ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 8, No. 1, 2013, *pp*. 041-054

An Alternative Approach for PDE-Constrained Optimization via Genetic Algorithm

Laxminarayan Sahoo[1], Asoke Kumar Bhunia[2], Debkumar Pal[2] and Birendra Nath Mandal[3]

[1] Department of Mathematics, Raniganj Girls' College, Raniganj-713358, India

2 Department of Mathematics, The University of Burdwan, Burdwan-713104, India

3 NASI Senior Scientist, Physics & Applied Mathematics Unit, Indian Statistical Institute, 203, B.T.Road, Kolkata - 700 108, India

(*Received September 10, 2012, accepted November 22, 2012*)

Abstract. This paper deals with an alternative approach for solving the PDE-constrained optimization problems. For this purpose, the problem has been discretized with the help of finite difference method. Then the reduced problem has been solved by advanced real coded genetic algorithm with ranking selection, whole-arithmetic crossover and non uniform mutation. The proposed approach has been illustrated with a numerical example. Finally, to test the performance of the algorithm, sensitivity analyses have been performed on objective function values with respect to different parameters of genetic algorithm.

Keywords: *PDE-Constrained optimization, optimal control, genetic algorithm.*

1. Introduction

PDE-Constrained optimization refers to the optimization of systems governed by partial differential equations. This means that the constraints of the optimization problems are partial differential equations. Most of the physical problems are modeled mathematically by a system of partial differential equations. Due to the complexity of the partial differential equations, analytical solutions to these equations do not exist in general. Hence to solve these partial differential equations, there is only one way to find the approximate numerical solutions.

Finding the solutions of constrained optimization problems is a challenging as well as demanding task in the competitive market situations due to globalization of market economy. Parallel research has been going on in PDE-simulation and numerical optimization. Currently there is a growing tendency of cooperation and collaboration among these two communities in order to achieve the best possible results in practical applications.

To the best of our knowledge, very few researchers have solved this type of problems in the last few years. In this connection, one may refer to the recent works of Hazra [1, 2], Hazra and Schulz [3], Hazra et al. [4], Biros and Ghattas [5], Emilio et al. [6], Griesse and Vexler [7], Rees, Stoll and Wathen [8], Rees, Dollar and Wathen [9] along with others.

Hazra [1] proposed simultaneous pseudo-timestepping as an efficient method for aerodynamic shape optimization. In this method, instead of solving the necessary conditions by iterative techniques, pseudo-time embedded nonstationary system is integrated in time until a steady state is reached. Hazra and Schulz [3] developed a method for the optimization problems with PDE constraints. This method can be viewed as a reduced SQP method in the sense that it uses a preconditioner derived from that method. The reduced Hessian in the preconditioner is approximated by a pseudo-differential operator. Hazra et al. [4] proposed a new method based on simultaneous pseudo-timestepping for solving aerodynamic shape optimization problem. The preconditioned pseudo-stationary state, costate and design equations are integrated simultaneously in time until a steady state is reached. The preconditioner used in this study is motivated by a continuous re-interpretation of reduced SQP methods. Biros and Ghattas [5] proposed a method for the steady-state PDE-constrained optimization, based on the idea of using a full space Newton solver combined with an approximate reduced space quasi-Newton SQP preconditioner. The basic components of this method are Newton solution of the first-order optimality conditions that characterize stationarity of the Lagrange function. Emilio et al. [6] solved the shape optimization problem in ship hydrodynamics using computational fluid dynamics. Griesse and Vexler [7] considered the efficient computation of derivatives of a functional which depends on the solution of a PDE-constrained optimization problem with inequality constraints. They derived the conditions under with the quantity of interest possesses first and second order derivatives with respect to the perturbation parameters. They developed an algorithm for the efficient evaluation of these derivatives with considerable savings over a direct approach, especially in case of high dimensional parameter spaces. To test the efficiency of the algorithm, numerical experiments involving a parameter identification problem for Navier-Stokes flow and an optimal control problem for a reaction-diffusion system have been presented. Hazra [2] proposed a method, called multigrid one-shot method, for solving state constrained aerodynamic shape optimization problems. Here multigrid strategy has been used to reduce the total number of iterations. Rees, Stoll and Wathen [8] illustrated how all-at-once methods could be employed to solve the problems from PDE-constrained optimization problems. In particular, they showed that both the problems with and without constraints lead to linear systems in saddle point form and also presented efficient preconditioning strategies for both problems. Rees, Dollar and Wathen [9] considered simple PDE-constrained optimization problem, viz. distributed control problems in which the constraint is either 2 or 3 - dimensional Poission PDE. Using discretization, the given constraints are converted to linear system with large dimension. To solve the systems, they introduced two optimal preconditioners for those systems which lead to convergence of symmetric Krylov subspace iterative methods in a number of iterations which does not increase with the dimension of the discrete problem. These preconditioners are block structured and involve standard multigrid cycles. The optimality of the preconditioned iterative solver is proved theoretically and verified computationally in several test cases.

The earlier mentioned methods are gradient based methods. However, these methods have some limitations. Among these limitations, one is that the traditional non-liner optimization methods very often stuck to the local optimum. To overcome these limitations, generally, soft computing methods like Genetic Algorithm, Simulated Annealing and Tabu search, are used for solving decision-making problems. Among these methods genetic algorithm is very popular. It is a well-known computerized stochastic search method based on the evolutionary principle “survival of the fittest” of Charles Darwin and natural genetics. It is executed iteratively on the set of real / binary coded solution called population. In each iteration, three basic genetic operations i.e., selection, crossover and mutation are performed. The concept of this algorithm was conceived by Prof. John Holland [10], University of Michigan, Ann Arbar in the year 1975. Thereafter, he and his students contributed much of the development of the subject. Goldberg [11] first popularized this subject by writing a text book. After that, several text books (Michalawicz [12], Gen and Chang [13], Sakawa [14], Eiben and Smith [15]) have been published in this area.

In this paper, an alternative approach has been proposed to solve the constrained optimization problem subject to the Poisson partial differential equations with Dirichlet boundary condition. In this approach, firstly the given problem has been discretized by finite difference method. Then the reduced problem has been solved by real coded advanced genetic algorithm. Next to illustrate the proposed approach, a numerical example has been solved. Finally, the performance of the proposed approach is tested by the sensitivity analysis on the best and mean objective function values with respect to the parameters of genetic algorithm.

2. The Problem

In this paper, we consider the optimization problem as follows:

$$\text{Minimize } J\left(y,u\right)=\frac{1}{2}\|y-\overline{y}\|_{L_2(\Omega)}^2+\frac{\beta}{2}\|u\|_{L_2(\Omega)}^2\ ,\ \beta>0 \tag{1}$$

for given $\bar{y}$ which represents the desired state and for some domain $\Omega \in R^d$ (d = 2,3) with boundary Γ. The state variable y and control variable u are related by the state equation

$$-\nabla^2 y = u \quad \text{in } \Omega$$
$$\text{with} \quad y = g \quad \text{on } \Gamma, \tag{2}$$

where g is the given function defined on the boundary Γ. In the above problem, the state equation (2) is simply a Poission equation with Dirichlet boundary condition. It is to be noted that the state variable y can be eliminated from (1) with the help of state equation (2). As a result, the function $J(y, u)$ in (1) is reduced to $J(y(u), u) = F(u)$ and the optimization problem (1) subject to (2) reduces to an unconstrained optimization problem as follows:

$$\text{Minimize } F(u) \tag{3}$$

If the function F (u) is minimized subject to so-called bound constraints

$$\underline{u}(x) \le u(x) \le \bar{u}(x) \text{ a.e in } \Omega, \tag{4}$$

the corresponding problem is called bound constrained optimization problem.

Let us define

$$u_{ad} = \{u \in L_2(\Omega) : \underline{u}(x) \le u(x) \le \bar{u}(x) \quad a.e. \quad in \quad \Omega\}$$

In this case, the PDE-constrained optimization problem can be written as

$$\underset{y,u}{\text{Minimize}} \ \frac{1}{2}\|y - \bar{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$$

$$\text{subject to} \quad -\nabla^2 y = u \quad in \ \Omega \tag{5}$$
$$y = g \quad on \ \Gamma$$
$$\text{and } \underline{u}(x) \le u(x) \le \bar{u}(x) \text{ a.e in } \Omega$$

For detail discussion of this type of problem, one may refer to the works of Hinze et al. [16], Rees et al. [8] and Rees et al. [9].

There are two approaches for solving the optimization problem (5). The first approach is optimize-then-discretize and second one is discretize-then-optimize. In this paper, we have used the second approach. For this purpose, we have applied the finite difference method to discretize the problem (5).

For discritization, we have considered two dimensional optimization problem. This means that the region $\Omega \in R^2$. Let $\Omega \equiv [a_1, b_1] \times [a_2, b_2]$, a_1, b_1, a_2, b_2 being constants. The region Ω is divided into $n_1 n_2$ equal sub-regions by a set of grid points (x_{1i}, x_{2j})

where $x_{1i} = a_1 + ih_1, \quad i = 0,1,2,...,n_1$

$x_{2j} = a_2 + jh_2, \ j = 0,1,2,...,n_2$

and $h_k = \dfrac{(b_k - a_k)}{n_k}, \ k = 1,2$

The grid spacing in x_1 and x_2 directions are denoted by h_1 and h_2 respectively.

Denoting $y(x_{1i}, x_{2j})$ by $y_{i,j}$, the finite difference approximations based on central difference for the partial derivative of $y(x_1, x_2)$ to both the space derivatives are given by

$$\left(\frac{\partial^2 y}{\partial x_1^2}\right)_{i,j} = \frac{y_{i+1,j} - 2y_{i,j} + y_{i-1,j}}{h_1^2}$$

$$\left(\frac{\partial^2 y}{\partial x_2^2}\right)_{i,j} = \frac{y_{i,j+1} - 2y_{i,j} + y_{i,j-1}}{h_2^{\,2}}, \qquad i,j = 1,2,\cdots,n-1$$

Hence at each grid point $\left(x_{1i}, x_{2j}\right)$, the equation $-\nabla^2 y = u$ becomes

$$\frac{y_{i+1,j} - 2y_{i,j} + y_{i-1,j}}{h_1^{\,2}} + \frac{y_{i,j+1} - 2y_{i,j} + y_{i,j-1}}{h_2^{\,2}} = -u_{i,j}, \quad i,j = 1,2,\cdots,n-1$$

which implies

$$2\left(h_1^2 + h_2^2\right) y_{i,j} - h_2^2\left(y_{i+1,j} + y_{i-1,j}\right) - h_1^2\left(y_{i,j+1} + y_{i,j-1}\right) = h_1^2 h_2^2 u_{i,j}, \quad i,j = 1,2,\cdots,n-1$$

Then the problem (5) reduces to

$$\text{Minimize } z = \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\left(y_{i,j} - \bar{y}_{i,j}\right)^2 + \frac{\beta}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} u_{i,j}^{\;2} \tag{6}$$

subject to $2\left(h_1^2 + h_2^2\right) y_{i,j} - h_2^2\left(y_{i+1,j} + y_{i-1,j}\right) - h_1^2\left(y_{i,j+1} + y_{i,j-1}\right) = h_1^2 h_2^2 u_{i,j}, \quad i,j = 1,2,\cdots,n-1$

The boundary conditions $y = g$ on Γ are reduced to

$$y_{i,0} = g_{i,0},\ y_{i,n} = g_{i,n}, \quad i = 0,1,2,\cdots,n \tag{7}$$

$$y_{0,j} = g_{0,j},\ y_{n,j} = g_{n,j}, \quad j = 0,1,2,\cdots,n$$

where $g_{i,j} = g\left(x_{1i}, x_{2j}\right)$

The bound constraints $\underline{u}(x) \le u(x) \le \bar{u}(x)$ a.e. in Ω can be written as

$$\underline{u}_{i,j} \le u_{i,j} \le \bar{u}_{i,j}, \quad i,j = 1,2,\cdots,n-1 \tag{8}$$

where $u_{i,j} = u\left(x_{1i}, x_{2j}\right)$

Special case: When $h_1 = h_2 = h$ (say).

In this case, (6)-(8) reduce to

$$\text{Minimize } z = \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\left(y_{i,j} - \bar{y}_{i,j}\right)^2 + \frac{\beta}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1} u_{i,j}^2 \tag{9}$$

subject to $4y_{i,j} - \left(y_{i+1,j} + y_{i-1,j}\right) - \left(y_{i,j+1} + y_{i,j-1}\right) = h^2 u_{i,j}, \quad i,j = 1,\ 2,\cdots,\ n-1$

The boundary conditions $y = g$ on Γ are reduced to

$$y_{i,0} = g_{i,0},\ y_{i,n} = g_{i,n}, \quad i = 0,1,2,\cdots,n \tag{10}$$

$$y_{0,j} = g_{0,j},\ y_{n,j} = g_{n,j}, \quad j = 0,1,2,\cdots,n$$

where $g_{i,j} = g\left(x_{1i}, x_{2j}\right)$

The bound constraints $\underline{u}(x) \le u(x) \le \bar{u}(x)$ a.e. in Ω can be written as

$$\underline{u}_{i,j} \le u_{i,j} \le \bar{u}_{i,j}, \quad i,j = 1,2,\ldots,n-1 \tag{11}$$

where $u_{i,j} = u\left(x_{1i}, x_{2j}\right)$

Now, if we consider three dimensional optimization problems, then after discretization by finite difference method, the reduced problem is given by

Minimize $z = \frac{1}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sum_{k=1}^{n-1}\left(y_{i,j,k} - \overline{y}_{i,j,k}\right)^2 + \frac{\beta}{2}\sum_{i=1}^{n-1}\sum_{j=1}^{n-1}\sum_{k=1}^{n-1}u_{i,j.k}^2$ (12)

subject to

$$6y_{i,j,k} - \left(y_{i+1,j,k} + y_{i-1,j,k}\right) - \left(y_{i,j+1,k} + y_{i,j-1,k}\right) - \left(y_{i,j,k+1} + y_{i,j,k-1}\right) = h^2 u_{i,j,k},$$

$$i, j, k = 1, 2, \cdots, n-1$$

The boundary conditions $y = g$ on Γ are reduced to

$$y_{i,0,k} = g_{i,0,k}, \quad y_{i,n,k} = g_{i,n,k}, \qquad i, k = 1, 2, \cdots, n-1 \tag{13}$$

$$y_{0,j,k} = g_{0,j,k}, \quad y_{n,j,k} = g_{n,j,k}, \qquad j, k = 1, 2, \cdots, n-1$$

$$y_{i,j,0} = g_{i,j,0}, \quad y_{i,j,n} = g_{i,j,n}, \qquad i, j = 1, 2, \cdots, n-1$$

The bound constraints $\underline{u}(x) \leq u(x) \leq \overline{u}(x)$ a.e. in Ω can be written as

$$\underline{u}_{i,j,k} \leq u_{i,j,k} \leq \overline{u}_{i,j,k}, \qquad i, j, k = 1, 2, ..., n-1 \tag{14}$$

where $u_{i,j,k} = u\left(x_{1i}, x_{2j}, x_{3k}\right)$.

Now we have to solve the reduced optimization problems (9) and (12). For this purpose, we have developed real coded advanced genetic algorithm:

3. Genetic Algorithm

Genetic algorithm is a stochastic search algorithm that simulates the process of natural evolution and natural genetics. The evolution process begins by taking randomly generated possible solutions of the problem. Each solution in the population is evaluated according to some fitness measure. Filter solutions in the population are selected and allowed to produce offspring by crossover and mutation operators. This process is repeated till the termination criterion is satisfied.

The different steps of this algorithm are described as follows:

Algorithm

Step-1. Initialize the parameters of genetic algorithm, bounds of variables and different parameters of the problem.

Step-2. $t = 0$ [t represents the number of current generation].

Step-3. Initialize $P(t)$ [$P(t)$ represents the population at t-th generation].

Step-4. Evaluate the fitness function for each chromosome of $P(t)$

Step-5. Find the best result from $P(t)$.

Step-6. t is increased by unity.

Step-7. If the termination criterion is satisfied go to Step-14, otherwise, go to next step.

Step-8. Select $P(t)$ from $P(t-1)$ by ranking selection process.

Step-9. Alter $P(t)$ by crossover, mutation operations.

Step-10. Evaluate the fitness function for each chromosome of $P(t)$.

Step-11. Find the best result from $P(t)$.

Step-12. Compare the best results of $P(t)$ and $P(t-1)$ and store better one.

Step-13. Go to Step-6:

Step-14. Print the result.

Step-15. End.

For implementing the above GA in solving the problems the following basic components are to be considered.

- GA Parameters
- Chromosome representation and initialization of population
- Evaluation of fitness function
- Selection process
- Genetic operators (crossover and mutation)
- Termination criterion

3.1. GA Parameters

There are different parameters used in the genetic algorithm, viz. population size (*p_size*), maximum number of generations (*m_gen*), crossover rate/probability of crossover (*p_cross*) and mutation rate/probability of mutation (*p_mute*). There is no hard and fast rule for choosing the population size for GA. However, if the population size is considered to be large, storing of the data in the intermediate steps of GA may create some difficulties at the time of computation with the help of computer. On the other hand, for very small population size, some genetic operations cannot be implemented. Particularly, mutation operator does not work properly as the mutation rate is very low. Regarding the maximum number of generations, there is no indication for considering this value. Generally, it is problem dependent. Particularly, it depends upon the number of genes (variables) of a chromosome in artificial genetics. Again, from the natural genetics it is obvious that the crossover rate is always greater than that of mutation rate. Usually, the crossover rate varies from 0.8 to 0.95 where as the mutation rate, as 0.05 to 0.2. Sometimes, it is considered as $1/n$ where n be the number of genes (or variables) of the individuals.

3.2. Chromosome representation and initialization

In the applications of GA, the appropriate chromosome (individual) representation of solution for the given problems is an important task to the users. There are different types of representations, viz. binary, real, octal, hexadecimal coding, available in the existing literature. Among these representations, real coding representation is very popular as this type of chromosome representation looks like a vector. In this representation, each component (gene) of the chromosome is the values of decision variables.

After the selection of chromosome representation, the next step is to initialize the chromosomes that will take part in the artificial genetic operations like natural genetics. This process produces population size number of chromosomes in which every component for each chromosome is randomly generated within the bounds of the corresponding decision variable. There are different procedures for generating a random number for each component of the chromosomes. In this work, we have used uniform distribution for this purpose.

3.3. Evaluation of fitness function

After getting a population of potential solutions, we need to check how good they are. So we have to calculate the fitness value for each chromosome. In evaluation, the value of objective function corresponding to the chromosome is taken as the fitness value of that chromosome.

3.4. Selection

The selection operator plays an important role in GA. Usually, it is the first operator applied to the population. The primary objective of this operator is to emphasize on the above average solutions and eliminate below average solutions from the population for the next generation under the well-known evolutionary principle “survival of the fittest”. In this work, we have used ranking selection scheme as selection operator. This selection operator indicates that only the ranking order of the fitness of the individuals within the current population determines the probability of selection. In this selection, the population is actually sorted from the best to worst fashion. The selection probability of each individual is determined according to the ranking rather than their fitness value. There are several methods for assigning the selection probability for each individual on the basis of ranking. Here we have considered the nonlinear ranking method. In this method, the selection probability p_i of the individual with rank *i* is obtained as

$$p_i = p_c(1-p_c)^{i-1}$$

where p_c be the selection probability of best individual.

3.5. Crossover

After selection process, other genetic operators like, crossover and mutation are applied to the resulting chromosomes (those which have survived). Crossover operator plays an important role to empower the GA. It operates on two or more parent chromosome solutions at a time and generates offspring by combining the features of all the parent chromosomes. In this operation, expected $[p\_cross * p\_size]$ ($*$ denotes the product and [ ] denotes the integral value) number of chromosomes will take part. In our work, we have used whole arithmetical crossover. The different steps of this operator are as follows:

Step-1. Find the integral value of $p\_cross * p\_size$ and store it in N.

Step-2. Select the chromosomes v_k and v_i randomly from population.

Step-3. The components v'_{kj} and v'_{ij} $(j = 1, 2, \cdots, n)$ of two offspring will be created by

$$v'_{kj} = \lambda v_{kj} + (1-\lambda) v_{ij}$$
$$v'_{ij} = (1-\lambda) v_{kj} + \lambda v_{ij}$$

where λ is a random number between 0 and 1.

Step-4. Repeat Step-2 and Step-3 for $\frac{N}{2}$ times.

3.6. Mutation

The aim of mutation operation is to introduce the random variations into the population. Sometimes, it helps to regain the information lost in earlier generations. Mainly, this operator is responsible for fine tuning of the system. This operator is applied to a single chromosome only. Usually, its rate is very low; otherwise it would defeat the order building being generated through selection and crossover operations. Mutation attempts to bump the population gently into a slightly better course i.e., mutation changes single or all the genes of a randomly selected chromosome slightly. In this work, we have used nonuniform mutation. If the gene v_{ik} of a chromosome v_i is selected for mutation and the domain of v_{ik} is $[l_{ik}, u_{ik}]$, then the reduced value of v_{ik} is given by

$$v'_{ik} = \begin{cases} v_{ik} + \Delta(t, u_{ik} - v_{ik}), & \text{if random digit is } 0. \\ v_{ik} - \Delta(t, v_{ik} - l_{ik}), & \text{if random digit is } 1. \end{cases}$$

where $k \in \{1, 2, ..., n\}$ and $\Delta(y)$ returns a value in the range $[0.y]$.

In our work, we have taken

$\Delta(t, y) = yr(1 - t/T)^b$ where *r* is a random number from $[0,1]$, *T* is the maximum generation number and *b* is the system parameter determining the degree of non-uniformity.

3.7. Termination criteria

The termination condition of GA is to stop the algorithm when either of three conditions is satisfied:

(i) the best individual does not improve over specified generations.

(ii) the total improvement of the last certain number of best solution is less than a pre-assigned small positive number.

(iii) the number of generations reaches maximum number of generation (*m_gen*).

4. Numerical Illustration

To illustrate the proposed GA for solving PDE-constrained optimization problem, we have considered the following example [Rees et al. (2010)].

$$\min_{y,u} \frac{1}{2}\|y-\bar{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$$

$$\text{s.t.} \quad -\nabla^2 y = u \quad \text{in } \Omega$$

$$y = \bar{y} \quad \text{on } \Gamma$$

where $\Omega = [0,1]^m$, $m = 2,3$

$$\bar{y} = \begin{cases} -x_1 \exp\left(-\left(\left(x_1-\frac{1}{2}\right)^2 + \left(x_2-\frac{1}{2}\right)^2\right)\right) & \text{if } (x_1,x_2)\in[0,1]^2 \\ -x_1 \exp\left(-\left(\left(x_1-\frac{1}{2}\right)^2 + \left(x_2-\frac{1}{2}\right)^2 + \left(x_3-\frac{1}{2}\right)^2\right)\right) & \text{if } (x_1,x_2,x_3)\in[0,1]^3. \end{cases}$$

The bounds $\underline{u}_a$ and $\bar{u}_b$ are defined as follows

$$\underline{u}_a = \begin{cases} -0.35 & \text{if } x_1 < 0.5 \\ -0.4 & \text{otherwise} \end{cases}$$

and

$$\bar{u}_b = \begin{cases} -0.1\ \exp\left(-\left(x_1^2 + x_2^2\right)\right) & \text{if } (x_1,x_2)\in[0,1]^2 \\ -0.1\ \exp\left(-\left(x_1^2 + x_2^2 + x_3^3\right)\right) & \text{if } (x_1,x_2,x_3)\in[0,1]^3. \end{cases}$$

Here we have taken the value of β as $\beta = 10^{-2}$.

Considering equal number of subintervals in each direction of Ω, we have discretized the problem by finite difference method and obtained two reduced problems, one for m=2 and another for m=3. Then, we have solved both the problems for different values of the number of subintervals with the help of GA with ranking selection process, whole arithmetic crossover and non uniform mutation. This GA has been coded in C programming language. The computational work has been done on a PC with Intel Core-2-duo 2.5 GHz Processor in LINUX environment.

In this computation, the following values of GA parameters have been used:

p_cross = 0. 8, *p_mute* = 0.1 and b = 2

For each computation, 20 independent runs have been performed by GA and the following measurements have been obtained for m = 2 and 3 separately.

(i) The value of the objective function

(ii) CPU time.

The results have been displayed in Table-1 and Table-2.

Table-1: Result of two dimensional problem

No. of division	P_size	M_gen	Best found objective function value	Mean objective function value	S.D. of objective	Mean time
5	10	500	0.05122	0.05122	2.13575×10^{-17}	0.0925
5	15	200	0.05122	0.05122	2.13575×10^{-17}	0.0515
5	20	200	0.05122	0.05122	2.13575×10^{-17}	0.077
5	20	100	0.05123	0.05124	7.88069×10^{-6}	0.0375
5	20	500	0.05122	0.05122	2.13575×10^{-17}	0.1825
5	15	300	0.05122	0.05122	2.13575×10^{-17}	0.082
10	10	500	0.21855	0.21871	8.6601×10^{-5}	1.3155
10	20	500	0.21836	0.21840	1.93309×10^{-5}	2.65
10	20	200	0.21952	0.21978	1.59456×10^{-4}	0.63288
10	15	200	0.21966	0.22048	8.78374×10^{-4}	0.7925
10	15	300	0.21891	0.21912	1.16566×10^{-4}	1.18444
20	10	500	0.94505	0.94787	1.94313×10^{-3}	15.147
20	20	500	0.93777	0.93945	8.00664×10^{-4}	30.248
20	20	200	0.95256	0.95502	1.68052×10^{-3}	12.244
20	15	300	0.93043	0.95001	4.80573×10^{-3}	13.7705

In Table 1 and 2, best found objective function values, mean and standard deviation of objective function values along with mean computational time have been shown for different number of divisions, population size (*p_size*) and maximum number of generations (*M_gen*). From the computational results of standard deviation values of objective of each case, it is evident that the simulation results are stable.

5. Sensitivity Analysis

To investigate the overall performance of the proposed GA for solving PDE-constrained optimization problem, sensitivity analyses have been carried out graphically for the best found and mean objective function values with respect to different GA parameters separately taking other parameters at their original values. For this purpose, we have considered the two dimensional optimization problem mentioned in the earlier section 'Numerical Illustration'. The corresponding results have been shown in Fig. 1-4. From Fig. 1, it is observed that the best found objective function value be the same for all the values of population size (*p_size*) greater than or equal to 15 whereas the mean objective function value be the same for the population size 35, 40, 45, 50. On the otherhand, in Fig. 2, it is seen that the best found objective function values be the same for all values of maximum number of generations greater than or equal to 475 whereas the mean objective function value be the same for the maximum number of
generation greater than or equal to 550. This means that the proposed GA is stable with respect to the population size as well as the maximum number of generations. In Fig. 3 – 4, the objective function values (best found as well as mean) have been compared with respect to the probability of crossover (*p_cross*) within the range from 0.60 to 0.95 and the probability of mutation (*p_mute*) within the range from 0.04 to 0.16 respectively. From these figures, it is clear that the proposed GA is also stable with respect to the probability of crossover as well as the probability of mutation.

Table-2: Result of three dimensional problem

No.of division	P_size	M_gen	Best found objective function value	Mean objective function value	S.D of objective	Mean time
5	10	500	1.29531	1.29599	3.16502×10^{-4}	0.604
5	20	500	1.29442	1.29462	1.24355×10^{-4}	1.239
5	20	200	1.29886	1.30022	6.74286×10^{-4}	0.495
5	15	200	1.29965	1.30173	8.90383×10^{-4}	0.365
5	15	300	1.2966	1.29782	6.8286×10^{-4}	0.546
10	10	500	14.47154	14.48215	4.65141×10^{-3}	16.169
10	20	500	14.44199	14.45144	4.33701×10^{-3}	32.4875
10	20	200	14.49715	14.50671	5.25776×10^{-3}	13.0425
10	15	200	14.5028	14.51442	6.29426×10^{-3}	9.7375
10	15	300	14.48634	14.49519	6.16286×10^{-3}	14.616
20	20	500	145.96196	146.05327	5.99274×10^{-2}	551.99
20	10	500	146.00694	146.08767	5.89379×10^{-2}	282.16
20	20	200	146.03197	146.10606	6.30338×10^{-2}	225.96
20	15	200	146.04698	146.11108	5.50818×10^{-2}	168.67
20	15	300	145.99485	146.09398	6.77513×10^{-2}	252.59

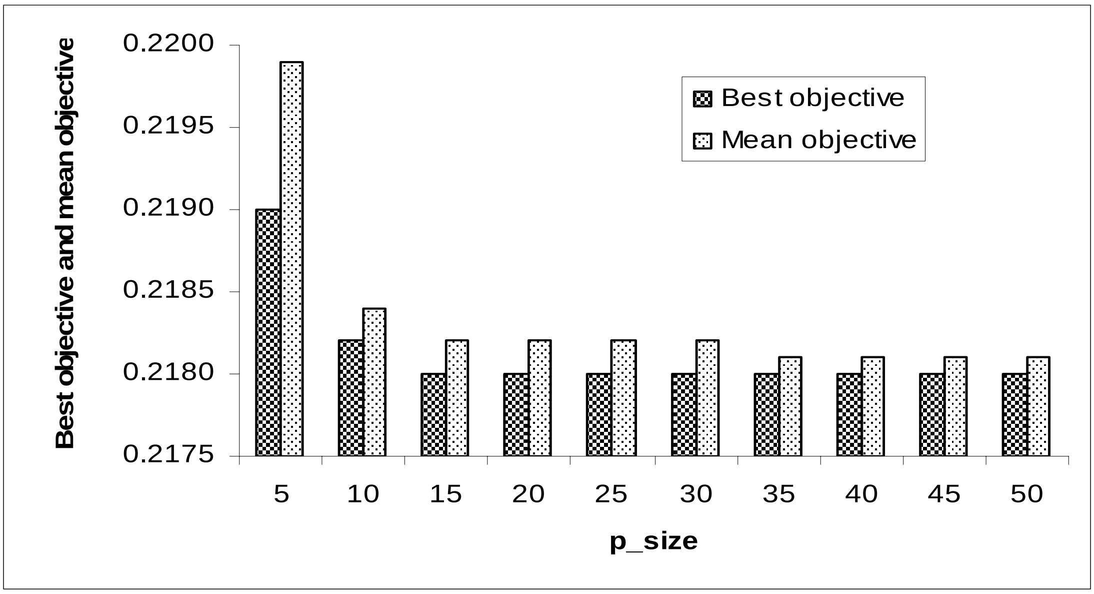

Fig. 1: Population size (*p_size*) vs. Best and mean objective function values

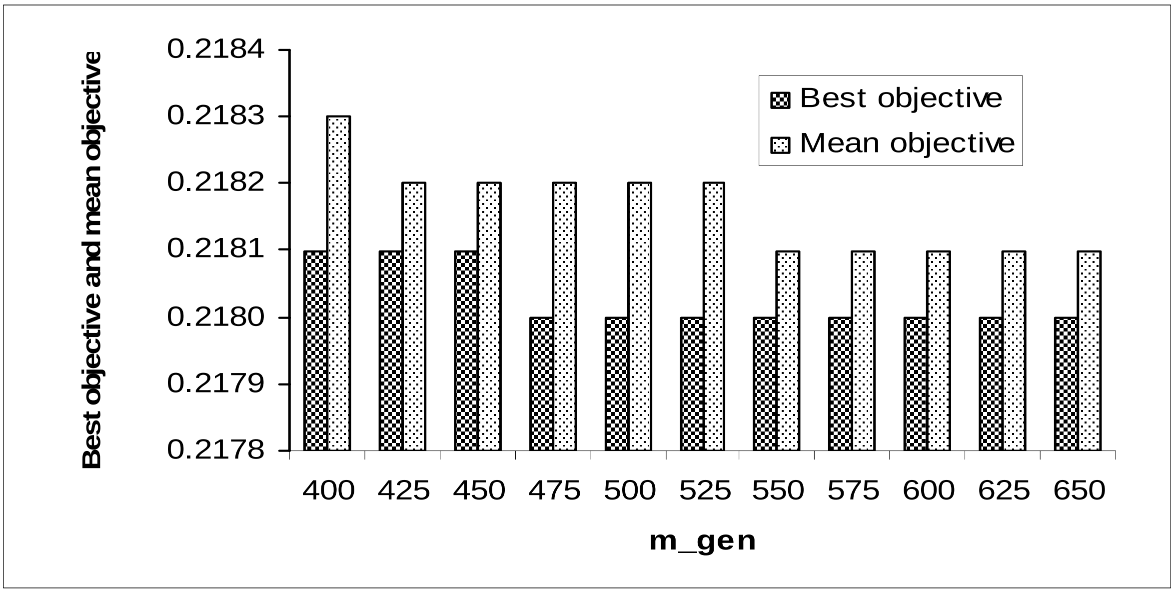

Fig. 2: Maximum no. of generation (*m_gen*) vs Best and mean objective function values

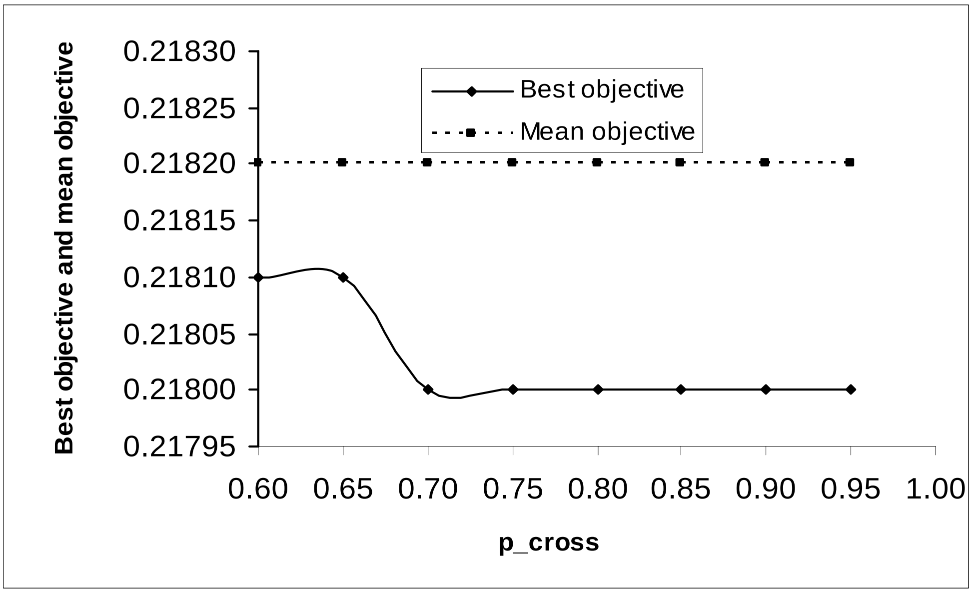

Fig. 3: Probability of crossover (*p_cross*) vs Best and mean objective function values

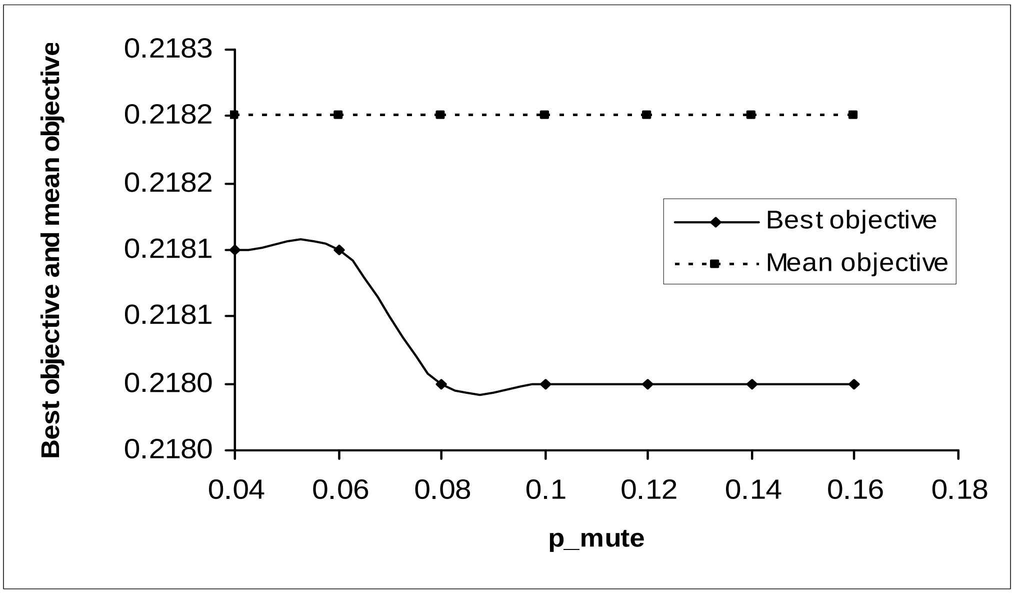

Fig. 4: Probability of mutation (*p_mute*) vs Best and mean objective function values

6. Conclusion

For the first time, we have proposed an alternative approach based on advanced genetic algorithm to solve the PDE-constrained optimization problems subject to Poisson partial differential equations. This approach does not require any derivative information. The aim of this study is to determine the global or close to global optimum (though the global optimality cannot be proved analytically with the help of proposed algorithm). From the study of sensitivity analysis, it has been observed that the proposed algorithm is efficient as well as stable. In this paper, the proposed approach is applied for solving constrained optimization problem with bound constraints. However, it can be applied for the problem with unknown search space also.

Now there is a question: why this problem is solved by GA though it can be solved by gradient based indirect method. The objective function of the optimization problem is quadratic, but the numbers of control variables as well as the state variable are larger. As a result, the computational cost will be high in case of gradient based indirect method than GA.

For future research, one may apply this approach for solving PDE-constrained optimization problems in the areas of fluid flow and aerodynamic shape optimization.

7. Acknowledgements

For this research, the second author would like to acknowledge the financial support provided by the Council of Scientific and Industrial Research (CSIR), New Delhi, India. The authors are also grateful to Dr. S. B. Hazra, University of Trier, Trier, Germany.

8. References

[1] S. B. Hazra, *An efficient method for aerodynamic shape optimization*, AIAA, pp.2004-4628, 2004.
[2] S. B. Hazra, *Multigrid one–shoot method for state constrained aerodynamic shape optimization*, SIAM J. Sci. Comput. 30 (2008), pp. 3220-3248.

[3] S. B. Hazra and V. Schulz, *Simultaneous pseduo-timestepping for pde-model based optimization problems*, BIT Numer. Math. 44 (2004), pp. 457-472.
[4] S. B. Hazra, V. Schulz, J. Brezillon, and N. R. Gauger, *Aerodynamic shape optimization using simultaneous pseudo-timestepping*, J. comp. Phys, 204 (2005), pp. 46-64.
[5] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur Methods for PDE-Constrained Optimization. Part I: The Krylov-Schur Solver*, SIAM J. Sci. Comput. 30(6) (2005), pp. 3220-3248.
[6] F. E. Campana, D. Peri, Y. Tahara and F. Stern, *Shape optimization in ship hydrodynamics using computational fluid dynamics*, Comput. Methods Appl. Mech. Eng. 196 (2006), pp. 634-651.

[7] R. Griesse and B. Vexler, *Numerical Sensitivity Analysis for the Quantity of Interest in PDE-Constrained Optimization*, SIAM J. Sci. Comput. 29(1) (2007), pp. 22-48.

[8] T. Rees, H. S. Dollar and A. J. Wathen, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput. 32 (2010), pp. 271-298.

[9] T. Rees, M. Stoll and A.Wathen, *All-At-Once Preconditioning in PDE-Constrained Optimization*, Kybernetika, 46(2) (2010), pp. 341 – 360.

[10] J. H. Holland, *Adaptation in Natural and Artificial System*, University of Michigan press, Ann Arbor, Cambridge, USA, 1975.

[11] D. E. Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning*; Addison-Wesley Publishing Co.Inc., MA, 1989.

[12] Z. Michalawicz, *Genetic Algorithm + Data structure = Evoluation Programs*, 3rd revised and extended edition, Springer Verlog, Berlin, Heidelberg, New york, 1996.

[13] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Optimization*, John Wiley & Sons Inc., 2000.

[14] M. Sakawa, *Genetic algorithms and fuzzy multiobjective optimization*, Kluwer Academic Publishers, 2002.

[15] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, Springer, 2003.

[16] M. Hinze, R. Pinnau, M. Ulbrich and S. Ulbrich, *Optimization with PDE- Constraints, Mathematical Modeling: Theory and Applications*, Springer-Verlag, New York, (2009).