

A Hybrid Genetic Algorithm for Constrained Optimization Problems

Da-lian Liu

Department of Basic Course Teaching, Beijing Union University, Beijing, China
Email: ldlluck@sina.com

Xiao-hua Chen and Jin-ling Du

Tourism Institute, Beijing Union University, Beijing, China
School of Management Engineering, Shan Dong Jianzhu University, Jinan, China
Email: {cxhldz@sina.com, sdaidjl@sdjzu.edu.cn}

Abstract—Genetic algorithm (GA) is a powerful method to solve constrained optimization problems (COPs). In this paper, a new fitness function based hybrid genetic optimization algorithm (NFFHGA) for COPs is proposed, in which a new crossover operator based on Union Design is presented, and inspired by the smooth function technique, a new fitness function is designed to automatically search for potential solutions. Furthermore, in order to make the fitness function work well, a special technique which keeps a certain number of feasible solutions is also used. Experiments on 6 benchmark problems are performed and the compared results with the best known solutions reported in literature show that NFFHGA can not only quickly converge to the optimal or near-optimal solutions, but also have a high performance.

Index Terms—Constrained optimization, genetic algorithm, fitness function, Uniform Design

I. INTRODUCTION

The constrained optimization problem (COP) is a kind of mathematical programming problem, and has become an important branch of operations research. It has been applied to wide variety of areas such as the military, economics, engineering, management science, and etc [1].

Without loss of generality, the general COP with n variables and m constraints can be written in the following standard form:

$$\begin{aligned} & \text{Min } f(x) \\ & \text{s.t. } g_i(x) \leq 0 \quad i = 1, 2, \dots, p ; \\ & \quad h_j(x) = 0 \quad j = p + 1, \dots, m ; \\ & \quad x \in S \subseteq R^n \end{aligned} \quad (1)$$

Where $x = (x_1, x_2, \dots, x_n)$ is an one-dimensional vector of n decision variables, $f(x)$ is an objective function, $g_i(x) \leq 0 \quad i = 1, 2, \dots, p$ are p inequality constraints and $h_j(x) = 0 \quad j = p + 1, \dots, m$ are $m - p$ equality constraints.

Set $F = \left\{ x : x \in S; g_i(x) \leq 0, h_j(x) = 0, \begin{array}{l} i = 1, 2, \dots, p, \\ j = p + 1, \dots, m \end{array} \right\}$ is called the

feasible region of problem (1) and x is called the feasible solution contained in set F .

Over the past decades, genetic algorithms (GAs) have been successfully used to solve COPs, and researchers have proposed a large number of constrained optimization genetic algorithms (e.g. [2] ~ [10]). Other techniques (e.g. [11] ~ [13]) are also adopted to solve COPs efficiently too. Most researchers pay more attention to how to handle constraints and much previous work has been done on constraint-handling techniques, which can be mainly classified into 3 categories. The first one is the method based on penalty function, the second one is the method based on multi-objective optimization techniques and the third one is the method based on biasing feasible over infeasible solutions[14]. For example, K. Deb proposed a method which combines the advantages of penalty function and the preference of the feasible solutions in [4]. Venkatraman and Yen presented a method based on a genetic, two-phase framework for solving COPs in [3]. In the first phase of the algorithm, the objective function is completely disregarded and the COP is treated as a constraint satisfaction problem. In the second phase, the simultaneous optimization of the objective function and the satisfaction of the constraints are treated as a bi-objective optimization problem.

Constraint-handling techniques are very important for COPs, at the same time, as a method based on genetic algorithms, the design of the genetic operators is even more important. With the rapid development of the artificial intelligence, there are almost no differences between the genetic algorithms (GAs) and the evolutionary algorithms (EAs), but the fitness function still plays a very important role in both two algorithms. EAs (or GAs) lacks a mechanism able to bias efficiently the search towards the feasible region in constrained search spaces, which has triggered a considerable amount of research and a wide variety of approaches have been suggested in the last few years to incorporate constraints into the fitness function of an evolutionary algorithm[2].

Coello [15] used a population-based approach to handle constraints in single-objective optimization problem. The fitness of this approach is the following:

if $g_j(x) \leq 0.0$ then $\text{fitness} = g_j(x)$
else if $v \neq 0$ then $\text{fitness} = -v$

else $fitness = f(x)$

where $g_j(x)$ refers to the j th constraint of the problem, v is the number of violated constraints ($v \leq m$) and $f(x)$ is the value of the objective function of the individual.

Since the fitness function depends on every constraint, the main drawback of this approach is that the number of subpopulations required increases linearly with the number of constraints of the problem.

Another fitness function was presented in [4] by Deb. Neglecting the details of the approach, it can be summed up into 3 points: (i) the feasible solution is better than the infeasible one; (ii) between two feasible solutions, the one with the better objective function value is preferred; (iii) between two infeasible solutions, the one with the smaller degree of constraint is preferred.

In fact, the most common idea to design fitness function for dealing with COPs is based on the 3 points mentioned above, and the main drawback of the idea is that it neglects the potential ability of the infeasible solutions. Considering that the global optimal solutions locate on or near the boundary of the feasible region for many constrained optimization problems, the infeasible solutions may be more close to the global optimal. The algorithm should be very efficient if the advantages of the infeasible solutions can be well used in it.

In this paper, a new hybrid genetic algorithm based on a new fitness function (NFFHGA) for solving constrained optimization problem is proposed. First, the constrained optimization problem is transformed into a two-objective preference optimization problem based on penalty function method (e.g. [16]). One objective function is the objective function of the original constrained optimization problem and the other is derived from the constraints by penalty function. Then, inspired by the smoothing technique (e.g., [17]), a new fitness function is designed and a strategy keeping a certain number of feasible solutions is proposed too. By combining the fitness function and the strategy, a new selection operator named NFFM is proposed. Moreover, a new crossover operator and mutation operator are designed. To enhance the efficiency of the crossover operator, one-dimensional search scheme is incorporated into the mutation operator. Numerical simulation demonstrates the validity and efficiency of the proposed algorithm.

This paper is organized as follows. Section II presents the basic concepts of the GAs. In section III, the detail of algorithm NFFHGA is presented. Section IV discusses the experimental results obtained.

II. GENETIC ALGORITHMS OVERVIEW

Genetic algorithms are a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. They were invented to mimic some of the processes observed in natural evolution. The father of the original Genetic Algorithm was John Holland who invented it in the early 1970's.

GA simulates the survival of the fittest among individuals over consecutive generation for solving a problem. Each generation consists of a population of

character strings that are analogous to the chromosome that we see in our DNA. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evolution.

Generally, the GA begins by defining the optimization variables and ends by testing for convergence. A path through the components of the GA is shown as a flowchart in Figure 1 which is described by Asef who lives in Islamic Republic of Iran.

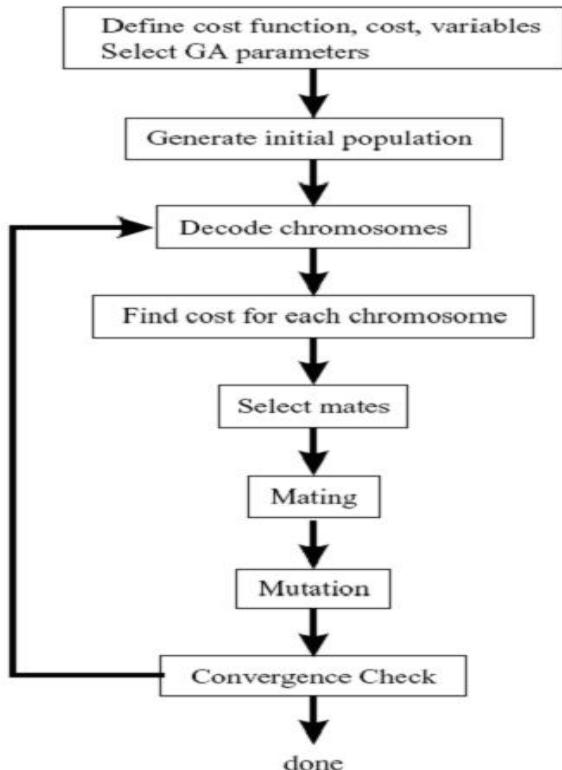


Figure 1. A path through the components of the GA

The definition of the optimization variables is mainly according to the problem itself, and an initial population is usually randomly generated. The following 3 genetic operators are the main part of the whole algorithm.

1. **Selection**, which equates to survival of the fittest.

The key idea of the selection operator is giving preference to better individuals and allowing them to pass on their genes to the next generation. The goodness of each individual depends on its fitness and the fitness function's design may be determined by an objective function or by a subjective judgment. Using selection will tend to fill the population with copies of the best individual from the population.

2. **Crossover**, which represents mating between individuals.

The crossover operator is the prime distinguished factor of GA from other optimization techniques. Its implementation details are as follow:

- Two individuals are chosen from the population according to the parameter of p_c which can be adjusted during the performance of the program;

- A crossover site along the bit strings is randomly chosen;
- The values of the two strings are exchanged up to this point;
- The two new offspring created from this mating are put into the next generation of the population.

By recombining portions of good individuals, this process is likely to create even better individuals. With the intensive study of this operator, more and more methods are presented which are not only confined to the idea mentioned above.

3. Mutation, which introduces random modifications.

With some low probability, a portion of the new individuals will have some of their bits flipped. Its purpose is to maintain diversity within the population and inhibit premature convergence.

III. NFFHGA ALGORITHM

A. Transformation

Converting a constrained optimization problem like (1) into an unconstrained optimization problem is a kind of simple and practical constraint handling methods, and penalty function method is a widely used and efficient method to deal with constraints. The penalty function is usually defined as the distance function between x and the feasible region. First, we define functions $p_j(x)$ as follows:

$$p_j(x) = \begin{cases} \max\{0, g_j(x)\} & 1 \leq j \leq p \\ |h_j(x)| & p+1 \leq j \leq m \end{cases}$$

where $p_j(x)$ is the violation degree of the j th constraint of problem (1) at point x .

Define function

$$p(x) = \sum_{j=1}^m p_j(x),$$

where $p(x)$ is the violation degree of all the constraints of problem (1) at point x and also the distance between x and the feasible region.

Define two functions

$$f_1(x) = f(x) \text{ and } f_2(x) = p(x).$$

Note that a point x minimizing $p(x)$ is equivalent to x being a feasible solution. Therefore, problem (1) can be transformed into the following two objective optimization problems

$$\min(f_1(x), f_2(x)) \quad (2)$$

The optimal solutions of problem (2) are the optimal solutions of problem (1) and for detail please refer to [16].

B. New Crossover Operator

Select a feasible solution x_1 randomly from the population and select another individual x_2 stochastically too. The individuals x_1 and x_2 are to do

crossover based on Uniform Design [18]. The details are as follows:

Find the center point of segment of two parent points, and construct a hyper sphere with that center point as center of the sphere, and half the length of the segment as radius.

Generate q uniformly distributed points on that hyper sphere by the uniform design method in [18] ;

Choose two best individuals from q points as the offspring of two parent points through the following method:

① If there are at least two feasible solutions in q points, then choose two individuals with the better objective function value from these feasible solutions;

② If there is only one feasible solution in q points, then keep it as one offspring point and choose the one with the smallest value of $f_2(x)$ from other $(q-1)$ points as another offspring point.

③ If there is no feasible solution in q points, then choose two individuals with the smaller values of $f_2(x)$ from these q infeasible solutions;

In addition, if there is no feasible solution in the population at the beginning of GA, then choose two points randomly from the population as parent points.

C. Improved Mutation Operator

Suppose that x is a solution undergoing mutation, the mutation is designed based on two cases:

- Case1: if x is a feasible solution, calculate the approximate gradient Δd of $f_1(x)$ at x as follows:

$$\Delta d = [\Delta d_1, \Delta d_2, \dots, \Delta d_n]^T,$$

where $\Delta d_i = \frac{f_2(x + \delta e_i) - f_2(x)}{\delta}$, e_i is a n -dimensional unit vector with its i -th element is one and others are zeros, $\delta > 0$ is a positive number small enough and $i = 1 \sim n$.

Generally, $d = -\Delta d$ is a descent direction used in the line search (for detail please refers to [17]). Determine whether the new individual after the line search is a feasible solution until an infeasible solution is gotten. Then the last feasible solution generated before the infeasible solution is the offspring of x and is denoted as \tilde{x} .

- Case2: if x is an infeasible solution, then

$$\tilde{x} = x + \frac{t}{T} \Delta x, \text{ where}$$

$\Delta x \sim N(0, \sigma^2) = (N(0, \sigma_1^2) \dots N(0, \sigma_n^2))$, i.e. Δx is an n -dimensional random variable obeying n -dimensional Gaussian distribution with mean $0 = (0, 0, \dots, 0)^T$ and variance $\sigma^2 = (\sigma_1^2, \dots, \sigma_n^2)^T$. t is the iteration number currently and T is the maximum iteration number

needed in the algorithm. Coefficient $\frac{t}{T}$ is added in the mutation operator and this can improve the ability braking away from the local optimum by enhancing the mutation capacity with the increasing of the iteration times.

D. Selection Operator NFFM

As described in the former part, selection operator plays an important role in the genetic algorithm. So many scholars pay more attention to it and many kinds of different selection operators are designed. The individual ranking based on Pareto intensity values is used in literature [19] and the selection operator in literature [20] is based on preference. But most of them either need to adjust sensitive coefficients or have complex operation. In order to overcome these shortcomings, and take full advantage of the infeasible solutions near the boundary, a selection operator named NFFM (New fitness function method) is proposed as follows.

New fitness function

According to formula (2), we define the following fitness function

$$F(x) = f_2(x) - [1 - \text{sign}(f_2(x))] [f_2(x) - f_1(x)] \quad (3)$$

From above formula, we can see that when $f_2(x) > 0$, x must be an infeasible solution and $\text{sign}(f_2(x)) = 1$, then $F(x) = f_2(x)$, i.e., the fitness function is $F(x) = f_2(x)$;

When $f_2(x) = 0$, x must be a feasible solution and $\text{sign}(f_2(x)) = 0$, then $F(x) = f_1(x)$, i.e. the fitness function is $F(x) = f_1(x)$.

The advantages of this fitness function are Obvious. At first, it can prefer the feasible solutions which make the objective function value smaller, and secondly, it can prefer the infeasible solutions with the smaller degrees of constraint violation automatically. But the drawback of this fitness function is also obvious: it doesn't certainly think the feasible solutions are better than the infeasible solutions, which means it may give up some feasible solutions whose $F(x)$ values are smaller than infeasible solutions and retained some infeasible solution.

Note that the optimal solutions of many constrained optimization problems lie on or near the boundary of the feasible region. It is necessary to keep some infeasible solutions scatter around the boundary of the feasible region which can enhance the possibility to get the optimal solution of problem (1). For example, in literature [21], a fixed proportion of infeasible solutions is kept. But the ratio of infeasible solution must be controlled correctly. In order to enhance the property of the new fitness function, we present a strategy keeping a certain number of feasible solutions.

E. Strategy Keeping A Certain Number of Feasible Solutions

In the k -th population, the number of feasible solutions Num can be defined according to following formula.

$$Num = \begin{cases} n & \text{if } \frac{n}{N} \leq c \\ cN + n' & \text{else} \end{cases} \quad (4)$$

where N is the population size, c ($c \geq 0.5$) is a parameter that can be adjusted according to the particular case. n is the number of feasible solution included in current population and n' is the number of feasible solutions selected by the fitness function $F(x)$ from other $(1-c) \times N$ individuals of the population.

This strategy ensures that all of the feasible solutions are better than infeasible solutions during early stage so that more feasible solutions are accumulated. During the later period, more than a half number of the population are occupied by feasible solution which can not only help keeping the optimal solution as soon as it is found but also guarantee the convergence of the algorithm. The more important character of this strategy is it keeps some infeasible solutions in the population when feasible solutions can not be get enough, which makes the algorithm has more opportunity to obtain the global optimal.

F. Hibrid Genetic Optimization Algorithm based on NFFM ((NFFHGA))

- Step1 Given Population size N . Generate initial population $P(0)$ based on a chaotic sequence, given crossover and mutation probability p_c and p_m . Set $k = 0$.
- Step2 Select parents from $P(k)$ for crossover with probability p_c and for each parent, generate offspring by crossover operator. The set of all offspring generated is denoted as $C(k)$.
- Step3 Select parents form $C(k)$ for mutation with probability p_m . Each selected parent generates an offspring by mutation operator. The set of all offspring generated is denoted as $M(k)$.
- Step4 Select the next generation population $P(k+1)$ among $P(k) \cup C(k) \cup M(k)$ by NFFM, Let $k = k + 1$.
- Step5 If stop criterion is satisfied, then stop. Otherwise, go to step 2.

IV EXPERIMENTAL RESULTS AND DISCUSSION

The first five benchmark problems denoted by $g01, g02, g04, g08$ and $g11$ are chosen from [22] and the last one is chosen from [2], and they are listed in the appendix. The proposed algorithm (denoted as NFFHGA for short) is executed for these problems under Matlab environment.

In the simulation, we take the following parameters: $N = 200$, $p_c = 0.5$, $p_m = 0.3$ and for each of the test problem, 30 independent runs were performed. We record the following data: Best optimal value (Best), mean best objective value (Mean) and the worst objective value (Worst) obtained in 30 runs and NA means the value is not available. The results are summarized in the following Tables.

TABLE I.

THE COMPARISON OF THE RESULTS OBTAINED BY NFFHGA AND SAFF, RY, AIRCES, SMES IN [17] WITH THE FIRST 5 TEST FUNCTIONS

TF	Methods	Best	Mean	Worst
g01	NFFHGA	-15.000	-15.000	-15.000
	SAFF	-15.000	-15.000	-15.000
	RY	-15.000	-15.000	-15.000
	AIRCES	-15.000	-15.000	-15.000
	SMES	-15.000	-15.000	-15.000
g02	NFFHGA	0.80351	0.76615	0.718358
	SAFF	0.80297	0.79010	0.76043
	RY	0.803515	0.781975	0.726288
	AIRCES	0.803575	0.779465	0.716312
	SMES	0.803601	0.785238	0.751322
g04	NFFHGA	-30665.539	-30665.539	-30665.539
	SAFF	-30665.539	-30665.539	-30665.539
	RY	-30665.539	-30665.539	-30665.539
	AIRCES	-30665.539	-30665.539	-30665.539
	SMES	-30665.539	-30665.539	-30665.539
g08	NFFHGA	-0.095825	-0.095825	-0.095825
	SAFF	-0.095825	-0.095825	-0.095825
	RY	-0.095825	-0.095825	-0.095825
	AIRCES	-0.095825	-0.095825	-0.095825
	SMES	-0.095825	-0.095825	-0.095825
g11	NFFHGA	0.750	0.750	0.750
	SAFF	0.750	0.750	0.750
	RY	0.750	0.750	0.750
	AIRCES	0.750	0.750	0.750
	SMES	0.750	0.750	0.750

From Table 1 it can be seen that for the test problems $g01$, $g04$ and $g08$, which mainly examine the comprehensive ability of an algorithm, the performance of NFFHGA is equal to that of any other methods listed above. For $g11$, which test the ability dealing with the constraints of an algorithm, the performance of NFFM is good enough too but its number of generations is only 23. The experimental results show that NFFHGA is not very efficient with problem $g02$ which is a high dimension problem where SMES ([23]) gets the best results and SAFF ([24]) is the worst of the four other methods in best optimal value. But the results of NFFHGA are not too far from the optima which indicate that NFFHGA is also effective.

TABLE II.

THE COMPARISON OF THE RESULTS OBTAINED BY NFFHGA AND THE METHODS PROPOSED IN [2] WITH THE LAST TEST FUNCTION.

Opt.	Methods	Best	Mean	Worst	St.D.
6059 .94	NFFHGA	5991.35	6117.96	6307.38	21.351
	COMOGA	6369.42	7795.41	9147.52	701.36
	HCVEGA	6064.72	6259.96	6820.94	170.25
	HCNPGA	6059.92	6172.52	6845.77	123.89

The statistical results of Table 2 indicate that NFFHGA presented the lowest standard deviation (St.D.) and the best median and average solutions in the same problem T6. It means that for T6, NFFHGA is extremely effective and efficient.

APPENDIX

TEST FUNCTION SUITE

T1: (g_{01}) : Minimize

$$f(x) = 5 \sum_{i=1}^4 x_i - 5 \sum_{i=1}^4 x_i^2 - \sum_{i=5}^{13} x_i$$

Subject to

$$g_1 = 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leq 0$$

$$g_2 = 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leq 0$$

$$g_3 = 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leq 0$$

$$g_4 = -8x_1 + x_{10} \leq 0$$

$$g_5 = -8x_2 + x_{11} \leq 0$$

$$g_6 = -8x_3 + x_{12} \leq 0$$

$$g_7 = -2x_4 - x_5 + x_{10} \leq 0$$

$$g_8 = -2x_6 - x_7 + x_{11} \leq 0$$

$$g_9 = -2x_8 - x_9 + x_{12} \leq 0$$

Where

$$0 \leq x_i \leq 1 \quad (i = 1, 2, \dots, 9), \quad 0 \leq x_i \leq 100 \quad (i = 10, 11, 12), \\ \text{and } 0 \leq x_{13} \leq 1.$$

The optimum solution is $x^* = (1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 1)$

where $f(x^*) = -15$

T2: (g_{02}) Maximize

$$f(x) = \frac{\sum_{i=1}^n \cos^4(x_i) - 2 \prod_{i=1}^n \cos^2(x_i)}{\sqrt{\sum_{i=1}^n i x_i^2}}$$

Subject to

$$g_1(x) = 0.75 - \prod_{i=1}^n x_i \leq 0$$

$$g_2(x) = \sum_{i=1}^n x_i - 7.5n \leq 0$$

Where $n = 20$ and $0 \leq x_i \leq 10 \quad (i = 1, \dots, 20)$. The global maximum is unknown; the best we found is $f(x^*) = 0.803619$

T3: (g_{04}) Minimize

$$f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 \\ + 37.293239x_1 - 40792.141$$

Subject to

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 +$$

$$0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$$

$$\begin{aligned}
g_2(x) &= -85.334407 - 0.0056858x_2x_5 - \\
&\quad 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0 \\
g_3(x) &= 80.51249 + 0.0071317x_2x_5 + \\
&\quad 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0 \\
g_4(x) &= -80.51249 - 0.0071317x_2x_5 - \\
&\quad 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0 \\
g_5(x) &= 9.300961 + 0.0047026x_3x_5 + \\
&\quad 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0 \\
g_6(x) &= -9.300961 - 0.0047026x_3x_5 - \\
&\quad 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0
\end{aligned}$$

Where $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$ and

$27 \leq x_i \leq 45$ ($i = 3, 4, 5$). The optimum solution is

$$x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$$

where $f(x^*) = -30665.539$

T4: (g_{08}) Maximize

$$f(x) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$$

Subject to

$$g_1 = x_1^2 - x_2 + 1 \leq 0$$

$$g_2 = 1 - x_1 + (x_2 - 4)^2 \leq 0$$

where $0 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 10$. The optimum is located at $x^* = (1.2279713, 4.2453733)$ where

$$f(x^*) = 0.095825$$

T5: (g_{11}) : Minimize

$$f(x) = x_1^2 + (x_2 - 1)^2$$

$$\text{Subject to } h(x) = x_2 - x_1^2 = 0$$

Where $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$. The optimum solution is $x^* = (\pm 1/\sqrt{2}, 1/2)$ where $f(x^*) = 0.75$

T6: (Design of a Pressure Vessel)

The description of the problem : A cylindrical vessel is capped at both ends by hemispherical heads as shown in Figure2. 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).

The problem can be stated as follows:

Minimize

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1161x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0,$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0,$$

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

$$g_4(x) = x_4 - 240 \leq 0,$$

Where $0.125 \leq x_1 \leq 10, 0.1 \leq x_2, x_3, x_4 \leq 10$.

The global maximum is unknown; the best we found is $f(x^*) = 6059.946341$

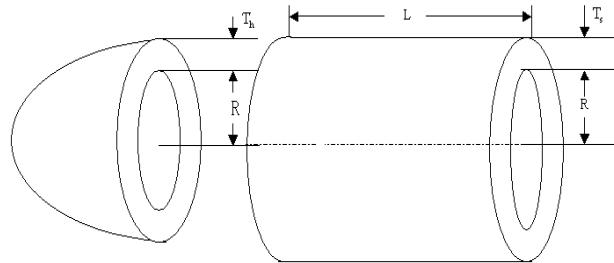


Figure2. Center and end section of the pressure vessel used for problem T6.

ACKNOWLEDGMENT

The authors would like to thanks the reviewers for their valuable comments and suggestions which greatly improve the paper.

REFERENCES

- [1] Erwie Zahara and CHIA-Hsin Hu, "Solving constrained optimization problems with hybrid particle swarm optimization," *Engineering Optimization*, Vol. 40, No. 11, pp.1031-1049, November 2008.
- [2] Efren Mezura-montes and Carlos A. Coello Coello, "A numerical comparison of some multiobjective-based techniques to handle constrained in genetic algorithms," *Technical Report, ENOCINV-01-2003*, México: Evolutionary Computation Group (EVOCINV), Computer Science Section, Electrical Engineering Department, CINVESTAVIPN, 2003.
- [3] Sangameswar Venkatraman and Gary G. Yen, "A genetic framework for constrained Optimization Using Genetic algorithms," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 4, pp. 424-435, August 2005.
- [4] Kalyanmoy Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, pp.311-338, 2000.
- [5] Michalewicz, Z., Janikow, C.Z.: FENOCOP:, "A genetic algorithm for numerical optimization problems with linear constraints," *Communications of the ACM*, Vol. 39, No. 175, December 1996.
- [6] T P Runarsson and X Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans on Evolutionary Computation*, Vol.4, No.3, pp.284-294, September 2000.
- [7] Mezura-Montes E and Coello Coello CA. "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Trans on Evolutionary Computation*, Vol.9, No.1, pp. 1-17, 2005.
- [8] Z Michalewicz and M Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary Computation*, pp.4:1-32, 1996.
- [9] Wang Yong, Cai Zi-xing, Zhou Yu-ren and Xiao Chi-xin, "Constrained optimization evolutionary algorithm," *Journal of Software*, Vol. 20 No. 1, pp.11-29, January 2009.
- [10] Mohamed Ali Wagdy and Sabry Hegazy Zaher, "Constrained optimization based on modified differential evolution algorithm," *Information Science*, Vol.194, pp.171-208, July 2012.
- [11] Byrd Richard H., Curtis Frank E. and Nocedal Jorge, "An inexact Newton method for nonconvex equality

- constrained optimization," *Mathematical Programming*, Vol.122, Issue 2, pp.273-299, April 2010.
- [12] Wu Jui -yu, "Solving constrained global optimization via artificial immune system," *International Journal on Artificial Intelligence Tools*, Vol.20, Issue 1, pp.1-27, February 2011.
- [13] Boussaïd Ilhem, Chatterjee Amitava and Ahmed-Nacer Mohamed, "Biogeography-based optimization for constrained optimization problem," *Computers & Operations Research*, Vol.39, Issue 12, pp.3293-3304, December 2012.
- [14] Yong Wang, Hui Liu, Zi-xing Cai and Yu-ren Zhou, "An orthogonal based constrained evolutionary optimization algorithm," *Engineering Optimization*, Vol. 39, No. 6, pp.715-736, September 2007.
- [15] Carlos A and Coello Coello. "Treating Constraints as objectives for single-objective evolutionary optimization," *Engineering Optimization*, 32(3),pp: 319-346, 2000.
- [16] Yiping Wang, Dalian Liu and Yiu-Ming Cheung, "Preference Bi-objective Evolutionary Algorithm for Constrained Optimization," *Computational Intelligence and Security*, part1, pp.184-191, 2005.
- [17] Wang Yu-ping, Liu Da-lian, "A global optimization evolutionary algorithm and its convergence based on a smooth scheme and line search," *Chinese Journal of Computers*, Vol. 29 No. 4, pp.670-675 , April 2006.
- [18] Kaitai Fang and Yuan Wang, The applications of number methods in uniform design, Science Publisher of China, 1996.
- [19] Zhou Yu-ren, Li Yuan-xiang, Wang Yong and Kang Li-shan, "A pareto strength evolutionary algorithm for constrained optimization," *Journal of Software*, Vol.14, No.7, pp.1243-1249, 2003.
- [20] Dragan Cvetkovic and Ian C. parmee, " Preferences and Their Application in Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, Vol.6, No.1, pp.42-56, 2002.
- [21] Lin Dan, Li Min-qiang, Kou Ji-song, "A GA based method for solving constrained optimization problem," *Journal of Software*, Vol.12, No.4, pp. 628-632, , April 2001.
- [22] Cai Zi-xing, Jiang Zhong-yang, Wang Yong and Luo Yi-dan, "A novel constrained optimization evolutionary algorithm based on orthogonal experimental design," *Chinese Journal of Computer*, Vol.33, No.5, pp. 855-864, May 2010.
- [23] Efren Mezura- Montes, Coello C C A, "A simple multimembered evolution strategy to solve constrained optimization problems," *IEEE Transaction on Evolutionary Computation*, Vol.9, No.1, pp.001-017, 2005.
- [24] Farmani R, Wright J A, "Self-adaptive fitness formulation for constrained optimization," *IEEE Transactions on Evolutionary Computation*, Vol.7, No. 5, pp.445-455, 2003.

Da-lian Liu was born in Hebei province, P.R. China, in 1978. She received the B.S. degree in HeBei Normal University, China, in 2001, and M.S. degree in Xidian University, China, in 2004. She was awarded as a lecturer of department of basic course teaching of Beijing Union University in 2004. Her research interests focus on optimization theory and method, intelligent computation and data mining, etc.