Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 5 (2015), pp. 2645-2655

http://www.ripublication.com

Statistical Investigation Of Optimization Methods For Solving Constrained Non-Linear Problems

Umme Maria[1], Radha Gupta[2], Rajendran D[3], Kokila Ramesh[4]

[1]M.Phil student in Mathematics at School of Engineering and Technology, Jain University.NH – 209, Kanakapura Main Road, Jakkasandra post, Ramanagara district, Karnataka 562112, India.
[2]Faculty and Head of the Department in Mathematics at School of Engineering and Technology, Jain University, Kanakapura, Bangalore.
[3]Senior Scientist, Animal Nutrition Division, National Institute of Animal Nutrition and Physiology, Bangalore.
[4]Faculty in Mathematics at School of Engineering and Technology, Jain University, Kanakapura, Bangalore.
[1]mariahumme@gmail.com[2]radha.gaurav.gupta@gmail.com
[3]rajnutri@yahoo.com[4]nimai.ramesh@gmail.com

ABSTRACT

Constrained non-linear optimization problems (CNLP) with bounds on the decision variables, where objective functions are minimized/maximized under given constraints frequently appear in real world. There are many traditional as well as heuristic algorithms to solve CNLP, most of which are based on numerical approximations. In the past two decades, the use and development of heuristic-based algorithms to solve CNLP have significantly grown. In fact, now a days Excel Solver and MATLAB® toolboxes are using GA as inbuilt function to solve these CNLP. The objective of this paper is to identify connections and contrasts between the Excel Solver, MATLAB® toolboxes and heuristics algorithms written by researchers to solve some of the constrained non-linear benchmark problems. The theoretical and graphical investigation of the computational results obtained using different techniques is discussed here. It is observed that Excel solver and MATLAB®toolboxes can effectively be used to solve CNLP up to 10 variables and 8 constraints with bounds on the decision variables.

Key words: Constrained Non-Linear problem, Evolutionary algorithms, Heuristic algorithms

1Corresponding author

INTRODUCTION

While modelling the real time scenarios, non-linear problems are encountered quite frequently and hence they attract a large amount of importance. The problems that we see in many spheres of life are usually found to be non-linear because of their complex nature.Solving these complex problems manually is a tedious work, and many times impossible due to the mathematical rigidity of the properties that needs to be satisfied.

Also, when compared to linear, non-linear problems are usually much more complicated and difficult to solve. Thus the usage of computers could be of great help to the researchers willing to make progress in this area. At times, a non-linear problem can be converted to a linear problem by making certain approximations/relaxations which have to be assumed in order to solve the problems. The solution thus obtained by this way will not be up to the mark and might not help in its implication to the real life scenario.

The general form of a non-linear problem is:

Optimize $f(X)$

Subject to constraints $g_j(X)$ $(\leq = \geq)$ b_j j=1,2,……., m

where: $X=(x_1,x_2,x_3,\ldots\ldots\ldots,x_n)$

The set of eight non-linear problems considered in this study have been taken from [8] where an attempt has been made to solve the CNLP by Genetic Algorithm (GA), and by Particle Swarm Optimization (CPSO) technique. These non-linear problems are solved using the MATLAB® toolbox (fmincon and GA) and the Excel solver, which too give promising results thus meeting the main objective of the study.

A comparative study of these results obtained with that solved by Particle Swarm Optimization (CPSO) and Genetic Algorithm (GA),has been investigated.

Also, a graphical analysis of all the techniques with respect to mean and confidence intervals in terms of achievement of the objective function, is exhibited here.

There are many evolutionary, algorithms and traditionaltechniques available in literature to solve Constrained Non-Linear Problems (CNLP) with bounds on the decision variables. The major challenge faced here is that one technique that is suitable for one particular problem may be highly inefficient for solving the other problem.Therefore, it becomes difficult to choose the appropriate technique to solve a particular problem. Thus the motive behind this comparative investigation, though not the best, is to get an idea to identify the appropriate technique to approach a particular problem thus making decision making an easier task.

To begin with one of the first algorithms, Random Search Technique (RST) was originally developed by Price in 1965, which was further improvised by Kusum Deep and C Mohan, who used Fortran 77 to solve the problems. Also since then, there have come many more versions and hybrids of RST which are based on programming languages like C, C++ and different methods [12]. Later on there came RST dependent algorithms like Genetic Algorithms (GA) and their hybrids [5], [10], [13], [14] and [15] which are based on evolution and which were universally used by many researchers to solve many complex non-linear problems. There are also many more evolutionary and heuristic algorithms and their hybrids and extensions which have

been used in the study of these complex non-linear problems like the tabu search algorithms [15], SQP [6], Particle Swarm Optimization (PSO) [1], [3], [5], [8], Ant Colony Optimization (ACO) [9], Imperialist Competitive Algorithms (ICA) [2], Line-up competition algorithms [11], Big bang-big crunch [7] and many more. There has also been a lot of work going on for non-linear problems that are unconstrained [4].
The data for the study done has been utilized from the work done by [8], which contains different types of non-linear problems. This paper tries to show by comparison that CPSO is an effective method for solving non-linear problems.

MATERIALS AND METHODS

This comparative study has many non-linear methods that have been used to solve the benchmark problems. The description of the methods is done below:

(i) Particle Swarm Optimization (PSO)-This is an evolutionary computation technique that has been developed by Kennedy and Eberhart. In this technique, the population of random solutions is initialized, after which we search for the optima by updating generations and then the potential solutions that follow the current optimum particles are flown through the problem.

(ii) Genetic Algorithm (GA) - A genetic algorithm (GA) is a method for solving constrained as well as unconstrained optimization problems using the principle of the biological evolution. The algorithm, inspired from the Darwin’s theory of evolution, repeatedly modifies a population of individual solutions. At each step, the genetic algorithm randomly selects individuals from the current population and uses them as parents to produce the children for the next generation. Over successive generations, the population "evolves" toward an optimal solution. It can solve problems in which the objective function is discontinuous, non-differentiable, stochastic, or highly nonlinear. It is a stochastic global searchmethod.

(iii) MATLAB®– fmincon and GA:MATLAB® stands for Matrix Laboratory.It is a language with high performance capability used for technical computing. There are two ways of solving problems here, one is by solving the problem by the coding method and the other is by making use of the toolbox. In the toolbox there are many tools to solve these problems that are computer oriented and hence makes the process of solving faster, comparatively lesser tedious and also the chances of making error is lesser when compared to manual solving of the problems. MATLAB® is also known for its graphics. There are a lot of plotting functions available in the MATLAB® toolbox that helps in giving a good visual representation of the solutions of the problem. The study here is making use of the optimization toolbox with the tools fmincon and Genetic Algorithm (GA).fmincon - It is used to find the minimum of constrained non-linear multivariable function. It finds a minimum of a constrained non-linear multivariable function, and by default is based on the SQP (Sequential Quadratic Programming) algorithm. The algorithm repeatedly modifies a population of individual solutions. It is deterministic by nature.

(iv) Evolutionary Solver (GA) - The new Evolution solver accepts Solver models defined in exactly the same way as the Simplex and GRG Solvers, but uses genetic algorithms to find its solutions. While the Simplex and GRG solvers are used for linear and smooth non-linear problems, the Evolutionary Solver can be used for any Excel formulas or functions, even when they are not linear or smooth non-linear.

GRG Nonlinear-Excel: The Generalized Reduced Gradient (GRG) Algorithm is used for optimizing the non-linear problems. *GRG* Solver is a deterministic local optimization method.

The GRG (Generalized Reduced Gradient) method -- used in the standard Excel Solver since 1990, and also included in the Premium Solver -- assumes that the objective function and constraints are smooth nonlinear functions of the variables. A smooth non-linear function has a smooth (possibly curved) graph with no sharp 'corners' or 'breaks.' For such problems, the GRG method is quite accurate and quite fast -- often 10 to 20 times faster than a genetic or evolutionary algorithm -- and yields a locally optimalsolution. This solution is provably 'best within the vicinity,' but does not rule out other,better solutions that may be far away from the initial values of the variables.

Here we make use of the spreadsheet to write the problem and then make use of the excelsolver which solves the problem. This technique not only provides us with the answer of the respective problem, but also gives information about the sensitivity, limit and feasibility of the solution obtained.

Here, the first two methods are the ones used by [8], to solve the benchmark problems. The third and the fourth methods are the ones used for solving the problems in this paper.

For the fmincon solver, the algorithm that has been used, is the Active set. The derivatives are taken as approximated by the solver. The start point is considered to be one for all the decision variables. For the options including the stopping criteria, the default values have been retained, and the level of display has been selected as iterative.

For the GA solver, the population type is considered to be a double vector, the population size is taken as 50, the number of generations is taken as 1000 and the trials are taken as 100, of which the most optimal is chosen as the solution. The level of display here too is taken as iterative and for the other options the default values have been utilized.

In Microsoft Excel, the solving method is considered to be GRG Non-linear, the population size here also is taken as 50, and the other options are considered to have taken the default values.

PROBLEMS

Problem 1 - Source : Ref [8]	**Problem 2 - Source : Ref [8]**
Min $5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$,	Min $x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + (x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$,
Sub To $0 \le 85.334407 + 0.0056858\, x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 \le 92$, $90 \le 80.51249 + 0.0071371x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 \le 110$, $20 \le 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 \le 25$, $78 \le x_1 \le 102$, $33 \le x_2 \le 45$, $27 \le x_i \le 45$, i = 3, 4, 5	Sub To $105 - 4x_1 - 5x_2 + 3x_7 - 4x_8 \ge 0$, $-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \ge 0$, $-10x_1 + 8x_2 + 17x_7 - 2x_8 \ge 0$, $-2x_1^2 - 2(x_2 - 1)^2 + 2x_1x_2 - 14x_5 + 6x_6 \ge 0$, $8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \ge 0$, $-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \ge 0$, $3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \ge 0$, $-0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \ge 0$, $-10 \le x_i \le 10$, i = 1,……,10
Problem 3 - Source : Ref [8]	**Problem 4 - Source : Ref [8]**
Max $\sin^3(2\pi x_1)\sin^3(2\pi x_2)$,	Min $(x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$,
Sub To $x_1^2 - x_2 + 1 \le 0$, $1 - x_1 + (x_2 - 4)^2 \le 0$, $0 \le x_1 \le 10$, $0 \le x_2 \le 10$	Sub To $127 - 2x_1^2 - 3x_2^4 - x_3 - 4x_4^2 - 5x_5 \ge 0$ $282 - 7x_1 - 3x_2 - 10x_3^2 - x_4 + x_5 \ge 0$, $196 - 23x_1 - x_2^2 - 6x_6^2 + 8x_7 \ge 0$, $-4x_1^2 - x_2^2 + 3x_1x_2 - 2x_3^2 - 5x_6 + 11x_7 \ge 0$, $-10 \le x_i \le 10$, i=1,……,7
Problem 5 - Source : Ref [8]	**Problem 6 – Source : Ref[8]**
Min $x_1 + x_2 + x_3$	Min $-a_1\exp\left(-a_2\sqrt{\frac{1}{n}\sum_{j=1}^{2} x_j^2}\right) - \exp\left(\frac{1}{n}\sum_{j=1}^{2}\cos(a_3x_j)\right) + a_1 + e$,

Sub To $1 - 0.0025(x_4 + x_6) \geq 0$, $1 - 0.0025(x_5 + x_7 - x_4) \geq 0$, $1 - 0.01(x_8 - x_5) \geq 0$, $x_1x_6 - 833.33252x_4 -$ $100x_1 + 83333.333 \geq 0$, $x_2x_7 - 1250x_5 - x_2x_4 + 1250x_4 \geq 0$, $x_3x_8 - 1250000\, x_3x_5 + 2500x_5 \geq 0$, $100 \leq x_1 \leq 10000$; $1000 \leq x_i \leq 10000$, i=2,3; $10 \leq x_i \leq 1000$, i=4,……,8	Sub To $-5 \leq x_j \leq 5$ j=1,2 Where the constants are set as, $a_1 = 20$, $a_2 = 0.2$, $a_3 = 2\pi$, e = 2.71282
Problem 7 - Source : Ref [8]	**Problem 8 - Source : Ref [8]**
Max $-2x_1^2 + 2x_1x_2 - 2x_2^2 + 4x_1 + 6x_2$,	Max $\dfrac{x_1^2x_2^2x_3^2}{2x_1^3x_3^2 + 3x_1^2x_2^2 + 2x_2^2x_3^3 + x_1^3x_2^2x_3^2}$
Sub To $2x_1^2 - x_2 \leq 0$, $x_1 + 5x_2 \leq 5$, $x_1, x_2 \geq 0$	Sub To $x_1^2 + x_2^2 + x_3^2 \geq 1$, $x_1^2 + x_2^2 + x_3^2 \leq 4$, $x_1, x_2, x_3 \geq 0$

In the different non-linear problems that are being considered in this paper, problem-1 is an interval optimization problem, problem-2includes inequality constraints, problem-3 is a multimodal problem, problem-4is a non-convex linear problem, problem-5 is a penalty hard problem, problem-6 is an exponential function (Ackley function), problem-7 is a quadratic problem with non-linear and linear constraints and problem-8is a non-linear fractional problem with non-linear constraints.

The values of the objective function found by the different techniques namely: Excel Solver, MATLAB (fmincon, GA) are given in Table 1 for the respective problems where all the decimal places are rounded off up to six digits. Table 2 gives the values of the decision variables for all the problems and the decimal places here are rounded off up to three digits.

TABLE 1. Comparative study results for objective function values

PROBLEM AND TYPE	EXCEL SOLVER (a)	MATLAB TOOLBOX (fmincon) (b)	MATLAB TOOLBOX (GA) (c)	CPSO		GA	
				BEST	DEVIATION FROM	BEST	DEVIATION
1) Min Z	-30665.5	-30665.538675	-30599.834194	-30664.7	(a): 0.000026 (b): 0.000027 (c): -0.002115	-30665.5	(a): 0 (b): 0.000001 (c): -0.002141
2) Min Z	20.22142	20.221422	22.774489	24.80818	(a): -0.184889 (b): -0.184889 (c): -0.081977	25.237	(a): -0.198739 (b): -0.198739 (c): -0.097575
3) Max Z	1	1	1	0.9999	(a): 0.000100 (b): 0.000100 (c): 0.000100	0.95825	(a): 0.043569 (b): 0.043569 (c): 0.043569
4) Min Z	683.981	680.630057	680.902748	680.667	(a): 0.004869 (b): -0.000054 (c): 0.000346	680.67	(a): 0.004864 (b): -0.000059 (c): 0.000342
5) Min Z	7049.248	7049.248020	-	7114.84	(a): -0.009219 (b): -0.009219	7115.00	(a): -0.009241 (b): -0.009241
6) Min Z	-0.00546	-0.005460	-0.004733	0.005451	(a): -2.001651 (b): -2.001687 (c): -1.868230	0.005456	(a): -2.000733 (b): -2.000769 (c): -1.867434
7) Max Z	6.613085	6.613085	6.614019	6.611034	(a): 0.000310 (b): 0.000310 (c): 0.000452	-	-
8) Max Z	0.148441	0.148441	0.148441	0.153728	(a): -0.034395 (b): -0.034394 (c): -0.034394	-	-

TABLE 2:Results for decision variables of all the problems

PROBLEM	DECISION VARIABLES	METHODS		
		EXCEL SOLVER	MATLAB (fmincon)	MATLAB (GA)
Problem 1	x_1	78	78	78.070
	x_2	33	33	33.298
	x_3	29.995	29.995	30.382
	x_4	45	45	45
	x_5	36.776	36.776	35.791
Problem 2	x_1	1.983	1.983	1.989
	x_2	3.212	3.211	2.887
	x_3	8.245	8.244	8.721
	x_4	5.196	5.196	5.138
	x_5	0.531	0.531	1.038
	x_6	2.058	2.058	3.013
	x_7	0.832	0.832	0.988
	x_8	10	10	10
	x_9	7.794	7.794	8.178
	x_{10}	8.764	8.765	9.375
Problem 3	x_1	1.25	1.75	1.75
	x_2	4.25	4.75	4.75

Problem 4	x_1	2.348	2.330	2.236
	x_2	1.935	1.951	1.944
	x_3	0	-0.477	-0.438
	x_4	4.298	4.366	4.409
	x_5	0	-0.624	-0.639
	x_6	1.048	1.038	1.161
	x_7	1.582	1.594	1.545
Problem 5	x_1	579.307	579.307	
	x_2	1359.97	1359.971	
	x_3	5109.971	5109.971	
	x_4	182.018	182.018	
	x_5	295.601	295.601	
	x_6	217.982	217.982	
	x_7	286.416	286.417	
	x_8	395.601	395.601	
Problem 6	x_1	0	0	0
	x_2	0	0	0
Problem 7	x_1	0.659	0.659	0.659
	x_2	0.868	0.868	0.868
Problem 8	x_1	1.044	1.046	1.045
	x_2	1.249	1.248	1.248
	x_3	1.162	1.161	1.162

RESULTS & DISCUSSIONS:
The advantage of using Excel Solver &MATLAB® toolboxes (fmincon and GA) is that one need not have the knowledge of any computer language to write the code for GA or any other heuristic algorithm. Both these Excel Solver &MATLAB® toolboxes have GA as inbuilt functions. An individual has to feed in the problem correctly and can get the expected solution without putting much effort.

Table 1 shows that:

- For problems 1and 4, the objective of the problem is exactly achieved by all the five techniques with negligible deviations, and also the solutions obtained by using theExcel solver and GA give the most minimum value for problem 1 and fmincon gives the most minimum value for problem 4 when compared to the results obtained by the other methods.
- For problem 2, the results have a deviation ranging from 8 to 20 percent.
- For problem 3, Excel solver, fmincon, GA toolbox and CPSO give exact results whereas GA used by [8] gives slightly lower results with 4% deviation.
- Problem 5, Excel solver and fmincon give most minimum results with only 1% deviation.
- For problem 6, Excel solver and fmincon give the minimum valued solutions.

- Problem 7 and 8 have not been solved using GA used by [8]and in the case of problem 7, all the methods give exact results with negligible deviations. For problem 8, CPSO gives the best solution with the deviation being almost 3%.

Whenever there is a difference in answer, these inbuilt functions are giving better answers.
As far as CPSO is concerned, this technique also gives near to optimal solutions in problems 1,3,4,6 and 8 and can be used effectively to solve CNLP.
This investigative study suggests that one does not have to worry about writing the code for solving small size CNLP. Inbuilt toolboxes & Excel solver are very efficient in giving the accurate results. But in the case of problem 5, the GA from MATLAB® toolbox did not give a solution to the problem, in this case the feasible point could not be found. However, the problems tested here are limited to 8 constraints and 10 variables with bounds on the decision variables. We are in the process of investigating the efficiency of these inbuilt toolboxes & Excel solver on large size CNLP.
Also this study suggests that one can try the hybrid of two heuristic approaches to get an improved technique. As both these inbuilt toolboxes & Excel solver use GA to give solutions, so looking at the efficiency of CPSO one can also try the usage of CPSO as an inbuilt function for both these packages.
In Figure 1, the first four and the sixth graph, I, II, III, IV& V represent the non-linear methods / techniques Excel solver, fmincon (MATLAB®), GA (MATLAB®), CPSO and GA used by [8] respectively. In the fifth graph I, II, III & IV represent the techniques i.e., Excel Solver, fmincon, CPSO and GA used by [8] respectively. In the 7^{th} and 8^{th} graph I, II, III, IV represent solutions got by using Excel Solver, fmincon, GA (MATLAB®), CPSO respectively.

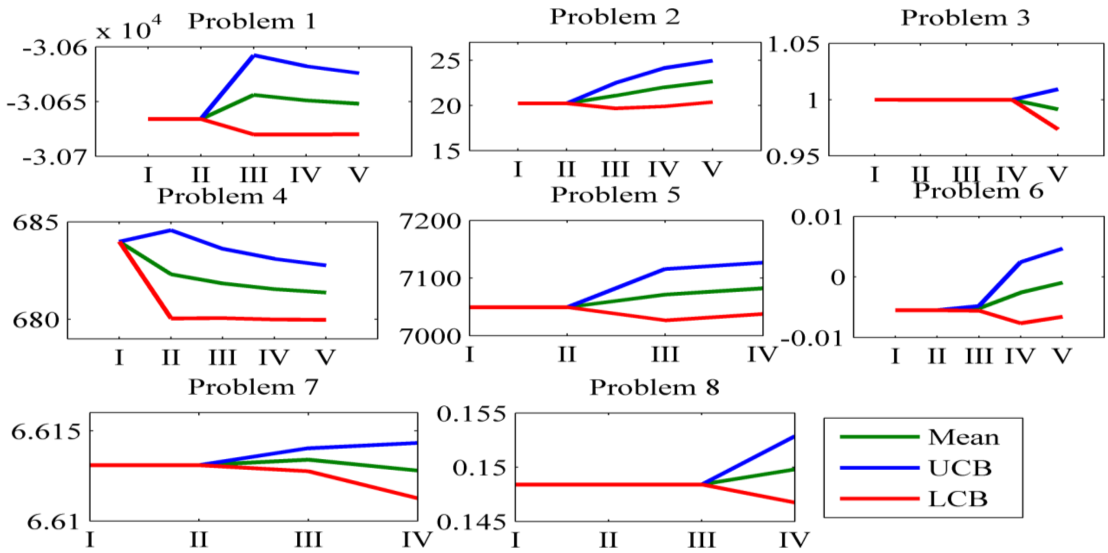

Figure 1: The variation of the mean w.r.t the different techniques and its confidence bands: UCB- Upper Confidence Band and LCB- Lower Confidence Band.

The graphical representation above, shows that the running mean of the objective functional value using different cases lies well within the 95% confidence bands. This infers that the methods used in the paper go along with the methods used by [8]. Therefore these methods can be used to solve any CNLP with limited no of variables and constraints.

CONCLUSION

The realistic problems we see are more often non-linear and also constantly varying which brings in demand for improvised techniques that give better optimal results. Though there are many techniques, there is still need for more robust techniques and also there are studies going on in this field to find techniques that are more efficient and also feasible. In the study that has been attempted here, the problems have been limited to a maximum of8 constraints and 10 variables with bounds on the decision variables, hence these techniques can be further tested on more complex problems.In the process of the study being done, a new technique can be found that can be useful to solve these problems and also by testing the technique on certain benchmarking problems, one can find whether the technique is useful and also whether it is robust by nature.

REFERENCES

1. MazdakShokrian and Karen Ann High. 2014. “Application of a multi objective multi-leader particle swarm optimization algorithm on NLP and MINLP problems,”Comput. Chem. Eng., 60, pp. 57-75.
2. Mahdi Abdollahi, AyazIsazadeh, and DavoudAbdollahi. 2013. “Solving the Constrained Nonlinear Optimization based on Imperialist Competitive Algorithm,”Int. J. Nonlinear. Sci., 15(3), pp. 212-9.
3. Yang Sun, Lingbo Zhang and XingshengGu. 2012. “A hybrid co-evolutionary cultural algorithm based on particle swarm optimization for solving global optimization problems,”Neurocomputing., 98, pp. 76-89.
4. Dexuan Zou, LiqunGao, Jianhua Wu and Steven Li. 2010. “Novel global harmony search algorithm for unconstrained problems,”Neurocomputing., 73, pp. 3308-18.
5. Kusum Deep. 2009. “Recent advances in optimization techniques and their applications,” Department Of Mathematics And Continuing Education Centre, Indian Institute of Technology.
6. Zhibin Zhu. 2008. “A sequential equality constrained quadratic programming algorithm for inequality constrained optimization,” J. Comput. Appl. Math., 212, pp. 112-25.
7. Osman K. Erol and Ibrahim Eksin. 2006. “A new optimization method: Big bang-big crunch,” Adv. Eng. Softw., 37, pp. 106-11.

8. Ying Dong, Jiafu Tang, Baodong Xu and Dingwei Wang. 2005. “An Application of Swarm Optimization to Nonlinear Programming,” Comput. Math. Appl., 49, pp. 1655-68.
9. Christian Blum. 2005. “Ant colony optimization: Introduction and recent trends,” Phys. Life. Rev., 2, pp. 353-73.
10. Richard Y. K. Fung, Jiafu Tang and Dingwei Wang. 2002. “Extension of a hybrid Genetic Algorithm for nonlinear programming problems with equality and inequality constraints,” Comput. Oper. Res., 29, pp. 261-74.
11. Liexiang Yan and Dexian Ma. 2001. “Global optimization of non-convex nonlinear programs using Line-up Competition Algorithm,”Comput. Chem. Eng., 25, pp. 1601-10.
12. C. Mohan and H. T. Nguyen. 1999. “A Controlled Random Search Technique Incorporating the Simulated Annealing Concept for Solving Integer and Mixed Integer Global Optimization Problems,” Comput. Optim. Appl., 14, pp. 103-32.
13. Jiafu Tang and Dingwei Wang. 1998. “A Hybrid Genetic Algorithm for a Type of Nonlinear Programming Problem,”Comput. Math. Appl., 36(5), pp. 11-21.
14. Takao Yakota, Mitsuo Gen, Takeaki Taguchi., and Yinxiu Li. 1995. “A Method for Interval 0-1 Nonlinear Programming Problem using a Genetic Algorithm,” Comput.Ind. Eng., 29(1-4), pp. 531-5.
15. Fred Glover, James P. Kelly and Manuel Laguna. 1995. “Genetic algorithms and Tabu search: Hybrids for optimization,” Comput. Oper. Res., 22(1), pp. 111-34.