CHAPTER 3

OVERVIEW OF EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION, SOFTWARE COMPLEXITY AND SOFTWARE USABILITY

3.1 MULTI-OBJECTIVE OPTIMIZATION

The process of concurrently optimizing more than one objective contradictory with each other subject to some constraints is called multi-objective optimization. There is a need to take a decision that is optimal in the company of trade-offs among more than one conflicting goals. One cannot identify a single point of solutions to optimize each objective simultaneously [22].

Optimization is a mathematical discipline that concerns to find out a substitute solution with most cost effective and maximum achievable performance under the limitations by maximizing the required factors and lowering the not required factors [33]. The meaning of maximization is to trying to achieve the uppermost or maximum outcome or result will generated with no regard to expense or cost. These days, optimization comprises a broad diversity of techniques commencing artificial intelligence and is utilized to perk up the business processes in almost all the industries.

3.2 EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION (EMO)

The term MO is, without loss of generality, there is no single solution and it should be best when measured on all objectives/ solution [14-19]. It minimizes n number of the components f_k, where if there exists no, perfect, feasible, unique answer however a minimization crisis, dominance can be given as:-

$$iff!\ u, v: \ , !\ \gamma: (0, 1), \#\ w : P : w \leq \square u + (1 - \square)v , \qquad \textit{Equation 1}$$

where the disparity applies component wise or else, P will be non-convex. Likewise, for concavity:

The definition of Concavity can be given as: a *well defined non-dominated set P will be concave iff*

$$!\ u, v : P , !\ \gamma : (0, 1) , \#\ w : P :: \square u + (1 - \square)v\ p< w . \qquad \textit{Equation 4}$$

The surfaces attained after establishing the trade-off might not be concave or convex. Therefore, the regions of local concavity as well as local convexity can generally be recognized in such trade-off surfaces.

The trade-off surface’s convexity is based on the fact that in what manner the objectives are scaled. As a result, the look for the best or an optimal solution is discarded from what comes as an observation in the case of one objective problem. Generally there is a requirement of decision making as well as searching. To facilitate the decision building and searching, there are four different approaches are acknowledged.

3.2.1 POSTERIOR ARTICULATION OF PREFERENCES

In order to discover every promising answer of the non conquered set, utilization of the user first choice can be done to decide the largely suitable known as decision making following the search. A large number of techniques are available that enable to discover the solution space [23]. The great reward with these kinds of methods is that the obtained solution is self-determining of the decision maker's preferences. The study has only to be execute ones, since the Pareto set would not vary providing the problem description are unaffected. Though, a few of these methods bear from a huge computational load.

3.2.2 PROGRESSIVE ARTICULATION OF PREFERENCES

At each step in this preference, a partial preference piece of the important information is delivered by the authorized decision maker to the start the process of optimization. The process of optimization and decision making happen at interleaved steps, thus it inward information and produces improved alternatives. Also, the decision-maker is not constantly giving input throughout the operation of the algorithm.

3.3 NO PREFERENCE

This approach is useful in solving a problem as well as provides a solution to the Decision Maker. This method doesn't consider any preference information. This Min-Max formulation has the fundamental of minimization of the calculated relative distance that is starting from a given candidate solution up to the desired utopian solution. Also, there is no preference related information from the decision- maker is essential. Though, the outcome is merely a single point on the obtained Pareto front, which the Decision Maker must have to recognize as the last and final generated solution.

Through the provision of the single objectives diverse weightings and altering the exponent in the distance formulation, dissimilar points on the obtained Pareto front might be established [24]-[26]. Though, then the preference related information provided from the decision-maker afterwards is required. In this scenario, this kind of formulation isn't generally considered in industrial design. Yet, this Min-Max formation could be considered jointly along with the other similar methods in order to uncover various points on the desired Pareto front for the interactive methods.

3.4 CONSTRAINTS OF OPTIMIZATION

The most suitable solution to a given real world problem might be controlled by a series of real world limitations forced of the desired decision variable. There are following two categories of constraints:

3.4.1 DOMAIN CONSTRAINTS

The first category is the Domain Constraints, which states the domain of the description of required objective function. Considering the case of control systems, the closed loop system steadiness will be specified as a case of a mentioned domain constraint, since the major and significant performance measures are not very clear for the unstable systems.

3.4.2 PREFERENCE CONSTRAINTS

This imposes additional limits on the solution of the problem. The known stability margin, in case, represents a (subjective) liking of the decision maker.

It is completely supposed that there is no less than a single point in the given U that fulfils each and every constraint, even though in the preparation that cannot for all time be certain. While the defined constraints may not be all concurrently fulfilled, the major issue is regularly seemed to recognize not a single solution as it occurs. In order to probably slow down the preference constraints, there is a range of constraints violated along with the level to which every defined constraint is desecrated are then engaged into consideration.

The Constraints may be considered as objectives of maximum priority, which have to be mutually satisfied prior to the optimization of the left over. Fulfilling several violated inequality constraints are obviously the multi-objective problem of minimizing the related functions awaiting the provided values (aim) is achieved. The idea of non inferiority is as a result, willingly appropriate and even predominantly suitable while constraints are themselves non commensurable.
In contrast, the problems categorized by well defined soft – objectives simply, are regularly reorganized as the constrained optimization problems (single objective) so that it can be resolved.

3.5 VISUALIZATION AND PERFORMANCE ASSESSMENT ISSUES

This is related with the graphical image of the trade-off data, that is, in case, if there are single or multiple sets.

The major objective that is present here is to correspond to the individual decision maker, the valuable information regarding to the most excellent established trade-off on the whole non-dominated solutions. In contrast, for achieving within reach into how fine an optimizer can be likely to execute upon any specified crisis, the extracted data by several long and tedious optimization runs have to be measured in its totality, which means this is wrong to believe that only the whole non-dominated solutions will be taken into consideration. As a result, a visualization process is being considered; the process of Visualization of the trade-off data commencing single run methods for posterior as well as progressive articulation of preferences that call for the trade-off information be expressed in front of the human decision maker in a shape that should be without difficulty understand [27].

Therefore, in order to attain the smooth image of the Pareto-set, the interpolating between the data is not usually right, primary as here there is usually zero assurance that it would truly be even, as well as next as actual generated solutions related to those in-between objective vectors, yet if they be present, but still not acknowledged.

Maximum, one possibly will wish to draft an edge unravelling those important points in objective space that are equals or dominates. This kind of edge can as well be viewed are identified to be achievable, providing the data is given [28].

While evaluating an optimizer, the individual is generally worried for the worth value of the resultant solutions that are capable to fabricate, as well as with the sum of calculation attempt it needed. Additionally, evaluation of how probable a single run is to generate high-quality outputs is also important.
If there is only one objective, the value of the end result of an optimization run is calculated straight by the objective value related with the finest solution established. Given that this is a single, scalar value, the division of the quality of end results generated by numerous runs could be effortlessly seen by the scatter plots, histograms and empirical distribution functions etc.

In case, if there exists multiple that is more than one objective, a dissimilar illustration is essential. A general method, also known as the approach of parallel coordinates, including the integrating an integer index i to every individual objective and then representing every non-dominated point with the help of a line linking the objective points. Along with this illustration, rival objectives having successive indices outcome in the crossing of lines, while non- concurrent lines point out non-rival objectives. Even though the sequencing of the objectives might be routinely resolute upon on the foundation of a number of measure of struggle (so that struggle will be at its maximum level among neighbouring objectives), being capable to vary this sequencing interactively is valuable as well, and not hard to execute.

In case, if there are many objectives, though, runs will usually generate changeable number of approximations. The importance of the trade-off description generated by each run depends on two factors. The first is the non-dominated points established and on how well they cover it up [30]. Merely the suggestions of how fine the individual objective points establish in every run inclined to be, however the valuable information on matter that how they inclined to be dispersed alongside the trade-off surface is gone. There are following two factors:

- Population based search techniques simulating the behaviour of natural evolution.
- Dealing with complex search spaces having robust and powerful search mechanism.

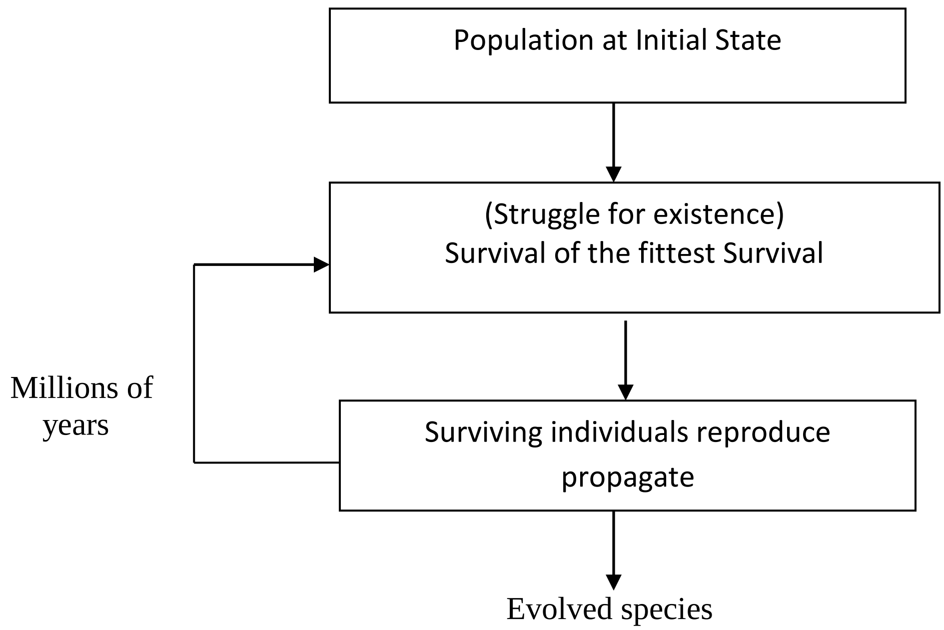

Figure 3.1 Natural Evolution

3.6 SOFTWARE COMPLEXITY

Complexity issue related to "Ease of Use" in the line with the end user. A high skilled user may find some task very easy than low skilled (related to computer related knowledge) user [6]. Complexity of the system varies from mere installation of the software to complexity verifying online in real time mode. It provides a relative comparison between the levels of difficulty in software products. For example, in Indian scenario, there are number of consumers who have high end systems but lacks of usage of all the software systems. It is divided into three parts Personal, Social and Technological [7]-[11]. The Personal factors identifies the traits of user depending upon his/her knowledge skills, user is able to perceive things and develops adaptability accordingly. When design a software module for user interface personal traits are given very high priority. Social factor deals with peer environment and their skill set.

3.6.1 COMPLEXITY ANALYSIS

It is a new approach to evaluating the usability of software that combines many of the advantages of both usability testing and usability inspection methods. Complexity analysis provides metrics that quantify usability and that are highly correlated to results gathered through usability testing [37]. In addition, complexity analysis shares the lightweight characteristics of usability inspection methods, enabling teams that develops software to frequently evaluate software usability throughout the development process.

Complexity analysis involves breaking down a user task into a set of constituent steps and then calculating a complexity metric for each step in the task relative to the type of user [64]. For example, Table 1 shows the complexity metric for each step of a fictitious installation task for the user role “social-networking parent”. The nine steps of the installation task in Table 1 are listed sequentially in the left-hand column, and their corresponding complexity metrics are in the right-hand column. The complexity metric for a step is a measure of how difficult it is for the targeted user to complete that step. You can think of the complexity metric as an inverse measure of usability – the higher the complexity metric, the lower the usability for that step. Therefore, “lower is better” when it comes to complexity metrics.

The overall complexity metric for a task is the sum of the complexity metrics for its constituent steps. Therefore, the complexity metric for the installation task depicted in Table 1 is 79.

The steps of an Installation Task with its complexity metric are shown in the table 3.1.

Table 3.1: Complexity metrics for the steps of a software installation task

S. No.	**Step**	**Complexity Measure**
1	Turn off firewall	15
2	Execute setup file	11
3	Select license option	3
4	Select type of installation	3
5	Specify installation directory	5
6	Confirm user name	11
7	Install program files	11
8	Specify preferences	3
9	Turn on firewall	17
	Total	**79**

3.7 SOFTWARE DELIVERABILITY (USABILITY ASPECT)

The software deliverability can be considered as the degree of the usability factor provided to the user of the system by the software. The software deliverability should be high in order to attain maximum value from the software. The business value of any software is highly affected by the software deliverability factor which later imposes several constraints on the software developers.

The term software usability is actually the level of comfort or the ease with which a user can work on the software. As discussed earlier that the software complexity varies from person to person and from software to software in a well defined and constrained scenario, the software usability also varies accordingly. Higher the complexity, lower will be the usability aspect of that particular software product.

Higher usability factor enabled software has higher market space capturing probability as comparison to those software products which has low software usability value. This directly affects the market status of the software as well as the company.

The usability factors of software can be categorized as:

- Efficiency:
- Learnability
- Satisfaction
- Memorability:
- Errors

The next factor Memorability can be defined as when users go back to the design following a period of not using it, how effortlessly can they re-establish the expertise? And the last factor satisfaction can be explained as how pleasing is it to utilize the design?

Six Basic Factors of Software Usability:-

1. Context Shifts
2. Navigational Guidance
3. Input parameters
4. System feedback

5. Error feedback
6. New concepts

1**. *Context shift*** occurs when the user crosses user interface, tool or product boundaries in order to perform a step.

*2***. *Navigational guidance*** refers to the support provided to a user for proceedings into a step (from the previous step) and through the step.

3. ***Input parameters*** are data supplied by the user to complete the step

4. ***System feedback*** is the system response to the user actions for a given step. Examples of system feedback include progress indication dialog boxes, confirmation of command execution and system generated reports.

5. ***Error feedback*** is the system response to common error situations the user may encounter.

6. ***New Concepts*** refers to background information on a specific topic that the user needs to understand in order to perform a step, and that the user has encountered for the first time in the context of their current task(s).