PERGAMON

Computers & Industrial Engineering 41 (2002) 423–440

computers & industrial engineering

www.elsevier.com/locate/dsw

Genetic algorithm to maximize a lower-bound for system time-to-failure with uncertain component Weibull parameters

David W. Coit[a,*], Alice E. Smith[b]

[a]*Department of Industrial Engineering, Rutgers University, 96 Frelinghuysen Rd., Piscataway, NJ 08854-8018, USA*
[b]*Department of Industrial and Systems Engineering, Auburn University, Auburn, AL 36849, USA*

Accepted 1 October 2001

Abstract

A genetic algorithm (GA) is used to solve the redundancy allocation problem when the objective is to maximize a lower percentile of the system time-to-failure distribution and the available components have random Weibull scale parameters. The GA searches the prospective solution space using an adaptive penalty to consider both feasible and infeasible solutions until converging to a feasible recommended system design. The objective function is intractable and a bi-section search is required as a function evaluator. Previously, this problem has most often been formulated to maximize system reliability instead of a lower-bound on system time-to-failure. Most system designers and users are risk-averse, and maximization of a lower percentile of the system time-to-failure distribution is a more conservative strategy (i.e. less risky) compared to maximization of the mean or median of the time-to-failure distribution. The only previous research to consider a lower percentile of system time-to-failure, also required that all component Weibull parameters are known. Those findings have been extended to address problems where the Weibull shape parameter is known, or can be accurately estimated, but the scale parameter is a random variable. Results from over 90 examples indicate that the preferred system design is sensitive to the user's perceived risk.

Keywords: Genetic algorithm; Redundancy allocation problem; Reliability optimization; Weibull parameters

1. Introduction

The redundancy allocation problem involves the selection of components and levels of redundancy to maximize some defined objective function. In this paper, the problem has been formulated to maximize a $\alpha \times 100\%$ lower percentile of the system time-to-failure distribution, where α is the system user's risk

* Corresponding author. Fax: +1-732-445-5467.
E-mail address: coit@rci.rutgers.edu (D.W. Coit).

PII: S0360-8352(01)00066-3

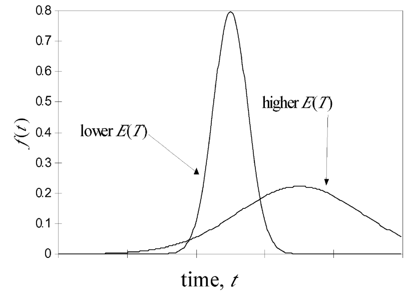

Fig. 1. Comparison of system time-to-failure density functions.

level. The formulation presented here is unique because users are not required to know the Weibull scale parameter explicitly. It is assumed that the Weibull scale parameter is distributed according to different defined distributions for the available components. A genetic algorithm (GA) is used to search the solution space and recommend solutions to the problem. This approach offers distinct benefits because it does not require the specification of a mission time and it incorporates designer and system user risk.

System designers and product users are generally risk-averse. If some tangible subset of the population fails very early, the product will be viewed as unreliable by many consumers, even if the mean or median time-to-failure is very high. This will be particularly true if the implications of failure are severe. A conservative design strategy is to select the design that maximizes a $(1 - \alpha) \times 100\%$ lower-bound on system time-to-failure, i.e. the $\alpha \times 100\%$ percentile of the system time-to-failure distribution. This provides better assurances that even the less reliable members of the population are satisfactory.

Consider the system time-to-failure distributions presented in Fig. 1 for two functionally equivalent systems. A risk-neutral system designer would prefer the system with the higher mean-time-to-failure (i.e. $E(T)$). For risk averse designers, the choice is less clear. If the implications of failure are very dire, then the choice in Fig. 1 with the lower $E(T)$ might actually be preferable. For this design alternative, there is a longer time period after first purchasing the product where the probability of failure is very small ($<\alpha \times 100\%$). After that, it may be necessary to replace the system or to perform preventive maintenance. α is a user selected risk level and depends on the consequence of an early failure.

2. Redundancy allocation problem

The design of new products involves the specification of performance requirements, the evaluation

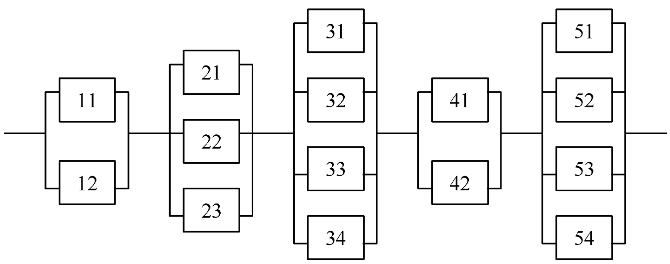

Fig. 2. Sample series–parallel system.

and selection of components to perform clearly defined functions and the determination of a system-level architecture. Detailed system engineering specifications prescribe minimum levels of reliability, maximum weight, maximum volume, etc. If the design is to be produced economically or within some specified budget, different design alternatives must be considered, resulting in a complex combinatorial optimization problem.

The redundancy allocation problem pertains to a system of s subsystems in series. For each subsystem, there are m_i functionally equivalent components, with different levels of cost, weight, reliability and other characteristics, which may be selected. There is an unlimited supply of each of the m_i choices. A minimum of one component must be chosen for each subsystem, but it is often advantageous to add redundant components. An example series–parallel system is depicted as Fig. 2. The use of redundancy improves system reliability but adds to system cost, weight, etc. There are system-level constraints and the problem is to select the design configuration that maximizes some stated objective function.

The redundancy allocation problem has been shown to be NP-hard by Chern (1992). It has been solved using dynamic programming (Fyffe, Hines, & Lee, 1968; Nakagawa & Miyazaki, 1981), integer programming (Bulfin & Liu, 1985; Gen, Ida, & Lee, 1990) and GAs (Campbell & Painton, 1996; Coit & Smith, 1996; Coit & Smith, 1998; Ida, Gen, & Yokota, 1994; Painton & Campbell, 1995; Rubinstein, Levitin, Lisnianski, & Ben-Haim, 1997).

System reliability is a convenient objective function but it is not appropriate for all design problems. The logarithm of system reliability for a series–parallel system is a separable function and dynamic programming or integer programming (with appropriate transformations) can be used to determine optimal solutions to the problem. When there is no mission time to compute component and system reliability, these algorithms are no longer applicable.

If there is no obvious choice for mission time, design evaluation and optimization should be based on comparisons of the system time-to-failure distributions (as opposed to the reliability for one distinct time period). It is common for many industries, including defense, railroad and automotive industries, to use the mean-time-to-failure as a performance measure. It is less common to use a $\alpha \times 100\%$ percentile of the time-to-failure distribution; however, this can be more informative because it incorporates risk. It is generally insufficient to design and manufacture products that are highly reliable 'on average'. It is necessary for a large percentage of the products to achieve some minimally acceptable performance level as well.

Campbell and Painton (1996) and Painton and Campbell (1995) have solved a reliability optimization problem that incorporates risk. In their work, component time-to-failure is distributed according to an exponential distribution, but the distribution parameter itself is a random variable. They used a GA to maximize a lower percentile of the mean time between failure (MTBF) for a fixed system structure, given defined component reliability improvement levels and repair assumptions.

Rubinstein et al. (1997) also use a GA to solve a redundancy allocation problem with uncertain component properties. In their work, the expected system reliability is maximized given the uncertainty. This is one of the few papers that explicitly recognize reliability estimation uncertainty. However, it assumes that decision makers are risk-neutral and that there is a well-defined mission time that can be used to compute reliability.

Ushakov and Harrison (1994) and Gnedenko and Ushakov (1995) present algorithms to maximize the median time-to-failure (used as a surrogate for MTTF). They present heuristics that can be applied if there is a significant amount of standby redundancy. Nakashima and Yamato (1977) solve an analogous problem to maximize the time period where system reliability remains above a preselected value. Their algorithm assumes that components have exponential time-to-failure, but that the distribution parameters are the decision variables to be determined in addition to the redundancy levels.

Coit and Smith (1998) used a GA to find solutions to the redundancy allocation problem to maximize a lower-percentile of the system time-to-failure distribution. In this work, Weibull distribution parameters were required for all components. In practice, this can be a difficult requirement to satisfy. The Weibull shape parameter is often known or can be estimated accurately, but it is often difficult to estimate the scale parameter. A more realistic problem formulation is to recognize the uncertainty of Weibull distribution parameters. Considering a Bayesian perspective, the uncertainty distribution can be considered as a prior distribution.

2.1. Formulation to maximize a lower-bound on system time-to-failure

A formulation to maximize a percentile of the system time-to-failure distribution is presented as Problem P1. $T(\mathbf{x})$ is the system time-to-failure for design solution $\mathbf{x}$ (with random component Weibull scale parameters). $T(\mathbf{x})$ is a random variable. The distribution of $T(\mathbf{x})$ depends on the components and redundancy levels selected for a particular system design. The time-to-failure for each available component is distributed according to a two-parameter Weibull distribution with a known shape parameter and a scale parameter distributed in accordance with some distribution.

T_{ij} = time-to-failure for the jth component used for subsystem i
$T_{ij} \sim \text{Weibull}(\lambda_{ij},\beta_{ij})$
$F(t; \lambda_{ij}, \beta_{ij}) = 1 - \exp(-\lambda_{ij} t^{\beta_{ij}})$
β_{ij} = Weibull shape parameter for jth component for subsystem i
λ_{ij} = Weibull scale parameter for jth component for subsystem i
$\lambda_{ij} \sim F(\cdot)$
$F(\cdot)$ = a defined distribution

Problem P1 is as follows.

Problem P1. $\max T_{1-\alpha}(\mathbf{x})$

subject to $\mathbf{A}\mathbf{x} \leq \mathbf{b}$

$$1 \leq \sum_{j=1}^{m_i} x_{ij} = n_i \leq n_{\max,i} \quad \forall\, i$$

$$x_{ij} \in \{0, 1, 2, \ldots\}$$

where, $T_{1-\alpha}(\mathbf{x})$ is the $\alpha \times 100\%$ percentile of marginal distribution for system time-to-failure, $\mathbf{x}$ the $(x_{11}, x_{12}, \ldots, x_{1,m1}, x_{21}, x_{22}, \ldots, x_{2,m2}, x_{31,\ldots}, x_{s,ms})^T$, x_{ij} the quantity of the jth available component used in subsystem i, m_i the number of available components for subsystem i, n_i the number of components used within ith subsystem, $n_{\max,i}$ the maximal allowable number of component within ith subsystem, s the number of subsystems, and α is risk level.

$\mathbf{A}$ is a $q \times r$ matrix where q is the number of linear constraints and r is the number of x_{ij} decision variables. $\mathbf{b}$ is a q-dimensional vector defining the constraint limits. Typical constraints may include a cost budget which can not be exceeded or the maximum acceptable weight or volume.

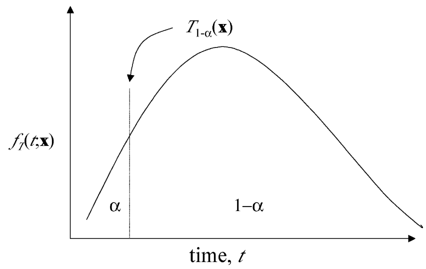

Fig. 3. Example probability density function for $T(\mathbf{x})$.

The marginal distribution for $T(\mathbf{x})$ is defined as follows. λ is a vector representation of the uncertain Weibull shape parameters.

$$f_T(t;\mathbf{x}) = \int_\lambda f_{T,\Lambda}(t,\lambda;\mathbf{x})\mathrm{d}\lambda \tag{1}$$

where $f_T(t;\mathbf{x})$ is the marginal distribution for system time-to-failure, depending on $\mathbf{x}$, $f_{T,\Lambda}(t,\lambda;\mathbf{x})$ is joint distribution for system time-to-failure and λ, depending on $\mathbf{x}$.

$T_{1-\alpha}(\mathbf{x})$ is a $(1-\alpha)\times 100\%$ lower-bound estimate of system time-to-failure considering the variability associated with component time-to-failure and the variability associated with the λ_{ij} values. The risk level, α, is user defined $(0 < \alpha < 1)$. More specifically, $T_{1-\alpha}(\mathbf{x})$ is the $\alpha \times 100\%$ percentile of the marginal distribution of system time-to-failure, found by integrating over all values of λ_{ij}.

Fig. 3 conceptually depicts a typical marginal *pdf* of the system time-to-failure, $T(\mathbf{x})$, and its lower percentile, $T_{1-\alpha}(\mathbf{x})$. The system marginal *pdf* will be complex and non-standard for almost all cases and will depend on the system structure function and the individual component distributions. $T_{1-\alpha}(\mathbf{x})$ represents the time value on the abscissa where the cumulative area under the marginal density function is α. The problem objective is to search over all feasible solutions, $\mathbf{x}$, to identify the design configuration which maximizes $T_{1-\alpha}(\mathbf{x})$. The optimal design configuration for a risk-neutral design will generally be different than the optimal solution for a risk-averse design (e.g. $\alpha = 0.05$).

A risk-neutral design objective involves maximization of the mean of a random variable, such as time-to-failure, as described by Bunn (1984). A risk-neutral design objective can often be closely approximated by using the median time-to-failure ($\alpha = 0.50$).

Problem P1 is more realistic than other formulations for many design problems, but there are inherent difficulties. Direct solution of this problem is difficult because there is no closed-form expression to compute $T_{1-\alpha}(\mathbf{x})$, which is intractable for any non-trivial problem. There are several possibilities to evaluate this expression including Monte Carlo simulation, bi-section search and numerical integration.

One possibility is to use Monte Carlo simulation to estimate $T_{1-\alpha}(\mathbf{x})$, similar to the approach used by Painton and Campbell (1995). Monte Carlo simulation is an approach that allows the closest-to-reality description of a system behavior (component functional and statistical dependencies, aging effects, etc.) and a straightforward treatment of parameter uncertainty. Recent research by Joyce, Withers, and Hickling (1998) and Cantoni, Marseguerra, and Zio (2000) have described the combination of Monte Carlo and GA in combinatorial optimization problems and have shown the feasibility of the approach in reliability allocation problems.

Monte Carlo simulation involves numerous samplings for each prospective solution considered by the

GA. Another approach was used to determine $T_{1-\alpha}(\mathbf{x})$ based on an equivalent problem formulation and a bi-section search. This approach did not require the random sampling associated with Monte Carlo simulation, although it did require multiple search iterations from a bi-section search. The bi-section search requires fewer iterations than Monte Carlo simulation.

To develop the alternative problem formulation, consider the following interpretation of $T_{1-\alpha}(\mathbf{x})$. In the expression that follows, $T_{1-\alpha}(\mathbf{x})$ is defined as the time value, designated as t', where the probability of system failure (which depends on $\mathbf{x}$) after t' is greater than $1-\alpha$.

$$T_{1-\alpha}(\mathbf{x}) = \{t' | \int_{\lambda_s} \cdots \int_{\lambda_1} \int_{t'}^{\infty} f_{T,\Lambda}(t,\lambda;\mathbf{x})\mathrm{d}t\,\mathrm{d}\lambda_1 \ldots \mathrm{d}\lambda_s = 1-\alpha\} \tag{2}$$

where, $\lambda_i = (\lambda_{i1}, \lambda_{i2}, \ldots, \lambda_{in_i})$, $\mathrm{d}\lambda_i = (\mathrm{d}\lambda_{i1}\,\mathrm{d}\lambda_{i2} \ldots \mathrm{d}\lambda_{in_i})$, and $f_{T,\Lambda}(t,\lambda;\mathbf{x})$ is the joint probability density function for system time-to-failure and λ.

The equation for $T_{1-\alpha}(\mathbf{x})$ can be simplified if the following equalities are considered.

$$\int_{\lambda_s} \cdots \int_{\lambda_1} \int_{t'}^{\infty} f_{T,\Lambda}(t,\lambda;\mathbf{x})\mathrm{d}t\,\mathrm{d}\lambda_1 \ldots \mathrm{d}\lambda_s = \int_{\lambda_s} \cdots \int_{\lambda_1} \int_{t'}^{\infty} f_{T,\Lambda}(t|\lambda;\mathbf{x}) f_\Lambda(\lambda)\mathrm{d}t\,\mathrm{d}\lambda_1 \ldots \mathrm{d}\lambda_s$$

$$= \int_{\lambda_s} \cdots \int_{\lambda_1} R(t'|\lambda;\mathbf{x}) f_\Lambda(\lambda)\mathrm{d}\lambda_1 \ldots \mathrm{d}\lambda_s = E_\Lambda[R(t'|\lambda;\mathbf{x})] \tag{3}$$

where, $R(t'|\lambda;\mathbf{x})$ is the system reliability, conditional on λ and depending on $\mathbf{x}$, and $E_\Lambda R(t'|\lambda;\mathbf{x})]$ is the expected value of $R(t'|\lambda;\mathbf{x})$ given the uncertainty of λ.

When the failure times for all components in a series–parallel system are independent, the lower-percentile of the system can be expressed as,

$$T_{1-\alpha}(\mathbf{x}) = \left\{ t' \left| \prod_{i=1}^{s} \left[1 - \prod_{j=1}^{n_i} \left(1 - E_\Lambda[R_{ij}(t'|\lambda_{ij})] \right) \right] = 1-\alpha \right. \right\} \tag{4}$$

$R_{ij}(t'|\lambda_{ij})$ is the reliability of the jth component for subsystem i, conditional on λ_{ij} and is equal to $\exp(-\lambda_{ij}(t')^{\beta_{ij}})$.

$E_\Lambda[R_{ij}(t'|\lambda_{ij})]$ is the expected value of $R_{ij}(t'|\lambda_{ij})$ considering the uncertainty of λ_{ij}.

If λ_{ij} is distributed in accordance with independent uniform distributions with parameters a_{ij} and b_{ij} ($a_{ij} \le \lambda_{ij} \le b_{ij}$), then $T_{1-\alpha}(\mathbf{x})$ can be expressed as,

$$E_\Lambda[R_{ij}(t'|\lambda_{ij})] = \frac{\mathrm{e}^{-a_{ij}(t')^{\beta_{ij}}} - \mathrm{e}^{-b_{ij}(t')^{\beta_{ij}}}}{(t')^{\beta_{ij}}(b_{ij}-a_{ij})} \quad \text{for } \lambda_{ij} \sim \text{Uniform}(a_{ij},\ b_{ij}) \tag{5}$$

$$T_{1-\alpha}(\mathbf{x}) = \left\{ t' \left| \prod_{i=1}^{s} \left[1 - \prod_{j=1}^{n_i} \left(1 - \frac{\mathrm{e}^{-a_{ij}(t')^{\beta_{ij}}} - \mathrm{e}^{-b_{ij}(t')^{\beta_{ij}}}}{(t')^{\beta_{ij}}(b_{ij}-a_{ij})} \right) \right] = 1-\alpha \right. \right\} \tag{6}$$

In this case, a_{ij} and b_{ij} are upper and lower bounds for λ_{ij} and all intermediate values are equally likely. A new optimization formulation was developed for the specific case where λ_{ij} is distributed in accordance with a uniform distribution. An analogous formulation could be developed if λ_{ij} is distributed in accordance with some other distribution (i.e. normal, gamma, triangular). The equivalent formulation is as follows.

Problem P2. max t'

$$\text{subject to} \quad \prod_{i=1}^{s}\left[1-\prod_{j=1}^{n_i}\left(1-E_\Lambda[R_{ij}(t'|\lambda_{ij})]\right)\right]=1-\alpha$$

$$\mathbf{Ax} \le \mathbf{b}$$

$$1 \le \sum_{j=1}^{m_i} x_{ij} = n_i \le n_{\max,i} \ \ \forall\, i$$

$$x_{ij} \in \{0,1,2,\dots\}$$

where

$$E_\Lambda[R_{ij}(t'|\lambda_{ij})] = \frac{\mathrm{e}^{-a_{ij}(t')^{\beta_{ij}}} - \mathrm{e}^{-b_{ij}(t')^{\beta_{ij}}}}{(t')^{\beta_{ij}}(b_{ij}-a_{ij})} \tag{7}$$

t' is a continuous variable which can be considered a pseudo-mission time. The objective becomes to find the solution vector, $\mathbf{x}$, associated with the largest pseudo-mission time that meets an additional constraint for the expected system reliability at time t'. By adding the expected system reliability constraint set, the problem was restated such that a GA can be applied. An optimal solution to Problem P2 (i.e. $\mathbf{x}$, t') consists of the optimal solution, $\mathbf{x}$, for Problem P1 and the optimal value of the objective function, $T_{1-\alpha}(\mathbf{x})$. While the Problem P2 formulation is more accommodating, it is still problematic because it remains difficult to compute t'.

$E_\Lambda[R(t|\lambda;\mathbf{x})]$ is monotonically decreasing with t for any coherent system. Also, the original constraints do not include t'. Thus, for any prospective solution vector, $\mathbf{x}$, the maximum t' can be found by increasing t' until $E_\Lambda[R(t|\lambda;\mathbf{x})]$ equals $1-\alpha$ (i.e. reliability constraint is tight).

For this formulation, component time-to-failure is distributed according to the Weibull distribution, although with random scale parameter. The Weibull distribution can model component hazard functions that are increasing, decreasing or constant with respect to time depending on β_{ij}. The scale parameter (λ_{ij}) is stochastic, but not changing with respect to time. An alternative formulation could have the Weibull distribution parameters changing as a function of time. This also could be accommodated by using functions for $\lambda_{ij}(t)$ and $\beta_{ij}(t)$ within Eq. (4). Neither the bi-section search or GA is developed for any specific distributional form, so any functions of the Weibull parameters, changing with time, could be accommodated.

3. Genetic algorithm

A solution algorithm for this problem is based on GA search. GA was developed by Holland (1975), and later advanced by Goldberg (1989) among others. The GA encodings and operators used here were originally defined by Coit and Smith (1996). They were used together with a bi-section search to determine t' (Coit & Smith, 1998) and an adaptive penalty function to enforce compliance to the constraints. The GA was coded using C++ and run on both a Pentium personal computer and a VAX platform.

GA involves the evaluation of a population of solutions, which are revised over successive generations. Each solution is represented in the population by the vector **x**. The crossover and mutation operators are used to introduce new prospective design solutions each generation. Crossover involves the selection of parent solution vectors and the recombination of those vectors to produce new prospective solutions. Parent selection is random, but biased by the ordinal objective function ranking within a current population. Solutions that have been observed to be superior are more likely to be chosen. Mutation involves the addition or removal of components in accordance with a preselected mutation rate. This prevents premature convergence to a local optima. The culling operator involves the selection of the p solutions with the highest penalized objective function from among the prior population and the newly formed solutions.

The encoding strategy from Coit and Smith (1996) was used. Each possible solution to the redundancy allocation problem is a collection of n_i parts in parallel ($k_i \leq n_i \leq n_{\max}$) for each subsystem. The m_i components have been indexed in descending order in accordance with their reliability (i.e. 1 representing the most reliable, etc.). Each of the s subsystems are represented by $n_{\max,i}$ positions in the solution vector. An index of $m_i + 1$ is assigned to a position where an additional component was not used (i.e. $n_i < n_{\max,i}$). The subsystem representations are then placed adjacent to each other to complete the vector representation. As an example, consider a system with $s = 3$, $m_1 = 5$, $m_2 = 4$, $m_3 = 5$ and $n_{\max,i}$ is 5 for all i. The following vector,

$$\mathbf{v}_q = (1\ 1\ 6\ 6\ 6\,|\,2\ 2\ 3\ 5\ 5\,|\,4\ 6\ 6\ 6\ 6)$$

represents a solution with two of the most reliable components used in parallel for the first subsystem; two of the second most reliable and one of the third most reliable components used in parallel for the second subsystem; and one of the fourth most reliable components used for the third subsystem.

The crossover and mutation operators from Coit and Smith (1996) were used. Parent solution vectors were selected based on the ordinal ranking of their objective function. A uniform random number, U, between 1 and $\sqrt{p}$ was selected and the solution with the ranking closest to U^2 is selected as a parent. The crossover operator retained all identical genetic information from both parents and then randomly selected, with equal probability, from either of the two parents for components that differed.

The mutation operator performs random perturbations to selected solutions. A predetermined number of mutations within a generation is set for each GA trial. Each value within the solution vector (which was randomly selected to be mutated) was changed with probability equal to the mutation rate. A mutated component was changed to an index of $m_i + 1$ with probability of 0.5 and to a randomly chosen component, from among the m_i choices, with probability of 0.5.

An overview of the GA approach is presented.

Step 0: Initialize Coit and Smith GA (1996)). Define GA control parameters:

- solution vector encoding scheme
- population size, p (number of prospective solutions evaluated at each generation)
- number of crossover operations per generation ($<p$)
- number of mutation operations per generation ($<p$) and mutation rate
- termination criteria (maximum number of generations, or stalled search criteria)

Step 1: Determine Initial Population. Repeat the following p times.

- randomly select integers n_i for $i = 1, \ldots, s$
- For $i = 1, \ldots, s$, select n_i components randomly and uniformly from the m_i choices
- determine objective function values of initial population

Step 2: Crossover Operation (Coit & Smith, 1996)
Step 3: Mutation Operation (Coit & Smith, 1996)
Step 4: Objective Function Determination—for each member of population:

- Bi-section search to determine t'
- Adaptive penalty function (Coit, Smith, & Tate, 1996)

Step 5: Culling/Ranking Operation
Step 6: Termination Criteria Satisfaction

- If satisfied, select best feasible solution in final population as recommended design
- If not satisfied, proceed to next generation—go to Step 2

To determine the maximum t' value for a particular solution, $\mathbf{x}$, a bi-section search was used to determine t' where $E[R(t'|\lambda; \mathbf{x})]$ equals $1 - \alpha$. Since system reliability is monotonically decreasing with t and there is generally some *a priori* knowledge of the system time-to-failure distribution, it is relatively easy through trial-and-error to find upper and lower bounds for t' to serve as starting points. These bounds are the times, t_L and t_H where $E[R(t_H|\lambda; \mathbf{x})] \leq 1 - \alpha \leq E[R(t_L|\lambda; \mathbf{x})]$ and $t_H = 10t_L$. After finding the appropriate starting points, a bi-section search is used to determine t' to any pre-determined level of accuracy. The bi-section search involves evaluation of the midpoint between the bounds, and then the midpoint successively becomes the lower or upper bound in the next iteration.

In the sample problems, the search for t' was terminated when the upper and lower bound were within 0.001%. Determination of the starting points generally required less than five function evaluations, and then a maximum of 20 additional function evaluations where required to estimate t' to within 0.001%. This is significantly more efficient than using Monte Carlo simulation to estimate $T_{1-\alpha}(\mathbf{x})$ to the same degree of accuracy. To make the search even more efficient, the bi-section search could be replaced by a Fibonacci series or golden section search, or a derivative-based search such as a Newton–Raphson search.

The adaptive penalty function was presented and described by Coit, Smith, & Tate, (1996). This penalty function was specifically developed to exploit information available from GA search and to be updated by the relative success of the search as it proceeds. It promotes a thorough search within a near-feasible-threshold (*NFT*) near the boundary between the feasible and infeasible regions. The penalized objective function then becomes $t' - P(\mathbf{x})$, where the penalty, $P(\mathbf{x})$, is given by,

$$P(\mathbf{x}) = (V_{\text{all}} - V_{\text{feas}}) \sum_i \left(\frac{\Delta b_i(\mathbf{x})}{\text{NFT}_i} \right)^2$$

where, $\Delta b_i(\mathbf{x})$ is the constraint violation for ith constraint ($\Delta b_i(\mathbf{x}) = 0$ if constraint is not violated), NFT_i the near feasible threshold for ith constraint.

$$\text{NFT}_i = \frac{\text{NFT}_{i,\text{o}}}{1 + \gamma_i g}$$

$\text{NFT}_{i,\text{o}}$, γ_i are penalty function constants and g is the generation number for GA search, $g \in \{1, 2, \ldots\}$.

V_{all} is the maximum unpenalized t' (feasible or infeasible) and V_{feas} is the maximum *feasible* t' found

Table 1
Example input parameters

i	Choice 1			Choice 2			Choice 3			Choice 4		
	$E(\lambda_{ij})$	β_{ij}	$E(T)$	?E(λ_{ij})	β_{ij}	$E(T)$	?E(λ_{ij})	β_{ij}	$E(T)$	?E(λ_{ij})	β_{ij}	$E(T)$
1	0.0051293	1.0	195	0.0229489	0.5	3798	0.0298237	0.5	2249	0.0000011	5.0	14
2	0.0051293	1.0	195	0.0006188	2.0	36	0.0007257	2.0	33	–	–	–
3	0.0008338	2.0	31	0.0333179	0.5	1802	0.0013926	2.0	24	0.0513930	0.5	757
4	0.0440385	0.5	1031	0.0000016	5.0	13	0.0018633	2.0	21	–	–	–
5	0.0051293	1.0	195	0.0195667	0.5	5224	0.0007257	2.0	33	–	–	–
6	0.0010050	1.0	995	0.0002020	2.0	62	0.0000003	5.0	19	0.0000004	5.0	17
7	0.0000006	5.0	16	0.0008338	2.0	31	0.0298237	0.5	2249	–	–	–
8	0.0000009	5.0	15	0.0000011	5.0	14	0.0000021	5.0	13	–	–	–
9	0.0010050	1.0	995	0.0030459	1.0	328	0.0040822	1.0	245	0.0000009	5.0	15
10	0.0105361	1.0	95	0.0513930	0.5	757	0.0186330	1.0	54	–	–	–
11	0.0040822	1.0	245	0.0005129	2.0	39	0.0061875	1.0	162	–	–	–
12	0.0010536	2.0	27	0.0162519	1.0	62	0.0198451	1.0	50	0.0023572	2.0	18
13	0.0001005	2.0	88	0.0000002	5.0	20	0.0030459	1.0	328	–	–	–
14	0.0001005	2.0	88	0.0000005	5.0	17	0.0000008	5.0	15	0.0000011	5.0	14

by the GA during generations 1 to $g-1$. This penalty function encourages the evaluation of infeasible solutions, which are near the feasible region. It has been demonstrated using many reliability optimization sample problems (Coit & Smith, 1996b) that a thorough GA search near the feasibility boundary, including both feasible and infeasible solutions, can result in superior final solutions in a large majority of test cases compared to search strategies which reject all infeasible solutions.

The algorithm continues for a pre-determined maximum number of generations (G) or until no additional improvement is observed (K generations without improvement in the best solution). The GA approach does not guarantee that the optimal solution is found, but GA has been demonstrated to produce very good results and consistently find the optimal solution (Cantoni et al., 2000; Coit & Smith, 1996). It is recommended that multiple runs be performed with different initial populations. Then, the best feasible solution should be used if all runs do not converge to the same solution. This is a precaution resulting from criticisms concerning the convergence capabilities of GAs. In practice, it was demonstrated, during the example problems in Section 4, that the standard deviation of the final GA solution is very low; thereby, demonstrating sound convergence capabilities.

4. Illustrative examples

The algorithm was demonstrated using modified versions of the 33 example problems solved by Nakagawa and Miyazaki (1981) and Coit and Smith (1998) at three α-levels ($\alpha = 0.50$, 0.10 and 0.05). When $\alpha = 0.50$, the problem closely approximates a risk-neutral design situation and involves the maximization of the median time-to-failure. When $\alpha = 0.10$ or 0.05, the problem is for risk-averse design scenarios.

A GA population size of 40 was used and ten different runs were made with different initial populations for each test case. For this problem, 18 children and 22 mutates were generated each generation.

Table 2
λ Distribution parameters for component choices

i	1			2			3			4		
	a_{ij}	b_{ij}	b_{ij}/a_{ij}	a_{ij}	b_{ij}	b_{ij}/a_{ij}	a_{ij}	b_{ij}	b_{ij}/a_{ij}	a_{ij}	b_{ij}	b_{ij}/a_{ij}
1	2.9×10^{-3}	7.4×10^{-3}	2.6	3.7×10^{-3}	4.2×10^{-2}	11.2	3.0×10^{-3}	5.7×10^{-2}	18.6	3.7×10^{-7}	1.7×10^{-6}	4.7
2	1.3×10^{-3}	8.9×10^{-3}	6.8	9.9×10^{-5}	1.1×10^{-3}	11.5	8.4×10^{-5}	1.4×10^{-3}	16.3	–	–	–
3	1.2×10^{-4}	1.5×10^{-3}	13.0	5.0×10^{-3}	6.2×10^{-2}	12.3	4.0×10^{-4}	2.4×10^{-3}	5.9	2.7×10^{-2}	7.5×10^{-2}	2.8
4	1.0×10^{-2}	7.8×10^{-2}	7.4	8.5×10^{-7}	2.4×10^{-6}	2.8	9.2×10^{-4}	2.8×10^{-3}	3.0	–	–	–
5	8.5×10^{-4}	9.4×10^{-3}	11.0	2.6×10^{-3}	3.6×10^{-2}	13.9	2.4×10^{-4}	1.2×10^{-3}	5.0	–	–	–
6	2.4×10^{-4}	1.8×10^{-3}	7.4	1.2×10^{-4}	2.8×10^{-4}	2.4	5.0×10^{-8}	5.5×10^{-7}	11.0	6.0×10^{-8}	7.6×10^{-7}	12.7
7	7.0×10^{-8}	1.2×10^{-6}	16.7	1.2×10^{-4}	1.5×10^{-3}	12.4	3.1×10^{-3}	5.7×10^{-2}	18.5	–	–	–
8	1.8×10^{-7}	1.7×10^{-6}	9.4	3.2×10^{-7}	1.8×10^{-6}	5.6	2.1×10^{-7}	4.0×10^{-6}	19.1	–	–	–
9	1.0×10^{-4}	1.9×10^{-3}	18.1	4.8×10^{-4}	5.6×10^{-3}	11.6	1.1×10^{-3}	7.1×10^{-3}	6.7	2.3×10^{-7}	1.6×10^{-6}	7.2
10	1.6×10^{-3}	1.9×10^{-2}	12.3	1.8×10^{-2}	8.5×10^{-2}	4.8	7.1×10^{-3}	3.0×10^{-2}	4.2	–	–	–
11	5.0×10^{-4}	7.7×10^{-3}	15.3	7.0×10^{-5}	9.6×10^{-4}	13.7	9.2×10^{-4}	1.1×10^{-2}	12.4	–	–	–
12	1.5×10^{-4}	1.9×10^{-3}	12.8	3.2×10^{-3}	2.9×10^{-2}	9.3	1.8×10^{-2}	2.2×10^{-2}	1.2	4.9×10^{-4}	4.2×10^{-3}	8.7
13	3.4×10^{-5}	1.7×10^{-4}	4.8	1.2×10^{-7}	2.8×10^{-7}	2.3	3.1×10^{-4}	5.8×10^{-3}	18.4	–	–	–
14	1.2×10^{-5}	1.9×10^{-4}	15.4	8.0×10^{-8}	9.4×10^{-7}	11.8	8.0×10^{-8}	1.6×10^{-6}	19.8	2.4×10^{-7}	1.9×10^{-6}	7.8

Table 3
GA performance

No	$\alpha = 0.5$				$\alpha = 0.1$				$\alpha = 0.05$			
	Max	Min (time)	Mean	Std dev	Max	Min (time)	Mean	Std dev	Max	Min (time)	Mean	Std dev
1	19.851	19.811	19.839	0.0166	15.089	15.023	15.054	0.0238	13.126	13.021	13.106	0.0430
2	19.842	19.794	19.830	0.0197	15.004	14.980	14.989	0.0103	13.038	12.911	13.005	0.0550
3	19.826	19.768	19.818	0.0192	14.971	14.743	14.890	0.0697	12.944	12.736	12.881	0.0589
4	19.812	19.762	19.795	0.0140	14.908	14.793	14.858	0.0307	12.878	12.758	12.840	0.0519
5	19.796	19.738	19.779	0.0191	14.872	14.628	14.772	0.0722	12.752	12.626	12.698	0.0459
6	19.779	19.735	19.762	0.0197	14.793	14.661	14.735	0.0534	12.659	12.549	12.609	0.0379
7	19.760	19.708	19.754	0.0164	14.699	14.587	14.650	0.0397	12.543	12.395	12.495	0.0435
8	19.735	19.666	19.713	0.0291	14.551	14.433	14.522	0.0379	12.428	12.258	12.376	0.0513
9	19.708	19.678	19.702	0.0107	14.554	14.323	14.436	0.0788	12.335	12.148	12.265	0.0580
10	19.705	19.630	19.683	0.0227	14.480	14.265	14.388	0.0922	12.269	12.049	12.152	0.0794
11	19.681	19.605	19.645	0.0266	14.397	14.219	14.282	0.0643	12.181	11.862	12.057	0.1070
12	19.654	19.500	19.617	0.0472	14.296	14.136	14.225	0.0582	12.087	11.829	11.995	0.0923
13	19.629	19.586	19.618	0.0156	14.172	14.120	14.152	0.0194	11.991	11.802	11.917	0.0741
14	19.590	19.557	19.576	0.0083	14.103	14.043	14.068	0.0293	11.796	11.604	11.720	0.0749
15	19.575	19.530	19.560	0.0148	14.043	13.840	13.917	0.0691	11.714	11.486	11.646	0.0682
16	19.550	19.456	19.507	0.0268	13.983	13.755	13.881	0.0885	11.571	11.423	11.497	0.0438
17	19.522	19.430	19.508	0.0300	13.906	13.653	13.747	0.0962	11.527	11.313	11.430	0.0581
18	19.497	19.404	19.479	0.0388	13.818	13.483	13.724	0.1189	11.324	11.214	11.287	0.0429
19	19.445	19.351	19.424	0.0377	13.713	13.378	13.543	0.1069	11.274	11.137	11.204	0.0425
20	19.425	19.373	19.394	0.0213	13.576	13.301	13.474	0.1110	11.192	11.099	11.141	0.0328
21	19.400	19.353	19.387	0.0188	13.422	13.257	13.376	0.0578	11.099	10.961	11.047	0.0425
22	19.371	19.281	19.349	0.0353	13.310	12.986	13.187	0.1169	11.044	10.895	10.957	0.0507
23	19.347	19.180	19.297	0.0807	13.172	12.977	13.086	0.0836	10.934	10.807	10.869	0.0386
24	19.294	19.046	19.240	0.0905	13.101	12.876	12.994	0.0916	10.846	10.731	10.791	0.0357
25	19.222	19.088	19.196	0.0467	13.013	12.796	12.924	0.0971	10.813	10.714	10.752	0.0351
26	19.185	19.025	19.116	0.0416	12.920	12.601	12.774	0.1264	10.733	10.626	10.681	0.0399
27	19.115	19.001	19.034	0.0369	12.815	12.686	12.783	0.0431	10.676	10.593	10.614	0.0241
28	19.062	18.995	19.040	0.0271	12.686	12.439	12.592	0.0996	10.604	10.483	10.555	0.0385
29	19.016	18.989	18.997	0.0095	12.615	12.395	12.542	0.0716	10.544	10.450	10.499	0.0251
30	18.983	18.941	18.951	0.0167	12.494	12.208	12.397	0.1046	10.453	10.346	10.398	0.0326
31	18.957	18.808	18.887	0.0472	12.403	12.153	12.349	0.0775	10.412	10.264	10.344	0.0357
32	18.903	18.777	18.802	0.0531	12.307	12.032	12.222	0.0983	10.330	10.203	10.289	0.0448
33	18.829	18.651	18.745	0.0747	12.186	12.013	12.151	0.0545	10.269	10.093	10.194	0.0604

The mutation rate was 0.05. The maximum number of generations (G) was 1200. The K parameter was not used so all GA runs consisted of 1200 generations.

The examples involve a system comprised of 14 subsystems with 3 or 4 component choices available for each subsystem. The system cost constraint is always 130. The weight constraint is varied incrementally from 191 to 159 to define the 33 problem variations (Nakagawa & Miyazaki, 1981).

c_{ij} and w_{ij} are the cost and weight of the jth available component for subsystem i. The system cost and weight are a linear combination of the components selected for the system design. The values for c_{ij} and w_{ij} were previously presented (Fyffe et al., 1968; Nakagawa & Miyazaki, 1981). Weibull distribution parameters were previously presented for this problem (Coit & Smith, 1998). Table 1 presents the

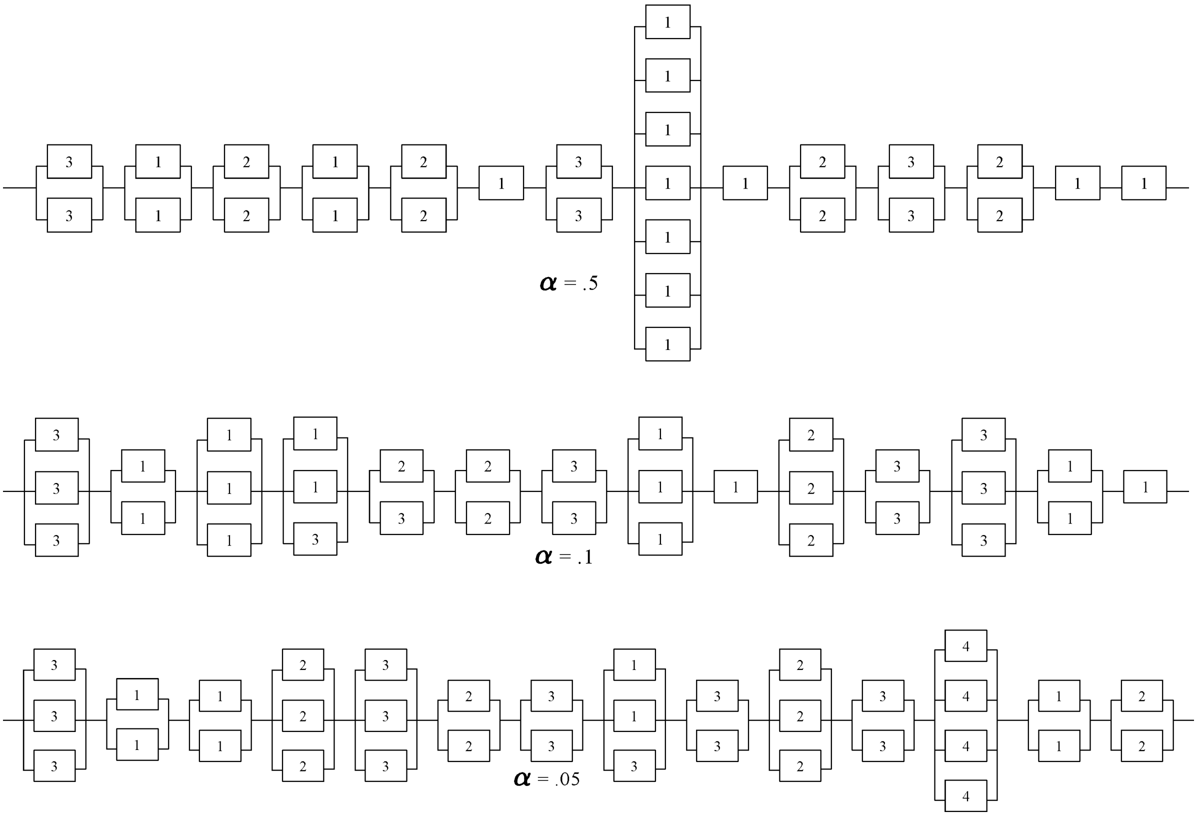

Fig. 4. Comparison of different risk levels (test case 26).

Weibull parameters slightly modified. The scale parameter in Table 1 is presented as $E(\lambda_{ij})$ instead of λ_{ij} because in this new formulation, it is considered as a random variable.

It was necessary to assign uniform distributions for the λ_{ij} values so the algorithm could be properly demonstrated. Previously λ_{ij} was presented as known parameter values, but they are now assumed to be $E(\lambda_{ij})$ corresponding to a uniform distribution. b_{ij}/a_{ij} ratios were selected and corresponding a_{ij} and b_{ij} values were computed based on the expected value of λ_{ij}, i.e. $(a_{ij} + b_{ij})/2$. They are presented in Table 2. The b_{ij}/a_{ij} ratios were selected from a uniform distribution between 1 and 20 to generate diverse distributions. If the ratio is small, it implies that there is little variability in the λ_{ij} values (i.e. λ_{ij} is known with a great deal of precision). If the ratio is large, there is more variability in the λ_{ij} values.

The GA for this problem was tested on all 33 test cases at three different α-levels. Ten GA trials were performed for each test case. The results are presented in Table 3 and Appendix A. The standard deviation in each case is less than 2% of the mean lower-bound on time-to-failure. Table 3 presents the minimum, average and maximum objective functions.

Example results are presented in Appendix A. An important observation is that the recommended design configurations are very different depending on the α-level. For none of the 33 cases was the recommended designs the same for different α-levels, and often, they varied significantly. It is also interesting that the solution which maximizes $T_{1-\alpha}(\mathbf{x})$ for $\alpha = 0.50$ is a poor choice for maximizing

$T_{1-\alpha}(\mathbf{x})$ at $\alpha = 0.05$. Similarly, the reverse is also true. As anticipated, there needs to be different design strategies depending on a user's risk profile.

To demonstrate the different solutions depending on α, Fig. 4 depicts the three solutions ($\alpha = 0.50$, 0.10 and 0.05) for test case no. 26. For some subsystems (2, 7, 11), the same components and redundancy levels are selected for all three cases. For others, there are significant differences. Consider subsystem 12; when $\alpha = 0.05$, the GA solution is to use four of component choice four in parallel. When $\alpha = 0.1$, the GA solution is to use three of component choice three in parallel, and finally when $\alpha = 0.5$, the GA solution is to use two of component choice two in parallel. By evaluating the component characteristics, it seems clear that for riskier applications, the preferred design strategies are to (1) use less-reliable components in redundant configurations if the uncertainty is less, and (2) select higher time-dependent component choices (higher β_{ij}) even if the expected time-to-failure is less. Another interesting subsystem is no. 8. For this subsystem, all three choices have relatively high β_{ij} and low expected failure time. For low α, the selections are not particularly interesting, but for the median system failure time ($\alpha = 0.5$), this subsystem becomes the 'weakest link' and it is necessary to devote significant resources (seven components in parallel) to increase the objective function. As anticipated, the risk has great influence on design behavior.

When $\alpha = 0.5$, the problem becomes very similar to the analogous problem previously solved by Coit and Smith (1998). The original problem considered the Weibull parameters to be known deterministic values. In this paper, the Weibull shape parameter is a random variable with the mean equal to the deterministic value presented by Coit & Smith (1998). The problems are similar, although not the same when $\alpha = 0.5$ (they are very different for $\alpha = 0.05$ or 0.1). Examining the GA solutions reveals that the same solution was determined for 26 of the 33 problems.[1]

It was not possible to precisely evaluate the computation time for the algorithms because they were run on a variety of platforms, including personal computers (Pentium) and VAX, and in both batch and interactive modes. However, to provide an approximate comparison, a typical GA trial to maximize a $(1-\alpha)\times 100\%$ lower-bound on reliability consumed less than one minute of CPU time (and usually less than one-half of a minute) on the VAX.

5. Conclusions

The use of GA optimization procedures in reliability allocation problems was illustrated for a case in which the failure times of the components are Weibull-distributed, with uncertain Weibull shape parameters distributed according to uniform distributions. The GA can be generalized to consider a broader range of problems. Any parametric distribution for component Weibull scale parameter can be incorporated into the algorithm. Additionally, the penalty function can accommodate any form of non-linear constraints.

The GA results indicate that consideration of component distribution uncertainty can impact the algorithm results. This result is logical and had been noted by previous researchers. However, the problem had not previously been implemented or demonstrated when a lower-bound on system time-to-failure is the objective function. In practice, system designers and reliability engineers do not always

[1] Tables 4 and 5 in Coit and Smith (1998) are missing an additional component three for subsystem 1 in each case except problem 21 ($\alpha = 0.05$) which is correct as stated.

Table A1
Recommended system design for $\alpha = 0.5$

No.	$T_{1-\alpha}(\mathbf{x})$	$C(\mathbf{x})$	$W(\mathbf{x})$	Component selections
1	19.851	130	191	333,11,22,11,22,1,33,1111111,1,222,33,233,11,11
2	19.842	129	190	333,11,22,11,22,12,33,1111111,1,222,33,3333,11,1
3	19.826	129	189	333,11,22,11,22,22,33,1111111,1,222,33,3333,11,1
4	19.812	129	188	333,11,22,11,22,12,33,1111111,1,222,33,222,11,1
5	19.796	129	187	333,11,22,11,22,22,33,1111111,1,222,33,222,11,1
6	19.779	128	186	333,11,22,11,22,22,33,1111111,1,222,33,223,11,1
7	19.760	127	185	333,11,22,11,22,22,33,1111111,1,222,33,233,11,1
8	19.735	126	184	333,11,22,11,22,22,33,1111111,1,222,33,333,11,1
9	19.708	125	183	33,11,22,11,22,22,33,1111111,1,222,33,233,11,1
10	19.705	124	182	333,11,22,11,22,1,33,1111111,1,222,33,233,11,1
11	19.681	123	181	333,11,22,11,22,1,33,1111111,1,222,33,333,11,1
12	19.654	122	180	33,11,22,11,22,1,33,1111111,1,222,33,233,11,1
13	19.629	121	179	33,11,22,11,22,1,33,1111111,1,222,33,333,11,1
14	19.590	121	178	333,11,22,11,22,1,33,1111111,1,12,33,233,11,1
15	19.575	120	177	333,11,22,11,22,1,33,1111111,1,22,33,233,11,1
16	19.550	119	176	333,11,22,11,22,1,33,1111111,1,22,33,333,11,1
17	19.522	118	175	33,11,22,11,22,1,33,1111111,1,22,33,233,11,1
18	19.497	117	174	33,11,22,11,22,1,33,1111111,1,22,33,333,11,1
19	19.445	117	173	33,11,22,11,23,1,33,1111111,1,22,33,333,11,1
20	19.425	117	172	333,11,22,11,22,1,33,1111111,1,22,33,233,1,1
21	19.400	116	171	333,11,22,11,22,1,33,1111111,1,22,33,333,1,1
22	19.371	115	170	33,11,22,11,22,1,33,1111111,1,22,33,233,1,1
23	19.347	114	169	33,11,22,11,22,1,33,1111111,1,22,33,333,1,1
24	19.294	114	168	33,11,22,11,23,1,33,1111111,1,22,33,333,1,1
25	19.222	115	167	33,11,12,11,23,1,33,1111111,1,22,33,333,1,1
26	19.185	113	166	33,11,22,11,22,1,33,1111111,1,22,33,22,1,1
27	19.115	112	165	33,11,22,11,22,1,33,1111111,1,22,33,23,1,1
28	19.062	112	164	33,11,22,11,23,1,33,1111111,1,22,33,23,1,1
29	19.016	114	163	333,1,22,11,22,1,33,1111111,1,22,33,333,1,1
30	18.983	113	162	33,1,22,11,22,1,33,1111111,1,22,33,233,1,1
31	18.957	112	161	33,1,22,11,22,1,33,1111111,1,22,33,333,1,1
32	18.903	112	160	33,1,22,11,23,1,33,1111111,1,22,33,333,1,1
33	18.829	113	159	33,1,12,11,23,1,33,1111111,1,22,33,333,1,1

know distribution parameters precisely. Furthermore, there is not always a well-defined mission time to compute system reliability. Therefore, this formulation and solution methodology presented in this paper are realistic and applicable for many actual problem domains.

The distributions for λ_{ij} were defined such that $E(\lambda_{ij})$ was equal to the λ_{ij} values from Coit and Smith (1998). Practitioners often do something similar, except in reverse, when data is encountered which indicates that λ_{ij} is subject to variability. They may determine the expected value or an average, and then use it as if it were a deterministic or exact value. The ramifications of such an assumption can be clarified by considering this comparison as it applies to Jensen's inequality (Billingsley, 1986) which states that,

$$\varphi(E(X)) \leq E(\varphi(X))$$

if $\varphi(X)$ is convex on an interval containing the range of X.

Table A2
Recommended system design for $\alpha = 0.1$

No.	$T_{1-\alpha}(\mathbf{x})$	$C(\mathbf{x})$	$W(\mathbf{x})$	Component selections
1	15.089	130	191	333,11,112,111,22,22,33,11111,1,222,33,3334,11,12
2	15.004	130	190	333,11,112,111,22,12,33,11111,1,222,33,223,11,12
3	14.971	130	189	333,11,112,111,22,22,33,11111,1,222,33,223,11,12
4	14.908	129	188	333,11,112,111,22,22,33,11111,1,222,33,233,11,12
5	14.872	130	187	333,11,111,111,22,22,33,11111,1,222,33,233,11,12
6	14.793	129	186	333,11,111,111,22,22,33,11111,1,222,33,333,11,12
7	14.669	127	185	333,11,111,111,22,22,33,11111,1,222,33,2334,11,1
8	14.551	128	184	333,11,111,113,22,22,33,11111,33,222,33,333,11,1
9	14.554	128	183	333,11,111,111,333,22,33,11111,1,222,33,223,11,1
10	14.480	127	182	333,11,111,111,333,22,33,11111,1,222,33,233,11,1
11	14.397	125	181	333,11,111,111,22,22,33,11111,1,222,33,233,11,1
12	14.296	124	180	333,11,111,111,22,22,33,11111,1,222,33,333,11,1
13	14.172	122	179	333,11,111,111,333,22,33,1111,1,222,33,3334,11,1
14	14.103	121	178	333,11,111,111,333,22,33,1111,1,222,33,3344,11,1
15	14.043	122	177	333,11,111,111,333,22,33,1111,1,222,33,223,11,1
16	13.983	121	176	333,11,111,111,333,22,33,1111,1,222,33,233,11,1
17	13.906	120	175	333,11,111,111,333,22,33,1111,1,222,33,333,11,1
18	13.818	118	174	333,11,111,111,22,22,33,1111,1,222,33,333,11,1
19	13.713	117	173	333,11,111,113,22,22,33,1111,1,222,33,333,11,1
20	13.576	117	172	333,11,111,113,23,22,33,1111,1,222,33,333,11,1
21	13.422	114	171	333,11,111,113,22,22,33,1113,1,222,33,333,11,1
22	13.310	114	170	333,11,111,113,23,22,33,1113,1,222,33,333,11,1
23	13.172	116	169	333,11,111,112,23,22,33,1113,1,222,33,333,11,1
24	13.101	113	168	333,11,111,113,333,22,33,111,1,222,33,333,11,1
25	13.013	111	167	333,11,111,113,22,22,33,111,1,222,33,333,11,1
26	12.920	111	166	333,11,111,113,23,22,33,111,1,222,33,333,11,1
27	12.815	113	165	333,11,111,112,23,22,33,111,1,222,33,333,11,1
28	12.686	112	164	333,11,111,112,23,22,33,111,1,222,33,334,11,1
29	12.615	110	163	333,11,111,114,333,22,33,111,1,222,33,333,11,1
30	12.494	108	162	333,11,111,113,22,22,33,111,1,222,33,333,1,1
31	12.403	109	161	333,11,11,112,23,22,33,111,1,222,33,333,11,1
32	12.307	110	160	333,11,111,112,23,22,33,111,1,222,33,333,1,1
33	12.186	109	159	333,11,111,112,23,22,33,111,1,222,33,334,1,1

This observation is directly relevant to many reliability engineering activities. Practitioners often assume Weibull or exponential distribution parameters are known exactly, even when empirical evidence indicates otherwise. Jensen's inequality indicates that this assumption is pessimistic, resulting in low reliability estimates.

Acknowledgements

David Coit's work was supported by NSF CAREER grant DMII-9874716.

Table A3
Recommended system design for $\alpha = 0.05$

No.	$T_{1-\alpha}(\mathbf{x})$	$C(\mathbf{x})$	$W(\mathbf{x})$	Component selections
1	13.126	130	191	333,11,111,111,333,22,33,1111,23,222,33,3334,11,12
2	13.038	129	190	333,11,111,111,333,22,33,1111,23,222,33,3344,11,12
3	12.944	130	189	333,11,111,111,333,22,33,1111,33,222,33,3344,12,12
4	12.878	130	188	333,11,111,111,333,22,33,1111,33,222,33,3444,11,12
5	12.752	128	187	333,11,111,111,333,22,33,1113,33,222,33,3344,11,12
6	12.659	127	186	333,11,111,111,333,22,33,1113,33,222,33,3444,11,12
7	12.543	126	185	333,11,111,111,333,22,33,1113,33,222,33,4444,11,12
8	12.428	125	184	333,11,111,113,333,22,33,1113,33,222,33,4444,11,12
9	12.335	125	183	333,11,111,111,333,22,33,111,33,222,33,3344,11,12
10	12.269	124	182	333,11,111,111,333,22,33,111,33,222,33,3444,11,12
11	12.181	123	181	333,11,111,111,333,22,33,111,33,222,33,4444,11,12
12	12.087	122	180	333,11,111,113,333,22,33,111,33,222,33,4444,11,12
13	11.991	124	179	333,11,111,112,333,22,33,111,33,222,33,4444,11,12
14	11.796	123	178	333,11,111,123,333,22,33,111,33,222,33,4444,11,12
15	11.714	122	177	333,11,111,112,23,22,33,111,33,222,33,4444,11,12
16	11.571	122	176	333,11,111,112,33,22,33,111,33,222,33,4444,11,12
17	11.527	120	175	333,11,111,112,333,22,33,111,1,122,33,4444,11,12
18	11.324	118	174	333,11,111,123,23,22,33,113,33,222,33,4444,11,12
19	11.274	124	173	333,11,11,2222,333,22,33,113,33,222,33,4444,11,12
20	11.192	121	172	333,11,111,122,333,22,33,113,33,222,33,4444,11,22
21	11.099	118	171	333,11,111,122,333,22,33,113,1,122,33,4444,11,12
22	11.044	117	170	333,11,111,122,333,22,33,113,1,222,33,4444,11,12
23	10.934	115	169	333,11,111,112,333,22,33,113,1,222,33,4444,11,22
24	10.846	121	168	333,11,11,2222,23,22,33,113,33,222,33,4444,11,22
25	10.813	116	167	333,11,111,122,333,22,33,113,1,222,33,4444,11,22
26	10.733	118	166	333,11,11,222,333,22,33,113,33,222,33,4444,11,22
27	10.676	117	165	333,11,111,222,333,22,33,113,1,222,33,4444,11,22
28	10.604	115	164	333,11,11,222,333,22,33,133,33,222,33,4444,11,22
29	10.544	114	163	333,11,111,222,333,22,33,133,1,222,33,4444,11,22
30	10.453	113	162	333,11,11,222,23,22,33,133,33,222,33,4444,11,22
31	10.412	113	161	333,11,11,222,33,22,33,133,33,222,33,4444,11,22
32	10.330	112	160	333,11,111,222,33,22,33,133,1,222,33,4444,11,22
33	10.269	110	159	333,11,11,222,333,22,33,133,1,222,33,4444,11,22

Appendix A. GA solutions

See Tables A1–A3.

References

Billingsley, P. (1986). *Probability and measure*, (2nd ed)New York: Wiley (p. 75).

Bulfin, R. L., & Liu, C. Y. (1985). Optimal allocation of redundant components for large systems. *IEEE Transactions on Reliability*, *R-34*, 241–247.

Bunn, D. (1984). *Applied decision analysis*, McGraw-Hill: New York.

Campbell, J. E., & Painton, L. A. (1996). Optimization of reliability allocation strategies through use of genetic algorithms. *Proceedings of 6th Symposium on Multidisciplinary Design and Optimization* (pp. 1233–1242). AIAA/NASA/ISSMO.

Cantoni, M., Marseguerra, M., & Zio, E. (2000). Genetic algorithms and monte carlo simulation for optimal plant design. *Reliability Engineering and System Safety*, *68*, 29–38.

Chern, M. S. (1992). On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, *11*, 309–315.

Coit, D. W., & Smith, A. E. (1996). Reliability optimization of series–parallel systems using a genetic algorithm. *IEEE Transactions on Reliability*, *45* (2), 254–260.

Coit, D. W., & Smith, A. E. (1996). Penalty guided genetic search for reliability design optimization. *Computers and Industrial Engineering*, *30* (4), 895–904.

Coit, D. W., & Smith, A. E. (1998). Redundancy allocation to maximize a lower percentile of the system time-to-failure distribution. *IEEE Transactions on Reliability*, *47* (1), 79–87.

Coit, D. W., Smith, A. E., & Tate, D. M. (1996). Adaptive penalty methods for genetic optimization of constrained combinatorial problems. *INFORMS Journal on Computing*, *8* (2), 173–182.

Fyffe, D. E., Hines, W. W., & Lee, N. K. (1968). System reliability allocation and a computational algorithm. *IEEE Transactions on Reliability*, *R-17*, 64–69.

Gen, M., Ida, K., & Lee, J. U. (1990). A computational algorithm for solving 0–1 goal programming with GUB structures and its application for optimization problems in system reliability. *Electronics and Communications in Japan, Part 3*, *73*, 88–96.

Gnedenko, B., & Ushakov, I. (1995). *Probabilistic reliability engineering*, New York: Wiley (pp. 505–511).

Goldberg, D. (1989). *Genetic algorithms in search, optimization and machine learning*, Reading, MA: Addison-Wesley.

Holland, J. (1975). *Adaptation in natural and artificial systems*, Ann Arbor, MI: University of Michigan Press.

Ida, K., Gen M., & Yokota, T. (1994). System reliability optimization with several failure modes by genetic algorithm. *Proceedings of 16th International Conference on Computers and Industrial Engineering* (pp 349–352).

Joyce, P. A., Withers, T. A., & Hickling P. J. (1998). Application of genetic algorithms to optimum offshore plant design. *Proceedings of ESREL* 98, Trondheim (Norway), June 16–19, pp. 665–671.

Nakagawa, Y., & Miyazaki, S. (1981). Surrogate constraints algorithm for reliability optimization problems with two constraints. *IEEE Transactions on Reliability*, *R-30*, 175–180.

Nakashima, K., & Yamato, K. (1977). Optimal design of a series–parallel system with time-dependent reliability. *IEEE Transactions on Reliability*, *26* (2), 119–120.

Painton, L. A., & Campbell, J. E. (1995). Genetic algorithms in optimization of system reliability. *IEEE Transactions on Reliability*, *44* (2), 172–178.

Rubinstein, R., Levitin, G., Lisnianski, A., & Ben-Haim, H. (1997). Redundancy optimization of static series–parallel reliability models under uncertainty. *IEEE Transactions on Reliability*, *46* (4), 503–511.

Ushakov, I., & Harrison, R. (1994). *Handbook of reliability engineering*, New York: Wiley (pp. 513–515).