UNIVERSIDAD NACIONAL DE COLOMBIA

Design optimization using Subset Simulation algorithm

by

Steven Vanegas Giraldo

DEPARTMENT OF CIVIL ENGINEERING
FACULTY OF ENGINEERING AND ARCHITECTURE
UNIVERSIDAD NACIONAL DE COLOMBIA
MANIZALES, COLOMBIA
JULY 2022

UNIVERSIDAD NACIONAL DE COLOMBIA

Design optimization using Subset Simulation algorithm

A text submitted in partial fulfillment of requirements for the degree of
CIVIL ENGINEER
in the
UNIVERSIDAD NACIONAL DE COLOMBIA

Author:
Steven Vanegas Giraldo[*]

Advisor:
Doktor der Technischen Wissenschaften
Diego Andrés Álvarez Marín[†]

DEPARTMENT OF CIVIL ENGINEERING
FACULTY OF ENGINEERING AND ARCHITECTURE
UNIVERSIDAD NACIONAL DE COLOMBIA
MANIZALES, COLOMBIA
JULY 2022

[*] stvanegasgi@unal.edu.co
[†] daalvarez@unal.edu.co

Design optimization using subset simulation

by

Steven Vanegas Giraldo

Abstract

In this document the representation of the principal problem of optimization th design of structures like a truss, is an important aspect in the actual representation in the problem about.

The conventional methods could be a problematic in aspect like of multimodal function

Keywords

Subset simulation,

Contents

List of Figures

List of Tables

Chapter 1

Introduction

What do?

A justification about and the objectives.

Qué se hace? Por qué se hace? justificación Objetivos

Chapter 2

Theoretical background

2.1 Preliminaries of optimization theory

In the context of the optimization, there are many methods where they make use of different approaches either in mathematical theory or inspired by some particular phenomenon founded in the world (e.g. animal behaviors). Each method in the spectrum of the optimizations methodologies has many advantages or disadvantages to the different problems that arise in the optimization process. The general diagram about methodologies is listed following.

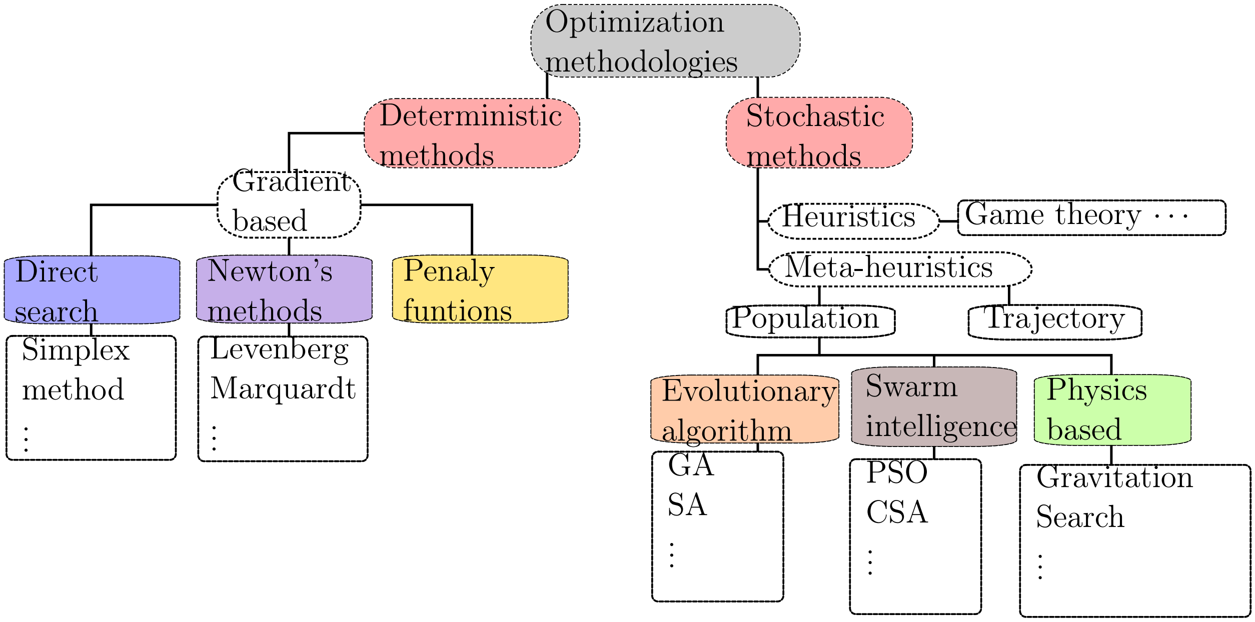

Figure **2.1**: General diagram of some optimization methodologies.

2.1.1 Main problem of the optimization

On a wide range of optimization problems from branches of engineering and science, it's possible to describe with a n-dimensional function ($h : \mathcal{X} \subseteq \Re^n \to \Re$) and the coherent values enclosed in the region $\mathcal{X}$ (variables's boundaries), in a deterministic model. The value of the *objective* or *cost* function $h(\boldsymbol{x})$ is computed by the value of n variables of problem $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ (Bertsekas, 1997). The general minimizing problem with respect to $\boldsymbol{x}$ is represented as

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathcal{X}} \quad & h(\boldsymbol{x}) \\ \text{s.t.} \quad & r_i(\boldsymbol{x}) = 0, \quad i = 1, \ldots, m_e \\ & g_j(\boldsymbol{x}) \leq 0, \quad j = 1, \ldots, m_i \end{aligned} \tag{2.1}$$

The objective function may have a m_e *equality constraints* functions $r_i(\cdot)$ and m_i *inequality constraints* functions $g_j(\cdot)$. When the objective and all the constraints are linear functions, the problem to be *linear programming*, otherwise, is said to be *nonlinear programming*. The *quadratic programming* is a particular case of nonlinear programming, when the cost function is quadratic and all the constraints are linear. The problem with $m_i = m_e = 0$ is unconstrained (Jorge Nocedal, 2006).
The *discrete case* is for all the optimizations problems when the variable $\boldsymbol{x}$ take meaning in it's context with discrete values.

Definition 2.1.1 ***(Feasible set)*** *A point $\boldsymbol{x} \in \Re^n$ is said feasible, if complies all the constraints functions $r_i(\boldsymbol{x})$ and $g_j(\boldsymbol{x})$ and in $\mathcal{X}$. The feasible set X are all the feasible points, that is, $X = \{\boldsymbol{x} \mid r_i(\boldsymbol{x}) = 0,\ \text{for } i = 1, \ldots, m_e;\ g_j(\boldsymbol{x}) \leq 0,\ \text{for } j = 1, \ldots, m_i;\ \boldsymbol{x} \in \mathcal{X}\}$.*

Definition 2.1.2 ***(Local minimum)*** *A feasible point $\boldsymbol{x}^* \in \Re^n$ is named a local minimum of a function $h(\boldsymbol{x})$, if there exist $\delta > 0$ such that $h(\boldsymbol{x}^*) \leq h(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X} \subseteq \Re^n$ satisfying $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \delta$. In the case when $\boldsymbol{x}^* \neq \boldsymbol{x}$, that is, $h(\boldsymbol{x}^*) < h(\boldsymbol{x})$ and satisfy $\|\boldsymbol{x} - \boldsymbol{x}^*\| < \delta$, $\boldsymbol{x}^*$ is called a strict local minimum.*

Definition 2.1.3 ***(Global minimum)*** *A feasible point $\boldsymbol{x}^* \in \Re^n$ is named a global minimum of a function $h(\boldsymbol{x})$, if $h(\boldsymbol{x}^*) \leq h(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X} \subseteq \Re^n$. In the case when $\boldsymbol{x}^* \neq \boldsymbol{x}$, that is, $h(\boldsymbol{x}^*) < h(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X} \subseteq \Re^n$, $\boldsymbol{x}^*$ is called a strict global minimum.*

In this paper the main optimization problem is treated in a minimization concept, however, the problem can be developed from the maximization point of view by setting the objective function by $-h(\boldsymbol{x})$, that is, $\max\limits_{\boldsymbol{x}\in\mathcal{X}} \; -h(\boldsymbol{x})$. An usefull property about the optimization problems is $-\min\limits_{\boldsymbol{x}\in\mathcal{X}} \; h(\boldsymbol{x}) = \max\limits_{\boldsymbol{x}\in\mathcal{X}} \; -h(\boldsymbol{x})$

2.1.2 Some metaheuristic methods

The real world optimizations problems in general have many variables and constraints, objective function behavior tend have severals local optimal, named *multimodal* problems. The traditional methods like the gradient based algorithms have some difficult respect search a global minimum, gradient and Hessian matrix calculations.

Metaheuristic methods are techniques nature-inspired, evolution strategy or physical process. Due to their functional ways based on diversification (exploring the feasible space with several solutions) and intensification (local reginal search where a good solution is found) could be a good option for solving a large multimodal optimizations problems (Yang, 2011). The metaheuristic methods don't guarantee to find the best solution in the feasible region, however, it generates a broader search and an approximate solution in a reduced time.

Some metaheuristic methods are presented briefly.

Particle swarm optimization with flyback technique

The *Particle swarm optimization* (PSO) is a method based on population and swarm intelligence, with influence in bird flocking fish schooling behavior and developed by (Kennedy & Eberhart, 1995). The main idea is to consider points $\boldsymbol{x}^k = [x_1, \ldots, x_n]^T \in \mathcal{X} \subseteq \Re^n$ as positions occupied by particles flying through space with a velocity of $\boldsymbol{V}^k = [V_1, \ldots, V_n]^T$ for one iteration k, the particles will update their position due to the influence of two attractor points, one the best position that has been obtained at the time by the particle $\boldsymbol{P}^k = [P_1, \ldots, P_n]^T$ (*personal knowledge*) and the best position found by the swarm $\boldsymbol{P}_g = [P_{g1}, \ldots, P_{gn}]^T$ (*social knowledge*).

$$V_{j,d}^{k+1} = \underbrace{\omega V_{j,d}^{k}}_{\text{Inertia}} + \underbrace{c_1\phi_1\left(P_{j,d}^{k} - x_{j,d}^{k}\right)}_{\text{Personal knowledge}} + \underbrace{c_2\phi_2\left(P_{g,d} - x_{j,d}^{k}\right)}_{\text{Social knowledge}} \tag{2.2}$$

$$x_{j,d}^{k+1} = x_{j,d}^{k} + V_{j,d}^{k} \tag{2.3}$$

The updates of the velocity and the new position for one dimension d of a given particle j are presented in equations (**2.2**) and (**2.3**) respectively. The inertia weight ω, ϕ_1 and ϕ_2 are two uniform random sequences generated from $\mathsf{U}(0,1)$. The cognitive coefficient c_1 and social coefficient c_2 controls the influences of personal and social knowledge.

The simple methodology *flyback technique* to include constraints in the PSO algorithm is introduced by (He, Prempain, & Wu, 2004) and developed in truss design optimization in (L. Li, Huang, Liu, & Wu, 2007). Consists of return to it's previous position any particle that is not within the feasible region (He et al., 2004). The PSO flyback algorithm in them (L. Li et al., 2007):

Algorithm 1 Particle swarm optimization flyback

```
procedure PSOFLYBACK($h_{\text{objective}}(\cdot), g_{\text{constraint}}(\cdot), N_{\text{particles}}, iter_{max}, c_1, c_2, \omega$)
    Set $k = 0$
    Set $\boldsymbol{P}_g^0 = Inf$
    for Each particle j in the population $N_{particles}$ do
        while (Out side of feasible set) do                    ▷ Satisfy constraints and boundaries
            Generate the current position $\boldsymbol{x}_j$
        end while
        Eval the objective function $h_{\text{objective}}(\boldsymbol{x}_j)$
        Set $\boldsymbol{P}_j^0$, if $\boldsymbol{P}_j^0 < \boldsymbol{P}_g^0$ set $\boldsymbol{P}_g^0 = \boldsymbol{P}_j^0$
    end for
    while $k < iter_{max}$ do                                   ▷ Termination condition
        for Each particle j in the population $N_{particles}$ do
            Update the velocity with equation (2.3) $\boldsymbol{V}_j^{k+1}$
            Update the position with equation (2.3) $\boldsymbol{x}_j^{k+1}$
            Eval $g_{\text{constraint}}(\boldsymbol{x}_j^{k+1})$
            If $\boldsymbol{x}_j^{k+1}$ is not in a feasible set, set $\boldsymbol{x}_j^{k+1} = \boldsymbol{x}_j^k$   ▷ Flyback technique
            Eval $h_{\text{objective}}(\boldsymbol{x}_j^{k+1})$
            If $f_{\text{objective}}(\boldsymbol{x}_j^{k+1}) < \boldsymbol{P}_j^k$, set $\boldsymbol{P}_j^k = h_{\text{objective}}(\boldsymbol{x}_j^{k+1})$
        end for
        Set $\boldsymbol{P_g} = \min\{\boldsymbol{P}_1^k, \ldots, \boldsymbol{P}_N^k\}$
    end while
    return $\boldsymbol{P_g}$ and $h_{\text{objective}}\boldsymbol{P_g}$
end procedure
```

2.2 Preliminaries of probability theory

In science we have two types of models of the world with phenomena, the certain or rational ones, where the result can be obtained a priori (e.g., determining a current through Ohm's law) and the random phenomena, where it is not certain what result can be obtained (e.g., rolling a die). In the following section are the preliminaries about some concept in the probability theory, which are part of the tools that study random phenomena. Andrey Nikolaevich Kolmogorov was a soviet mathematician, who systematized the current concepts of probability in the book (Kolmogorov, Morrison, & Bharucha-Reid, 1956), integration of set theory and axioms system. The section is centering in the following bibliography (Taboga, 2017), (Bertsekas & Tsitsiklis, 2008), (Leon-Garcia, 2008), (Rubinstein & Kroese, 2016), (Chan, 2021) and (Ghahramani, 2004).

2.2.1 Some initial definitions

Definition 2.2.1 ***(Experiment)*** *An* experiment *is a situation where don't have certain what could be the outcome (e.g., rolling a die). The experiment has the following features:*

i). All the possible outcomes ares know a priori.

ii). The experiment could be repeated under the same conditions.

iii). It's not certain what result can be obtained.

Definition 2.2.2 ***(Sample space Ω)*** *A* sample space *of a experiment, is a set Ω that contains all the possibles outcomes of that experiment. Usually an element (outcome) ω of the sample space is denoted by low case grek letters (i.g., ω, θ, ...).*

Definition 2.2.3 ***(Event)*** *An* event *is a subset $\mathcal{E}$ in the sample space Ω.*

Definition 2.2.4 ***(Mutually exlusive events)*** *A* collection of events $\{\mathcal{A}_1, \ \ldots, \ \mathcal{A}_n\}$ *are mutually exclusive if for each pairwise of events their interception is empty, that is, $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$, for all $i \neq j$ (i.e., the occurrence of i-th event don't implies the occurrence of the rest events of collection, are disjoint).*

Definition 2.2.5 ***(Collectively exhaustive events)*** *A* collection of events $\{\mathcal{A}_1, \ \ldots, \ \mathcal{A}_n\}$ *is said they are collectively exhaustive if their union is the sample space Ω, i.e. $\bigcup_{i=1}^{n} \mathcal{A}_i = \Omega$.*

For the elements of the sample space Ω, ***mutually exclusive*** means that an unique outcome of a try is possible and the ***collectively exhausty*** means that for a try of the experiment, the outcome obtained is in the sample space.

Definition 2.2.6 ***(Event space)*** *A event space $\mathscr{F}$ is named a power set and σ-field of a sample space Ω of a experiment.*

Definition 2.2.7 ***(A probability law)*** *A law or mesure of probability of a experiment, denoted by $\mathbb{P}(\cdot)$, is a function with domain a event space $\mathscr{F}$ and codomain some real value in the interval $[0, 1]$, $\mathbb{P} : \mathscr{F} \to [0, 1]$, and also satisfy the following proprieties* [*]*:*

[*] The three proprieties are know how the axioms of probability

*i). **(Normalization)**: The probability of sample space Ω is equal to* 1*,* $\mathbb{P}(\Omega) = 1$*.*

*ii). **(Non-negativity)**: The probability of all element $\mathcal{E}$ that belongs to the event space $\mathscr{F}$ is greater than or equal to* 0*,* $\mathbb{P}(\mathcal{E}) \geq 0$*.*

*iii). **(Additivity)**: For a sequence of events $\{\mathcal{A}_i\}$ for $i = 1, 2, \ldots,$ mutually disjoint events, (i.e., $\mathcal{E}_j \bigcap \mathcal{E}_k = \emptyset$ for $j \neq k$), satisfy:*

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} \mathcal{E}_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(\mathcal{E}_i)$$

Definition 2.2.8 ***(A probability space)*** *A probability space of a experiment, denoted by $(\Omega, \mathscr{F}, \mathbb{P})$, consists a sample space Ω, event space $\mathscr{F}$ and a law o measure probability $\mathbb{P}(\cdot)$.*

2.2.2 Conditional probability

The conditional probability is an important subject in the probability theory. Allowed compute probabilities when is know partial information about an event of the experiment (e.g., the certain about a test was negative (false alarm)).

Definition 2.2.9 ***(Conditional probability)*** *given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and the event $\mathcal{D}$, with probablity mesasure non negative* $\mathbb{P}(D) > 0$ *belongs to event space. Is said a conditional probability $\mathbb{P}(C|D)$ in arbitrary element $\mathcal{C}$ in event space.*

$$\mathbb{P}(\mathcal{C}|\mathcal{D}) = \frac{\mathbb{P}(\mathcal{C} \cap \mathcal{D})}{\mathbb{P}(\mathcal{D})}$$

The expression $\mathbb{P}(C|D)$ is read as the probability of event $\mathbb{P}(C)$ given that event $\mathbb{P}(D)$ occurred. The conditional probability complies with the 3 axioms of probability (i.e., it's a law or measure of probability). When the probability of the event $\mathcal{D}$, is equal to 0 is said that the conditional probability is not define.

Definition 2.2.10 ***(Multiplication rule)*** *given a probability space $(\Omega, \mathscr{F}, \mathbb{P})$ and a sequence of event $\{\mathcal{A}_1, \ldots, \mathcal{A}_n\}$ belongs to event space, with* $\mathbb{P}\left(\cap_{j=1}^{n-1} \mathcal{A}_j\right) > 0$ *for all* $j = 1, \ldots, n-1$*, satisfy:*

$$\mathbb{P}\left(\bigcap_{i=1}^{n} \mathcal{A}_i\right) = \mathbb{P}(\mathcal{A}_1) \prod_{j=2}^{n} \mathbb{P}\left(\mathcal{A}_j \middle| \bigcap_{k=1}^{j-1} \mathcal{A}_k\right) = \mathbb{P}(\mathcal{A}_1)\, \mathbb{P}(\mathcal{A}_2|\mathcal{A}_1) \cdots \mathbb{P}\left(\mathcal{A}_n \middle| \bigcap_{j=1}^{n-1} \mathcal{A}_j\right)$$

2.2.3 Random variables

Definition 2.2.11 ***(Random variable)*** *given an experiment with a probability space denoted by* $(\Omega, \mathscr{F}, \mathbb{P})$*, is said a random variable* X *a function maps an outcome belongs to the sample space to a real number* $\Re$*,* $X : \Omega \rightarrow \Re$*. Usually a random variables are denoted by capital letters (e.g.,* X*,* Y*,* X_1*,* X_2*, ...). There are two types of random variables:*

i). ***Discrete random variable*** *when* X *has a countable range (finite or infinite), is said that* X *is a discrete random variable.*

ii). ***Continuous random variable*** *when* X *has a uncountable range, is said that* X *is a continuous random variable* [*]*.*

Note that a function of a randon variable $g(X)$ is also a randon variable, $Y = g(X)$ where $g : \Re \rightarrow \Re$. The numerical value tha get a random variable is denoted by lower case letter (i.g., x, y, ...). In some case, a experiment with a sample space Ω could be describe by multiples random variables a whose value depends on the outcome of the experiment. A easy form to describe these random variables is with the help of ***random vectors***.

Definition 2.2.12 ***(Random vector)*** *given a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *with a sample space* Ω*. A random vector* $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$*, with some value* $\boldsymbol{x} = [x_1, x_2, \ldots, x_r]$*, is a function with domain the sample space to the* r*-dimensional real vector* $\Re^r$*,* $\mathbf{X} : \Omega \rightarrow \Re^r$*.*

2.2.4 Mass and density function

One of the ways to characterize random variables, discrete or continuous, is through the probability of the values that such variable could take, assimilated as events. Through the mass function and probability density it's possible to perform this characterization.

Definition 2.2.13 ***(Probability mass function (PMF))*** *given a discrete random variable* X*, (PMF) is a real-value function* $p_X(\cdot)$ *define in* $\Re$ *,* $p_X : \Re \rightarrow [0, 1]$*, that specifies the probability of obtaining the event* $\{X(\omega) = x\}$*.*

$$\mathbb{P}\left(\{X = x\}\right) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) = x\}\right) = p_X(x)$$

[*] In the case the continuous randon variables, the event space must be satisfy the special characteristic of a Borel field (is a σ-field $\mathcal{B}\,(-\infty, b] = \{x : -\infty < x \leq b\}$), which it could generated a unions, intersections and complements of intervals of the form $(-\infty, b]$, (Leon-Garcia, 2008, page 75)

Definition 2.2.14 ***(Probability density function (PDF))*** *is a real-value function $f_X(\cdot)$ define in $\Re$, $f_X : \Re \to [0, \infty)$, that maps the density of probability to obtaining the event $\{X \in [a, b]\}$.*

$$\mathbb{P}\left(\{X \in [a, b]\}\right) = \mathbb{P}\left(\{\omega \in \Omega | X(\omega) \in [a, b]\}\right) = \int_a^b f_X(x)dx$$

The notation $X \sim f$ or $X \sim p$ means that the random variable X is distributed according to (PDF) or (PMF) (e.g., $X \sim \mathsf{N}$ the random variable X has a normal distribution).

2.2.5 Joint probability mass and density function

A group random variables $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$ associate with a same experiment could be characterize by a joint probability mass function (for a discrete case) , or by a joint probability density function (for a continuous case).

Definition 2.2.15 ***(Joint probability mass function (Joint PMF))*** *given a collection of discrete random variables denoted by a random vector $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$ that describe a same experiment with a sample space Ω, is a function $p_{\mathbf{X}}(\boldsymbol{x})$ [*] in the r-dimensional real vector $\Re^r$ to interval $[0, 1]$, $p_{\mathbf{X}} : \Re^r \to [0, 1]$, that describe a probability of the event of $\mathbf{X} = \boldsymbol{x}$ take values $(\{X_1 = x_1\} \cap \{X_2 = x_2\} \cap \cdots \cap \{X_r = x_r\})$.*

$$\mathbb{P}\left(\{X_1 = x_1\} \cap \{X_2 = x_2\} \cap \cdots \cap \{X_r = x_r\}\right) = p_{\mathbf{X}}(\boldsymbol{x})$$

Definition 2.2.16 ***(Joint probability density function (Joint PDF))*** *given a collection of continuous random variables denoted by a random vector $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$ that describe a same experiment with a sample space Ω, is a function $f_{\mathbf{X}}(\boldsymbol{x})$ [†] in the r-dimensional real vector $\Re^r$ to interval $[0, 1]$, $f_{\mathbf{X}} : \Re^r \to [0, 1]$, that describe a probability of the randon vector $\mathbf{X}$ in the region $\mathcal{G}$ a subset of $\Re^r$. The region $\mathcal{G}$ is equal to $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_r, b_r]$.*

$$\mathbb{P}\left(\mathbf{X} \in \mathcal{G}\right) = \int_{a_1}^{b_1} \cdots \int_{a_r}^{b_r} f_{\mathbf{X}}(\boldsymbol{x})dx_r \cdots dx_1$$

2.2.6 Marginal functions

Definition 2.2.17 ***Marginal PMF*** *given collection of discrete random variables denoted by a random vector $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$ and the joint (PMF) $p_{\mathbf{X}}(\boldsymbol{x})$ that describe a same*

[*] The notation $p_{\mathbf{X}}(\boldsymbol{x}) = p_{X_1, X_2, \ldots, X_r}(x_1, x_2, \ldots, x_r)$

[†] The notation $f_{\mathbf{X}}(\boldsymbol{x}) = f_{X_1, X_2, \ldots, X_r}(x_1, x_2, \ldots, x_r)$

experiment with a sample space Ω. The marginal PMF of the random vector $\mathbf{X}$ is define by:

$$p_{X_i}(x) = \sum_{(x_1,\ldots,x_r)\in R_x : x_i = x} p_{\mathbf{X}}(\boldsymbol{x})$$

Where $R_{\mathbf{X}}$ is a set of the all possible vector that the random variable $\mathbf{X}$ could be take.

Definition 2.2.18 ***Marginal PDF*** *given collection of continuous random variables denoted by a random vector $\mathbf{X} = [X_1, X_2, \ldots, X_r]^T$ and the joint (PDF) $f_{\mathbf{X}}(\boldsymbol{x})$ that describe a same experiment with a sample space Ω. The marginal PDF of the random vector $\mathbf{X}$ is define by:*

$$f_{X_i}(x_i) = \underbrace{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty}}_{(r-1)times} f_{\mathbf{X}}(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_r) dx_r \ldots dx_{i+1} dx_{i-1} \ldots dx_1$$

2.2.7 Conditional (PMF) and (PDF)

The conditional also can be applice to the (PMF) and the (PDF).

Definition 2.2.19 ***(Conditional probability mass function)*** *given two discrete random variables X and Y with a joint (PMF) $p_{X,Y}(x,y)$ and marginal (PMF) of Y $p_Y(y)$, is said that the function $p_{X|Y=y} : \Re \to [0,1]$ is a conditional probability mass function of X given $Y = y$ if:*

$$p_{X|Y=y}(x|y) = \frac{\mathbb{P}\left(\{X = x\}, \{Y = y\}\right)}{\mathbb{P}\left(\{Y = y\}\right)} = \frac{p_{X,Y}(x,y)}{p_Y(y)}$$

Definition 2.2.20 ***(Conditional probability density function)*** *given two continuous random variables X and Y with a joint (PDF) $f_{X,Y}(x,y)$ and marginal (PDF) of variable Y $f_Y(y)$, is said that the function $f_{X|Y=y} : \Re \to [0,1]$ is a conditional probability density function of X given $Y = y$ if:*

$$f_{X|Y=y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

2.3 Independence

The independence of two event $\mathcal{C}$ and $\mathcal{D}$ is developed intuitive when the additional information about the event $\mathcal{D}$ (occurred or not occurred) not influence in the occurrence of the event $\mathcal{C}$ and vice versa (i.e., $\mathbb{P}\left(\mathcal{C}|\mathcal{D}\right) = \mathbb{P}\left(\mathcal{C}\right)$ and $\mathbb{P}\left(\mathcal{D}|\mathcal{C}\right) = \mathbb{P}\left(\mathcal{D}\right)$).

Definition 2.3.1 ***(Independece of two event)*** *given a probability space* $(\Omega, \mathscr{F}, \mathbb{P})$ *and two events* $\mathcal{C}$ *and* $\mathcal{D}$ *that belongs to the event space* $\mathscr{F}$*, the two events are said independence event if satisfy the following* [†]*:*

$$\mathbb{P}\left(\mathcal{C} \cap \mathcal{D}\right) = \mathbb{P}\left(\mathcal{C}\right) \mathbb{P}\left(\mathcal{D}\right)$$

Definition 2.3.2 ***(Mutually independece of collection of events)*** *given a probability space denoted by* $(\Omega, \mathscr{F}, \mathbb{P})$ *and a collection of events* $\{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_n\}$ *belongs to the event space* $\mathscr{F}$*, are said independence events if satisfy for each subset* $\mathcal{S}$ *of* $\{1, 2, \ldots, n\}$*:*

$$\mathbb{P}\left(\bigcup_{i \in \mathcal{S}} \mathcal{A}_i\right) = \prod_{i \in \mathcal{S}} \mathbb{P}\left(\mathcal{A}_i\right)$$

Definition 2.3.3 ***(Mutually independece of random variables)*** *given a collection of discrete or continuous random variables, denoted by a randon vector* $\mathbf{X} = [X_1, X_2, \ldots, X_n]$*, with a joint (PMF)* $p_{\mathbf{X}}(\boldsymbol{x})$ *or (PDF)* $f_{\mathbf{X}}(\boldsymbol{x})$*, is said that the variables* $X_1, X_2, \ldots, X_n$ *are mutually independence if:*

$$\begin{aligned} p_{\mathbf{X}}(\boldsymbol{x}) &= \prod_{i=1}^{n} p_{X_i}(x_i) \quad \textit{for a discrete case} \\ f_{\mathbf{X}}(\boldsymbol{x}) &= \prod_{i=1}^{n} f_{X_i}(x_i) \quad \textit{for a continuous case} \end{aligned}$$

Where $p_{X_i}(x_i)$ is the marginal (PMf) of the random variable X_i and $f_{X_i}(x_i)$ is the marginal (PDF) of the random variable X_i.

[†] The situation of $\mathbb{P}\left(\mathcal{C} \cap \mathcal{D}\right) = 0$ not implicate that the two event are independence

Chapter 3

Subset Simulation for design optimization

The *subset simulation* (SS) technique is well known in the area of extreme case and analysis of reliability of structures. The subset simulation methodologies has promising results in the calculation of small failure probabilities.

The basic of subset simulation method consists in in transform a small failure probability into a product of a number of not-so-small conditional failure probabilities (Wang, 2014) and the generate sample through *Markov Chain Monte Carlo* (MCMC) simulation for computing the conditional failure probabilities. This idea can be transposed to the field of optimization in that finding a minimum or a maximum can be considered as an extreme case for some function $h(\boldsymbol{x})$.

3.1 Optimization problem into the (SS) approach

The expression (**2.1**) in the optimization problem, considers the function $h(\boldsymbol{x})$ how a deterministic model about some process in the real word for some variable $\boldsymbol{x} = [x_1, \ldots, x_n]^T$. Changing the deterministic model into probability point of view, transforms the nature variables into a random variables $\boldsymbol{X} = [X_1, \ldots, X_n]^T$, where the randon variables $\boldsymbol{X}$ has their own probability density function $f_{\boldsymbol{X}}(\boldsymbol{x})$ (i.e., joint probability density function) and this convert the value function in a random variables (i.e., $h(\boldsymbol{x})$ is a random phenomena) characterized by some (PDF) $f_{\boldsymbol{H}}(\boldsymbol{x})$. In the terminology of probability, the algorithm proposed in (H.-S. Li & Au, 2010) may be applicable into a main problem of optimization in (**2.1**). Giving that $h(\boldsymbol{x})$ is a random variable, you can define for some threshold h what is the probability that the function takes a value less than h, that is, the region of interest is define by $F = \{\boldsymbol{x} \mid h(\boldsymbol{x}) < h\}$ express in the integral formulation by

$$\mathbb{P}\left(F\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h\right) = \int_F f_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{X} \qquad \textbf{(3.1)}$$

Once built the probability model, considered a value function in global minimum h^* of the cost function $h(\boldsymbol{x})$ and the event denoted by the region $F = \{\boldsymbol{x} \mid h(\boldsymbol{x}) < h^*\}$. For computes the probability of the event F (i.e. $\mathbb{P}\left(F\right)$), a construction of intermediate events such that $F_1 \supset \cdots F_i \cdots \supset F_m = F$ can be used to decomposing the main region in frequent events $F_i = \{\boldsymbol{x} \mid h(\boldsymbol{x}) < h_i\}$ for some sequence of thresholds $h_1 > \cdots > h_i > \cdots > h_m$ (see Fig. 3.1 for schematic two dimensional case).

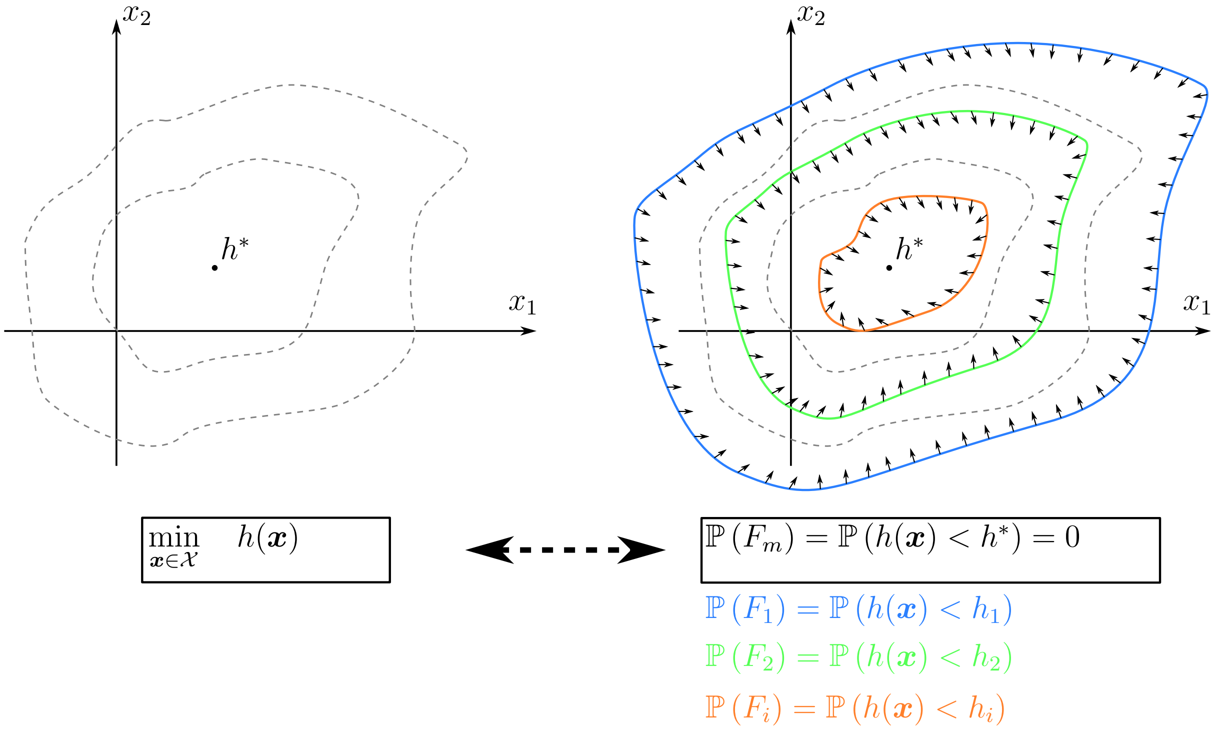
x_2
x_1
h^*
x_2
x_1
h^*
$\min_{\boldsymbol{x} \in \mathcal{X}} \quad h(\boldsymbol{x})$
$\mathbb{P}\left(F_m\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h^*\right) = 0$
$\mathbb{P}\left(F_1\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h_1\right)$
$\mathbb{P}\left(F_2\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h_2\right)$
$\mathbb{P}\left(F_i\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h_i\right)$

Figure **3.1**: Transformation between optimization main problem and probability model.

Note that the $\mathbb{P}\left(F_m\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h^*\right) = \mathbb{P}\left(h(\boldsymbol{x}) < h_m\right) = 0$ (h^* it's a global minimum) means that no value of $\boldsymbol{x}$ has a function value less than h^*, the probability $\mathbb{P}\left(F_m\right)$ can be expressed into a conditional probabilities using multiplication rule 2.2.10 by

$$\mathbb{P}\left(F_m\right) = \mathbb{P}\left(\bigcap_{i=1}^{m} F_i\right) = \mathbb{P}\left(F_1\right)\prod_{j=2}^{m}\mathbb{P}\left(F_j \middle| \bigcap_{k=1}^{j-1} F_k\right) = \mathbb{P}\left(F_1\right)\mathbb{P}\left(F_2|F_1\right)\cdots\mathbb{P}\left(F_n \middle| \bigcap_{k=1}^{m-1} F_k\right) \tag{3.2}$$

Combining the sequence of calculations from each term in (3.2), that is, each conditional probability $\mathbb{P}\left(F_i \mid F_{i-1}\right)$ for $i > 1$ and the probability $\mathbb{P}\left(F_1\right)$ of the first region or *first level*, give the result that the probability of the random variable $h(\boldsymbol{x})$ can take a value less that the global minimum function h^*. For the computing the probability of the event F_m in (3.2) we can use simulations by generating samples from conditional probabilities. The (SS) use the advantage of a Markov Chain Monte Carlo methods efficiently and from arbitrary (PDF).

It should be noted that the thresholds $h_1 > \cdots > h_i > \cdots > h_m$ cannot be known apriori, however, they can be estimated as simulations are carried out to obtain samples closer and closer to a region where the global minimum is found.

3.1.1 Constraints-handling

In variable design problems it's common to find that the problem is specified as in (2.1). However, the constraints are only detailed by m_i inequality constraints $g_j(\boldsymbol{x})$.

The handling of the constraints in the optimization problem must be adapted to the new approach that was appropriate for the (SS) method. (H.-S. Li & Au, 2010) adopt a technique that use *constraint fitness function* and *constraint violation function.*

To generate samples in the feasible region in a search space, it is important to know when a sample is violating a constraint and to quantify how severe this violation is. The constraint fitness function helps to developed that task, whose function is giving by

$$F_{con}(\boldsymbol{x}) = -\max_{j} \quad v_j(\boldsymbol{x}) \tag{3.3}$$

Using the auxiliary constraint violation function to discern for a j constraint the magnitude of violation, that is

$$v_i(\boldsymbol{x}) = \begin{cases} 0 & \text{if} \quad g_i(\boldsymbol{x}) \leq 0 \\ g_i(\boldsymbol{x}) & \text{if} \quad g_i(\boldsymbol{x}) > 0 \end{cases} \tag{3.4}$$

In (3.3) give for all sample $\boldsymbol{x}$ that complies all the constraints of the design optimization a value of zero, that is, $F_{con}(\boldsymbol{x}) = 0$. The worst constraint violation has the most negative value, otherwise, the value is closer to zero.

3.2 Generating samples

The calculation of the probability $\mathbb{P}\left(F_m\right)$ in terms of a computer simulation can be a very small quantity and a simulation through a *Direct Monte Carlo Simulation* (DMCS) would not generate sufficient samples to calculate this probability with precision. The reason for separating the problem to conditional probabilities where their event is more frequent and therefore can be evaluated in a more efficient way.

For each step of simulation, or *level*, the samples are getting closer and closer to being near a global optimum, that is, $h_i \to h^*$ and $\mathbb{P}\left(F_m\right) \to 0$. It is not known in advance how many level are needed to perform the calculation in (3.2) (i.e., what is the value of m), but it is possible to estimate them as simulations and sample generation are carried out.

3.2.1 Double-criterion sorting algorithm

In the proposal algorithm in (H.-S. Li & Au, 2010), the *double-criterion sorting* plays an important role in the selection of *seeds* for sample generation at the following levels and use two criterions for sorted the samples generated, both the value performance of the cost function and the violation of the constraints are taken into account.

Considered a set of N samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$. The first priority is the feasible samples in the top of the list, sorting by the value of fitness constraint function (3.3), the list can be constructed as

$$\underbrace{F_{con}(\boldsymbol{x}_1)}_{\text{Best value}} \geq F_{con}(\boldsymbol{x}_2) \geq \cdots \geq \underbrace{F_{con}(\boldsymbol{x}_N)}_{\text{Worst value}} \tag{3.5}$$

Them, only the feasible samples are sorted by their objective function value (i.e., the values of $F_{con}(\boldsymbol{x}) = 0$), looking for a less values function first (minimization case). The infeasible samples aren't sorted.

3.2.2 Initial considerations for simulation

For the generating samples, it assumed that the set of random variables $\{X_1, \ldots, X_n\}$ are mutually independent, that is, their (JPDF) are equal to $f_{\boldsymbol{X}} = f_{X_1}(x_1) \ldots f_{X_n}(x_n)$ and select the appropriate level probability p_0.

Consider N the number of samples for level and *level probability* $p_k \in (0, 1)$ for the different levels $k = 1, 2, \ldots, m$.

The length of the Markov Chain expressed by $N_{c_k} = p_{k-1} * N$ and the number of samples per chain is denoted by $N_{s_k} = 1/p_{k-1}$ for a level k. Note that the choice of the level probability should be such that the length and the number of samples per string should be positive integers. Each sample within the chain must generate the number of samples specified by N_{s_k}, therefore the product $N_{s_k} N_{c_k}$ of must equal the number of samples per level N.

3.2.3 Direct Monte Carlo Simulation (DMCS) level 0

In the initial level, level 0, the samples generating is do it by the direct Monte Carlo simulation. Each sample $\boldsymbol{x}_j$ in the set of N independent samples $\{\boldsymbol{x}_{1,(0)}, \boldsymbol{x}_{2,(0)}, \ldots, \boldsymbol{x}_{j,(0)}, \ldots, \boldsymbol{x}_{N,(0)}\}$, has n independent components (i.e., $\boldsymbol{x}_{j,(0)} = \{x^{(1)}_{j,(0)}, x^{(2)}_{j,(0)}, \ldots x^{(n)}_{j,(0)}\}$) distributed according with the (PDFs) $f_{X_i}(x_i)$ for $i = 1, \ldots, n$, subindex (0) indicate the level 0.

In the level 0, because the samples are generated without taking into account the value of the constraint fitness function, some samples at this level may not be in the feasible region.

For each sample in the set, evaluated the constraint and cost function of the problem. Use the double-criterion for sort the samples. Set the thresholds $h_1 = h(\boldsymbol{x}_{Np_0})$ and $F_{c_1} = F_{con}(\boldsymbol{x}_{Np_0})$ attributed to the (Np_0)th sample, $\boldsymbol{x}_{Np_0}$.

Select the region of interest F_1 (i.e., the first event), represented as

$$F_1 = \begin{cases} \{h(\boldsymbol{x}_{j,(0)}) \leq h_1\} & \text{if} \quad F_{c_1} = 0 \\ \{F_{con}(\boldsymbol{x}_{j,(0)}) \geq F_{c_1}\} & \text{if} \quad F_{c_1} < 0 \end{cases} \qquad \textbf{(3.6)}$$

For construction the samples in the event F_1 are the first Np_0th samples. These samples will be the seeds in the generation of $N(1 - p_0)$ samples for the next level, that is

$$\{\underbrace{\boldsymbol{x}_{1,(0)}, \boldsymbol{x}_{2,(0)}, \ldots, \boldsymbol{x}_{Np_0,(0)}}_{\text{seeds level 1}}, \underbrace{\ldots, \boldsymbol{x}_{j,(0)}, \ldots \boldsymbol{x}_{N,(0)}}_{\text{samples to be generated}}\} \qquad \textbf{(3.7)}$$

The term $\mathbb{P}\left(F_1\right)$ in (3.2), the first level, can be compute using

$$\mathbb{P}\left(F_1\right) \approx p_0 = \frac{1}{N}\sum_{s=1}^{N} I_{F_1}(\boldsymbol{x}_{s,(0)}) \tag{3.8}$$

Where the *indicator function* I_{F_1} take a value of 1 when $\boldsymbol{x}_{s,(0)} \in F_1$, otherwise, is zero.

3.2.4 Modified Metropolis-Hasting algorithm (MMH)

For generating samples for the next conditional probabilities $\{\mathbb{P}\left(F_{k+1}|F_k\right)\}$, for $k \geq 1$, is used the algorithm *Modified Metropolis-Hasting algorithm* (MMH) used in (Au & Beck, 2001), (Au & Beck, 2003), that is, a method included into the methods Markov Chain Monte Carlo simulation allow generated samples according to an arbitrary probability distribution.

The Modified Metropolis-Hasting algorithm is a variant of the *Metropolis-Hasting* (MH), solving the problems of the classic (MH) for generate when simulating random vectors with many independent components (Au & Beck, 2001).

The simulation of Markov Chain Monte Carlo gives the $N(1-p_{k-1})$th samples, where, the first $N * p_{k-1}$th samples are the length of the chain and the seeds for the generation of conditional samples at next simulation level. The length of the chain for a level, N_{c_k}, with N_{s_k} samples for each sample in the chain, is a strategy, where, the ergodic problems are less than if the Nth samples chain were generated (Wang, 2014).

The region of interest for the simulation level $k = 1, \ldots, m-1$ in the (MCMC), is

$$F_k = \begin{cases} \{h(\boldsymbol{x}_{j,(k-1)}) \leq h_k\} & \text{if} \quad F_{c_k} = 0 \\ \{F_{con}(\boldsymbol{x}_{j,(k-1)}) \geq F_{c_k}\} & \text{if} \quad F_{c_k} < 0 \end{cases} \tag{3.9}$$

And the thresholds are equal $h_k = h(\boldsymbol{x}_{Np_{k-1}})$ for the function value and $F_{c_k} = F_{con}(\boldsymbol{x}_{Np_{k-1}})$ attributed to the $\boldsymbol{x}_{Np_{k-1}}$ sample. The conditionals probabilities for the differents level (i.e., the terms $\mathbb{P}\left(F_{k+1}|k\right)$) in the (3.2), are computed by

$$\mathbb{P}\left(F_{k+1}|F_k\right) \approx p_k = \frac{1}{N}\sum_{s=1}^{N} I_{F_{k+1}}(\boldsymbol{x}_{s,(k)}) \tag{3.10}$$

The strategy for determining the regions F_k for $k = 1, \ldots, m-1$ are consistent with the double-criteria sorting methodology, looking first for the feasible samples or at least for those with the lowest value of constraint fitness function.

The (MMH) method used the (PDF) of the variables for n components, that is, $f_{X_i}(\cdot)$ and a proposal (PDF) $p^*_{X_i}(\cdot \mid \cdot)$ for $i = 1, \dots, n$ with easy sample generation. The pseudo code for (MMH) method is the following

Algorithm 2 Modified Metropolis-Hasting method

```
procedure MMH($\boldsymbol{x}_{j,(k)}, \{f_{X_i}(\cdot)\}, \{p^*_{X_i}(\cdot \mid \cdot)\}, F_k$)
    Generate a candidate $\boldsymbol{\xi}_{(k+1)} = [\xi^{(1)}_{(k+1)}, \xi^{(2)}_{(k+1)}, \dots, \xi^{(n)}_{(k+1)}]^T$
    for Each dimension $i = 1 \dots, n$ do
        Generate a candidate component $\xi^{(i)}_{(k+1)} \sim p^*\left(\cdot \mid x^{(i)}_{j,(k)}\right)$
        Eval the acceptance ratio $r^{(i)}_{k+1} = \frac{f_{X_i}\left(\xi^{(i)}_{(k+1)}\right) p^*\left(x^{(i)}_{j,(k)} \middle| \xi^{(i)}_{(k+1)}\right)}{f_{X_i}\left(x^{(i)}_{j,(k)}\right) p^*\left(\xi^{(i)}_{(k+1)} \middle| x^{(i)}_{j,(k)}\right)}$
        Set $u = \mathsf{U}(0,1)$
        if $u \leq \min\left(r^{(i)}_{k+1}, 1\right)$ then
            Set the $i$th component of $\boldsymbol{\xi}_{(k+1)}$ equal to $\xi^{(i)}_{(k+1)}$
        else
            Set the $i$th component of $\boldsymbol{\xi}_{(k+1)}$ equal to $x^{(i)}_{(k+1)}$
        end if
    end for
    Accept or reject the candidate
    if $\boldsymbol{\xi}_{(k+1)} \in F_k$ then
        Accept it as the next sample, that is, $\boldsymbol{x}_{(k+1)} = \boldsymbol{\xi}_{(k+1)}$
    else
        Reject it and take the current seeds, that is, $\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{j,(k)}$
    end if
    return The new sample $\boldsymbol{x}_{(k+1)}$
end procedure
```

3.3 The proposal algorithm

The proposal algorithm in (H.-S. Li & Au, 2010) use the subset simulation technique for solving the optimization problem, specially the design optimization.

3.3.1 Artificial (PDF) for design variables

The set of (PDFs) $\{f_{X_i}\}$ with $i = 1, \dots, n$ for n components, that characterizes the random variables $\boldsymbol{X} = [X_1, \dots, X_n]^T$, are selected by

$$f(x^{(i)};\mu^{(i)},\sigma^{(i)},x_l^{(i)},x_u^{(i)}) = \frac{\phi\left(\frac{x^{(i)}-\mu}{\sigma}\right)}{\Phi\left(\frac{x_u^{(i)}-\mu}{\sigma}\right)-\Phi\left(\frac{x_l^{(i)}-\mu}{\sigma}\right)} \tag{3.11}$$

Where $\phi(\cdot)$ is the *standard Gaussian* distribution and $\Phi(\cdot)$ is the cumulative distribution function. The equation (**3.11**) shows the *truncated normal distribution*, and it's value is within the interval determined by it's lower $x_l^{(i)}$ and upper $x_u^{(i)}$ limits. The mean $\mu^{(i)}$ and standard deviation $\sigma^{(i)}$ affects the search domain. If no are available information in advance about the possible region where the global minimum is located, the mean can be placed in the center of the definition bounds (H.-S. Li & Au, 2010), that is, $\mu^{(i)} = \frac{x_l^{(i)}+x_u^{(i)}}{2}$.

The standard deviation of the function in (**3.11**) controls the range to be explored and it has an influence on the efficiency (H.-S. Li & Au, 2010). The size of the exploration region is proportional to the magnitude of $\sigma^{(i)}$ (i.e., a large value of standard deviation develops a wide dispersion of samples, otherwise, a small value of standard deviation develops a localized distribution). The strategy used in (H.-S. Li & Au, 2010) adopts the three sigma applied in engineering, that is, $\sigma^{(i)} = \frac{\left|x_u^{(i)}-x_l^{(i)}\right|}{6}$ for each component. A wide scan given by a large value of the standard deviation will require more levels to converge.

3.3.2 Convergence criterion

The criterion of convergence is based in the standard deviation for a dimension (i) of the samples $\widehat{\sigma}_k^{(i)}$ for a level k (H.-S. Li & Au, 2010). Due to a large difference in the order of magnitude that some variables may have, it is preferable to use a convergence criterion taking into account the n boundaries of these variables as follows

$$\max\left(\left\{\sigma_{k,\text{bouds}}^{\widehat{(1)}},\ldots,\sigma_{k,\text{bouds}}^{\widehat{(n)}}\right\}\right) = \max\left(\left\{\left|\frac{\sigma_k^{\widehat{(1)}}}{x_u^{(1)}-x_l^{(1)}}\right|,\ldots,\left|\frac{\sigma_k^{\widehat{(n)}}}{x_u^{(1)}-x_l^{(n)}}\right|\right\}\right) \leq \varepsilon \tag{3.12}$$

For any tolerance value ε established by the user. When the algorithm converge, them the standard deviation of the samples for component tends to zero. An additional stopping criterion, is to limit the number of level simulations by a maximum number of k_{max}.

3.3.3 Level probability

The probability level p_k has an important influence for the convergence when the (SS) method is used for optimization. Is adopted in (H.-S. Li & Au, 2010) a determination of the probability of level depending on the standard deviation of the samples calculated as follows

$$p_k = \begin{cases} p_0 = 0.5 & \text{if} \quad \text{Is initial level} \\ p_k = 0.2 & \text{if} \quad \min\left(\left\{\sigma_k^{(1)}, \dots, \sigma_k^{(n)}\right\}\right) < 0.1 \\ p_k = 0.1 & \text{if} \quad \min\left(\left\{\sigma_k^{(1)}, \dots, \sigma_k^{(n)}\right\}\right) < 0.01 \end{cases} \qquad (\mathbf{3.13})$$

To encourage an initial exploration and progressively move towards the global minimum, a decrease in the probability of level is conducive to develop this aspect.

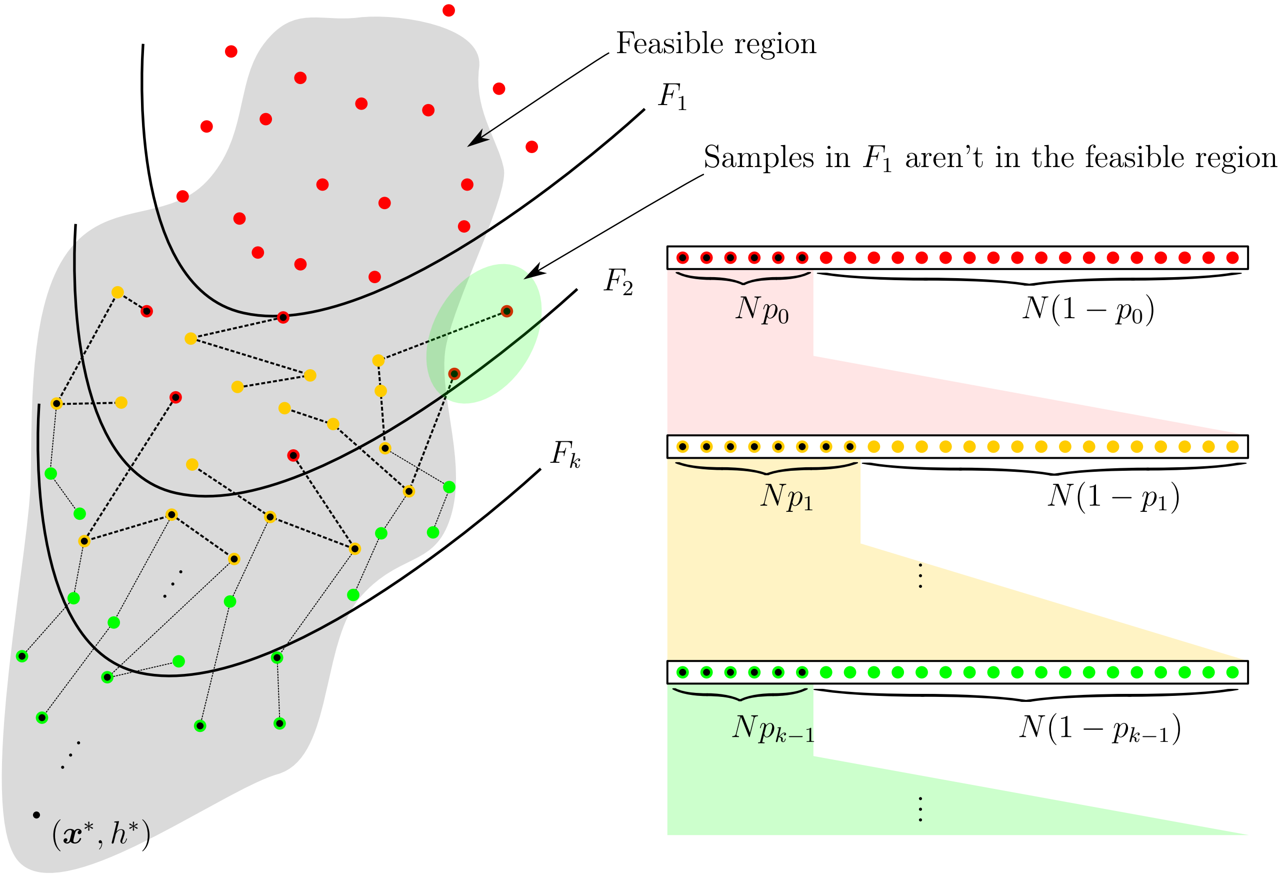

Figure **3.2**: Schematic of the basic idea of (SS) under an optimization process.

The general pseudo code for the proposal algorithm in (H.-S. Li & Au, 2010) is presented

Algorithm 3 Subset simulation for optimization algorithm.

```
procedure SSOptimization($h_{\text{obj}}(\cdot), g_{\text{constraint}}(\cdot), N, k_{max}, \varepsilon$)
    Set $k = 0$                                                        ▷ Level 0
    Set $p_k = 0.5$                                   ▷ Initial level probability $p_0$
    Set the (PDFs) $\{f_{X_i}(\cdot)\}$, $\{p^*_{X_i}(\cdot \mid \cdot)\}$ for $n$ component $i = 1, \ldots, n$
    Generate the $N$ samples by (DMCS) $\{\boldsymbol{x}_{1,(0)}, \boldsymbol{x}_{2,(0)}, \ldots, \boldsymbol{x}_{N,(0)}\}$
    for $j = 1, \ldots, N$ do
        for Each dimension $i = 1, \ldots, n$ do
            Generate a sample from $x^{(i)}_{j,(0)} \sim f_{X_i}(\cdot)$
        end for
        Eval the cost function $h_{\text{obj}}(\boldsymbol{x}_{j,(0)})$
        Eval the constraint fitness function $F_{con}(\boldsymbol{x}_{j,(0)})$, equation (3.3)
    end for
    Sort the samples by the double-criterion $\{\boldsymbol{x}_{1,(0)}, \boldsymbol{x}_{2,(0)}, \ldots, \boldsymbol{x}_{N,(0)}\}$
    Sort the list $\{h_{\text{obj}}(\boldsymbol{x}_{1,(0)}), \ldots, h_{\text{obj}}(\boldsymbol{x}_{N,(0)})\}$
    Sort the list $\{F_{con}(\boldsymbol{x}_{1,(0)}), \ldots, F_{con}(\boldsymbol{x}_{N,(0)})\}$
    Compute the standard deviation $\widehat{\sigma}_k$ of the samples sorted for each component
    Compute the standard deviation $\widehat{\sigma}_{k,\text{bounds}}$ of the samples sorted for each component
    while $k < k_{max}$ or $\varepsilon > max\left(\left\{\widehat{\sigma}^{(1)}_{k,\text{bounds}}, \ldots, \widehat{\sigma}^{(n)}_{k,\text{bounds}}\right\}\right)$ do   ▷ Termination condition
        Set $k = k + 1$                                            ▷ Next level
        Compute $N_{c_k} = N * p_{k-1}$                               ▷ Chain length
        Compute $N_{c_k} = \frac{1}{p_{k-1}}$                          ▷ Samples per chain
        Set the thresholds for the level $h_k = h_{\text{obj}(\boldsymbol{x}_{Np_{k-1}})}$ and $F_{c_k} = F_{con}(\boldsymbol{x}_{Np_{k-1}})$
        for Each sample in the chain $\{\boldsymbol{x}_{1,(k)}, \ldots, \boldsymbol{x}_{Np_{k-1},(k)}\}$, $j = 1, \ldots, Np_{k-1}$ do
            for $s = 1, \ldots, N_{s_k} - 1$ do
                Use (MMH) method to generate $N_{s_k} - 1$ samples
                Set $F_k$ with the thresholds $\{h_k, F_{c_k}\}$
                $\boldsymbol{x}_{k+1} = \text{MMH}\left(\boldsymbol{x}_{j,(k)}, \{f_{X_i}(\cdot)\}, \{p^*_{X_i}(\cdot \mid \cdot)\}, F_k\right)$   ▷ New sample
            end for
        end for
        Sort the samples by the double-criterion $\{\boldsymbol{x}_{1,(k+1)}, \boldsymbol{x}_{2,(k+1)}, \ldots, \boldsymbol{x}_{N,(k+1)}\}$
        Sort the list $\{h_{\text{obj}}(\boldsymbol{x}_{1,(k+1)}), \ldots, h_{\text{obj}}(\boldsymbol{x}_{N,(k+1)})\}$
        Sort the list $\{F_{con}(\boldsymbol{x}_{1,(k+1)}), \ldots, F_{con}(\boldsymbol{x}_{N,(k+1)})\}$
        Compute the standard deviation $\widehat{\sigma}_{k+1}$ of the samples sorted for each component
        Compute the standard deviation $\widehat{\sigma}_{k+1,\text{bounds}}$ of the samples sorted for each component
        Select the $p_k$ according to (3.13)
    end while
    return $\boldsymbol{x}_1$, $h_{\text{obj}}(\boldsymbol{x}_1)$
end procedure
```

Chapter 4

Numerical experiments

4.1 Some optimizations design problems

In this section some test models in the area of design optimization are presented following.

4.1.1 Welded beam design problems (WBD)

The design optimization problem of welded beam design problem has the goal to minimize the construction cost of a welded beam (H.-S. Li & Au, 2010). The problem contains two and five constraints of linear nonlinear character respectively, depending on the shear stress ($\tau(\boldsymbol{x})$), bending stress ($\sigma(\boldsymbol{x})$), buckling load on the bar ($P(\boldsymbol{x})$) and their respectively deflection ($\delta(\boldsymbol{x})$) and some geometrical constraints ($g_3(\boldsymbol{x})$), ($g_4(\boldsymbol{x})$) and ($g_5(\boldsymbol{x})$), expressed in mathematical form as

$$\begin{aligned}
\min_{\boldsymbol{x}\in\mathcal{X}} \quad & f\left(\boldsymbol{x}\right) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2) \\
\text{s.t.} \quad & \mathrm{g}_1(\boldsymbol{x}) = \tau(\boldsymbol{x}) - \tau_{\max} \leq 0 \\
& \mathrm{g}_2(\boldsymbol{x}) = \sigma(\boldsymbol{x}) - \sigma_{\max} \leq 0 \\
& \mathrm{g}_3(\boldsymbol{x}) = x_1 - x_4 \leq 0 \\
& \mathrm{g}_4(\boldsymbol{x}) = 0.10471x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0 \\
& \mathrm{g}_5(\boldsymbol{x}) = 0.125 - x_1 \leq 0 \\
& \mathrm{g}_6(\boldsymbol{x}) = \delta(\boldsymbol{x}) - \delta_{max} \leq 0 \\
& \mathrm{g}_7(\boldsymbol{x}) = P - P_c(\boldsymbol{x}) \leq 0
\end{aligned} \qquad \mathbf{(4.1)}$$

Where

- $\tau(\boldsymbol{x}) = \sqrt{(\tau')^2 + \frac{2\tau'\tau''x_2}{2R} + (\tau'')^2}$

- $\tau' = \frac{P}{\sqrt{2}x_1x_2}$
- $\tau'' = \frac{MR}{J}$
- $M = P\left(L + \frac{x_2}{2}\right)$
- $R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$
- $J = 2\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]$
- $\sigma(\boldsymbol{x}) = \frac{6PL}{x_4x_3^2}$
- $\delta(\boldsymbol{x}) = \frac{4PL^3}{Ex_3^2x_4}$
- $P_c(\boldsymbol{x}) = \frac{4.013E\sqrt{x_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$
- $P = 6000$ [lb$_\mathrm{f}$]; $L = 14$ [in]; $E = 30 \times 10^6$ [psi]; $G = 12 \times 10^6$ [psi]; $\tau_{\max} = 13600$ [psi]; $\sigma_{\max} = 30000$ [psi]; $\delta_{\max} = 0.25$ [in].

The design variables $\boldsymbol{x} = [x_1, x_2, x_3, x_4]^T = [h, l, t, b]^T$ and their respectively bound or search space $\mathcal{X} = \{0.1 \leq x_1 \leq 2.0, 0.1 \leq x_2 \leq 10.0, 0.1 \leq x_3 \leq 10.0, 0.1 \leq x_4 \leq 2.0\}$ are schematized in

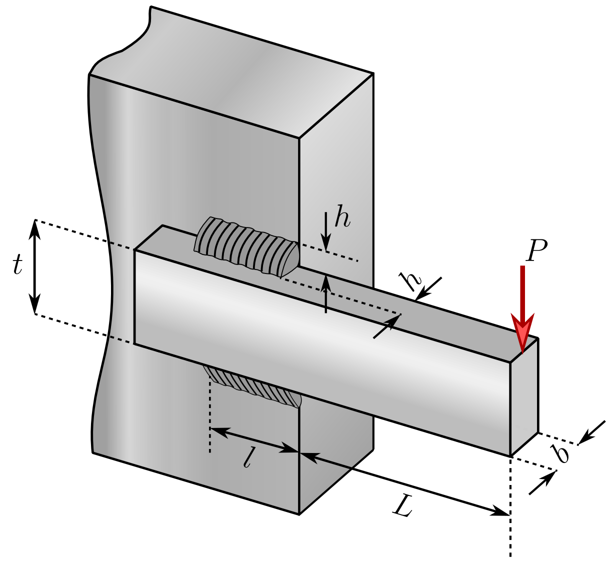

Figure **4.1**: Outline of the dimensions of the welded beam design problem.

4.2 Subset simulation under test

To test the benefits of the (SS) method in solving design problems, simulations of the design problems were performed for a variation of the number of samples taking the discrete values of $N = \{100, 200, 300, 400, 500\}$ And performing a number of 30 simulations per sample number.

Table **4.1**: Statistical performs on design problems for (30 runs) and $\varepsilon = 10^{-4}$ for the values of objective function of simulations in the method (SS).

Model	Index	N = 100	N = 200	N = 300	N = 400	N = 500
(WBD)	Best	1.724854	1.724853	1.724853	1.724853	1.724853
	Mean	2.108491	1.773646	1.736167	1.730329	1.724853
	Worst	2.780010	2.057823	1.846201	1.824922	1.724854
	Std	0.285344	0.087655	0.031720	0.021363	2.536104e-7
Truss 25 bars	Best	545.384962	545.244064	545.250728	545.244512	545.201210
	Mean	547.602937	545.793618	545.773894	545.594412	545.496710
	Worst	557.275184	548.798790	548.570497	546.945024	546.228210
	Std	2.706134	0.648599	0.681307	0.396783	0.259226

Table **4.2**: Statistical performs on design problems for (30 runs) and $\varepsilon = 10^{-4}$ for the number of levels of simulations in the method (SS).

Model	Index	N = 100	N = 200	N = 300	N = 400	N = 500
(WBD)	Best	173.00	183.00	183.00	184.00	184.00
	Mean	258.27	272.27	235.93	215.20	196.40
	Worst	374.00	427.00	418.00	430.00	236.00
	Std	51.68	84.59	76.85	57.12	10.32
Truss 25 bars	Best	106.00	169.00	169.00	177.00	180.00
	Mean	244.17	237.13	235.60	226.63	239.40
	Worst	547.00	389.00	363.00	297.00	369.00
	Std	76.17	56.60	45.14	29.55	52.54

Chapter 5

Conclusions

References

Au, S.-K., & Beck, J. L. (2001). Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, *16*(4), 263-277. doi: https://doi.org/10.1016/S0266-8920(01)00019-4

Au, S.-K., & Beck, J. L. (2003). Subset simulation and its application to seismic risk based on dynamic analysis. *Journal of Engineering Mechanics*, *129*(8), 901–917.

Bertsekas, D. P. (1997). *Nonlinear programming* (Vol. 48) (No. 3). Taylor & Francis.

Bertsekas, D. P., & Tsitsiklis, J. N. (2008). *Introduction to probability* (Vol. 1). Athena Scientific.

Chan, S. H. (2021). *Introduction to probability for data science.* Michigan Publishing.

Ghahramani, S. (2004). *Fundamentals of probability, with stochastic processes, 3rd edition* (3rd Edition ed.). Prentice Hall.

He, S., Prempain, E., & Wu, Q. H. (2004). An improved particle swarm optimizer for mechanical design optimization problems. *Engineering Optimization*, *36*(5), 585-605. Retrieved from https://doi.org/10.1080/03052150410001704854 doi: 10.1080/03052150410001704854

Jorge Nocedal, S. W. (2006). *Numerical optimization* (2nd ed ed.). Springer.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. , *4*, 1942-1948 vol.4. doi: 10.1109/ICNN.1995.488968

Kolmogorov, A., Morrison, N., & Bharucha-Reid, A. (1956). *Foundations of the theory of probability.* Chelsea Publishing Company.

Leon-Garcia, A. (2008). *Probability, statistics, and random processes for electrical engineering (3rd edition)* (3rd ed.). Prentice Hall.

Li, H.-S., & Au, S.-K. (2010). Design optimization using subset simulation algorithm. *Structural Safety*, *32*(6), 384-392. (Modeling and Analysis of Rare and Imprecise Information) doi: https://doi.org/10.1016/j.strusafe.2010.03.001

Li, L., Huang, Z., Liu, F., & Wu, Q. (2007). A heuristic particle swarm optimizer for optimization of pin connected structures. , *85*(7), 340-349. doi: https://doi.org/10.1016/j.compstruc.2006.11.020

Rubinstein, R. Y., & Kroese, D. P. (2016). *Simulation and the monte carlo method.* John Wiley & Sons.

Taboga, M. (2017). *Lectures on probability theory and mathematical statistics.* Createspace.

Wang, S.-K., Yu; Au. (2014). *Engineering risk assessment and design with subset simulation* (1st ed.). Wiley.

Yang, X. (2011). Metaheuristic Optimization. *Scholarpedia*, *6*(8), 11472. (revision #91488) doi: 10.4249/scholarpedia.11472