Chapter 10

Introduction to Optimization Design

10.1 Introduction

In the previous chapters, we discussed how to model uncertainty by probability theory. We also introduced commonly used uncertainty analysis techniques for quantifying the impact of the uncertainty of model input on the model output (performance). Our ultimate goal is to use the knowledge we have gained from uncertainty analysis to manage and mitigate the effects of uncertainty at the design level. Therefore, we can ensure that a design be robust and safe against various uncertainties. The commonly used probabilistic design methodologies include reliability-based design, robust design, and Design for Six Sigma. Since all of these methods need to use optimization during the design process, a brief introduction to optimization design will be given in this chapter. We will then discuss reliability-based design and robust design in Chapters 11 and 12 respectively.

Instead of providing a comprehensive presentation of optimization design techniques, this chapter is intended to present introductory materials about optimization design. It will ensure a reader acquire basic working knowledge that is necessary for optimization modeling, the use of optimization software, and the analysis of optimization results. To help one easily understand the optimization techniques, a graphical means is employed in some cases instead of providing mathematical details. After finishing this chapter and associated homework, one is expected to be able to formulate an engineering optimization problem and solve it with optimization software.

10.2 Optimization Design

Optimization is a design tool that assists designers automatically to identify the optimal design from a number of possible options, or even from an infinite set of options. Optimization design is increasingly applied in industry since it provides engineers a cheap and flexible means to identify optimal designs before physical deployment. Optimization capabilities have also been increasingly integrated with CAD/CAM/CAE software such as Adams, Nastran, and OptiStruct.

Even in our daily life, we are constantly optimizing our goals (objectives) within the limit of our resources. For example, we may minimize our expenditure or maximize our saving while maintaining a certain living level. When shopping for a car, we may try to meet our preference (performance of the car, safety, fuel economy, etc.) maximally on the condition that the price does not exceed what we can afford. It is the same case in engineering design where we optimize performances of the product while meet all the design requirements.

The general process of optimization design is given in Fig. 10.1.

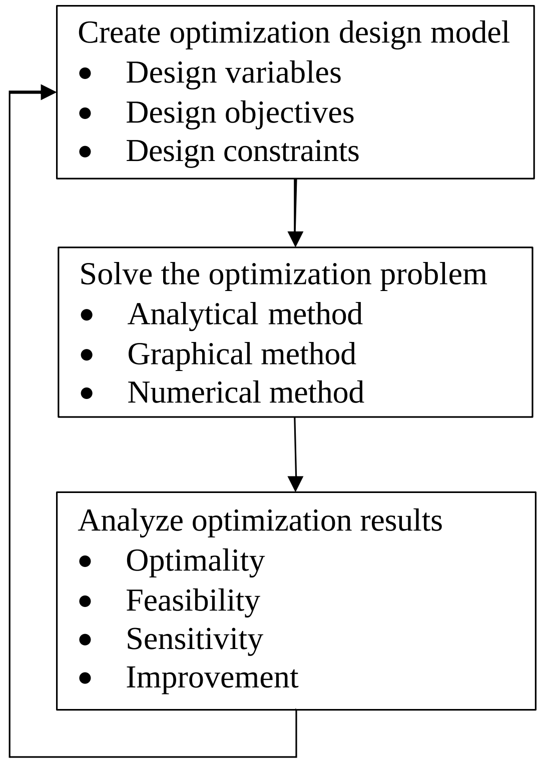

Figure 10.1 The Process of Optimization Design

The first step of optimization design is to create an optimization model in mathematical formulations. This step is called optimization modeling. In this step, several decisions are to be made, such as what will be optimized, what design variables will be changed to produce an optimal design, and what requirements should be met. Modeling is the most important step in optimization design, and designers may spend a significant portion of time on modeling during the optimization process.

The second step is solving the optimization model. Three methods are usually used, including analytical method, graphical method, and numerical method. Methods of solving optimization problems will be discussed in Section 10.5.

The last step is the posterior analysis. In this step, designers perform some analyses on the optimal solution. The following questions will be answered.

- Is the design optimal?
- Is the design feasible?
- Is the design reasonable?
- What design variables are most important to the design performances?
- How would the further improvement be made by modifying the optimization model?

As shown in Fig. 10.1, the optimization process is iterative. If the design solution is not satisfactory, designers will modify the optimization model and repeat the procedure until a satisfactory design is found.

10.3 Optimization Modeling

We will first present a general mathematical optimization model and then discuss the individual components of the optimization model.

Optimization model

A standard optimization model is given by

$$\begin{cases} \underset{\mathbf{d}}{\text{minimize}}\, f(d_1,\ d_2,\ \cdots,\ d_n) \\ \text{subject to} \\ \quad g_i(d_1,\ d_2,\ \cdots,\ d_n) \le 0,\ i=1,2,\cdots,n_i \\ \quad h_j(d_1,\ d_2,\ \cdots,\ d_n) = 0,\ j=1,2,\cdots,n_e \\ \quad d_k^l \le d_k \le d_k^u,\ k=1,2,\cdots,n \end{cases} \tag{10.1}$$

where $\mathbf{d}=(d_1,\ d_2,\ \cdots,\ d_n)$ is the vector of design variables that is to be determined during the design; $f(d_1,\ d_2,\ \cdots,\ d_n)$ is a design objective function that is to be minimized; $g_i(d_1,\ d_2,\ \cdots,\ d_n)$ is an inequality constraint function; $h_j(d_1,\ d_2,\ \cdots,\ d_n)$ is an equality constraint; d_k^l and d_k^u are lower bound and upper bound of design variable d_k, respectively.

The above model can be interpreted as follows: find an optimal set of design variables $\mathbf{d}=(d_1,\ d_2,\ \cdots,\ d_n)$ over the range $d_k^l \le d_k \le d_k^u$ $(k=1,2,\cdots,n)$ that minimizes the design objective function $f(d_1,\ d_2,\ \cdots,\ d_n)$ while satisfies the design constraints $g_i(d_1,\ d_2,\ \cdots,\ d_n) \le 0$ and $h_j(d_1,\ d_2,\ \cdots,\ d_n) = 0$. It is noted that there are three basic components in an optimization problem – design variables, design objective, and design constraint. We will discuss each of them below in more depth.

Design Variables

A design variable is also called a decision variable or control variable. A design variable is under the control of a decision maker (designer) and could have an impact on the solution of the optimization problem. Essentially, a design is determined by a set of design variables. Different combinations of design variables represent different designs. For example, if we design a cubic vessel, the vessel can be represented by a set of design variables, length *l*, width *w*, height *h*, and thickness *t* (see Fig. 10.2). The design variables will therefore consist of $\mathbf{d} = (l,\ w,\ h,\ l)$. Design 1 by $\mathbf{d}_1 = (100cm,\ 30\text{cm},\ 40\text{cm},\ 1\text{cm})$ is considerably different from Design 2 by $\mathbf{d}_2 = (500cm,\ 100\text{cm},\ 160\text{cm},\ 2\text{cm})$ since they have different attributes or performances (volume, cost, strength, etc).

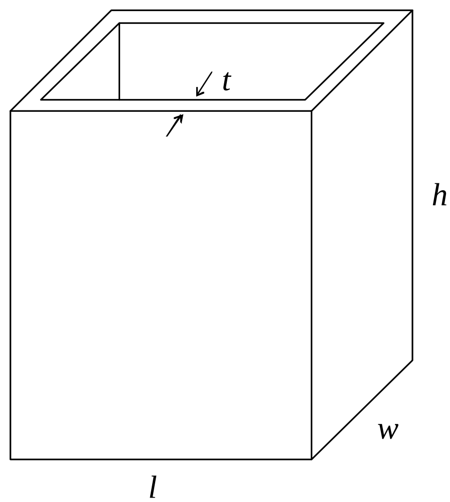

Fig. 10.2 A Cubic Vessel

The goal of the design optimization is to find the best combination of design variables that optimizes designer's preference (design objective) and maintains certain requirements (constraints). For example, for the aforementioned vessel design, we may want to find an optimal set of design variables, length *l*, width *w*, high *h*, and thickness *t*, to minimize the material consumption (or equivalently, the surface area).

A design variable can be in the following forms.

- A continuous variable. In the above vessel design problem, the dimensional design variables are continuous design variables.
- An integer. The number of the teeth of a gear and the type of materials (type 1, type 2, and so on) are examples of the integer design variables.
- A discrete variable. The variable can take values from only a discrete real set. For example, when designing a standard component, designers are required to choose the design variables from a list of recommended values from design standards or design codes.

Design Objective

A design objective represents the desires of the decision makers (designers), for example, to maximize profit or minimize cost. In other words, the design objective can be considered as a criterion to compare whether or not a given design is better than others. In optimization design, a design objective is represented by a mathematical function in term of design variables, and such a function is called objective function, which is given by $f(\mathbf{d})$ in Eq. 10.1. Certain design variable combination determines the optimal value of the objective function.

Choosing an appropriate design objective is important; different design objectives may lead to totally different design results. Examples of design objective include

- Maximizing yield (productivity), strength, robustness, reliability, durability, or safety.
- Minimizing cost, weight, volume, manufacturing time, or probability of failure.

The objective function in the standard optimization model in Eq. 10.1 is to be minimized. If a designer wishes to maximize a objective function $f(\mathbf{d})$, the problem of $\text{Maximize}\, f(\mathbf{d})$ can be easily converted to the standard minimization problem in Eq. 10, because $\text{Maximize}\, f(\mathbf{d})$ is equivalent to $\text{Minimize}\, -f(\mathbf{d})$.

Design Constraint

Designer's desires (for example, increasing the profit) cannot be optimized infinitely since there are limited resources that can be used in product development. The limited resources and other restrictions imposed by government and corporate regulations have to be met strictly. These requirements are expressed by constraint functions in optimization design. A constraint function is also expressed in a mathematical form in terms of design variables. As shown in the optimization model in Eq. 10.1, a constraint function can be an inequality constraint $g_i(d_1, d_2, \cdots, d_n) \le 0$ or an equality constraint $h_j(d_1, d_2, \cdots, d_n) = 0$.

Examples of design constraints include the following.

- The maximum stress should be less than the strength.
- The maximum deflection should be less than an allowable value.
- The probability of failure should be below an acceptable level.
- The cost should not exceed the budget.

Compared to a design objective, a design constraint is "rigid" since it must be satisfied strictly. The former is more "flexible" since we want it as small (or large) as permitted by the design constraints.

In sum, when creating an optimization model, three components must be included: *design variables*, *a design objective*, and *design constraints*.

Example 10.1: Optimization modeling

A company plans to manufacture a product. The total of $5000 funding is available to purchase labor and material. The unit prices of labor and material are $10 and $20, respectively. If d_1 units of labor and d_2 units of material are pursed, the company will produce d_1d_2 units of the product. Formulate a mathematical model that maximizes the quantity of the product that the company can manufacture.

Since the units of the labor and material are to be determined, the design variables are taken as $\mathbf{d} = (d_1,\ d_2)$. The objective is to maximize the quantity of the product, which is given be d_1d_2. Therefore, the objective function is formulated by $f(\mathbf{d}) = d_1d_2$. d_1 and d_2 should satisfy the budget constraint, and this results in the constraint function $g(\mathbf{d}) = 10d_1 + 20d_2 \le 5000$. The optimization model is then given by

$$\begin{cases} \underset{\mathbf{d}}{\text{maximize}}\ f(\mathbf{d}) = d_1d_2 \\ \text{subject to} \\ \quad g(\mathbf{d}) = 10d_1 + 20d_2 \le 5000 \\ \quad d_1 \ge 0 \\ \quad d_2 \ge 0 \end{cases} \tag{10.2}$$

To express the optimization problem in the standard model in Eq. 10.1, we can minimize a new objective $f(\mathbf{d}) = -d_1d_2$, which is equivalent to maximizing d_1d_2. Then the optimization model in a standard form is given by

$$\begin{cases} \underset{\mathbf{d}}{\text{maximize}}\ f(\mathbf{d}) = -d_1d_2 \\ \text{subject to} \\ \quad g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 \le 0 \\ \quad d_1 \ge 0 \\ \quad d_2 \ge 0 \end{cases} \tag{10.3}$$

10.4 Basic Terminologies

Before discussing how to solve an optimization problem specified in Eq. 10.1, we will use the above example to explain the basic terminologies of design optimization. When studying the following definitions, please refer to Fig. 10.3, which illustrates the concepts graphically with the above example.

Design space: The design space is the domain defined by the design variables. In the example, the design space is a two-dimensional real set $\mathbf{d} = (d_1,\ d_2)$.

Design point: Any point in the design space is a design point. A design point represents a design option. When n design variables are involved, a design point is an n-dimensional vector. In the above example, a design point is expressed by $\mathbf{d} = (d_1,\ d_2)$.

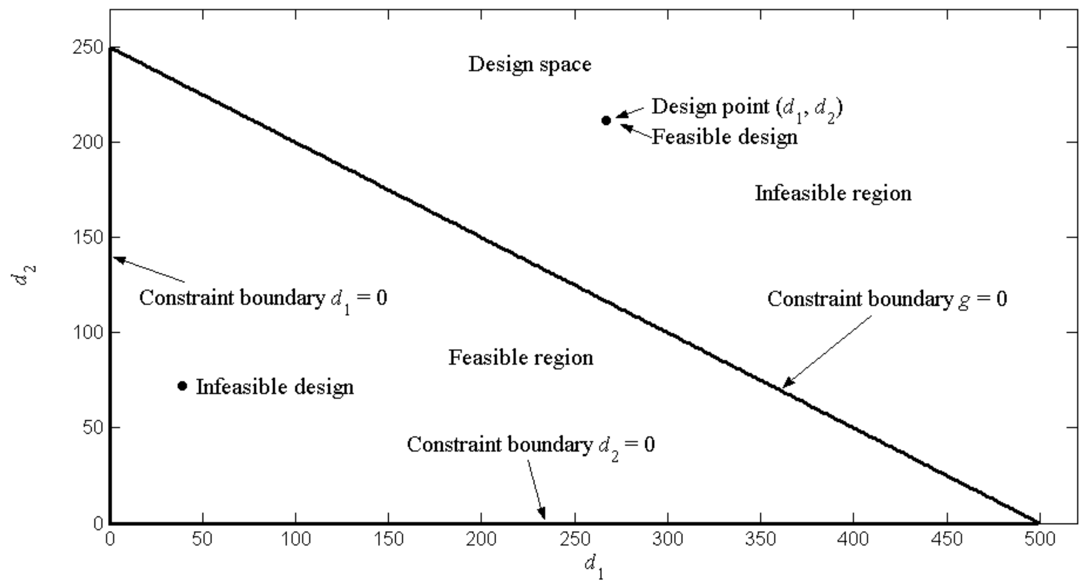

Figure 10.3 Basic Terminologies

Feasible design (point): A feasible design (point) represents a design that satisfies all the design constraints. In others words, if a design point $\mathbf{d} = (d_1,\ d_2,\ \cdots,\ d_n)$ meet all the constraints in Eq. 10.1, $g_i(\mathbf{d}) \le 0\ (i = 1,2,\cdots,n_i)$, $h_j(\mathbf{d}) = 0\ (j = 1,2,\cdots,n_e)$, and $d_k^l \le d_k \le d_k^u\ (k = 1,2,\cdots,n)$, then the design specified by $\mathbf{d} = (d_1,\ d_2,\ \cdots,\ d_n)$ is feasible.

Infeasible design (point): An infeasible design (point) represents a design that violates at least one design constraint.

Constraint boundary: A constraint boundary is a domain of design variables, where the corresponding constraint function reaches its limit. For example, for the constraint function $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 \le 0$, its boundary is $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0$.

Feasible region: The feasible region is the domain composed of feasible design points. All the constraints are satisfied in the feasible region.

Infeasible region: The infeasible region is the domain composed of all the infeasible design points. It is noted that the design space is divided into feasible and infeasible regions by all the constraint boundaries.

Active constraint: If a constraint reaches its limit at a design point, then the constraint is said to be active at that design point. For example, if at some design point,

$g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0$, then the constraint $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 \le 0$ is active at that design point.

10.5 Solve Optimization Problems

There are many ways to solve an optimization problem. We will discuss analytical, graphical, and numerical methods in the following subsections.

Analytical method

If a design $\mathbf{d} = (d_1, d_2, \cdots, d_n)$ is an optimal point, then the following conditions (*Kuhn-Tucker conditions*) hold,

$$\begin{cases} \nabla f(\mathbf{d}) + \sum_{i=1}^{n_i} w_i \nabla g_i(\mathbf{d}) + \sum_{j=1}^{n_e} v_j \nabla h_j(\mathbf{d}) = \mathbf{0} \\ w_i g_i(\mathbf{x}) = 0 & \text{for } i = 1,2,\cdots,n_i \\ w_i \ge 0 & \text{for } i = 1,2,\cdots,n_i \\ h_j(\mathbf{x}) = 0 & \text{for } j = 1,2,\cdots,n_e \end{cases} \tag{10.4}$$

where ∇ stands for the gradient vector and is given by

$$\nabla f(\mathbf{d}) = \left[\frac{\partial f}{\partial d_1}, \frac{\partial f}{\partial d_2}, \cdots, \frac{\partial f}{\partial d_n} \right]^T \tag{10.5}$$

$$\nabla g(\mathbf{d}) = \left[\frac{\partial g}{\partial d_1}, \frac{\partial g}{\partial d_2}, \cdots, \frac{\partial g}{\partial d_n} \right]^T \tag{10.6}$$

and

$$\nabla h(\mathbf{x}) = \left[\frac{\partial h}{\partial d_1}, \frac{\partial h}{\partial d_2}, \cdots, \frac{\partial h}{\partial d_n} \right]^T \tag{10.7}$$

w_i and v_i in Eq. 10.4 are undetermined constants.

Alternatively, we can use the following equivalent conditions without the notation of gradient.

$$\begin{cases} \dfrac{\partial f(\mathbf{d})}{\partial d_i} + \sum_{j=1}^{n_i} w_j \dfrac{\partial g_j(\mathbf{d})}{\partial d_i} + \sum_{k=1}^{n_e} v_k \dfrac{\partial h_k(\mathbf{d})}{\partial d_i} = 0, \quad i = 1,2,\cdots,n \\ w_j g_j(\mathbf{d}) = 0 \qquad \text{for } j = 1,2,\cdots,n_i \\ w_j \geq 0 \qquad \text{for } j = 1,2,\cdots,n_i \\ h_k(\mathbf{d}) = 0 \qquad \text{for } k = 1,2,\cdots,n_e \end{cases} \tag{10.8}$$

We can obtain the optimal solution by solving Eqs. 10.4 or 10.8. To solve the equations, we need to calculate the derivatives of the objective and constraint functions. The simultaneous equations in Eqs. 10.4 or 10.8 are usually nonlinear. Because of the complexities, the analytical method is used only if the problem is very simple.

Example 10.2: Using the analytical method to solve Example 10.1

The standard optimization model in Eq. 10.3 will be used to solve the problem. There is one inequality constraint, and there is no equality constraint.

The derivatives of the objective and constraint function are calculated by

$$\frac{\partial f}{\partial d_1} = -d_2 ,$$

$$\frac{\partial f}{\partial d_2} = -d_1 ,$$

$$\frac{\partial g}{\partial d_1} = 10 ,$$

and

$$\frac{\partial g}{\partial d_2} = 20 .$$

Using Kuhn-Tucker condition $\nabla f(\mathbf{d}) + \sum_{i=1}^{n_i} w_i \nabla g_i(\mathbf{d}) + \sum_{i=1}^{n_e} v_i \nabla h_i(\mathbf{d}) = \mathbf{0}$ in Eq. 10.4 or $\frac{\partial f(\mathbf{d})}{\partial d_i} + \sum_{j=1}^{n_i} w_j \frac{\partial g_j(\mathbf{d})}{\partial d_i} + \sum_{k=1}^{n_e} v_k \frac{\partial h_k(\mathbf{d})}{\partial d_i} = 0, \quad i = 1,2,\cdots,n$ in Eq. 10.8, we have

$$\frac{\partial f}{\partial d_1} + w_1 \frac{\partial g}{\partial d_1} = -d_2 + 10 w_1 = 0 \tag{10.9}$$

and

$$\frac{\partial f}{\partial d_2} + w_1 \frac{\partial g}{\partial d_1} = -d_1 + 20 w_1 = 0 \tag{10.10}$$

From $w_i \nabla g_i(\mathbf{d}) = 0$, we have

$$w_1\left(10d_1 + 20d_2 - 5000\right) = 0 \tag{10.11}$$

Solving Eqs. 10.9 through 10.11 yields $d_1 = 250$, $d_2 = 125$ and $w_1 = 12.5$. Therefore the optimal solution is $d_1 = 250$, $d_2 = 125$, $f(\mathbf{d}) = -d_1 d_2 = -31250$, and $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0$. The solution indicates that if the company purchases 250 units of labor and 125 units of material, it will produce a maximum number of units (31250) of the product with the condition that the cost is equal to $5000.

Graphical Method

If the optimization problem is relatively simple, for example, there are only one deign variable or two design variables, all the objective and constraint functions can be visualized with a 1-D or 2-D plot. One may easily identify the optimal point from the graph.

The procedure of solving an optimization problem graphically is as follows.

- Plot constraint boundaries and identify the feasible design region.
- Plot the contours of objective function and identify the direction along which the objective function increases or decreases.
- Identify the location of the optimal point. Usually, the optimal point is located on one or more constraint boundaries. In other words, the optimal point is usually a tangent point between a contour of the objective function and some constraint boundaries.
- Solve the tangent point if needed.

Let us use Example 10.1 again to demonstrate how to solve an optimization problem graphically.

Example 10.3: Solve Example 10.1 graphically

The constraint boundaries of $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 \le 0$, $d_1 = 0$ and $d_2 = 0$, and contours of objective function $f(\mathbf{d}) = d_1 d_2 = 31250, 20000, 12800, 5000$ are plotted in Fig. 10.4. The contours of the objective function indicate that the objective function increases when both design variables increase. Therefore, moving the design point to the direction pointing toward upper-right will produce higher objective function values. Considering that a design point must remain within feasible region, we see that the optimal point is the tangent point of constraint boundary $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0$ and one of the objective function contours.

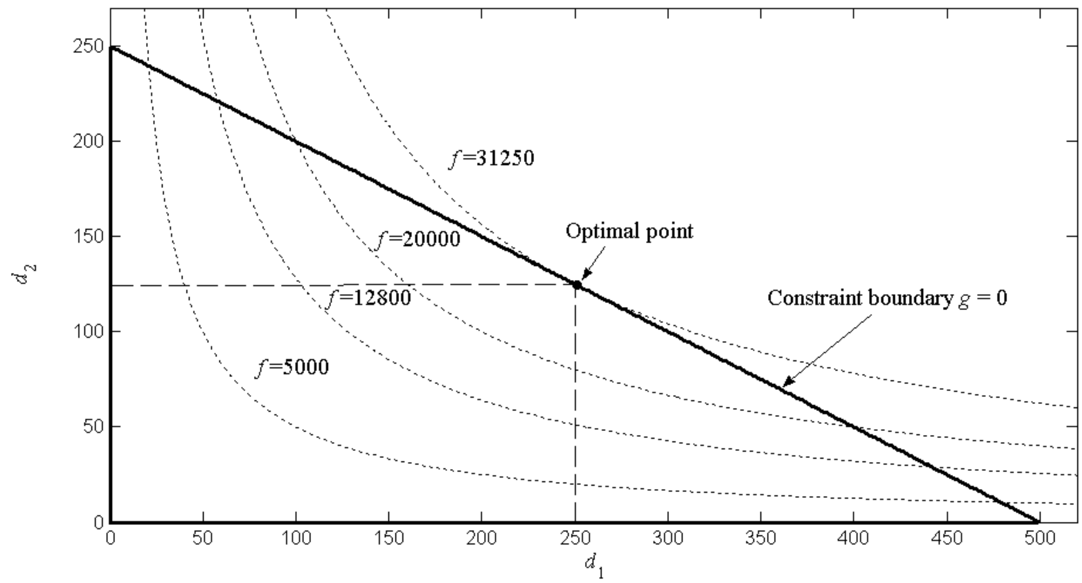

Figure 10.4 Graphic Method

In this case, it may not be easy to obtain the tangent point by just simply looking at the plot. To find the tangent point, we solve following simultaneous equations.

$$g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0 ,$$

and

$$f(\mathbf{d}) = d_1 d_2 = \mathrm{c} ,$$

where c is a constant.

Eliminating variable d_2 from the two equations, we obtain

$$d_1^2 - 500d_1 + 2c = 0 .$$

The roots of the above equation are

$$d_1 = \left(500 \pm \sqrt{250000 - 8c}\right)/2 .$$

Since a unique solution exists (recall that we are looking for a tangent point), we have

$$250000 - 8c = 0 ,$$

which results in $c = 31250$ and $d_1 = 250$. The equation $g(\mathbf{d}) = 10d_1 + 20d_2 - 5000 = 0$ then results in $d_2 = 125$. The solution is identical to the one obtained from analytical method in Example 10.2.

Numerical Method

Most engineering design optimization problems cannot be solved by analytical or graphical methods since a large number of design variables are involved and objective and constraint functions are very complicated. For example, in vehicle engine design, there are hundreds of design variables and constraints. The only practical way to solve complex optimization problems is using numerical methods. Next, we will introduce the commonly used numerical methods which are based on the gradient of objective and constraint functions.

With a numerical method, the optimal design starts from an initial design point (starting point) that represents an initial design. The numerical optimizer evaluates the objective function and constraint functions and their derivatives. Based on the function values and the derivatives, the optimizer generates a search direction along which the objective function will likely to descend. A step size along the descent direction will be searched such that the objective function will decrease to the lowest possible value without violating any constraints. Then, at the next iteration, the current design point moves along the search direction with the specified step size. A new design point is then obtained.

The optimizer evaluates the objective function and constraint functions again at the new design point and checks whether the solution converges. If the convergence is not reached, the optimizer generates a new search direction and the step size for the next iteration. This procedure repeats until optimal solution is found.

Let the current design point be $\mathbf{d}^k$, where k is the iteration counter. The search direction $\mathbf{a}^k$ (an n dimensional vector) and the step size along the direction b^k (a scalar) are generated. Then, a new design point $\mathbf{d}^{k+1}$ is given by

$$\mathbf{d}^{k+1} = \mathbf{d}^k + b^k \mathbf{a}^k . \quad (10.12)$$

Eq. 10.12 is graphically illustrated in Fig. 10.5 for a two-dimensional problem.

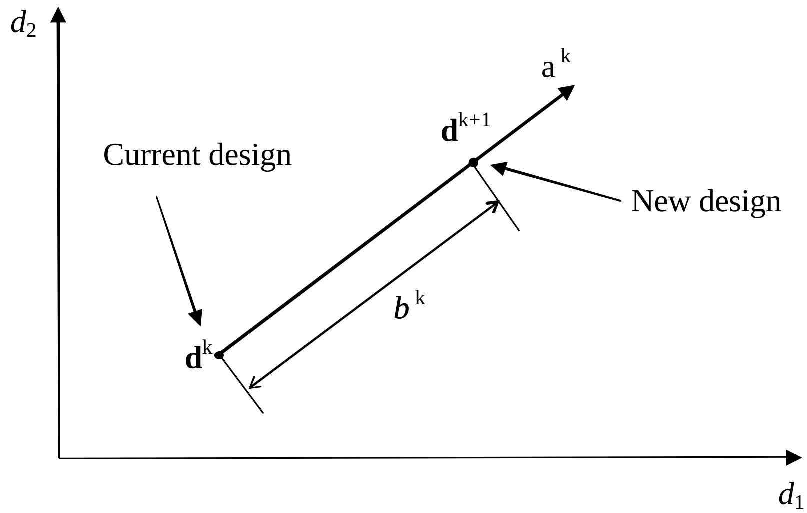

Figure 10.5 Optimization Search

The search process terminates if the convergence has been researched. Some commonly used stopping criteria are listed below, where ***e*** is small positive number.

1) The distance between two consecutive points is small.

$$\left\| \mathbf{d}^{k+1} - \mathbf{d}^{k} \right\| \le e_1 ,$$

where $\|\cdot\|$ stands for a distance. Or,

$$\left| d_i^{k+1} - d_i^{k} \right| \le e_1 \ (i = 1, 2, \cdots, n) .$$

2) The difference between the objective functions at two consecutive points is small.

$$\left| f(\mathbf{d}^{k+1}) - f(\mathbf{d}^{k}) \right| \le e_1 .$$

Other criteria such as Kuhn-Tucker conditions may be also used.

Most of the optimization algorithms use the above strategy to search the optimal point. Different optimization algorithms use different approaches to generate the search direction and step size. This is the reason there exist dozens of popular optimization algorithms. Since the derivatives are evaluated at each design point, the methods are also termed as gradient-based methods. In addition to providing the objective and constraint functions, a user can also provide the functions (equations) of derivatives of the objective and constraint functions. If the analytical derivatives are not provided, the optimizer will evaluate the derivatives numerically. The optimization process can be fully automatic when optimization software is used. The general flowchart of the gradient based methods is given in Fig. 10.6.

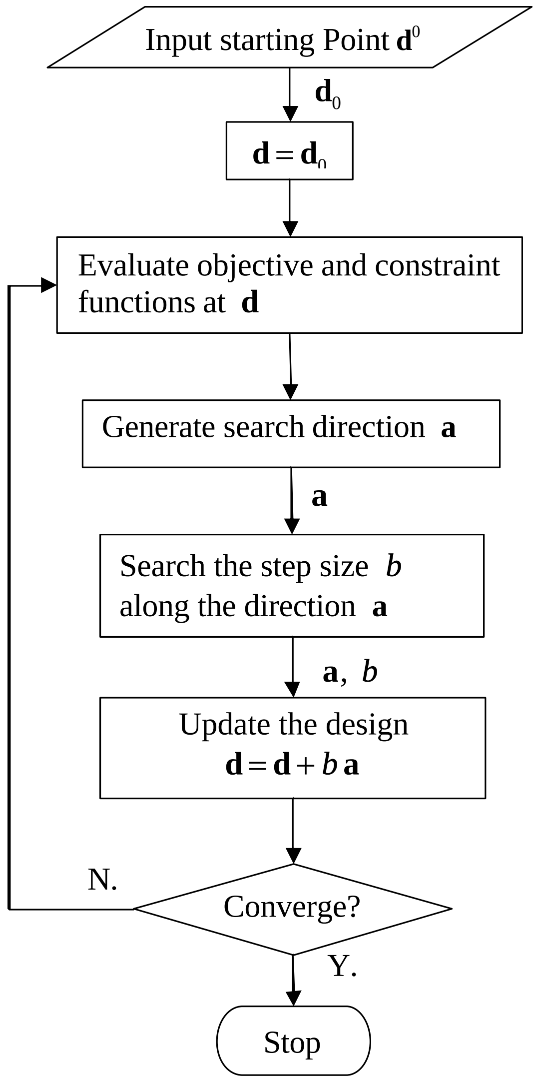

Figure 10.6 Flowchart of Gradient-Based Optimization Methods

It is worthwhile noting that the optimal solution may depend on where the optimization search process starts when multiple local optimal points exist. As shown in Fig. 10.7, d_1^*, d_2^* and d_3^* are three optimal points where the objective function reaches the minimum values locally. d_3^* is the global optimal point since the objective function has the minimum values in the entire design space. If the optimization starts from different stating points, d_1^0, d_2^0 and d_3^0, the search process will end up with the optimal points d_1^*, d_2^* and d_3^*, respectively. Therefore, to obtain a global optimal solution, different starting points may be used. Another strategy is to use so-called global optimization algorithms, which have more chance to find the global optimal solution than non-global optimization ones.

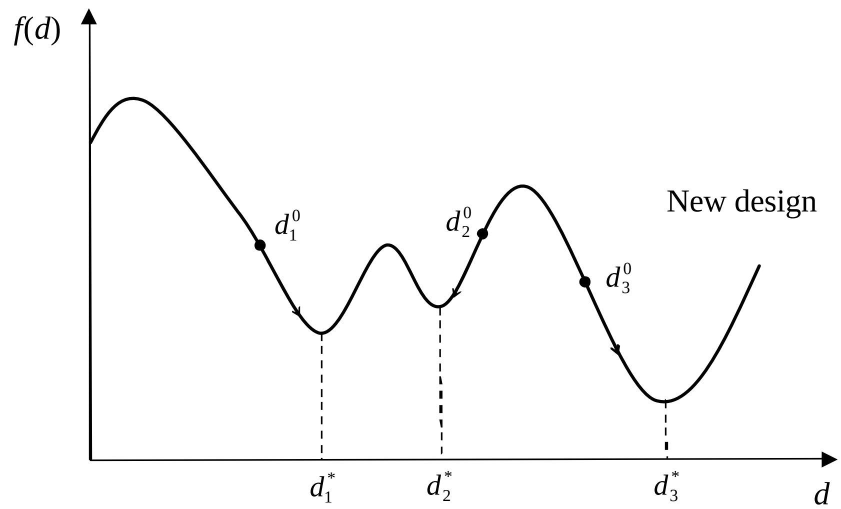

Figure 10.7 Optimal Point and Starting Points

The discussions on how to use two optimization tools, Microsoft Excel and Matlab Optimization Toolbox, are given in the appendices.

Appendix 10.1 Optimization by Excel

The optimizer in Excel is called Solver and is located on the Tools pull-down menu from the main menu bar (see Fig. 10.A.1).

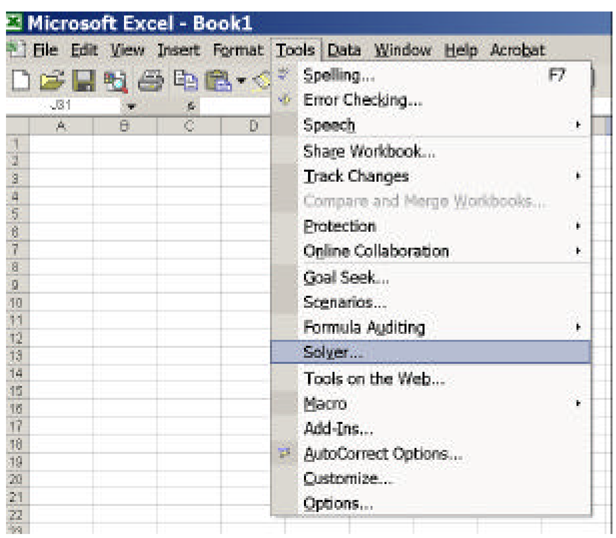

Figure 10.A.1 Solver in Excel

If you do not see the Solver, please do the followings. Click on "Tools" on the menu bar and choose Add-Ins (see Fig. 10.A.1), then a dialog window comes up as shown in Fig. 10.A.2 and check "Solver Add-in" box. Exit the window by pressing "OK".

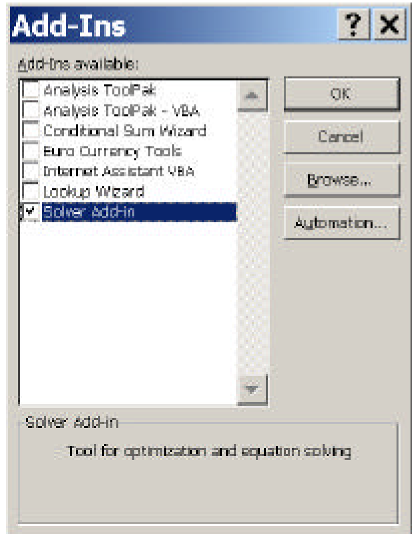

Figure 10.A.2 Add-Ins Window

Next let us see how to set up an optimization problem in Excel.

After clicking on "Solver" from "Tools" from Excel main menu bar, you will see the following setup window (Fig. 10.A.3) and use it to set up an optimization model. Let us examine each part of the dialog box one by one.

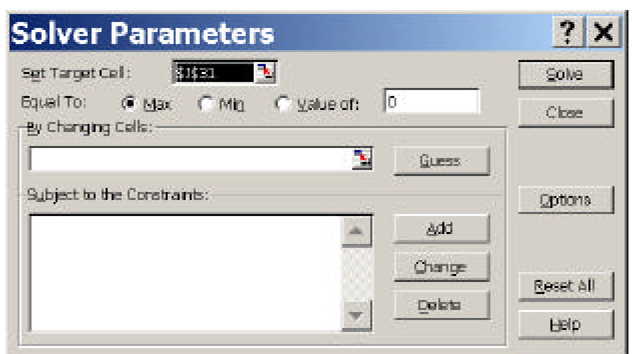

Figure 10.A.3 Solver parameters

Set Target Cell is where you indicate the objective function that is to be optimized. This cell must contain a formula that depends on design variables (defined in other cells in the line of "By changing cell").

Equal to gives you the option of defining either a maximization problem (use Max) or a minimization problem (use Min). We will not use "value of". "By Changing Cells" is where you indicate which cells contain the design variables. You must fill in each cell with a value. These values represent the starting point for the optimization.

Guess guesses all nonformula cells referred to by the formula in the Set Target Cell box, and places their references in the "By Changing Cells" box.

Subject to the Constraints is used to impose constraints on the design variables. Under the heading of "Subject to the Constraints", you have options to add, change, or delete constraints. After you choose "Add", a dialog box will appear as follows (Fig. 10.A.4). You put a constraint function under "Cell Reference" and the limit under "Constraint" and choose the relationship between them by checking one from "<=", ">=", or "=".

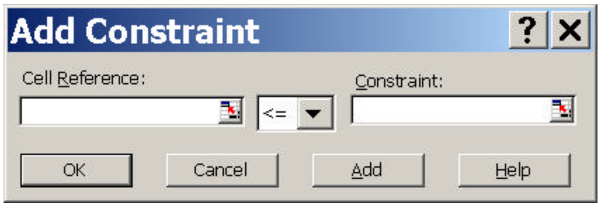

Figure 10.A.4 Add constraints

Options displays the Solver Options dialog box, where you can load and save problem models and control advanced features of the solution process. We will use default values and will not change them.

Reset All clears the current problem settings, and resets all settings to their original values.

After you finish above settings, click on "Solve" and you will see a new window popping up as shown in Fig. 10.A.5. If you choose "Keep Solver Solution", the initial design point will be replaced by the optimal point in the adjustable cells. If choose "Restore Original Values", the initial design point will be remain in the adjustable cells (the cells for design variables). For "Reports', please use "Answer" which will generate a report with the target cell (objective) and the adjustable cells (design variables) with their original and final values, constraints, and information about the constraints.

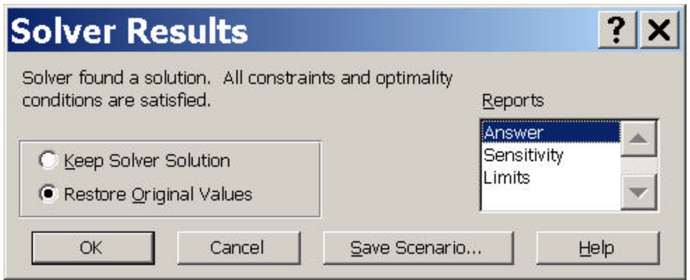

Figure 10.A.5 Set Up How the Results to Be Reported

Example 10.4: Solve Example 10.1 using Excel

The problem setting for the problem is shown in Fig. 10.A.6 and the report of optimal solution is shown in Fig. 10.A.7. The same result is obtained as in Examples 10.2 and 10.3.

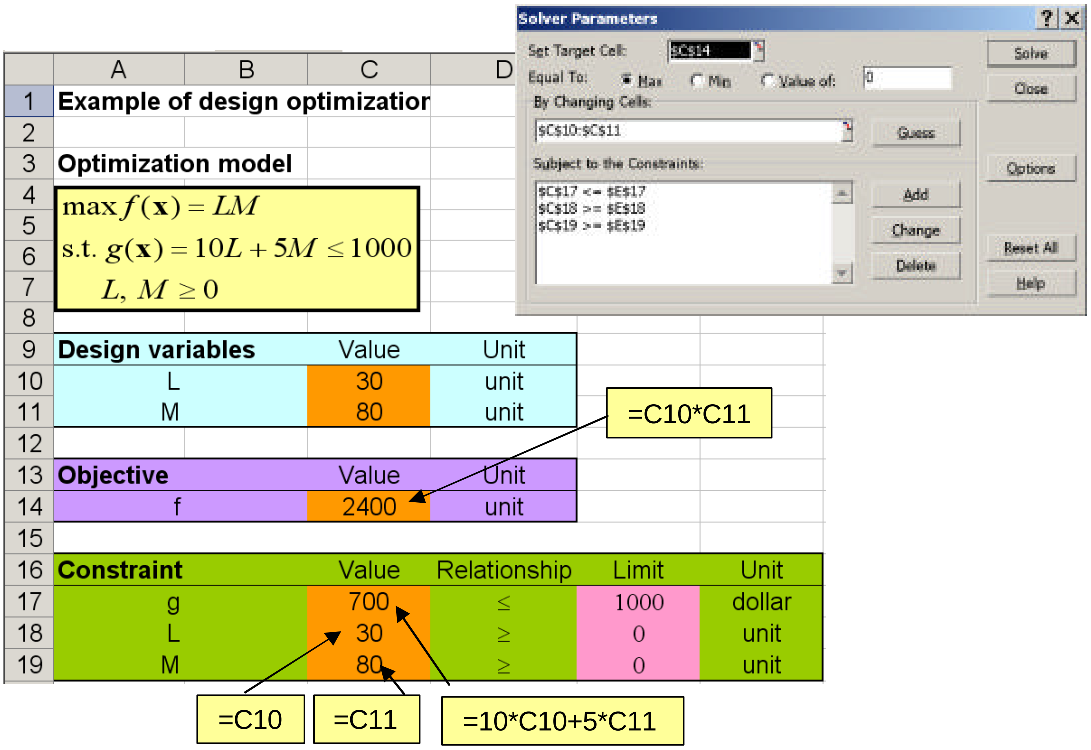

Figure 10.A.6 Set up the optimization problem

Microsoft Excel 10.0 Answer Report
Worksheet: [optimization_example.xls]Sheet1
Report Created: 10/26/2003 6:26:27 PM

Target Cell (Max)

Cell	Name	Original Value	Final Value
$C$14	f Value	2400	5000

Adjustable Cells

Cell	Name	Original Value	Final Value
$C$10	L Value	30	50
$C$11	M Value	80	100

Constraints

Cell	Name	Cell Value	Formula	Status	Slack
$C$17	g Value	1000	$C$17<=$E$17	Binding	0
$C$18	L Value	50	$C$18>=$E$18	Not Binding	50
$C$19	M Value	100	$C$19>=$E$19	Not Binding	100

Figure 10.A.7 Optimal Solution Report

Appendix 10.2 Optimization by Matlab Optimization Toolbox

MATLAB Optimization toolbox provides functions that perform optimization on the various types of problems. We can use either **fmincon** or **constr** functions to solve an optimization problem. Since constr no longer exists in the latest version of Matlab, we will only introduce **fmincon**.

MATLAB optimization mainly consists of three MATLAB files

- Main function
- Objective function
- Constraint function.

Main Function: The purpose of the main function is to initialize the independent variables, to set the optimization options, to call the optimizer **fmincon** function, and to plot the results.

The Syntax for the **fmincon** function is as follows:

d = fmincon(fun,d0,A,b,Aeq,beq,lb,ub,nonlcon,options) minimizes the objective function specified by **fun**.

d0 is the initial value for the design variable. The design is subjected to linear inequality constraints such that **A*d<=b**, linear equality constraints such that **Aeq*d=beq**, and nonlinear inequality constraints **c(x)** and equality constraints **ceq(d)** such that **c(d)<=0** and **ceq(d)=0.** Both inequality constraints **c(d)** and equality constraints **ceq(d)** are defined in function **nonlcon**. **lb** and **ub** are upper and lower bounds for the design variables **d**. Different from the standard formulation given in Eq. 10.1, the Matlab Optimization distinguishes nonlinear constraints from linear constraints, and the optimization model is then given by

$$\begin{cases} \underset{\mathbf{d}}{\text{minimize}}\ f(\mathbf{d}) & \\ \text{subject to} & \\ \quad \mathbf{A}\cdot\mathbf{d} \le \mathbf{0} & \text{Inequality linear constraint} \\ \quad \mathbf{A}\cdot\mathbf{eqd} = \mathbf{0} & \text{Equality linear constraint} \\ \quad c_i(\mathbf{d}) \le 0,\ i = 1,2,\cdots,n_i & \text{Inequality nonlinear constraint} \\ \quad ceq_j(\mathbf{d}) = 0,\ j = 1,2,\cdots,n_e & \text{Equality nonlinear constraint} \\ \quad lb_k \le d_k \le ub_k,\ k = 1,2,\cdots,n & \end{cases} \tag{10.13}$$

Example 10.5 Use Matlab to solve the cantilever beam design

A cantilever beam to be designed is illustrated in Fig. 10.A.8.

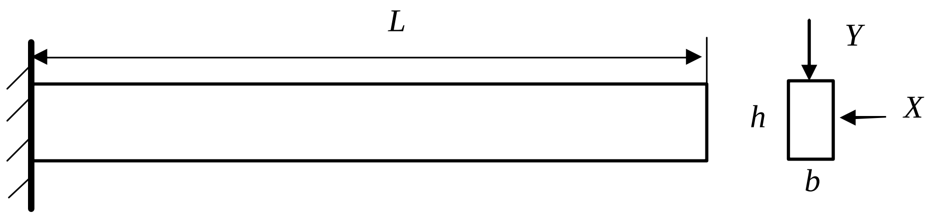

Figure 10.A.8 A Cantilever Beam

The objective of the problem is to minimize the weight or equivalently the cross-sectional area

$$f(\mathbf{d}) = bh,$$

where b and b are width and height of the cross section, respectively, and the design variables are $\mathbf{d} = (b,\ h)$.

Two constraints are considered. The first constraint is that the maximum stress at the fixed end of the cantilever beam is less than the yield strength $S = 35000\,psi$.

$$g_1(\mathbf{d}) = \frac{6L}{bh}(\frac{X}{b} + \frac{Y}{h}) - S \le 0,$$

where $X = 500lb$ and $Y = 1000lb$ are external forces; $L = 100''$ is the length of the beam.

The second constraint is that the tip displacement does not exceed an allowable value D_0,

$$g_2(\mathbf{d}) = \frac{4L^3}{E}\sqrt{\left(\frac{X}{b^3h}\right)^2 + \left(\frac{Y}{bh^3}\right)^2} - D_0 \le 0,$$

where $D_0 = 2.5''$ and $E = 29e6\,psi$ is the material modulus of elasticity.

The bounds for the design variables are $1 \le b \le 10$ and $1 \le h \le 20$, respectively.

The optimization model is then given by

$$\begin{cases} \underset{\mathbf{d}=(b,h)}{\text{minimize}}\ f(\mathbf{d}) \\ \text{subject to} \\ \quad g_1(\mathbf{d}) = \dfrac{6L}{bh}(\dfrac{X}{b}+\dfrac{Y}{h}) - S \le 0 \\ \quad g_2(\mathbf{d}) = \dfrac{4L^3}{E}\sqrt{\left(\dfrac{X}{b^3h}\right)^2+\left(\dfrac{Y}{bh^3}\right)^2} - D_0 \le 0 \\ \quad 1 \le b \le 10 \\ \quad 1 \le h \le 20. \end{cases} \tag{10.17}$$

The starting point is set to $\mathbf{d} = (b,\ h) = (2,\ 2)$ and the results from Matlab are

```
The optimal point = 2.0465      4.0931
The objective function = 8.3766
The constraint functions = -6.0856e-008     -0.3022
```

The optimal point is $\mathbf{d} = (b,\ h) = (2.0465, 4.0931)$, the objective function is $f(\mathbf{d}) = 8.3766$, the constraint functions are $g_1(\mathbf{d}) = \text{-6.0856e-8}$, and $g_2(\mathbf{d}) = \text{-0.3022}$. Since $g_1(\mathbf{d})$ is very close zero, it is an active constraint at the optimal.

The Matlab source codes are given below.

Main function

```
%Example 10.4 in ME301
d0=[2,2]; %starting point, t=h=2;
lb=[1,1]; %lower bounds for design variables;
ub=[10,20]; %upper bounds for design variables;
option=optimset('display','iter'); %set options to show the optimization history
d=fmincon('obj_fun',d0,[],[],[],[],lb,ub,'constr_fun',option); %call the optimizer
%Analysis at the optimal point
t=d(1);
h=d(2);
obj=t*h;
c=constr_fun(d); %calculate the constraint functions
disp(['The optimal point = ',num2str(d)]);
disp(['The objective function = ',num2str(obj)]);
dis p(['The constraint functions = ',num2str(c)]);
```

Objective function

```
function obj=obj_fun(d)
%Objective function
b=d(1);
h=d(2);
obj=b*h;
```

Constraint function

```
function [c,ceq]=constr_fun(d)
%Constraint function
b=d(1);
h=d(2);
X=500;
Y=1000;
E=29E6;
D0=2.5;
L=100;
S=35000;
c(1)=6*L/b/h*(X/b+Y/h)-S;%1st constraint
c(2)=4*L^3/E*((X/b^3/h)^2+(Y/b/h^3)^2)^0.5-D0;%2nd constraint
ceq=[]; %no equality constraint
```