**PAPER WRITING**
**CAP 707**
**(Spring term 2014-15)**


***IMPLEMENTATION***
***RESEARCH TOPIC: ENHANCEMENT AND OPTIMIZATION OF SOFTWARE TESTING TECHNIQUES***

Submitted by:
**Shubham Derashri (11407483)**
**Harman Deep Kaur (11410286)**
**Sukhneet Kaur (11408818)**
**Mintu Sharma (11410951)**

Under the Guidance of

**MR.PUNEET MANGLA**
**School Of Computer Applications**
**Lovely Professional University**
**Phagwara (Punjab, India)**

# INDEX

# ABSTRACT

Software testing techniques provides a means to reduce errors, cut maintenances and overall cost of project. Software testing tools are doing there working according software development life cycle.

Today we have a lot of software testing techniques in the market. Selection of tools is totally based on the project requirement and commercial tools or open source tool (free tools). Software testing methods and levels are also concern through of software testing tools ,because without using testing techniques we can't do anything in order to achieve the quality and proper maintenance of the software.

# INTRODUCTION

The topic of our research is the various software testing techniques that are available presently. The motto of this literature review is to briefly review the already done research in this particular field. Software testing is the major aspect in order to deliver the final product to the end user who is actually going to use it, it is obviously expected by his/her that whatever product he purchased from the company should not come with any error or miscalculations problems. So every software developing company has an special team which is responsible for the testing of the product at various levels and this team uses some standard techniques in order to evaluate or test the software. Testing typically consumes 40-50% of development efforts, and consumes more effort for systems that require higher levels of reliability, it is a significant part of the software engineering , the proportion of time devoted to testing increased. As the amount of maintenance and upgrade of existing systems grow, significant amount of testing will also be needed to verify systems after changes are made.

Despite advances in formal methods and verification techniques, a system still needs to be tested before it is used. Testing remains the truly effective means to assure the quality of a software system of non-trivial complexity as well as one of the most intricate and least understood areas in software engineering .Testing, an important research area within computer science is likely to become even more important in the future.

In different publications, the definition of testing varies according to the purpose, process, and level of testing described. Miller gives a good description of testing in:

*"The general aim of testing is to affirm the quality of software systems by systematically exercising the software in carefully controlled circumstances."[1]*

Miller's description of testing views most software quality assurances activities as testing. He contends that testing should have the major intent of finding errors. A good test is one that has a high probability of finding an as yet undiscovered error, and a successful test is one that uncovers an as yet undiscovered error.

Testing is primarily done on software as well as in web for testing client and server architecture. Software testing is one of the major and primary techniques for achieving high quality software. Software testing is done to detect presence of faults, which cause software failure. However, software testing is a time consuming and expensive task It consumes almost 50% of the software system development resources.

So this was the introduction to the software testing it states that why testing is necessary and what are the aspects of the testing expected by the client and the developer.

# PROBLEM BACKGROUND

In present, testing techniques have gradually involved from the practice of single programmers or small development teams into a systematic, managed engineering discipline. Not only have there been numerous researches on testing techniques, but also more and more considerable industry practices. There are testing classes taught in universities. There have been special testing teams, test managers, and tester job positions open to professional testers; there have been training programs and complete procedures for testing in large enterprises; and there are increasing number of companies and vendors doing testing work for other companies.[2]

However, despite the numerous research results (quite a lot of them are really sound) testing remains an awkward, time -consuming, cost-ineffective chunk of work that is always not very satisfying in most industry practices. Only a small number of the research results have been utilized successfully in industry practices so that the test process can be greatly improved or automated. The most common testing exercises in industry are static analysis including code inspections, peer reviews, walkthroughs. Not enough testing tools can be applied directly on industry projects and products without being largely modified and even re-developed. Test plans are still written by hand, while test environment remained simple and crude. It is always the case that testing ends up being a must-end activity because the project runs out of budget and is beyond deadline. This inconsistency of testing research and practice has been called the ñtesting gap.ò

Above mentioned statements are defining the current problem background which we have identified during the literature review of the given topic as per work done on the topic in previous semester. The problem which we have identified is currently known to us as ñTesting Gap Problemò.

As written in the above paragraphs what we know about testing gap is that it is a problem which is generally ignored by the organizations due to the lack of time and resources, however, inspite of sometimes organizations having proper time and resources but then also they ignore this problem due to the improper estimation of cost associated with the particular testing technique which they have implemented to test a specific software. So, in the further sections of this research paper, we are going to propose an algorithm by considering the testing gap as linear

programming problem to find the optimum values of the time and resources associated with the particular testing technique and hence to minimize the total cost associated with the testing of a specific software.

# PRESENT WORK/IMPLEMENTATION

So by understanding the problem of Testing Gap what we understood is that testing is a still a time consuming and a very stressed field for an organization. The main problem which is stated as testing gap is that the testing team is always in a confusion to use which kind of testing technique to be used for the testing of a particular software. As the testing includes the 40-50% of the total cost of the software development ,so it is highly recommended for a testing team to choose the testing technique wisely which can hence reduce the total cost of testing of a particular technique, hence for the same we are proposing a algorithm by taking the optimization techniques as base(In our algorithm we are taking Simplex method as base),hence by optimizing the cost of techniques to minimum we can now opt to choose from a set of testing technique to use in the software testing which will provide the best and cost efficient results.

## PROPOSED ALGORITHM

The algorithm which we are going to propose we consider a hypothetical situation in which the őZô function is the cost function related with the testing techniques used for testing. Following are some assumptions which we have taken in order to implement the algorithm.

(i) For a specific software only four testing techniques will be used for testing.
(ii) Each technique possess a specific cost with it and hence we are going to derive all these cost functions in terms of a linear programming problem.
(iii) The number of constraints will always be equal to the number of testing techniques that is 4 in this algorithm.

The following are the constraints which we are considering in this algorithm :

a.) The number of resources (i.e. the number of testers working on it).
b.) The time duration (i.e. the amount of time in days to complete the testing).
c.) The feasibility factor of that technique (in percentage).
d.) Accuracy of the result of the technique (in percentage).

Let us consider the testing techniques as x1,x2,x3 and x4 respectively and the cost associated with them as C1,C2,C3 and C4 respectively, deriving this into an LPP the Z function which we will need to minimize is as follows :

$Z=C1x1+C2x2+C3x3+ C4x4$

Let us consider the resources for testing techniques as p1,p2,p3,p4 respectively for x1,x2,x3 and x4. The total no. of resources should not exceed P (any positive number).

And t1,t2,t3 and t4be the time associated with x1,x2,x3 and x4 respectively, and should not exceed T(any positive number).

Also, f1,f2,f3,f4 be the feasibility factor for each x1,x2,x3 and x4 respectively, and should not exceed F(any positive number).

Also, a1,a2.a3 and a4 be the accuracy factor for each x1,x2,x3 and x4 respectively. It can have maximum value A(by default we are considering A= 100 for the sake of simplicity.

So, the final LPP along with constraints is as follows:

Minimize: $Z=C1x1+C2x2+C3x3+ C4x4$

Subject to :

$p1x1+p2x2+p3x3+p4x4<=P$

$t1x1+t2x2+t3x3+t4x4<=T$

$f1x1+f2x2+f3x3+f4x4<=F$

$a1x1+a2x2+a3x3+a4x4>=A$

now by using SIMPLEX(minimization) Method of Optimization we can easily minimize the Z function to minimize the total cost by getting the optimum values for the above constraint to determine the best fit values of the no. of resources, time duration, feasibility factor and hence the accuracy of a particular technique.

This algorithm will lead to the following result:

1.) The Best case scenario
2.) The worst case scenario.
3.) The optimal scenario.

Now further stretching this algorithm where we will consider that clients are going to change their requirement at a particular moment of time, this will also affect the constrantsôs current

behavior and the calculations, in order to refract the changes in the final solution we may either go for the complete solution again by solving the function with new constraints, but it is a very time consuming and non operationally feasible technique, hence we will choose the Post Optimality Analysis criteria to solve the problem with the changed constraints. Hence the algorithm is completed.

## MAIN ALGORITHM FOR REDUCING TESTING GAP

1.) Consider the cost function to be minimized(taking 4 testing techniques)
2.) Determine the magnitude of the constraints(i.e. resources ,time, feasibility-factor, accuracy)
3.) Form it as an LPP
4.) Solve using SIMPLEX(minimization) Optimization method
5.) Repeat 4 till you get:
    5.1)All the necessary conditions satisfying the Big M minimization LPP solution
    5.2) Best Testing Technique for a specific software
    5.3) Worst Testing Technique for a Specific Software
    5.4) Optimal Values for all the constraints for a particular technique
6.) Evaluate the Cost function with optimal values of x1,x2,x3 and x4 hence getting the total Minimized cost
7.) convert all the values into absolute values(i.e. ignoring the negative signs)
8.) Check for the Changed Userôs requirement if yes then go for step 9 else go to step 10
9.) Post Optimal Analysis
10.) End(Getting the Optimized values).

## IMPLEMNTING ALGORITHM WITH SOME REAL TIME PROBLEMS

**Dummy Problem 1: (when single technique is opted)**

**ABC Company is following software testing techniques in order to test a software. The data along with constraints(as required by the algorithm) is given as follows:**

| CONSTRAINTS | TESTING TECHNIQUES | | | |
|---|---|---|---|---|
| | A | B | C | D |
| RESOURCES | 2 | 1 | 3 | 2 |
| TIME | 3 | 4 | 2 | 4 |
| FEASIBILITY FACTOR | 1 | 3 | 2 | 4 |
| ACCURACY | 10 | 25 | 25 | 10 |
| COST ASSOCIATED(k units) | 4 | 5 | 9 | 2 |

Now let us solve this by converting the data first converting it into an LPP and hence adding the constraints according to organization conditions Hence the equation formed for the scenarios is as follows:

Minimize Z= 4x1+5x2+9x3+2x4

Subject to:

2x1+x2+3x3+2x4<=10 (resource constraint)

3x1+4x2+2x3+5x4<=21(Time Constraint)

x1+3x2+2x3+4x4<=12(Feasibility factor constraint)

20x1+25x2+25x3+10x4<=100(Accuracy constraint)

Now multiplying both sides of Z function with -1 we will get

-Z = -4x1-5x2-9x3-2x4

=> -Z+4x1+5x2+9x3+2x4=0

As all the constraints are of less than type so we need to add slack variables in order to make the constraints as consistent linear equations

Hence the constraints will be:

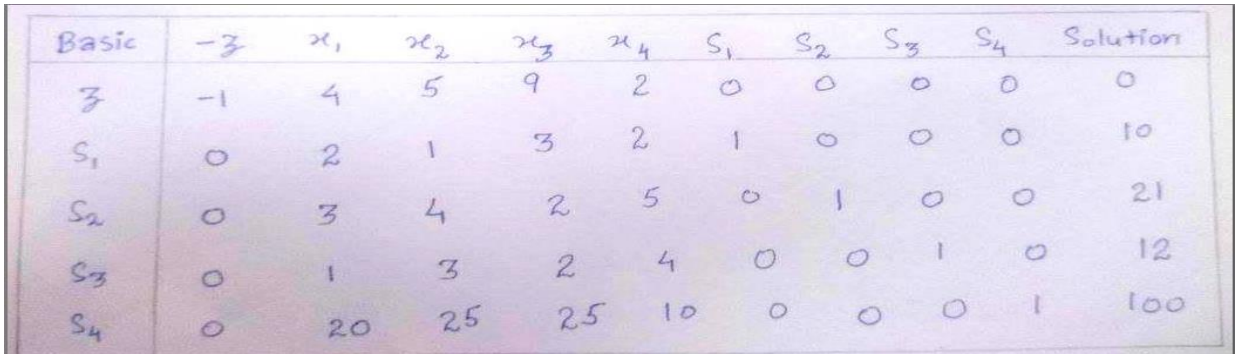
2x1+x2+3x3+2x4+s1=10

3x1+4x2+2x3+5x4+s2=21

x1+3x2+2x3+4x4+s3=12

20x1+25x2+25x3+10x4+s4=100, where s1,s2,s3,s4 are the slack variables.

Hence the first table for simplex minimization is as follows:



| Basic | $-Z$ | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| $Z$ | -1 | 4 | 5 | 9 | 2 | 0 | 0 | 0 | 0 | 0 |
| $S_1$ | 0 | 2 | 1 | 3 | 2 | 1 | 0 | 0 | 0 | 10 |
| $S_2$ | 0 | 3 | 4 | 2 | 5 | 0 | 1 | 0 | 0 | 21 |
| $S_3$ | 0 | 1 | 3 | 2 | 4 | 0 | 0 | 1 | 0 | 12 |
| $S_4$ | 0 | 20 | 25 | 25 | 10 | 0 | 0 | 0 | 1 | 100 |

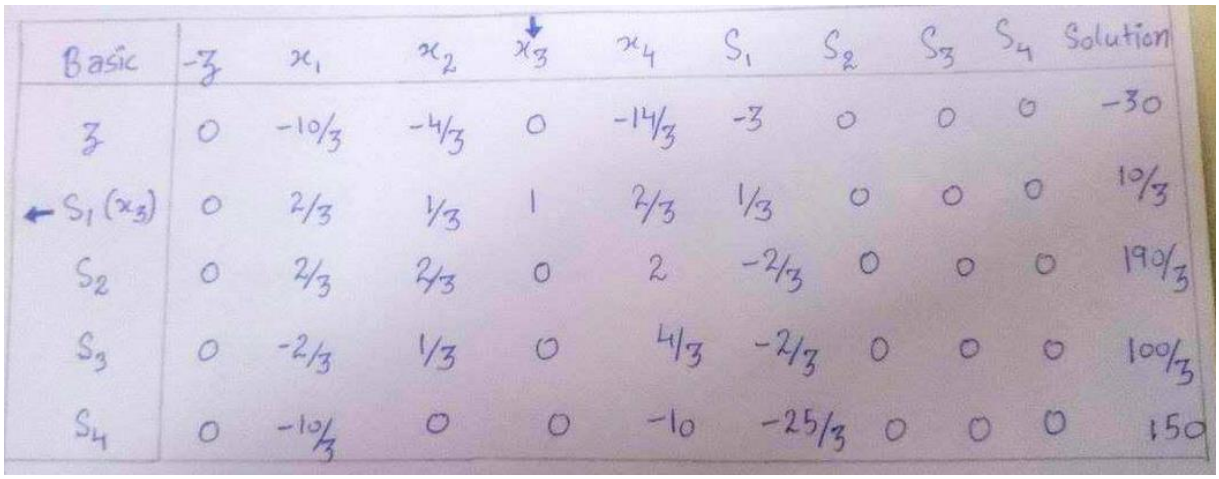
Now we can observe that this table is not optimized because it is not satisfying the conditions of optimiality for simplex minimum i.e. (all elements of Z row is non positive),so we will perform iteration of simplex to optimize it.

Here x3(most positive) will be entering variable and s3 will be leaving variable(least positive intercept)

New pivot row= current pivot row/pivot element

New Row = (Current Row-Pivot Coloumn Coefficient)×new pivot row

Hence the new table will be:



| Basic | -Z | $x_1$ | $x_2$ | $x_3$ ↓ | $x_4$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | 0 | $-10/3$ | $-4/3$ | 0 | $-14/3$ | -3 | 0 | 0 | 0 | -30 |
| ← $S_1$ ($x_3$) | 0 | $2/3$ | $1/3$ | 1 | $2/3$ | $1/3$ | 0 | 0 | 0 | $10/3$ |
| $S_2$ | 0 | $2/3$ | $2/3$ | 0 | 2 | $-2/3$ | 0 | 0 | 0 | $190/3$ |
| $S_3$ | 0 | $-2/3$ | $1/3$ | 0 | $4/3$ | $-2/3$ | 0 | 0 | 0 | $100/3$ |
| $S_4$ | 0 | $-10/3$ | 0 | 0 | -10 | $-25/3$ | 0 | 0 | 0 | 150 |

**Final Result:**

Now we can see that this table is optimized as each entry in Z row is non positive and hence we are getting the optimized table after 1 iteration only so for this particular case

The cost ï Z=-30 i.e. Z=30K and Optimum testing Technique which will give the minimum value of Z is C(x3 is entered) and the resources should be used as the $1/3^{rd}$ of the total resources .

By following the above results the organization ABC can overcome the Testing Gap problem and hence testing can be done effectively on the specific software a company is using.

**Dummy Problem2:(while Opting Multiple Techniques)**

**XYZ Company is following software testing techniques in order to test a software. The data along with constraints(as required by the algorithm) is given as follows:**

| CONSTRAINTS | TESTING TECHNIQUES | | | |
|---|---|---|---|---|
| | A | B | C | D |
| RESOURCES | 1 | -2 | -2 | 4 |
| TIME | 2 | -1 | 1 | -3 |
| FEASIBILITY FACTOR | 4 | -2 | 1 | -1 |
| ACCURACY | 10 | 20 | 30 | 30 |
| COST ASSOCIATED(k units) | 8 | 6 | 3 | -3 |

Minimize Z= 8x1+6x2+3x3-3x4

Subject to:

x1-2x2-2x3+4x4<=40 (resource constraint)

2x1-x2+x3-3x4<=8(Time Constraint)

4x1-2x2+3x3-x4<=10(Feasibility factor constraint)

10x1+20x2+30x3+30x4<=100(Accuracy constraint)

Now multiplying both sides of Z function with -1 we will get

-Z = -8x1-6x2-3kx3+3x4

=> -Z+8x1+6x2+3x3-3x4=0

As all the constraints are of less than type so we need to add slack variables in order to make the constraints as consistent linear equations

Hence the constraints will be:

x1-2x2-2x3+4x+s1=40

2x1-x2+x3-3x4+s2=8

4x1-2x2+3x3-x4+s3=10

10x1+20x2+30x3+30x4+s4=100, where s1,s2,s3,s4 are the slack variables.

the first table for simplex minimization is as follows:

| Basic | Z | $x_1$ | $x_2$ | $x_3$ | $x_4$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | -1 | 8 | 6 | 3 | -3 | 0 | 0 | 0 | 0 | 0 |
| $S_1$ | 0 | 1 | 2 | -2 | 4 | 1 | 0 | 0 | 0 | 40 |
| $S_2$ | 0 | 2 | -1 | 1 | -3 | 0 | 1 | 0 | 0 | 8 |
| $S_3$ | 0 | 4 | -2 | 1 | -1 | 0 | 0 | 1 | 0 | 10 |
| $S_4$ | 0 | 10 | 20 | 30 | 30 | 0 | 0 | 0 | 1 | 100 |

Now we can observe that this table is not optimized because it is not satisfying the conditions of optimiality for simplex minimum i.e. (all elements of Z row is non positive),so we will perform iteration of simplex to optimize it.

Here x1(most positive) will be entering variable and s3 will be leaving variable(least positive intercept)

New pivot row= current pivot row/pivot element

New Row = (Current Row-Pivot Coloumn Coefficient)×new pivot row

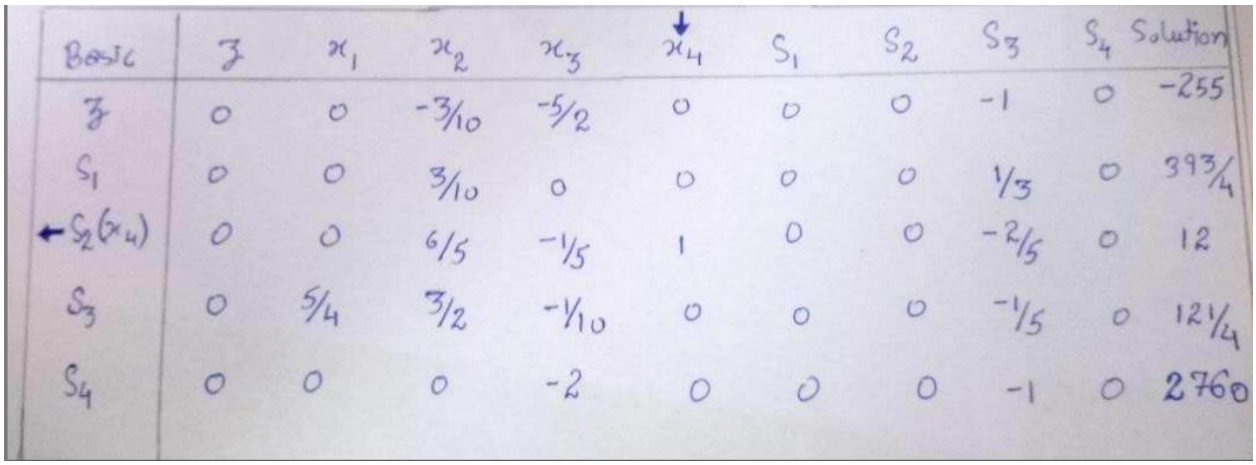
Hence the new table will be:

| Basic | Z | ↓ $x_1$ | $x_2$ | $x_3$ | $x_4$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | 0 | 0 | 1 | -5/4 | 5/4 | 0 | 0 | -2 | 0 | -20 |
| $S_1$ | 0 | 0 | -1/2 | -3/4 | -3/4 | 0 | 0 | -1/4 | 0 | 195/2 |
| $S_2$ | 0 | 0 | 3/2 | -1/4 | 5/4 | 0 | 0 | -1/2 | 0 | 15 |
| ← $S_3$ ($x_1$) | 0 | 1 | -1/2 | 1/4 | -1/4 | 0 | 0 | 1/4 | 0 | 5/2 |
| $S_4$ | 0 | 0 | -5 | 5 | -5 | 0 | 0 | -5/2 | 0 | 225 |

Again we can observe that this table is not optimized because it is not satisfying the conditions of optimality for simplex minimum i.e. (all elements of Z row is non positive),so we will perform iteration of simplex on the above table to optimize it.

Here x4(most positive) will be entering variable and s2 will be leaving variable(least positive intercept)

Again applying the same formula for simplex method we got the table of iteration 2 as:



| Basic | Z | $x_1$ | $x_2$ | $x_3$ | $x_4$ ↓ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | Solution |
|---|---|---|---|---|---|---|---|---|---|---|
| Z | 0 | 0 | -3/10 | -5/2 | 0 | 0 | 0 | -1 | 0 | -255 |
| $S_1$ | 0 | 0 | 3/10 | 0 | 0 | 0 | 0 | 1/3 | 0 | 393/4 |
| ← $S_2(x_4)$ | 0 | 0 | 6/5 | -1/5 | 1 | 0 | 0 | -2/5 | 0 | 12 |
| $S_3$ | 0 | 5/4 | 3/2 | -1/10 | 0 | 0 | 0 | -1/5 | 0 | 121/4 |
| $S_4$ | 0 | 0 | 0 | -2 | 0 | 0 | 0 | -1 | 0 | 2760 |

**Final Result:**

Now we can see that this table is optimized as each entry in Z row is non positive and hence we are getting the optimized table after 1 iteration only so for this particular case

The cost ï Z=-255 i.e. Z=255units and Optimum testing techniques un this case is more than one, for this project we can see that two testing techniques are participating and hence by using the proportion of A as 121/4 and D as 12 we can get the minimum optimized value for the testing and the absolute feasibility factor should be used 1 and 2/5 for A and D respectively.

By following the above results the organization ABC can overcome the Testing Gap problem and hence testing can be done effectively on the specific software a company is using.

So that in this way we deployed the algorithm for opting of multiple selection of testing technique also, and in this way the algorithm can be implied on any general case of software testing.

# RESULTS AND DISCUSSIONS

By the above algorithm which we just proposed and tried to implement it giving the following results:

1.) The best testing technique(s).
2.) The proportion in which testing is done with the selected technique
3.) The constraint which will acts as deciding factor that what are the maximum optimized values for the selected testing technique from the selected techniques.
4.) The value of minimized cost.

5.) The exact optimized values for the constraints and proportion for a testing technique.
6.) By getting these values the testing gap problem can be very much eradicated because the problem of testing gap deals only within the proper selection of technique, maintaining cost and constraints.

The following major points can be discussed while implementing the algorithm

1.) Degeneracy in Simplex Tables
2.) Circular Solutions
3.) Unbounded problems

These are the limitations for the algorithm which can reduce the quality of service for this proposed algorithm. Currently we are working on the problem of circular solutions which often come into the picture when we tried to solve some scenarios, rest 2 can be handled using the respected solution algorithm for the degeneracy and unbounded solutions.

Hence these are the major points of discussion which we need to consider in order to maintain the QOS of algorithm and also the accuracy of results of algorithm.

# CONCLUSION AND FUTURE SCOPE

So we can say that the result of algorithm is solving the identified problem up to a very much extent and hence the algorithm can be proposed in order to produce the best testing results which will really increase the quality of software and increase the robustness and reliability of the software, which results into the more satisfaction of end user which is the ultimate goal for any software development organization, also it will also reduce the maintenance cost of the software because if testing is done efficiently at the developing stage then there is very less chance of software to produce the errors which can effect the quality of software however further improvements are required and it is must in field of software development so maintenance costs is a vital cost for a software. But using the algorithm can maintain the balance between cost and utilization of constraints in the software development, hence we can say that the algorithm is providing the results which is capable of balancing the cost and other constraints of software testing with best possible results and also increasing the quality and reliability of software.

In future we are also trying to improve the algorithm by reducing its complexity to minimum so that the algorithm can be implied for every case which a organization can be facing during the software testing. More constraints are in process to be identified in order to maximize the accuracy of algorithm, As the no. of constraints increases the no. of possible testing technique considerations will also increase results into providing the more number of choices to test software by selecting from more options. This algorithm will help very much in maximizing the quality of software testing.

So by this we are winding up Algorithm For Reducing Testing Gap Problem at this point only. However future improvements and accuracy checks are needed on the algorithm to make the algorithm more efficient.

# REFERENCES

**Research papers:**

[1].) E. F. Miller, ñIntroduction to Software Testing Technology,ò Tutorial: Software Testing & Validation Techniques, Second Edition, IEEE Catalog No. EHO 180-0, pp. 4-16

[2].) www.cs.cmu.edu/luluo/Courses/17939Report.pdf