1. Introduction

Reliability is central to productivity and effectiveness of real world industrial systems [22, 35]. To maximize productivity, the systems should always be available. However, it is difficult for an industrial system, comprising several complex components to survive over time since its reliability directly depends on the characteristics of its components. Failure is inevitable, such that system reliability optimization has become a key subject area in industry. Developing effective system reliability optimization is imperative. Two approaches for system reliability improvement are: (i) using redundant elements in subsystems, and (ii) increasing the reliability of system components.

Reliability-redundancy allocation maximizes system reliability via redundancy and component reliability choices [23], with restrictions on cost, weight, and volume of the resources. The aim is to find a trade-off between reliability and other resource constraints [22]. Thus, for a highly reliability system, the main problem is to balance reliability enhancement and resource consumption. A number of approaches in the literature focus on the application of metaheuristic methods for solving system reliability optimization problems [9, 7, 27, 15, 33, 34, 10, 13]. However, real-life reliability optimization problems are complex:

(i) management goals and the constraints are often imprecise;
(ii) problem parameters as understood by the decision maker may be vague; and,
(iii) historical data is often imprecise and vague.

Uncertainties in component reliability may arise due to variability and changes in the manufacturing processes that produce the system component. Such uncertainties in data cannot be addressed by probabilistic methods which deal with randomness. Therefore, the concept of fuzzy reliability is more promising [2, 4, 5, 6, 30, 31]. Contrary to
probabilistic models, fuzzy theoretic approaches address uncertainties that arise from vagueness of human judgment and imprecision [26, 3, 28, 1, 13, 14].

A number of methods and applications have been proposed to solve fuzzy optimization problems by treating parameters (coefficients) as fuzzy numerical data. [31, 11, 20, 21, 24]. In a fuzzy multi-criteria environment, simultaneous reliability maximization and cost minimization requires a trade-off approach. Metaheuristics are a potential application method for such complex problems [9]. Population based metaheuristics are appropriate for finding a set of solutions that satisfy the decision maker’s expectations. This calls for interactive fuzzy multi-criteria optimization which incorporates preferences and expectations of the decision maker, allowing for expert judgment. Iteratively, it becomes possible to obtain the most satisfactory solution.

In light of the above issues, the aim of this research is to address the system reliability optimization problem for a complex bridge system in a fuzzy multi-criteria environment. Specific objectives of the research are (1) to develop a fuzzy multi-criteria decision model for the problem; (2) to use an aggregation method to transform the model to a single-criteria optimization problem; and, (3) to develop a multi-criteria optimization method for the problem.

The rest of the paper is organized as follows: The next section describes the problem formulation for the complex bridge system. Section 3 provides a general fuzzy multi-criteria optimization modelling approach. In Section 4, a fuzzy multi-criteria genetic algorithm approach is proposed. Computational experiments, results and discussions are presented in Section 5. Section 6 concludes the paper.

2. Problem formulation

This section presents the mathematical formulation for the reliability optimization for a complex bridge system. In the real world, a typical complex bridge system [23] comprises five components or subsystems. The general structure of the complex bridge system is illustrated in Fig. 1.

![Fig. 1. The schematic diagram of the complex bridge system](image)

The aim is to maximize system reliability, subject to multiple linear constraints. In this respect, we present the following notations and assumptions;

**Notations:**
- \(m\) the number of subsystems in the system
- \(n_i\) the number of components in subsystem \(i, 1 \leq i \leq m\)
- \(\Xi(n_1, n_2, \ldots, n_m)\), the vector of the redundancy allocation for the system
- \(r_i\) the reliability of each component in subsystem \(i, 1 \leq i \leq m\)
- \(\rho\) the vector of the component reliabilities for the system
- \(q_i\) the failure probability of each component in subsystem \(i, 1 \leq i \leq m\)
- \(R(n_i) = 1 - q_i^{n_i}\), the reliability of subsystem \(i, 1 \leq i \leq m\)
- \(R_i\) the system reliability
- \(g_i\) the \(\rho\) constraint function
- \(w_i\) the weight of each component in subsystem \(i, 1 \leq i \leq m\)
- \(v_i\) the volume of each component in subsystem \(i, 1 \leq i \leq m\)
- \(c_i\) the cost of each component in subsystem \(i, 1 \leq i \leq m\)
- \(V\) the cost of the system
- \(W\) the upper limit on the weight of the system
- \(\alpha, \beta\) parameters in constraint functions of subsystem \(i\)
- \(\eta\) the system reliability

**Assumptions**

1. The availability of the components is unlimited;
2. The weight and product of weight and square of the volume of the components are deterministic;
3. The redundant components of individual subsystems are identical;
4. Failures of individual components are independent;
5. All failed components will not damage the system and are not repaired.

The problem can be formulated as a mixed integer nonlinear programming model as follows [8, 34, 35]:

\[
\begin{align*}
\text{Max} & \quad \eta(r,a) = R_1 R_2 + R_3 R_4 + R_5 R_6 + 2R_2 R_4 R_6 R_8 - 2R_2 R_4 R_6 - R_2 R_4 R_8 - R_2 R_4 R_5 - R_2 R_4 R_6 - R_2 R_4 R_8 \\
\text{Subject to:} & \\
& g_1(r,a) = \sum_{i=1}^{n} w_i q_i^2 \leq V \\
& g_2(r,a) = \sum_{i=1}^{n} a_i (-1000/ln n)^{\beta} (n_i + \exp(n_i/4)) \leq C \\
& g_3(r,a) = \sum_{i=1}^{n} w_i \exp(n_i/4) \leq W \\
& 0 \leq r_i \leq 1, \quad n_i \text{ positive integer}, \quad 1 \leq i \leq m
\end{align*}
\]

where, \(\eta(\cdot)\) denotes the system reliability, and expressions \(g_1(\cdot), g_2(\cdot),\) and \(g_3(\cdot)\) represent the total volume, cost, and weight of the system, respectively.

In the next section, we propose a general approach to fuzzy multi-criteria optimization, in the context of system reliability optimization.

3. Fuzzy multi-criteria optimization modelling

In a fuzzy environment, the aim is to find a trade-off between reliability, cost, weight and volume. A common approach is to simultaneously maximize reliability and minimize cost. Constraints are transformed into objective functions, so that reliability and other cost functions can be optimized jointly. This is achieved through the use of membership functions, which are easily applicable and adaptable to the real life decision process.

In general, the fuzzy multi-criteria optimization problem can be represented by the following [13, 29]:

\[
\begin{align*}
\text{Min} & \quad \tilde{\eta}(x) \\
\text{Subject to:} & \\
& g_k(x) \leq \alpha_k \quad \text{or} \quad \alpha_k \geq 0 \quad z = 1, 2, \ldots, p \\
& u_q \leq x_q \leq \bar{u}_q \quad q = 1, 2, \ldots, Q
\end{align*}
\]

where, \(x = (x_1, x_2, \ldots, x_Q)^T\), is a vector of decision variables that optimize a vector of objective functions, \(\tilde{\eta}(x) = [\tilde{\eta}_1(x), \tilde{\eta}_2(x), \ldots, \tilde{\eta}_n(x)]\) are
4. Fuzzy set theory permits gradual assessment of membership, in terms of a suitable function that maps to the unit interval $[0,1]$. Membership functions such as Generalized Bell, Gaussian, Triangular and Trapezoidal can represent the fuzzy membership [31]. Linear membership functions can provide good quality solutions with much ease, including the widely recommended triangular and trapezoidal membership functions [6, 8, 11, 30, 31]. Thus, we use linear functions to define fuzzy memberships of objective functions (or decision criteria).

Let $a_i$ and $b_i$ denote the minimum and maximum feasible values of each objective function $\eta_i(x)$, $i = 1, 2, \ldots, h$, where $h$ is the number of objective functions. Let $\mu_{\eta_i}$ denote the membership function corresponding to the objective function $f_i$. Then, the membership function corresponding to minimization and maximization is defined based on satisfaction degree. Fig. 2 illustrates the linear membership functions defined for minimization and maximization.

\begin{align*}
\mu_{\eta_i}(x) &= \begin{cases} 
1 & \eta_i(x) \leq b_i \\
\frac{b_i - \eta_i(x)}{b_i - a_i} & a_i \leq \eta_i(x) \leq b_i \\
0 & \eta_i(x) \geq b_i
\end{cases} \\
&\quad \quad \text{(3)}
\end{align*}

The linear membership function shown in Fig. 2(b) is suitable for representing profit functions that should be maximized. The membership function is represented as follows;

\begin{align*}
\mu_{\eta_i}(x) &= \begin{cases} 
1 & \eta_i(x) \geq b_i \\
\frac{\eta_i(x) - a_i}{b_i - a_i} & a_i \leq \eta_i(x) \leq b_i \\
0 & \eta_i(x) \leq a_i
\end{cases} \\
&\quad \quad \text{(4)}
\end{align*}

Having defined the fuzzy model using membership functions, the corresponding crisp model is formulated. Fuzzy evaluation enables FMGA to cope with infeasibilities which is otherwise impossible with crisp formulation. This gives the algorithm speed and flexibility, which ultimately improves the search power of the algorithm.

4. A fuzzy multi-criteria genetic algorithm approach

FMGA is an improvement from the classical genetic algorithm (GA). GA is a stochastic global optimization technique that evolves a population of candidate solutions by giving preference of survival to quality solutions, while allowing some low quality solutions to survive, to maintain diversity in the population [18]. Each candidate solution is coded into a string of digits, called chromosomes. New offspring are obtained from probabilistic genetic operators, such as selection, crossover (at probability $p_c$), mutation (at a probability $p_m$), and inversion [16]. A comparison of new and old (parent) candidates is done based on a given fitness function, retaining the best performing candidates into the next population. Characteristics of candidate solutions are passed through generations using genetic operators. The overall flow of the FMGA is presented in Fig. 3.

**Algorithm 1. The FMGA Pseudo-Code**

1. Begin
2. Input FMGA parameters: $p_c$, $p_m$, popsize, $w_1$, $w_2$, $w_3$, ...
3. Initialize pop: $i = 0, \Pi(0)$
4. Repeat
5. Selection()
6. Evaluation ($F(0)$)
7. Create temporal population, $temppop(0)$
8. Crossover()
9. Select 2 chromosomes from temppop($0$)
10. Apply crossover operator, repair if necessary
11. Mutation()
12. Mutate $P(0)$
13. Add offspring to $newpop(0)$
14. Replacement()
15. Compare successively, $newpop(0)$ and $oldpop(0)$ strings;
16. take the ones that fare better;
17. Select the rest of the strings with probability 0.52
18. Diversification()
19. Calculate population diversity $h_i$
20. While ($i < k$)
21. $diversity(0)$
22. Recalculate $h_i$
23. End While
24. Evaluation ($F(0)$)
25. New population()
26. $oldpop(0) = newpop(0)$
27. Advance population, $i = i + 1$
28. Until (termination criteria satisfied)
29. End

4.1. Chromosome coding

Traditionally, candidate solutions were encoded as binary strings. In the FMGA, each candidate solution is encoded into a chromosome using the variable vectors $n$ and $r$. An integer variable $n_i$ is coded as a real variable and transformed to the nearest integer value upon objective function evaluating.

4.2. Initialization and evaluation

An initial population of the desired size, $pop$, is randomly generated. FMGA then computes the objective function for each string (chromosome). The string is then evaluated according to the overall objective function in the model.

To improve flexibility and to incorporate the decision maker’s preferences into the model, we introduce user-defined weightings, $w = [w_1, w_2, \ldots, w_h]$. We use the max-min operator to aggregate the membership functions of the objective functions, incorporating expert opinion. Thus, from models (1) and (2), constraints $g_1(\cdot)$, $g_2(\cdot)$, and $g_3(\cdot)$ which represent volume, cost, and weight, respectively, are transformed into objective functions using the fuzzy membership functions. This leads to a multiple criteria system reliability optimization model, consisting of five criteria namely, reliability, volume, cost, and weight. In addition, the model is converted into a single objective optimization model as follows:

**Fig. 2. Fuzzy membership functions for $\eta_i(x)$**

**Fig. 3. The overall pseudo-code of the FMGA**
Max \( \left( \frac{1 \land \lambda_1(x)}{w_1} \right) \land \left( \frac{1 \land \lambda_2(x)}{w_2} \right) \land \ldots \land \left( \frac{1 \land \lambda_h(x)}{w_h} \right) \)

Subject to:
\[ \lambda_t(x) = \mu_{\eta_t}(x) \quad w_t \in [0,1] \quad t = 1, \ldots, h \]
\[ v_q \leq x_q \leq \bar{v}_q \quad q = 1, 2, \ldots, Q \]

Here, \( \mu_{\eta_t}(x) \) is a set of fuzzy regions that satisfy the objective functions \( \lambda_t \) which denote the degree of satisfaction of the \( t \)th objective; \( x \) is a vector of decision variables; \( w_t \) is the weighting of the \( t \)th objective function; and symbol “\( \land \)” is the aggregate min operator. Thus, expression (1) gives the minimum between 1 and \( \lambda_t(x)/w_t \). Though \( \lambda_t(x) \) are in the range [0,1], the value of \( \lambda_t(x)/w_t \) may exceed 1, howbeit, the final value of \( (1 \land \lambda_t(x)/w_t) \) will always lie in [0,1]. A FMGA approach is used to solve the model.

4.3. Selection and crossover

Several selection strategies have been suggested in [16]. The remainder stochastic sampling without replacement is preferred; each chromosome \( j \) is selected and stored in the mating pool according to the expected count \( e_j \):

\[ e_j = \frac{f_j}{\sum_{j=1}^{pop} f_j} \]

where, \( f_j \) is the objective function value of the \( j \)th chromosome. Each chromosome receives copies equal to the integer part of \( e_j \), while the fractional part is treated as success probability of obtaining additional copies of the same chromosome into the mating pool.

Genes of selected parent chromosomes are partially exchanged to produce new offspring. We use an arithmetic crossover operator which defines a linear combination of two chromosomes [25][29]. Assume a crossover probability of 0.41. Let \( p_1 \) and \( p_2 \) be two parents randomly selected for crossover. Then, the resulting offspring, \( q_1 \) and \( q_2 \), are given by the following expression;

\[ q_1 = e p_1 + (1-e) p_2 \]
\[ q_2 = (1-e) p_1 + e p_2 \]

where, \( e \) represents a random number in the range [0,1].

4.5. Mutation

Mutation is applied to every new chromosome so as to maintain diversity of the population, howbeit, at a very low probability. A uniform mutation probability rate of 0.032 is applied.

4.6. Replacement

At each generation or iteration, new offspring may be better or worse. As a result, nonperforming chromosomes should be replaced. A number of replacement strategies exist in the literature, e.g., probabilistic replacement, crowding strategy, and elitist strategy [26]. The proposed FMGA uses a hybrid of these strategies.

4.7. Termination

The FMGA algorithm uses two termination criteria to stop the iterations: when the number of generations exceeds the user-defined maximum iterations, and when the average change in the fitness of the best solution over specific generations is less than a small number, which is \( 10^{-5} \).

5. Computational illustrations

This section presents the computational experiments, results and discussions based on benchmark problems in [17, 19].

5.1. Computational experiments

We use the parameter values in [23] and define the specific instances of the problems as shown in Table 1.

<table>
<thead>
<tr>
<th>i</th>
<th>( 10^3 q_i )</th>
<th>( \beta_i )</th>
<th>( W_i )</th>
<th>( w_i )</th>
<th>( V )</th>
<th>( C )</th>
<th>( W )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.330</td>
<td>1.5</td>
<td>1</td>
<td>7</td>
<td>110</td>
<td>175</td>
<td>200</td>
</tr>
<tr>
<td>2</td>
<td>1.450</td>
<td>1.5</td>
<td>2</td>
<td>8</td>
<td>110</td>
<td>175</td>
<td>200</td>
</tr>
<tr>
<td>3</td>
<td>0.541</td>
<td>1.5</td>
<td>3</td>
<td>8</td>
<td>110</td>
<td>175</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>8.050</td>
<td>1.5</td>
<td>4</td>
<td>6</td>
<td>110</td>
<td>175</td>
<td>200</td>
</tr>
<tr>
<td>5</td>
<td>1.950</td>
<td>1.5</td>
<td>2</td>
<td>9</td>
<td>110</td>
<td>175</td>
<td>200</td>
</tr>
</tbody>
</table>

The FMGA was implemented in JAVA on a 3.06 GHz speed processor with 4GB RAM. The FMGA crossover and mutation parameters were set at 0.45 and 0.035, respectively. A two-point crossover was used in this application. The population size was set to 20, and the maximum number of generations was set at 500. The termination criteria was controlled by either the maximum number of iterations, or the order of the relative error set at \( 10^{-5} \), whichever is earlier. Whenever the best fitness \( f^* \) at iteration \( t \) is such that \( f_t - f^* \leq \varepsilon \) is satisfied, then five best solutions are selected; where \( \varepsilon \) is a small number, which was set at value \( \varepsilon = 10^{-5} \) for the computational experiments.

Expression (5) is used to solve benchmark problems. A fuzzy region of satisfaction is constructed for each criterion, that is, system reliability, cost, volume, and weight, denoted by \( \lambda_1, \lambda_2, \lambda_3, \lambda_4 \), respectively. By using the constructed membership functions together with their corresponding weighting vectors, an equivalent crisp optimization formulation is obtained [29]:

Max \( \left( \frac{1 \land \lambda_1(x)}{w_1} \right) \land \left( \frac{1 \land \lambda_2(x)}{w_2} \right) \land \ldots \land \left( \frac{1 \land \lambda_h(x)}{w_h} \right) \)

Subject to:
\[ \lambda_t(x) = \mu_{\eta_t}(x) \quad t = 1, \ldots, 4 \]
\[ 0.5 \leq \eta_t \leq 1 - 10^{-6} \quad \eta_t \in [0,1] \]
\[ 1 \leq \eta_{t+1} \leq 10 \quad \eta_{t+1} \in \text{positive integers} \]
\[ 0.5 \leq R_{t+1} \leq 1 - 10^{-6} \quad R_{t+1} \in [0,1] \]

The set \( \omega = \{ \omega_1, \omega_2, \omega_3, \omega_4 \} \) are user-defined weightings in the range \([0,2]\) that indicate the bias towards specific decision criteria. To illustrate, given the weighting set \( \omega = [1,1,1,1] \), the expert user expects no bias towards any criterion. On the contrary, set \( \omega = [1,0.4,0.4,0.4] \) indicates preferential bias towards the region with higher reliability values as compared to the rest of the criteria equally weighted at 0.4. Consequently, the decision process considers the expert opinion and preferences of the decision maker.

Rather than prescribing a single solution to the user or decision maker, the FMGA interactively provides a population of near-optimal solutions.
Two experiments were conducted: Experiment 1 and Experiment 2.

5.1.1 Experiment 1
The aim of experiment 1 was to demonstrate the performance of the FMGA algorithm over time. As such, the algorithm was executed for 500 iterations, to show the results of intermediate solutions over time. A graphical analysis of the results was presented to show the performance behaviour of the algorithm.

5.1.2 Experiment 2
This purpose of experiment 2 was to make a comparative analysis of the performance of the FMGA algorithm against best known algorithms in the literature. Thus, the algorithm was executed 25 times, and the best five solutions were selected. The experimental results were compared with best known algorithms in [17] and [19], based on four performance criteria namely, reliability \( R_s \), cost \( C_s \), weight \( W_s \), and volume \( V_s \).

For further comparative analysis, an improvement measure is defined. The percentage improvement \( I \) is defined according to the following expression:

\[
I = \left( \frac{v_s - v_{best}}{v_{best}} \right) \times 100\% \tag{9}
\]

where, \( v_s \) and \( v_{best} \) represent the FMGA solution value and the best known solution from literature. Computational results and the ensuing discussions are presented in the next section.

5.2 Computational results and discussions
This section presents the results of the computational experiments outlined in the previous section.

### Table 2. Minimum and maximum values of objective functions

<table>
<thead>
<tr>
<th>( b_s )</th>
<th>( n_1 )</th>
<th>( n_2 )</th>
<th>( n_3 )</th>
<th>( n_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>180</td>
<td>190</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>60</td>
<td>70</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

5.2.1. Experiment 1 results
Computational results from experiment 1 showed the performance of FMGA as compared to other best known algorithms. The best five FMGA solutions were compared with the best results obtained from the literature [8][35].

Tables 3 presents the best five FMGA solutions, and the best known solutions obtained from [8] (with system reliability \( R_s = 0.999958830 \), cost \( C_s = 175.00 \), weight \( W_s = 195.7352300 \), and \( V_s = 92.00 \)). It can be seen that, based on system reliability, cost, weight and volume, the five FMGA solutions are better than the best known results, except for a single weight value from solution \( S_4 \) (that is, 196.98273245) which is slightly higher than the best known (that is, 195.7352300).

Further, all the five best FMGA solutions outperformed the solutions in [35], based on all performance criteria. This indicates that, overall, the FMGA performs better than the previous algorithms.

Table 4 presents the percentage improvement of the FMGA solutions, using the best known results as benchmarks. The improvements in reliability, cost, weight and volume are denoted by \( I_R, I_C, I_W \) and \( I_V \).
Table 4. Percentage improvement of FMGA solutions over best known results

<table>
<thead>
<tr>
<th>Improvement</th>
<th>S₁</th>
<th>S₂</th>
<th>S₃</th>
<th>S₄</th>
<th>S₅</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>I₉</td>
<td>0.0034</td>
<td>0.0027</td>
<td>0.0017</td>
<td>0.0029</td>
<td>0.0014</td>
<td>0.0024</td>
</tr>
<tr>
<td>I₈</td>
<td>0.0003</td>
<td>0.0000</td>
<td>0.0076</td>
<td>0.0001</td>
<td>0.1046</td>
<td>0.0363</td>
</tr>
<tr>
<td>I₇</td>
<td>-0.6402</td>
<td>7.9698</td>
<td>9.3603</td>
<td>15.5321</td>
<td>0.1025</td>
<td>6.4649</td>
</tr>
<tr>
<td>I₆</td>
<td>27.1739</td>
<td>17.3913</td>
<td>21.7391</td>
<td>34.7826</td>
<td>15.2174</td>
<td>23.2609</td>
</tr>
</tbody>
</table>

The FMGA method addresses the conflicting multiple objectives of the problem, giving a trade-off between the objectives; The method gives a population of alternative solutions, rather than prescribe a single solution; The method is practical, flexible and easily adaptable to problem situations.

In view of the above advantages, FMGA is a useful decision support tool for the practicing decision maker in system reliability optimization, especially in a fuzzy environment.

6. Conclusions

Decision makers in system reliability optimization seek to satisfy reliability enhancement and cost minimization. In a fuzzy environment, management goals and constraints are often imprecise and conflicting. One most viable and useful option is to use a fuzzy satisficing approach that includes the preferences and expert judgments of the decision maker. This study provided a multi-criteria optimization of a complex bridge system. Using fuzzy multi-criteria evaluation, the model is converted into a single-objective model. Thus, FMGA uses fuzzy evaluation to find the fitness of candidates in each population. Illustrative computational experiments showed that the FMGA approach is highly capable of providing near optimal solutions.

Contrary to single-objective approaches which optimize system reliability only, FMGA provides satisficing solutions in the presence of fuzzy multiple criteria. Furthermore, the algorithm provides a population of good alternative solutions, which offers the decision maker a wide choice of practical solutions and an opportunity to consider other practical factors not included in the formulation. Therefore, the approach gives a robust method for system reliability optimization.

A fuzzy based approach is especially essential, given that, at design stage, the desired design information is not precisely known, which makes the problem rather ill-structured. As such, reliance on human experience and expert information is unavoidable. FMGA uses fuzzy theory concepts to effectively model the vagueness and imprecision of the expert knowledge, taking into account the conflicting multiple criteria. Computational results and comparative analysis showed that the proposed algorithm is more effective than best known algorithms in the literature.

Acknowledgement

This research work was financed by the University of Johannesburg, Johannesburg, South Africa, in the year 2015.

References


