An Effective Risk Analysis in Requirement Engineering Using Modified Goal Model

Shankar Nayak Bhukya and Suresh Pabboju

1Faculty of Computer Science and Engineering, Osmania University, Hyderabad, Telangana, India.

2CBIT, Osmania University, Hyderabad, Telangana State, India.

Abstract

Goal-Orientation provides a rich framework for analyzing risks that occur in software projects in Requirements Engineering. This paper intends a modified goal model for the effective analysis of risks. The reliable (high probability of success) goals are obtained with the aid of Genetic Algorithm. Here Genetic Algorithm is utilized for deriving optimum candidate solution. Candidate solution is the collection or set of goals. When the selected Goals are the preeminent, the particular candidate solution is also the best. In our research, we utilize Genetic Algorithm for deriving excellent output. Genetic Algorithm has been a search and also optimization method according to Darwin’s principle that prefers the Best, and removes the Rest. A Genetic algorithm sustains a inhabitants of candidate solutions to the issue at hand, and facilitates it emerge by applying iteratively a series of stochastic operators for deriving a series of optimum solutions. Experimental outcomes present the efficacy of the suggested customized goal design.

Key Words: Goal-orientation, requirement analysis, genetic algorithm, darwin’s principle, optimal solution, candidate solution.
1. Introduction

Software project stoppage rates stay high alarmingly despite heaving investments in information systems and also their significance for current organizations. These projects are having huge risk activities because of the hasty pace of technological modifications and the organizational modifications they may enforce thence risk management has been salient for project victory [1]. It is a general experience, apart from what procedure is adopted, that gathering requirements is delayed through a premature leap for designing and also coding. Such leap is obliged by project managers, and, oddly, customers who comprehend poorly the aftermaths of omitting requirements [2]. The quantitative assessment of early software systems in the life cycle has not yet been a general practice for several software projects. Here, the software engineering community is lacking quantitative assessment techniques integrated properly amidst the software principles and the contemporary development techniques [3].

Risks can cover up a broad spectrum of disagreeable situations, like security hazards, security pressure and data inexactnesses amidst others. The elicitation and also assessment of such risks is the key requirements engineering procedure [4]. Risk assessment must thus be the key Requirements engineering procedure [9]. Requirements engineering has been a procedure related methodology for signifying, realizing, relating, modeling, documenting and sustaining software requisitions in software life cycle which assist for comprehending the issue better [5]. Requirements engineers are at the forefront of uncertainty problems. The adequacy and “sufficient” completeness of the identified software requirements, environment assumptions and appropriate domain properties is generally uncertain. The fulfillment rate of that requirements and also presumptions is also uncertain [6]. While goals express why a system is needed, threats tell us why security for our system is needed. Yet, you will frequently discover that intentions and intimidation are managed in separated modeling procedures are not perhaps being influenced by each other at all [7].

Software development projects, which are provided their miscellaneous and conceptual nature, proffer distinctive challenges and also risks. Projects have been salient to the actualization of performing business strategy of organization since projects are a way through which the company policy is executed. In such light projects have been unsafe and managers must capture suitable actions for restricting them from this hazardous status [8]. Incompleteness frequently provides the aftermath from lacking expectancy of unanticipated conditions below which the software must adequately behave. A normal inclination for conceiving over idyllic systems restricts unfavorable conditions from being appropriately identified and, whilst likely and critical, rectified by suitable countermeasures [9]. Regrettably, in several cases hazard mitigation needs the complete design revision and, initial requisition. For the later phases of the software development procedures Shifting risk assessment can provide the clear
benefit of considering risk deductions as an essential segment of the original design [11].

Usually risk assessment is utilized for investigating every consideration, that direct to the frailer in the program. It is a techniques and methodologies to document the effect of extenuation policies and to judge system exigency [12]. The general comprehending is that assaying risk needs methodologies for identifying the sources of the measures which drive a system or else an organization towards the states which can reveal it to risk. Thence, apart from the direct events which direct to hazardous conditions, the sequels of action directing such events, and even more significantly, the intent of such actions must be recognized and well comprehended [10]. Risks frequently evolve from a normal inclination for believing that the software and also it’s environ will forever perform as anticipated; no requisitions are engineered for the cases of the positive assumption that does not hold. Requisitions completeness thence provides risk assessment at the core of the Requirements Engineering procedures [13].

The sketch formation of the paper is planned as follows: Section 2 reviews the associated works regarding the suggested technique. In sections 3, a concise discussion regarding the suggested methodology is provided, section 4 assesses the Experimental outcomes and section 5 provides conclusion to the paper.

2. Literature Review

Shareeful Islam et. al [14] suggested a Goal-driven Software Development Risk Management Model (GSRM) and also its unambiguous incorporation into the requisitions engineering phase and also an empirical assessment outcome of utilizing GSRM as a project. They integrated the case analysis methodology with action research hence the outcomes from the case analysis directly contributed for managing the analyzed project risks and also identified ways for enhancing the intended methodology. The data was accumulated from multiple sources and analyzed both in a qualitative and also in quantitative way. When risk factors were beyond the power of the project manager and project environment, it was difficult to control those risks. The project capacity influenced every dimension of hazard. GSRM had been a sensible risk management methodology which is utilized in an industrial context. The consequences were matched against other analyzed outcomes for generalizing findings and also recognized contextual components. A prescribed early phase risk management practicing provided former warning associated with the issues that prevail in a project, and also it provides contribution to the complete project triumph.

Ebrahim Bagheri and Ali A. Ghorbani [15] provided a correspondence betwixt requirement qualifications and interpreted propositional conviction bases. Through that analogy, they were able to analyze the contents of a provided set of requirement collections known as viewpoints and specified whether they
were incomplete, illogical, or incompatible below a closed-world reasoning presumption. Regarding the introduced requisition collections’ properties, they denoted a perspective integration game by which the discrepancy of non-canonical requirement condition was determined. The proposed properties were conceptually categorized into two groups, namely essential and contingent properties. Essential properties reflected the attributes of a requirement collection being considered in vacuum, whereas contingent properties revealed its characteristics with regards to the remaining of its peer collections, which describe the same universe of discourse. The game comprises numerous rounds of conciliation and was executed by two salient functions, called choice and also augmentation functions. The aftermath of that game was a sequel of inconsistency-free requirement compilations that is amalgamated for creating a distinctive fair agent of the provided requirement compilations.

Vittorio Cortellessa et. al [16] devised a technique for assessment of performance-related risk component, which initiated from desecrations of performance requisitions (termed, performance failures). Also, in that paper, they proposed a technique for annotating UML diagrams having risk associated attributes and also translated those diagrams as designs that were equipped to be assayed. The risk component assessment that the merged the likelihood of a performance failure and also its severity. The technique focused the min scenarios in the comosoftware/hardware system and the bottleneck components in the scenarios which had the higher service demands. That was an important feedback for software designers, that (based on this information) devised more effort to the design and the implementation (or to the acquisition, in case of COTS) of several critical components. Their methodology laid on calculation of the asymptotic bounds expressed as functions of the customers number that took only some arithmetic operations. As the amount of calculations was self-determining the resources number in the design and the range of customer populations, the performance-related risk evaluation presented in that paper scales very well.

Richard W. Woolridge et. al [17] enhanced the Outcome-Based Stakeholder Risk Assessment Model (OBSRAM) for providing the realistic guidance to recognize and manage project risks evolving from the stakeholders. Here, OBSRAM proffered the project team a step-by-step procedure to recognize stakeholders throughout requirements engineering, recognizing stakeholder impacts on the project, recognizing the project’s influence on stakeholders, and assaying the risks that their possible negative responses pose. They demonstrated OBSRAM utilizing a case analysis of a simulated airline-crew-scheduling system project which focused to mitigate aircraft ground turnaround time to 30 minutes or else less. OBSRAM assured that more resources and also consideration were dedicated for eliminating the influence and perception shortages of stakeholders providing the hugest probable risk to the project and also the issue domain.
J.Pernstal et al. [18] suggested a lightweight Requirements Engineering structure and elucidated and assayed its industrial applicability responding to the demands of a Swedish automotive company to enhance particular issues in inter-departmental requisitions coordination and also communication in huge-scale improvement of software-intensive systems. Moreover, a case analysis procedure and a powerful validation were utilized for enhancing and assaying the structure in close teamwork with their manufacturing partner, concerning three actual-life cases in a continuing car project. Understanding and criticism were compiled through explanation whilst utilizing the framework and also from 10 senior industry professionals of a questionnaire and also in-depth following-up interviews.

The knowledge and criticism regarding utilizing the framework exposed that it was appropriate and related for the industry and efficient and beneficial method for resolving actual issues in organizing and also communicating requisitions recognized at the case company. Anyhow, other concerns, like accessibility to required resources and also competences in the former enhancement phases, were recognized whilst utilizing the technique, which permitted for former preemptive action for being taken.

David and Marc [19] suggested few of the significant empirical concerns that evolve in software procedures simulation modeling.

They initially addressed problems regarding real-world data utilized for (1) establishing input parameters for a software procedures simulation design, and (2) launch real organizational outcomes against the model’s outcomes (i.e., outputs) is matched.

The challenges comprise tiny sample sizes on the input side, considering variability and also outliers, lacking required data, limply defined metrics, and so on. On the output side, the paper dealt with (1) confirmation and also validation of the design, and (2) quantitative procedures for assaying design outputs assisting managerial decision making containing financial performance utilizing Net Present Value (NPV), multi-criteria utility functions, and also Data Envelopment Analysis (DEA).

The paper aimed on the stochastic modeling utilizing Monte Carlo simulation. Here, the paper was grounded in the experiences of authors’ practical application, and significant points were enlightened by examples derived from that field task.

3. Modified Goal Model

Modified goal model discovers the reliable goals with the assistance of Genetic Algorithm (GA). Figure 1 elucidates the flow of risk evaluation in the intended approach.
3.1. Encoding

Encoding (Binary Encoding, Octal Encoding and Hexadecimal Encoding) techniques can be utilized for deriving results. Here Binary Encoding is utilized for this issue. In Binary system Bits are used where each Bit denotes one gene. Group of all these genes is called chromosomes. All variables are created into strings. Here each string is a collection of bits. No. of bits required for each string is firmed by utilizing equation (1)

\[ 2 \exp(q) \geq \left[ \frac{u}{x(l)} \Delta x \right] + 1 \] (1)

Where, \( 2 \exp(q) \) is the number of bits required for strings, \( x(u) \) is the upper frontier and \( x(l) \) is the lower limit. Here candidate solution are generated which is combination of some node for fulfilling the prime node. The candidate solutions are subjected to Genetic Algorithm for obtaining optimum solution

3.2 Fitness Function

Fitness function is used in Genetic Algorithm for selecting the best from provided set \( F(x) = f(x) \) for maximization problem. Fitness function value of the string is known as string’s fitness. Accuracy is calculated using the equation (2).

\[ \frac{[X(U) - X(L)]}{[2 \ pow(4) - 1]} \] (2)

Where, \( X(U) \) is the higher limit and \( X(L) \) is the lesser limit
The binary coding and the parallel angles are provided by the equation (3).

\[ X(i) = X(L) + \left( U - X(L) \right) \frac{\text{pow}(4) - 1}{4} S(i) \]  

(3)

Where, \( S(i) \) is the decoded value of the \( i^{th} \) chromosome, \( X(U) \) is the higher limit of the binary string and \( X(L) \) is the lesser limit of the binary string. Here, every binary strings have been of length 4 bits.

### 3.3 Angle Calculation

For the calculation of angles, a 4-bit string is taken which has minimum value of 0 and a maximum value of 15. All these are ranging from 0000 to 1111. Now angles among the chromosomes can be calculated as follows:

- For the 0th chromosome ‘0000’ angle is computed utilizing the given
  \[ X(U) = 90, \quad X(L) = 0 \]

Here binary bits are used, in which has 4 bits, so

\[ X(0) = X(L) + \left( U - X(L) \right) \frac{\text{pow}(4) - 1}{4} S(0) = 0 \]

Here s(0) is the decoded value of ‘0000’ i.e., 0. So angle of the 0th chromosome is 0 degrees.

- For the 1st chromosome ‘0001’ Angle is calculated as given below.
  \[ X(1) = X(L) + \left( U - X(L) \right) \frac{\text{pow}(4) - 1}{4} S(1) = 6 \]

Here s(1) is the decoded value of ‘0001’ i.e., 1. So angle of the 1st chromosome is 6 degrees.

Calculation of angles is very salient to get best results. When two individuals are chosen in first phase of Genetic Algorithm, it is forever recommended for selecting individuals with greater angle difference to get best output. Since individuals selected with less angle difference or nearby angles then there is always a possibility of having many similarities in common and this reason may not proffer good results. This is equal to marrying two individual who are close relatives and this provide the aftermath in unhealthy child in most of the cases according to medical science. The same reason also applies to angles in Genetic Algorithm. So in the commencing phase of Genetic Algorithm when two individual are chosen for producing offspring which requires being unique and best, it is necessary that both individuals should have large angle difference.

This cannot be possible practically in our research since the number of goals selected in first phase is very small which is 18. In real time practical application we may have many goals and this will give us choice to select individuals with high difference angles and obtain best output.

### 3.4 Phases of Genetic Algorithm

Genetic algorithm uses 3 basic operators:
- Reproduction /Selection.
- Cross over.
- Mutation.

**Reproduction/Selection**

This is the primary step in Genetic Algorithm. Tournament selection method is utilized in this research topic. In Tournament Method, individuals are haphazardly chosen from the population and also the greatest of the individuals has been returned like a parent. Holding tournament competition amidst every individual is accomplished in this phase.

Here $G_1, G_2, \ldots, G_{10}$ individual variables are available.

**Step 1:** Choose two individuals haphazardly with divergent angles $\phi_2$ and $\phi_5$

Here $\phi_2$ and $\phi_5$ are fitness functions of $G_2$ and $G_5$ respectively. Here $\phi_2$ is selected with low Risk Score.

**Step 2:** Next select two individuals at random with different angles $\phi_4$ and $\phi_8$

Here $\phi_4$ and $\phi_8$ are fitness functions of $G_3$ and $G_8$ respectively. Here $\phi_8$ is selected with low Risk Score.

**Step 3:** Select another two individuals at random with different angles $\phi_1$ and $\phi_3$

Here $\phi_1$ and $\phi_3$ are fitness functions of $G_1$ and $G_3$ respectively. Here $\phi_3$ is selected with low Risk Score.

Similarly, other populations are chosen from the mating pool. From above, 3, 7, 8 and 6 are chosen only once. 1 and 4 are selected twice.

<table>
<thead>
<tr>
<th>Individuals</th>
<th>Selected</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 and 7</td>
<td>7</td>
</tr>
<tr>
<td>1 and 6</td>
<td>6</td>
</tr>
<tr>
<td>1 and 2</td>
<td>2</td>
</tr>
<tr>
<td>4 and 9</td>
<td>4</td>
</tr>
<tr>
<td>8 and 3</td>
<td>8</td>
</tr>
</tbody>
</table>

**Table 1: Selected Individuals**

**Calculation of Risk Score**

Table 2 depicts the calculation of score values for risk.
Table 2: Risk Score Calculation

<table>
<thead>
<tr>
<th>Goal</th>
<th>Cost</th>
<th>Impact</th>
<th>Probability</th>
<th>Probability Number</th>
<th>Risk Score ($P \times L$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>G1</td>
<td>20</td>
<td>L</td>
<td>5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>G2</td>
<td>10</td>
<td>O</td>
<td>4</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>G3</td>
<td>20</td>
<td>L</td>
<td>5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>G4</td>
<td>20</td>
<td>L</td>
<td>5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>G5</td>
<td>20</td>
<td>L</td>
<td>5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>G6</td>
<td>20</td>
<td>O</td>
<td>4</td>
<td></td>
<td>28</td>
</tr>
<tr>
<td>G7</td>
<td>0</td>
<td>R</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>G8</td>
<td>0</td>
<td>R</td>
<td>3</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>G9</td>
<td>20</td>
<td>L</td>
<td>5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>G10</td>
<td>10</td>
<td>O</td>
<td>4</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>G11</td>
<td>5</td>
<td>O</td>
<td>4</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>G12</td>
<td>5</td>
<td>O</td>
<td>4</td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>G15</td>
<td>25</td>
<td>L</td>
<td>5</td>
<td></td>
<td>40</td>
</tr>
</tbody>
</table>

First, reproduction operator chooses haphazardly a duo of two individual strings to mate, next a cross-site is chosen at random with the string length and also position values are swapped betwixt two strings followed by the cross-over. For instance, the two chosen strings in a mating pair is A=1111 and B=0000

**Crossover**

If the haphazard choice of a cross-site has been b two, next the new strings followed by cross-over might be $A^*$=1100 and $B^*$=0011. It is a single – site over. Here, a cross-site is randomly selected along the length of mated strings and also bits close to the cross-sites are substituted.

<table>
<thead>
<tr>
<th>Strings before mating</th>
<th>Cross site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent–1 → [1][0][1][1]</td>
<td>[1][1][1]</td>
</tr>
<tr>
<td>Parent–2 → [0][1][0][1]</td>
<td>[0][0][1]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Strings after mating</th>
<th>Cross site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Child–1 → [1][0][1][1]</td>
<td>[0][0][1]</td>
</tr>
<tr>
<td>Child–2 → [0][1][0][1]</td>
<td>[1][1][1]</td>
</tr>
</tbody>
</table>

**Matrix Crossover**

Strings are represented as a single dimensional array. Two strings in length 4 are concatenated for creating an individual.Whilst a cross over possibility of $P_c$ is utilized only $100P_c$ percent in the population are utilized in the cross over operation and $100(1 – P_c)$ percentage of the population remains since it is in the current population.
Although the best 100(1 – P_c ) percentage of the population is deterministically copied to the fresh population, it is usually preferred at random. A cross over operation is mainly responsible for the search of new strings.

**Inversion and Deletion**

In inversion, a string from the population is selected and bits between two random sites are inverted as follows:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Substring 1**

**Substring 2**

- **Linear+end - inversion:** it performs linear inversion with a specified probability of 0.75. If linear inversion was not performed, the end inversion would be performed with equivalent probability of 0.125 at either left or right end of the string.
- **Continuous inversion:** is utilized with specified inversion probability to each new individual when it is created.
- **Mass inversion:** no inversion takes place until a new population is created.

**Deletion and Duplication**

Any 2 or 3 bits at random, in order, is chosen and the former bits have been duplicated.

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>[1] 0</th>
<th>0</th>
<th>[1] 0</th>
<th>1</th>
<th>[0] Before Deletion</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>[1] 0</td>
<td>0</td>
<td>[1] 0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>[1] 0</td>
<td>0</td>
<td>[1] 0</td>
<td></td>
<td>[0] Duplication</td>
</tr>
</tbody>
</table>

**Deletion and Regeneration:** Genes between two cross-sites are deleted and regenerated.

| 1 | 0 | 0 | 1 | 1 | 0 | [1] |
| 1 | 0 | 0 | 1 | | 1 | Deletion |
| 1 | 0 | 1 | 1 | 0 | 1 | [1] | Regeneration |
Cross-over and Inversion

Before cross

0] [0] [1] [1] [0] | [0] [1] parent – 1

1] [1] [1] [0] [0] | [1] [1] parent – 2

After cross over

0] [1] [1] [0] [0] | [0] [1]

1] [0] [1] [1] [0] | [1] [1]

After cross over and

1] [0] [1] [1] [0] | [1] [1] inversion

Mutation Operator

After cross over, strings are subjected to mutation. Mutation of a bit associates flipping it, modifying 0 to 1 and vice versa having a minimal mutation probability Pm. The bit-wise mutation is bit-by-bit performed through flipping a coin having a probability of Pm. It is utilized for sustaining diversity of the population. The mutation in a bit does not influence the mutation probability of other bits.

Mutation Rate

Mutation Rate is utilized to calculate number of bits to be muted. The mutation operator conserves the diversity amidst the population. They are lesser in nature. Usually, simple Genetic algorithms uses population size of 20 to 200 with a mutation rate varies from 0.001 to 0.5.

Bitwise Operators

Binary coding is used in coding mechanism to generate algorithm structure. It involves coding of real variables to binary strings.

One’s Compliment Operator

It is ~ (unary operator). It changes 0 to 1 and vice versa.

For example: □ 1 □ 2

A = 0100 0001 → 4 1 24 16

~A = 1011 1110 → 11 14 66 84

Logical Bit-wise Operators

Bit-wise AND (&) operator: it returns 1 if both the bits are 1, otherwise returns zero.
Parent 1a = 1010 1010 → 10 10
Parent 2b = 1100 0011 → 12 3
Child a & b = 1000 0010 → 8 2

**Bit-wise exclusive –OR (^) operator**

Parent 1a = 1010 1010 → 10 10
Parent 2b = 1100 0011 → 12 3
Child a & b = 1000 0010 → 6 9

**Bit-wise OR (|) operator**

It returns a 1 if one or more bits have a value of 1 otherwise returns zero.

Parent 1a = 1010 1010 → 10 10
Parent 2b = 1100 0011 → 12 3
Child a & b = 1000 0010 → 13 11

**Shift Operators**

There have been two shift operators, left-shift(<<) and right-shift(>>) operators

- **Shift left operator (<<):** It causes all the bits in the initial operand for being transferred to the left by no. of positions denoted through the second operand.
  
  A = 1010 0110 → 10 6
  
  A<<2 = 1001 1000 → 9 8

- **Shift right operator (>>):** It performs reverse operation of the above operator.
  
  A = 1010 0110 → 10 6
  
  A>>2 = 0010 1001 → 2 9

**Masking**

Masking has been a procedure where a provided pattern is changed into another bit pattern through logical bit-wise operations. The innovative bit pattern has been one of the operands in the bit-wise operation. The next operand is termed mask, is a specifically chosen bit pattern which proffers the required transformation.

**Bit-Wise Operators Used In GA**

Populations are selected randomly for matting and on each pair bit-wise AND
and bit-wise OR operators are performed. Similarly,

AND and exclusive – OR (or) OR and exclusive - OR

The above operations can be performed to produce children or population for the subsequent generation.

4. Result and Discussion

This section discusses each stage’s results in the genetic algorithm, which includes reproduction/selection, crossover and mutation. The overall output of the genetic algorithm is also analyzed.

<table>
<thead>
<tr>
<th>Binary coding</th>
<th>Angle</th>
<th>Binary coding</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0</td>
<td>1001</td>
<td>54</td>
</tr>
<tr>
<td>0001</td>
<td>6</td>
<td>1010</td>
<td>60</td>
</tr>
<tr>
<td>0010</td>
<td>12</td>
<td>1011</td>
<td>66</td>
</tr>
<tr>
<td>0011</td>
<td>18</td>
<td>1100</td>
<td>72</td>
</tr>
<tr>
<td>0100</td>
<td>24</td>
<td>1101</td>
<td>78</td>
</tr>
<tr>
<td>0101</td>
<td>30</td>
<td>1110</td>
<td>84</td>
</tr>
<tr>
<td>0110</td>
<td>36</td>
<td>1111</td>
<td>90</td>
</tr>
<tr>
<td>0111</td>
<td>42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>48</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4 depicts the binary coding and its corresponding angles.

The angles for the corresponding encoding are calculated using the equation (3).

Selection
- Two pairs are selected haphazardly.
- The Risk score function is regarded like the fitness function score for the suggested Genetic Algorithm for optimizing the candidate solution.

Random solutions

<table>
<thead>
<tr>
<th>G3</th>
<th>G4</th>
<th>G8</th>
<th>G10</th>
<th>G6</th>
<th>G7</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
<td>G3</td>
<td>G5</td>
<td>G6</td>
<td>G7</td>
<td>G10</td>
</tr>
</tbody>
</table>

Cross over

In the above figure, the first two represent the two random candidate solutions, which are considered for the crossover process. Cross over point is chosen at random.

<table>
<thead>
<tr>
<th>Offspring 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>G2</td>
</tr>
</tbody>
</table>
Offspring 2

| G3 | G4 | G8 | G6 | G7 | G10 |

**Mutation**

Here, any character is haphazardly chosen from previous output and a new character is allocated to the selected one by replacing the existing. The figure presents the throughput of mutation.

**Offspring 1**

| G2 | G3 | G5 | G6 | G7 | G10 |

**New Offspring**

| G2 | G3 | G5 | G6 | G9 | G10 |

The output is given to first stage again and again and repeated n number of times for best results. Output obtained at the end of Genetic Algorithm after repeating n number of times is given below.

**Analysis of GA**

Solutions obtained after Genetic Algorithm is described below.

**Table 8: Candidate Solutions**

<table>
<thead>
<tr>
<th>Candidate solutions</th>
<th>Cost</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>S11-[G4, G7, G8, G9, G10]</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>S22-[G4, G6, G7, G8, G9]</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>S33-[G2, G7, G8, G9, G10]</td>
<td>21</td>
<td>8</td>
</tr>
<tr>
<td>S44-[G3, G4, G5, G6, G7]</td>
<td>22</td>
<td>5</td>
</tr>
<tr>
<td>S55-[G2, G3, G4, G5, G6, G7, G8]</td>
<td>35</td>
<td>4</td>
</tr>
<tr>
<td>S66-[G2, G4, G5, G6, G7, G8, G9, G10]</td>
<td>36</td>
<td>3</td>
</tr>
</tbody>
</table>

**Table 9: Cost to Risk Analysis**

Table 8 shows the generated candidate solutions and table 9 depicts the cost to risk analysis.

Genetic algorithms are in fact comprised of candidate solutions. The chromosomes present in each of the iteration represent candidates. The chromosome with the greatest risk score is not contained in the candidate solution list.
5. Conclusion

Genetic Algorithm is utilized for deriving optimized candidate solutions. For gleaning minimum probability in risk we subject the obtained throughput in former chapter to the initial phase of Genetic Algorithm. The obtained throughput from Goal Risk model is subjected to Extraction tree and Appropriate Algorithm. The output of this phase is given to Genetic Algorithm to obtain optimum Candidate solutions. New concept of angles and Risk score is used to select individuals from the given population. Selecting individuals are very significant since they decide the quality of the solution and also quality of next population. If best is selected as input then there is hundred percent guarantees that the best will be the output which is according to Darwin’s standard of natural Selection. The following step is use of tournament method which is utilized for selecting the individual for reproduction. The other two phases of Genetic Algorithm namely Crossover and mutation is applied to get perfect solutions. For deriving minimum probability and minimum impact in case of risk we are calculating the impact factor for the candidate solutions obtained from Genetic Algorithm in the following research.

Reference


[7] Per Hakon Meland, Erlend Andreas Gjære, Stephane Paul, The


[18] Joakim Pernstål, Tony Gorschek, Robert Feldt, Dan Florén, “Requirements communication and balancing in large-scale
software-intensive product development, Information and Software Technology 67 (2015), 44-64.
