International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

# Introduction to Optimization Toolbox in MATLAB to Optimize Different Types of Optimization Problem

Y. D. Shahakar
Department of Electrical Engineering
SGBAU , Amravat
PRPCE & M,Amravati 444602, India.

Dr. A.S. Telang
Department of electrical Engineering
SGBAU , Amravati
PRPCE & M,Amravati 444602, India.

D. A. Shahakar
Department of Electrical Engineering
SGBAU ,Amravati
PRPCE & M,Amravati 444602, India

***Abstract* -** Optimization plays an important role in our day to day life. There are number of techniques of optimizations are available to find the solution in various field like engineering as well as non engineering areas. There are various toolbox of optimization which are available in MATLAB. MATLAB is useful software in which number of toolboxes like signal processing , optimization, neural network , statistics etc are used to optimize the solution . Here optimization toolbox has been listed which are very useful to determine the optimum solution of different types of engineering as well as non engineering optimization problem. Classical optimization methods require more iteration and time to find the solution. So in case of large number of variables , it is not possible to apply such classical techniques. Instead of using classical/conventional optimization methods , here we are using optimization toolbox to optimize solution in accurate manner.

***Keywords*** : **Optimization toolbox , LPP,NLPP , Problem based Solver**

## I. INTRODUCTION

Optimization plays an important role in our day to day life. The word “optimum” is Latin, and means “the ultimate ideal;” similarly, “optimus” means “the best.” Therefore, to optimize refers to trying to bring whatever we are dealing with towards its ultimate state. There are number of engineering and non engineering applications of optimization like in civil, mechanical, electrical engineering, transportation etc. There are number of optimization methods are available to find the exact optimum solution in less time and iteration. Conventional methods as well as intelligence based methods which are very useful to find the optimum solution for any type of problems.

Optimization is define as the technique of obtaining the best possible solutions out of the available alternatives under the given circumstances. **Optimization** is the mathematical discipline which is concerned with finding the maxima and minima of functions, possibly subject to constraints. Optimization is a precise procedure using design constraints and criteria to enable the planner to find the optimal solution. Optimization techniques have been applied in numerous fields to deal with different practical problems. ***Optimization*** is an important tool in making decisions and in analyzing physical systems. In mathematical terms, an ***optimization problem*** is the problem of finding the *optimum* solution from among the set of all *feasible* solutions.

There are various optimization toolbox available in MATLAB. These tools are very much useful to find optimum solution for different types of optimization problem. There are two approaches of optimization problem in MATLAB optimization toolbox like problem based optimization and solver based optimization. These toolboxes are useful to find exact optimum solution Optimization deals with selecting the best option among a number of possible choices that are feasible or don't violate constraints. MATLAB can be used to optimize parameters in a model to best fit data, increase profitability of a potential engineering design, or meet some other type of objective that can be described mathematically with variables and equations.

Mathematical optimization problems may include equality constraints (e.g. =), inequality constraints (e.g. <, <=, >, >=), objective functions, algebraic equations, differential equations, continuous variables, discrete or integer variables, etc

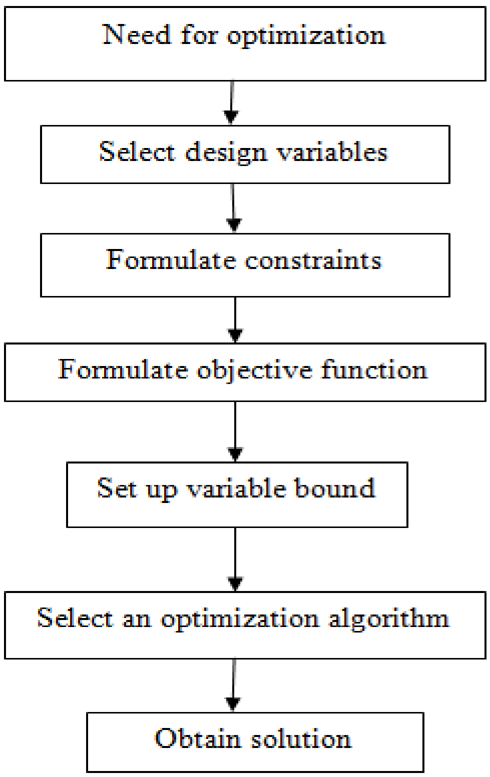
## II. FLOWCHART FOR OPTIMAL DESIGN

A flowchart means the separate steps of a process in sequential order. It is a generic tool that can be adapted for a wide variety of purposes, and can be used to describe various processes. There are various recent optimization algorithms which can apply to determine the optimum solution. From this algorithm,

 DOI: 10.55041/IJSREM12155 

we can design/draw flowchart. So it is very easy to apply flowchart for any type of problems. To optimize the solution, proper techniques/methods plays an important role in any optimization problem or any other type of problems

**Flowchart:**




By using flowchart , for any type of problem , we can optimize the solution.

## III.MATLAB OPTIMIZATION TOOLBOX

Optimization Toolbox is an optimization software package developed by MathWorks. It is an add-on product to MATLAB, and provides a library of solvers that can be used from the MATLAB environment. The toolbox was first released for MATLAB in 1990. Optimization toolbox is a function that extends the capability of MATLAB. To maximize or minimize objective functions, Optimization Toolbox™ provides functions for finding the value of unknown parameters.
It is collection of function that extend the capability of MATLAB. The toolbox includes unconstraint optimization, constraint non linear optimization , include goal attainment problem , minimax problems and semi infinite minimization problem and semi infinite minimization problem ,non linear least square and curve fitting , non linear system of equation solving, constraints linear least square, specialized algorithms for large scale problems.
MATLAB's Optimization Toolbox consist of

- Fsolve Implementation
- Improving Code .

Optimization Toolbox™ has two approaches to solving optimization problems or equations: problem-based and solver-based. Before you start to solve a problem, you must first choose the appropriate approach.

**A] Comparison between Problem Based & Solver Based Approach**

**Table I: Comparative Analysis**

| Problem Based Approach | Solver Based Approach |
|---|---|
| In the problem-based approach, you use symbolic-style variables to create optimization expressions and constraints | In the solver-based approach, you must place all variables into a single vector, which can be awkward, especially with variables of large or differing dimensions. |
| Represents the objective and constraints symbolically | Represents the objective and constraints as functions or matrices |
| Requires translation from problem form to matrix form, resulting in a longer solution time | Does not require translation from problem form to matrix form, resulting in a shorter solution time |
| Automatically calculates and uses gradients of objective and nonlinear constraint functions in many cases, but does not calculate Hessians | Allows direct inclusion of gradient or Hessian, but does not calculate them automatically |
| The solve function automatically chooses a solver that can handle your objective and constraints | In the solver-based approach you must choose an appropriate solver. |
| No Global Search or Multi Start. | Currently, to use these solvers you must use the solver-based approach. |
| No multiobjective problems | To solve multiobjective problems |

| | |
|---|---|
| | using gamultiobj or pareto search, use the solver-based approach. |
| No custom data types | o use a custom data type with ga or simulannealbn d, you must use the solver-based approach. |
| No checkpoint file for surrogate opt | Use the solver-based approach for checkpoint files in surrogate opt |
| No initial point or population for ga, particle swarm, or surrogate opt | See Initial Points for Global Optimization Toolbox Solvers. |
| No visual interface | The Optimize Live Editor task currently applies only to the solver-based approach. |

## VI. Optimization Toolbox Solvers and Methods

Optimization Toolbox™ solvers are grouped into four general categories:

- **Minimization**

  Solvers for minimization type are used to find minimum of the objective function near a starting point x0. They consist of an linear programming problem, non linear programming problem unconstrained optimization, quadratic programming, cone programming etc

  **Multiobjective minimization**

  Solvers for multiobjective minimization type are used to find either minimize the maximum value of a set of functions (fminimax), or to find a location where a collection of functions is below some specified values (fgoalattain).

- **Equation solvers**

  Equation solvers are used to find a solution to a scalar- or vector-valued nonlinear equation $f(x) = 0$ near a starting point x0. Equation-solving can be considered a form of optimization because it is equivalent to finding the minimum norm of $f(x)$ near x0.

- **Least-Squares Solvers**

  Least square solvers are used to

  minimize a sum of squares. This type of problem frequently arises in fitting a model to data. The solvers address problems of finding nonnegative solutions, finding bounded or linearly constrained solutions, and fitting parameterized nonlinear models to data.

**Global Optimization Toolbox Functions**

Global optimization refers to searching for the global optima. A global optimization algorithm, also calleda global search algorithm, is intended to locate a global optima. Global Optimization Toolbox provides functions that search for global solutions to problems that contain multiple maxima or minima. You can improve solver effectiveness by adjusting options and, for applicable solvers, customizing creation, update, and search functions.

Global solvers overcome the weakness through methods of range bounding i.e. interval analysis and convex analysis and range reduction techniques i.e. linear programming and constraint propagation within a branch-and-bound framework to find the global solutions to non-convex models.

Following are the different optimization toolbox available in MATLAB applicable for various optimization methods.

**Table II: Optimization Toolbox with Methods**

| Toolbox | Methods |
|---|---|
| fminsearch, fminunc | Unconstraint minimization problems |
| fmincon | Constraint minimization problems |
| linprog | Linear programming problems |
| *intlinprog,* | mixed-integer linear programming problems |
| *quadprog* | quadratic programming problems |

**Optimization Problem Set up:**
**Choosing a Solver**

**Table III : Global Optimization Toolbox its Applicability**

| Toolbox | Applicability |
|---|---|
| *patternsearch* | Miinimum of function using pattern search |
| *ga* | Miinimum of function using genetic algorithm |
| *gamultiobj* | Miinimum of multiple functions using genetic algorithm |
| *particleswarm* | Particle swarm optimization |

| | |
|---|---|
| *simulannealbnd* , | Minimum of function using simulated annealing algorithm |

### Set Options

**Table IV: Options for Optimization Problems**

| | |
|---|---|
| optimoptions | Create optimization options |
| resetoptions | Reset options |
| optimtool | Select solver and optimization options, run problems |

### IV. Computational Difficulties

Any optimization problem can be solved by using the classical methods .When we apply all these tabular methods, it requires too much computation time to solve large-scale problems. Because of complicated calculations, it requires more time and iteration. So it is necessary to apply another reliable method, which reduces computation time & iterations to get the exact optimum solution.

In MATLAB, there are various optimization toolboxes are available, which are very much useful to solve linear programming, nonlinear programming problem, quadratic programming , integer programming, single and multivariable constraint and unconstraint optimization problem to find the accurate optimum solution in less time and iterations. Classical techniques are time consuming and required more number of iterations to get the solution. So to solve same problems , we can implement another simplified tools to get the optimal solution.

To overcome all above difficulties, there are various toolboxes are available in MATLAB to solve any type of optimization problem consisting of large number of variables. When we apply the optimization toolbox in MATLAB for a particular problem , there is need to identify only it is linear or non linear optimization problem.

### V MATLAB Optimization Toolbox Example:

#### A] Linear Programming Example

| General Form | Standard Form |
|---|---|
| Maxf = $5X_1 - 4X_2 + 3X_3$<br>STC<br>$2X_1 + X_2 - 6X_3 = 2$<br>$6X_1 + 5X_2 + 10X_3 \le 76$<br>$8X_1 - 3X_2 + 6X_3 \le 50$<br>$X_1, X_2, X_3 \ge 0$ | Minf = $5X_1 - 4X_2 + 3X_3$<br>STC<br>$2X_1 + X_2 - 6X_3 = 2$<br>$X_1 + 5X_2 + 10X_3 \le 76$<br>$8X_1 - 3X_2 + 6X_3 \le 50$<br>$X_1, X_2, X_3 \ge 0$ |

| MATLAB Code Using Optimization Tools | |
|---|---|
| | |
| f=[-5 4 -3];<br>A=[6 5 10; 8 -3 6];<br>b=[76 50];<br>Aeq = [2 1 -6];<br>beq=[20];<br>lb=[0 0];<br>[x,fval]=linprog (f, A, b,[],[],lb) | X= 6.25 , 0<br><br>fval= -31.23 |

#### B] Linear Programming Example-Solver Based Approach

| General Form | Standard Form |
|---|---|
| Maxf = $5X_1 + 3X_2$<br>STC<br>$-3X_1 - 5X_2 \ge -15$<br>$5X_1 + 2X_2 \le 10$<br>$X_1, X_2 \ge 0$ | Minf = $-5X_1 - 3X_2$<br>STC<br>$3X_1 + 5X_2 \le 15$<br>$5X_1 + 2X_2 \le 10$<br>$X_1, X_2 \ge 0$ |
| **Input Data** | |
| f=[-5 4 -3]<br>A=[6 5 10; 8 -3 6]<br>b=[76 50]<br>Aeq = [2 1 -6]<br>beq=[20]<br>lb=[0 0] | |

### Steps to solve above problem

1. In MATLAB command window, click optimtool
2. Select an optimization solver
3. Select algorithm if any specific method mentioned.
4. Select coefficient of objective function, typically the function you want to minimize.
5. Select constraints, if any.
6. Set options, or use the default options
7. Click on start
8. Optimization running & optimal solution found.

**Optimal solution is -12.36842105263158**

#### C] Unconstrained Example

$$Minf(x) = e^{x_1}(4x_1 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1)$$

**Solution:**

**M-file %**

objective function

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 04 | April - 2022 Impact Factor: 7.185 ISSN: 2582-3930

L = @(x) exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1);

u0=[-1,1]; % Initial guess

[x,fval,exitflag,output]=fminunc(L,u0)

Results; Optimization terminated: relative infinity-norm of gradient less than options. Tol

Fun. x = 0.5000 -1.0000 fval = 1.0983e-015

exitflag = 1

output =

iterations: 8

funcCount: 66

stepsize: 1

firstorderopt: 7.3704e-008

algorithm: 'medium-scale: Quasi-Newton line search'

message: [1x85 char]

**Example M-file for Graphics**

%fun_

plot [x,y]=meshgrid(0:0.1:1,-2:0.1:0);

z=exp(x).*(4.*x.^2 + 2.*y.^2 + 4.*x.*y+2.*y+1);

figure, mesh(x,y,z) Lee EE529 xlabel('x(1)'),ylabel('x(2)'),zlabel('f(x)'),title('Unconstrained Minimization') figure,

mesh(x,y,z)

xlabel('x(1)'),ylabel('x(2)'),zlabel('f(x)')

title('Unconstrained Minimization-

View along x(1)') view(0,0)

figure, mesh(x,y,z)

xlabel('x(1)'),ylabel('x(2)'),zlabel('f(x)')

title('Unconstrained Minimization-View along x(2)')

view(90,0)

Graphic Results

0 0.2 0.4 0.6 0.8 1 -2 -1.5 -1 -0.5 0 0 5 10 15 x(1)

**Unconstrained Minimization**

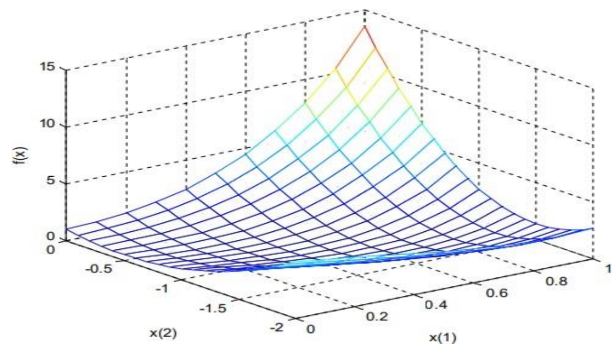



Fig. 1 : Mesh Plot of Unconstraint Optimization problem

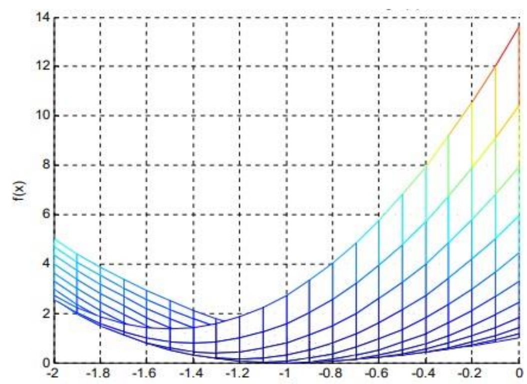



Fig. : 1 (a) Mesh Plot looking along x(1) axis

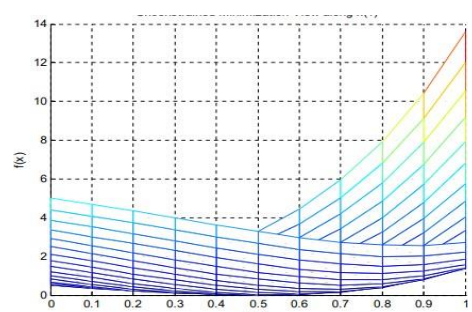



Fig. : (b)Mesh Plot looking along x(2) axis

From the above problem analysis , it is clear that optimization toolbox approaches provide a better solution than a conventional/tabular methods. It also provide exact optimum solution in less time and iteration

Optimized code has faster execution speed, utilizes the memory efficiently and gives better performance.

**VI Conclusion**

Optimization plays an important role in any type of problems. From the above discussion, it is clear that, there are various optimization toolboxes are available in MATLAB. Here we discussed about the problem based and solver based approaches. For any type of optimization problem consisting of large number of design variables and constraints, tabular methods are more complicated & takes more computation time & iterations and possibility to get inaccurate solution because of more calculations . It is important to show the effects of optimization toolbox to get the exact optimum solution in minimum time & iterations. Proposed methodology i. e optimization toolboxes consisting of problem based and solver based approaches plays an important role for any type of optimization problem, which is the best , more useful , required less computation time & most accurate method. Many real world problems like industrial , transportation , production , Research etc consisting of large number of variable & constraints can be easily solve by using solver based or problem based methods approaches. Optimization solvers help improve decision-making around planning, allocating and scheduling scarce resources. They embed powerful algorithms that can solve mathematical programming models, constraint programming and constraint-based scheduling models. Also we can explored the use of toolbox as an economic , tool to reduce the complexity over computation using introduced algorithm.

**References:**

[1] G. B. Dantzig,"Linear Programming and Extensions", Princeton University Press, Princeton, NJ. 1963.
[2] John Walkenbach, "Excel 2007 Formulas", John Wiley and Sons. 2007
[3] J.J. and Wright, S.J."Optimization software guide". Society for Industrial and Applied Mathematics. 1993.
[4] T. Cliff Ragsdale, "Spreadsheet Modeling and Decision Analysis, 6th Edition". SOUTH-WESTERN, Cengage Learning. 2011.
[5]N.Tripathi," A Study of Determination of Energy for Open Channel Flows Under Sluice Gate", Bulletin of the Marathwada Mathematical Society,13(2), pp. 48-52, 2012.
[6] Namrata Tripathi," Common fixed point theorem of a countable family of λ demicontractive mapping in Banach spaces", International Journal of Mathematics Trends and Technology, 11(2), pp. 89-96, 2014.
[7] Namrata Tripathi, "Some Suzuki-type fixed point theorems for generalized multivalued mappings and applications, South Asian Journal of Mathematics (SAJM), 2(3), pp. 205-209, 2012
[8] Nikolaos Ploskas & Nikolaos Samaeas "Linear Programming Using MATLAB (Springer Optimization & its applications)" 1st edition 2017.
[9] Singereshu S. Rao, "Engineering Applications: Theory & Practice", Fourth Edition ,John Wiley & Sons.
[10] D.S. Hira. and P.K. Gupta. Operations research, Seventh revised Edition 2014. ISBN: 81-219-0281-9 . [11] Taha, H. A. , " Operations Research An Introduction " , Macmillan publishing co., Inc. United States , 1982..
[12] E.R. Barnes, A Variation on Karmarkar's Algorithm for Solving Linear Programming problems, Mathematical Programming 36, 1986, p. 174-182 .
[13] Hillier, Frederick S and Lieberman, Gerald. Introduction to Operations Research. 2005. Boston: McGraw-Hill. ISBN: 0- 07-123828-X.
[14] Barnes,J, W, and R.M. Crisp, "Linear Programming: A Survey of General Purpose Algorithms,"AIIE Transactions,
[15] Namrata Tripathi & Namita Shrivastav "Optimization problem solved by different platform say optimum toolbox (MATLAB) & Excel solver" International Research Journal of Engineering & Technology. vol 4 , issue 9 sep 2017 .
[16] Optimization in Practice with MATLAB, Achille Messle.
[17] D. Gay, A Variant of Karmarkar's Linear Programming Algorithm for Problems in Standard form, Mathematical Programming 37 (1987) 81-90
[18] E.D. Anderson and K.D. Anderson. Presolving in Linear Programming, Mathematical Programming, 71 (1995), pp. 221-245.
[19] Charles Copper & Henderson,1963, An Introduction to Linear programming, John Willy,New York.
[20] Optimization Toolbox™ 4 User's Guide.
[21] Cesar Parez Lopez "MATLAB Optimization Technique:
Springer.
[22] opova O., Popov B., Romanov D., Evseeva M. Optimel: Software for selecting the optimal method SoftwareX, 6 (2017), pp. 231-236, 10.1016/j.softx.2017.08.001 Article

## Appendix

**Solvers available for different types of problems**

| Problem Type / Solver | LP | NLP | QP | MILP | MINLP | SOCP | Linear Least Squares | Nonlinear Least Squares |
|---|---|---|---|---|---|---|---|---|
| linprog | Yes | No | No | No | No | No | No | No |
| intlinprog | Yes | No | No | Yes | No | No | No | No |
| quadprog | Yes | No | Yes | No | No | Yes | Yes | No |
| coneprog | Yes | No | No | No | No | Yes | No | No |
| lsqlin | No | No | No | No | No | No | Yes | No |
| lsqnonneg | No | No | No | No | No | No | Yes | No |
| lsqnonlin | No | No | No | No | No | No | Yes | Yes |
| fmincon | Yes | Yes | Yes | No | No | Yes | Yes | Yes |
| fminunc | Yes | Yes | Yes | No | No | Yes | Yes | Yes |
| patternsearch | Yes | Yes | Yes | No | No | Yes | Yes | Yes |
| ga | Yes | Yes | Yes | Yes | Yes | No | Yes | Yes |
| particleswarm | Yes | Yes | Yes | No | No | No | Yes | Yes |
| simulannealbnd | Yes | Yes | Yes | No | No | No | Yes | Yes |
| surrogateopt | Yes | Yes | Yes | Yes | Yes | Yes | Yes | Yes |