints are handled by using dominance as a selection criterion. The main advantage of the approach is that it requires a low number of evaluations of the objective function to obtain competitive results. However, it requires a ranking process at each generation besides a clustering algorithm which is used to initialize the societies.

3. OUR APPROACH

The aim of the approach is to add simple modifications to the DE algorithm to adapt it to deal with constrained search spaces and also to improve its performance by generating more offspring per parent. At each generation, each parent will be able to generate n_o offspring. Among these newly generated solutions, the best of them will be selected to compete against its parent. Then, each parent will increase its chances to generate a fitter offspring. The selection of the best child is completely deterministic based on the three following criteria previously proposed by Deb [2]:

1. Between 2 feasible solutions, the one with the highest fitness value wins.
2. If one solution is feasible and the other one is infeasible, the feasible solution wins.
3. If both solutions are infeasible, the one with the lowest sum of constraint violation is preferred $\sum max(0,g_i(X))\ \forall i, , i=1,\ldots,n$,

After the best offspring is selected, it will compete against its corresponding parent and the best of them will survive for the next generation. However, unlike Deb's approach [2] and traditional DE [8], in this case, the comparison between parent and best offspring is performed with a stochastic element. Based on a parameter called selection ratio S_r the best solution is chosen based only on the value of the objective function value. In the remaining $1-S_r$ selections, the best solution is chosen based on the three criteria mentioned before. In this way, the best feasible solutions will remain in the population, besides those solutions with a promising value of the objective function, regardless of feasibility.

This mechanism, coupled with the DE's way of finding new search directions (based on the distribution of solutions in the current population) will allow the algorithm to explore the search space and its feasible region in a better way as to obtain better solutions. The details of our approach are presented in Figure 1.

4. EXPERIMENTS AND DISCUSSION

Our experimental design has two parts: (1) to compare our approach against a traditional EA with different types of penalty function approaches and (2) to compare our results against state-of-the-art approaches and against a traditional DE approach which generates only one offspring per parent. We selected four well known engineering design problems to use them in the experiments. The details of each problem are presented in an Appendix at the end of this document.

For the first set of experiments, we implemented four typical penalty approaches: (1) A zero fitness value to infeasible solutions (death penalty) [10], (2) static penalty (fixed penalty factor during all the process) [5], (3) dynamic penalty (the penalty factor has a low value at the beginning and a high value at the end of the process) [6]

```
Begin
    G=0
    Create a random initial population $\vec{x}_G^i \quad \forall i, i = 1, \ldots, NP$
    Evaluate $f\left(\vec{x}_G^i\right) \quad \forall i, i = 1, \ldots, NP$
    For G=1 to MAX_GENERATIONS Do
        F=rand[0.3,0.9]
        For i=1 to NP Do
➔           For k=1 to $n_0$ Do
                Select randomly $r_1 \neq r_2 \neq r_3 \neq i$
                $J_{rand}$ = randint(1,D)
                For j=1 to n Do
                    If (rand[0,1) < CR or j = $j_{rand}$) Then
                        $\text{child}_j = x_{j,G}^{r_3} + F\left(x_{j,G}^{r_1} - x_{j,G}^{r_2}\right)$
                    Else
                        $\text{child}_j = x_{j,G}^{i}$
                    End If
                End For
                If k>1 Then
➔                   If (child is better than $\text{child}_j = \vec{u}_{G+1}^i$) (based on the three criteria)) Then
                        $\vec{u}_{G+1}^i = \text{child}$
                    End If
                Else
                    $\vec{u}_{G+1}^i = \text{child}$
                End If

            End For
➔               If flip(Sr) Then
➔                   If ( $f\left(\vec{u}_{G+1}^i\right) \leq f\left(\vec{x}_{G+1}^i\right)$ Then
                    $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
                Else
                    $\vec{x}_{G+1}^i = \vec{x}_G^i$
                End If
            Else
➔               If ( $\vec{u}_{G+1}^i$ is better than $\vec{x}_G^i$ (based on the three criteria)) Then
                    $\vec{x}_{G+1}^i = \vec{u}_{G+1}^i$
                Else
                    $\vec{x}_{G+1}^i = \vec{x}_G^i$
                End If
            End If
        End For
        G=G+1
    End For
End
```

Figure 1. Our algorithm. The steps modified with respect to the original DE algorithm are marked with an arrow. randint(min,max) returns an integer value between min and max. rand[0,1) returns a real number between 0 and 1. Both functions adopt a uniform probability distribution. flip(W) returns 1 with probability W.

Table 1. Comparison of statistical results for the penalty-based approaches and our approach. “-“ means no feasible solutions found. A result in **boldface** means a better result. “*(X)” means that only in X runs (out of 30) feasible solutions were found.

a) Welded beam design

	Death Penalty	**Static**	**Dynamic *(28)**	**Adaptive *(27)**	**Our approach**
Best	1.739736	1.792248	1.781232	1.792266	**1.724852**
Mean	2.104756	2.023434	2.138872	2.164542	**1.724853**
Worst	2.803005	2.739448	2.904370	3.018553	**1.724854**
St. Dev.	3.0E-1	2.2E-1	2.8E-1	3.4E-1	**1.0E-15**

b) Pressure vessel design

	Death Penalty	**Static**	**Dynamic *(22)**	**Adaptive *(21)**	**Our approach**
Best	6172.421387	-	6162.862793	6292.51022	**6059.701660**
Mean	7417.028727	-	7042.828564	7703.780354	**6059.701660**
Worst	10477.677734	-	7798.198242	10830.894278	**6059.701660**
St. Dev.	9.6E+2	-	5.3E+2	1.6E+3	**1.0E-12**

c) Tension/Compression Spring design

	Death Penalty	**Static**	**Dynamic**	**Adaptive**	**Our approach**
Best	0.012719	0.012753	0.012702	0.012692	**0.012665**
Mean	0.014665	0.014636	0.013998	0.014002	**0.012666**
Worst	0.018139	0.018918	0.017044	0.016661	**0.012674**
St. Dev.	1.4E-3	1.6E-3	1.0E-3	1.2E-4	**2.0E-6**

d) Speed reducer design

	Death Penalty	**Static**	**Dynamic**	**Adaptive**	**Our approach**
Best	-	-	-	-	**2996.356689**
Mean	-	-	-	-	**2996.367220**
Worst	-	-	-	-	**2996.390137**
St. Dev.	-	-	-	-	**8.2E-3**

Table 2: Comparison of results with respect to two state-of-the-art approaches and a traditional DE approach. A result in boldface means a better result. “NA” means not available.

Problem	**Stats**	**Ray & Liew [9]**	**He et al. [4]**	**Our approach**	**Traditional DE**
Welded beam	Best	2.385435	2.380957	**1.724852**	1.904312
	Mean	3.255137	2.381932	**1.724853**	2.237370
	St. Dev.	9.6E-1	5.2E-3	**1.0E-15**	2.3E-1
	Evaluations	33000	30000	**24000**	24000
Pressure vessel	Best	6171.00	6059.7143	**6059.701660**	7247.938477
	Mean	6335.05	6289.92881	**6059.701660**	8854.318896
	St. Dev.	NA	3.1E+2	**1.0E-12**	1.3E+3
	Evaluations	**20000**	30000	24000	24000
Ten./Comp. Spring	Best	0.012669	**0.012665**	**0.012665**	0.012851
	Mean	0.012923	0.012702	**0.012666**	0.014119
	St. Dev.	5.9E-4	4.1E-5	**2.0E-6**	1.0E-3
	Evaluations	25167	**15000**	24000	24000
Speed reducer	Best	**2994.744241**	NA	2996.356689	3064.211426
	Mean	3001.758264	NA	**2996.367220**	3244.569010
	St. Dev.	4.0E+0	NA	**8.2E-3**	2.0E+2
	Evaluations	54456	NA	**24000**	24000

and finally an adaptive penalty (the penalty factor is updated based on the behavior of the population) [3].

30 independent runs per technique per problem were performed. The number of evaluations of the objective function was fixed to 24,000 for the four penalty-based approaches and also for our approach. For the penalty-based approaches we used a typical EA: a gray-coded genetic algorithm with roulette wheel selection, one point crossover, uniform mutation and elitism incorporated. The population size was 100 individuals and the number of generations 240.

The rate of crossover was 0.6 and the mutation rate was 0.01. The parameters for the static, dynamic and adaptive approaches and also for the genetic algorithm were defined after a trial-and-error process.

The reported parameters were those which provided the best results and they are the following: Static approach: fixed penalty factor = 1000. Dynamic approach: $\alpha=2$, $\beta=2$, C=0.5. Adaptive approach: $\beta_1=2.0$, $\beta_2=4.0$, k=50, $\delta_{initial}=5000$. Our DE-based approach used the following parameters: NP = 60, MAX_GENERATIONS=80 (24,000 evaluations of the objective function), CR=0.9, $n_o=5$ and $S_r=0.45$, F randomly generated within [0.3,0.9].

Discrete variables were handled by just truncating the real value to its closest integer value. The statistical results of the 30 independent runs are shown in Table 1. All these parameters were defined after previous experiments in order to obtain the best performance.

Based on the results obtained, our approach was able to provide the most robust ("better" mean and standard deviation values) and the best quality results ("better" best solution found) in all four engineering design problems adopted. In fact, none of the penalty-based approaches was able to find feasible solutions for the speed reducer design problem. Besides, the dynamic penalty approach could not find feasible solutions for the pressure vessel problem and it could not find feasible solutions in two and eight runs for the welded beam and the pressure vessel problems respectively.

Furthermore, the adaptive approach could not find feasible solutions for the welded beam and for the pressure vessel problems in three and nine runs, respectively. On the other hand, our approach consistently found feasible solutions in each run for all design problems.

The overall results suggest that our approach provided the most consistent performance, while the penalty-based approaches were competitive in some problems, but in others the results were poor. This behavior indeed reflects the need to update the penalty factors according to the problem to be solved. This negative effect is not present in our approach, because with the same set of parameters, avoiding the use of a penalty function and with the diversity mechanism, the approach finds good feasible solutions consistently. In fact, regardless of the penalty values adopted, calibrating the parameters for the genetic algorithm was more difficult than defining the values for the DE approach, which is a very interesting advantage of our approach.

Table 3. Details of the best solution found

Problem 1. Welded beam

	Ray & Liew[9]	**He et al. [4]**	**Our approach**
x_1	0.244438	0.244369	0.205730
x_2	6.237967	6.217520	3.470489
x_3	8.288576	8.291471	9.036624
x_4	0.244566	0.244369	0.205730
$g_1(x)$	-5760.110471	-5741.176933	-0.000335
$g_2(x)$	-3.245428	0.000001	-0.000753
$g_3(x)$	-0.000128	0.000000	-0.000000
$g_4(x)$	-3.020055	-3.022955	-3.432984
$g_5(x)$	-0.119438	-0.119369	-0.080730
$g_6(x)$	-0.234237	-0.234241	-0.235540
$g_7(x)$	-13.079305	-0.000309	-0.000882
f(x)	**2.38119**	**2.380956**	**1.724852**

For the second set of experiments we compared the performance of our approach against those provided by the last two approaches discussed in Section 2. We used He's and Ray's approaches because they solved the same set of test problems. In this case, the number of evaluations required by each approach is included and it is used as a comparison criterion. We also added the results obtained with a traditional DE approach with the same set of initial parameters used by our approach and with the same constraint handling mechanism. The only difference between traditional DE and our approach is that in the first, only one offspring per parent is generated and in our approach we generate 5 offspring per parent. The parameter values used in our approach are exactly the same used in the previous experiment. The comparison of the statistical results and the number of evaluations required by each approach are presented in Table 2. In Tables 3, 4, 5 and 6 we provide the details of the best solution found by each state-of-the-art technique and our approach for the four test problems respectively.

The results show that our approach clearly outperforms the two compared approaches in the welded beam design problem ("better" best, mean and standard deviation values, requiring a lower number of evaluations to provide such a performance). For the pressure vessel problem, Ray's approach required the lowest number of evaluations. However, our approach provided clearly "better" best, mean and standard deviation values than those provided by Ray's technique by just using 4000 more evaluations. For the spring design problem, He's approach provided the same best solution found by our approach by using just 15,000 evaluations of the objective function. Nonetheless, it is important to note that He's approach requires to generate an initial feasible population. In contrast, our approach starts only with random solutions, regardless of feasibility. Furthermore, our approach provided "better" mean and standard deviation values, which imply more robustness of our approach.

Table 4. Details of the best solution found

Problem 2. Pressure vessel

	Ray & Liew[9]	**He et al. [4]**	**Our approach**
x_1	0.8125	0.8125	0.8125
x_2	0.4375	0.4375	0.4375
x_3	41.9768	42.098446	42.098446
x_4	182.2845	176.636052	176.636047
$g_1(x)$	-0.0023	-0.000000	0.000000
$g_2(x)$	-0.0370	-0.035881	-0.035881
$g_3(x)$	-23420.5966	-0.000000	-0.000002
$g_4(x)$	-57.7155	-63.363948	-63.363949
f(x)	**6171.0**	**6059.7143**	**6059.701660**

For the speed reducer problem, Ray's approach provided the "best" best result, but it required more than twice the number of evaluations used by our approach. Besides, our best solution is close to the value provided by Ray's approach and also our mean and standard deviation values are clearly better, showing again, the robustness of the approach. Finally, our approach provided better results in all cases in all criteria with respect to the traditional DE approach despite the fact that both share the same diversity mechanism.

Table 5. Details of the best solution found

Problem 3. Tension/Compression spring

	Ray & Liew[9]	**He et al. [4]**	**Our approach**
x_1	0.0521602	0.051690	0.051688
x_2	0.368159	0.356750	0.356692
x_3	10.648442	11.287126	11.290483
$g_1(x)$	-0.000000	-0.000000	-0.000000
$g_2(x)$	-0.000000	0.000000	-0.000000
$g_3(x)$	-4.075805	-4.053827	-0.727747
$g_4(x)$	-0.719787	-0.727706	-4.053734
f(x)	**0.012669**	**0.012665**	**0.012665**

The details of the best solutions found shown in Tables 3, 4, 5 and 6 seem to emphasize the ability of the approach, based on the intensive use of the DE operator in one parent, to explore solutions close to already good solutions and to improve the quality of the final result. This behavior was found mostly in the beam and the pressure vessel design problems (Tables 3 and 4).

The overall results of this second experiment suggest that our approach was able to provide very competitive results compared with those provided by two state-of-the-art approaches based on the quality, robustness and computational cost measured by the number of evaluations of the objective function. Furthermore, the chance to generate more offspring per parent provided an improvement in the quality and robustness of the results obtained.

The conclusions for both experiments provide empirical evidence about the importance of allowing each solution to reproduce more than once in the same generation and the convenience of using a mechanism to deal with constraints by avoiding the use of a penalty function and favoring diversity.

Table 6. Details of the best solution found

Problem 4. Speed reducer

	Ray & Liew [9]	**Our approach**
x_1	3.500000	3.500010
x_2	0.700000	0.700000
x_3	17	17
x_4	7.327602	7.300156
x_5	7.715321	7.800027
x_6	3.350267	3.350221
x_7	5.286655	5.286685
$g_1(x)$	NA	-0.073918
$g_2(x)$	NA	-0.198001
$g_3(x)$	NA	-0.499144
$g_4(x)$	NA	-0.901471
$g_5(x)$	NA	-0.000005
$g_6(x)$	NA	-0.000001
$g_7(x)$	NA	-0.702500
$g_8(x)$	NA	-0.000003
$g_9(x)$	NA	-0.583332
$g_{10}(x)$	NA	-0.051345
$g_{11}(x)$	NA	-0.010856
f(x)	**2994.744241**	**2996.356689**

9. CONCLUSIONS AND FUTURE WORK

We have presented a DE-based approach to solve engineering design problems. The approach was based on the more frequent use of the DE operator per parent. The constraint handling technique was based on simple selection criteria and a diversity mechanism to maintain promising solutions regardless on feasibility. The results obtained in two different experiments show that the proposed approach outperforms four different penalty-based techniques when tested in four engineering design problems. Also, in the second experiment, the approach obtained better results than a traditional DE approach and it provided very competitive results against two state-of-the-art approaches which neither use a penalty function as a constraint-handling mechanism.

As future paths of research we will explore a mechanism to adapt the parameter that controls the number of offspring per parent and we will test our approach in other real-world problems with a high dimensionality.

ACKNOWLEDGMENTS

The second author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project number 45683.

REFERENCES

1. Bäck T., 1996, Evolutionary Algorithms in Theory and Practice. Oxford University Press, New York, 1996.
2. Deb, K., 2000, An efficient constraint handling technique method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 186(2/4):311—338.
3. Hadj-Alouane A.B. and Bean. J.C., 1997, A Genetic Algorithm for the Multiple-Choice Integer Program. Operations Research 45:92—101.
4. He S., Prempain E. and Wu O.H., 2004, An Improved Particle Swarm Optimizer for Mechanical Design Optimization Problems. Engineering Optimization, 36(5):585--605, October.
5. Hoffmeister, F. and Sprave, J., 1996, Problem-independent handling of constraints by use of metric penalty functions. In L.J. Fogel, P.J. Angeline, and T. Bäck, editors, Proceedings of the Fifth Annual Conference on Evolutionary Programming (EP'96), pages 289--294, San Diego, California, February. The MIT Press.
6. Joines, J. and Houck, C., 1994, On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GAs. In D. Fogel, editor, Proceedings of the first IEEE Conference on Evolutionary Computation, pages 579--584, Orlando, Florida. IEEE Press.
7. Coello Coello C.A. and Mezura-Montes E., 2002, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Advanced Engineering Informatics, 16(3):193—203, July.
8. Price, K.V., 1999, .An Introduction to Differential Evolution. In D. Corne, M. Dorigo, and F. Glover, editors, New Ideas in Optimization, pages 79--108. Mc Graw-Hill, UK.
9. Ray T. and Liew K., 2003, Society and Civilization: An Optimization Algorithm Based on the Simulation of Social Behavior. IEEE Transactions on Evolutionary Computation}, 7(4):386--396, August.
10. Schwefel H.P., 1981, Numerical Optimization of Computer Models. John Wiley & Sons, Great Britain.
11. Storn R., 1999, System Design by Constraint Adaptation and Differential Evolution. IEEE Transactions on Evolutionary Computation}, 3(1):22--34, April.

APPENDIX

Full description of the four problems used in the experiments:
Problem 1: (Design of a Welded Beam)
A welded beam is designed for minimum cost subject to constraints on shear stress (τ), bending stress in the beam (σ), buckling load on the bar (P_c), end deflection of the beam (δ), and side constraints. There are four design variables: *h* (x_1), *l* (x_2), *t* (x_3) and *b* (x_4). See Figure 2.

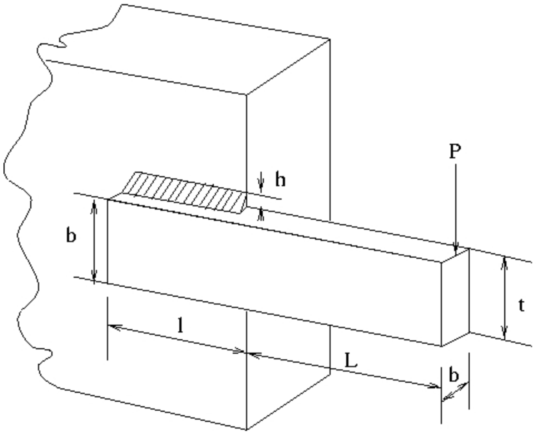

Figure 2: Welded beam

The problem can be stated as follows:
Minimize:

$$f(\vec{x}) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2)$$

Subject to:

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\max} \leq 0$$

$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\max} \leq 0$$

$$g_3(\vec{x}) = x_1 - x_4 \leq 0$$

$$g_4(\vec{x}) = 0.10471 x_1^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$$

$$g_5(\vec{x}) = 0.125 - x_1 \leq 0$$

$$g_6(\vec{x}) = \delta(\vec{x}) - \delta_{\max} \leq 0$$

$$g_7(\vec{x}) = P - P_c(\vec{x}) \leq 0$$

where:

$$\tau(\vec{x}) = \sqrt{(\tau')^2 + 2\tau'\tau'' \frac{x_2}{2R} + (\tau'')^2}\,, \qquad \tau' = \frac{P}{\sqrt{2} x_1 x_2}\,,$$

$$\tau'' = \frac{MR}{J}\,,\ M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}\,,$$

$$J = 2\left\{\sqrt{2} x_1 x_2 \left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}, \qquad \sigma(\vec{x}) = \frac{6PL}{x_4 x_3^2}\,,$$

$$\delta(\vec{x}) = \frac{4PL^3}{Ex_3^3 x_4},$$

$$P_c(\vec{x}) = \frac{4.013E\sqrt{\dfrac{x_3^2 x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$

$P=6000$ lb, $L=14$ in, $E=30\text{x}10^6$ psi, $G=12\text{x}10^6$ psi, $\tau_{max}=13{,}600$ psi, $\sigma_{max}=30{,}000$ psi, $\delta_{max}=0.25$ in, where $0.1 \le x_1 \le 2.0$, $0.1 \le x_2 \le 10.0$, $0.1 \le x_3 \le 10.0$ and $0.1 \le x_4 \le 2.0$.

Problem 2: (Design of a Pressure Vessel)

A cylindrical vessel is capped at both ends by hemispherical heads. The objective is to minimize the total cost, including the cost of the material, forming and welding. There are four design variables: T_s (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel, not including the head). T_s and T_h are integer multiples of 0.0625 inch, which are the available thicknesses of rolled steel plates, and R and L are continuous. See Figure 3.

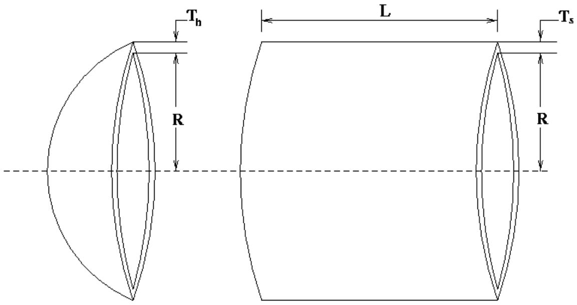

Figure 3: Pressure Vessel

The problem can be stated as follows:

Minimize:

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to:

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0$$

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

$$g_4(\vec{x}) = x_4 - 240 \le 0$$

where $1 \le x_1 \le 99$, $1 \le x_2 \le 99$, $10 \le x_3 \le 200$ and $10 \le x_4 \le 200$.

Problem 3: (Minimization of the Weight of a Tension/Compression String)

This 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 (x_2), the wire diameter d (x_1) and the number of active coils N (x_3). See Figure 4.

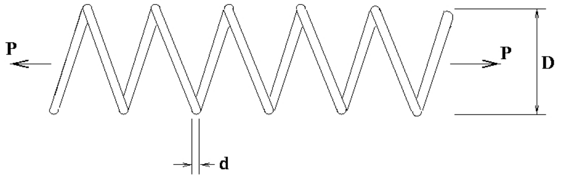

Figure 4: Tension/Compression Spring

Formally, the problem can be expressed as:

Minimize:

$$f(\vec{x}) = (N+2)Dd^2$$

Subject to:

$$g_1(\vec{x}) = 1 - \frac{D^3N}{71785d^4} \le 0$$

$$g_2(\vec{x}) = \frac{4D^2 - dD}{12566(Dd^3 - d^4)} + \frac{1}{5108d^2} - 1 \le 0$$

$$g_3(\vec{x}) = 1 - \frac{140.45d}{d^2N} \le 0$$

$$g_4(\vec{x}) = \frac{D+d}{1.5} - 1 \le 0$$

where $0.05 \le d \le 2$, $0.25 \le D \le 1.3$ and $2 \le N \le 15$.

Problem 4: (Minimization of the Weight of a Speed Reducer)

The weight of the speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surfaces stress, transverse deflections of the shafts and stresses in the shafts. The variables $x_1, x_2, \ldots, x_7$ are the face width, module of teeth, number of teeth in the pinion, length of the first shaft between bearings, length of the second shaft between bearings and the diameter of the first and second shafts. The third variable is integer, the rest of them are continuous.

Minimize:

$$f(\vec{x}) = 0.7854 x_1 x_2^2 \left(3.3333 x_3^2 + 14.9334 x_3 - 43.0934\right)$$

$$-1.508 x_1 \left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right)$$
$$+ 0.7854\left(x_4 x_6^2 + x_5 x_7^2\right)$$

subject to:

$$g_1(\vec{x}) = \frac{27}{x_1 x_2^2 x_3} - 1 \le 0$$

$$g_2(\vec{x}) = \frac{397.5}{x_1 x_2^2 x_3^2} - 1 \le 0$$

$$g_3(\vec{x}) = \frac{1.93 x_4^3}{x_2 x_3^2 x_6^4} - 1 \le 0$$

$$g_4(\vec{x}) = \frac{1.93 x_5^3}{x_2 x_3 x_7^4} - 1 \le 0$$

$$g_5(\vec{x}) = \frac{\left(\left(\frac{745 x_4}{x_2 x_3}\right)^2 + 16.9 x 10^6\right)^{1/2}}{110.0 x_6^3} - 1 \le 0$$

$$g_6(\vec{x}) = \frac{\left(\left(\frac{745 x_5}{x_2 x_3}\right)^2 + 157.5 x 10^6\right)^{1/2}}{85.0 x_7^3} - 1 \le 0$$

$$g_7(\vec{x}) = \frac{x_2 x_3}{40} - 1 \le 0$$

$$g_8(\vec{x}) = \frac{5 x_2}{x_1} - 1 \le 0$$

$$g_9(\vec{x}) = \frac{x_1}{12 x_2} - 1 \le 0$$

$$g_{10}(\vec{x}) = \frac{1.5 x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(\vec{x}) = \frac{1.1 x_7 + 1.9}{x_5} - 1 \le 0$$

where $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$ and $5.0 \le x_7 \le 5.5$.